

THE GENERIC MAPPING TOOLS

GMT Man Pages

Release 5.1.2

**P. Wessel, W. H. F. Smith,
R. Scharroo, J. Luis, and F. Wobbe**

May 04, 2015

1	Core modules	1
1.1	blockmean	1
1.2	blockmedian	3
1.3	blockmode	5
1.4	filter1d	8
1.5	fitcircle	11
1.6	gmt_shell_functions.sh	12
1.7	gmt.conf	14
1.8	gmt	28
1.9	gmt2kml	43
1.10	gmt5syntax	48
1.11	gmtcolors	49
1.12	gmtconnect	60
1.13	gmtconvert	63
1.14	gmtdefaults	66
1.15	gmtget	67
1.16	gmtinfo	68
1.17	gmtlogo	71
1.18	gmtmath	71
1.19	gmtselect	79
1.20	gmtset	83
1.21	gmtsimplify	85
1.22	gmtspatial	87
1.23	gmtswitch	90
1.24	gmtvector	92
1.25	gmtwhich	95
1.26	grd2cpt	96
1.27	grd2rgb	99
1.28	grd2xyz	101
1.29	grdblend	104
1.30	grdclip	107
1.31	grdcontour	108
1.32	grdcut	113
1.33	grdedit	116
1.34	grdfft	118
1.35	grdfilter	122

1.36	grdgradient	126
1.37	grdhisteq	129
1.38	grdimage	131
1.39	grdinfo	135
1.40	grdlandmask	137
1.41	grdmask	140
1.42	grdmath	143
1.43	grdpaste	151
1.44	grdproject	153
1.45	grdraster	155
1.46	grdreformat	157
1.47	grdsample	161
1.48	grdtrack	164
1.49	grdtrend	168
1.50	grdvector	170
1.51	grdview	173
1.52	grdvolume	177
1.53	greenspline	179
1.54	isogmt	184
1.55	kml2gmt	185
1.56	makecpt	186
1.57	mapproject	188
1.58	nearneighbor	193
1.59	project	195
1.60	ps2raster	199
1.61	psbasemap	203
1.62	psclip	211
1.63	pscoast	213
1.64	pscontour	218
1.65	pshistogram	223
1.66	psimage	226
1.67	pslegend	228
1.68	pslib	233
1.69	psmask	244
1.70	psrose	247
1.71	psscale	250
1.72	pstext	253
1.73	pswiggle	257
1.74	psxy	259
1.75	psxyz	268
1.76	sample1d	276
1.77	spectrum1d	278
1.78	sph2grd	282
1.79	sphdistance	285
1.80	sphinterpolate	288
1.81	sphtriangulate	290
1.82	splitxyz	292
1.83	surface	295
1.84	trend1d	298
1.85	trend2d	301
1.86	triangulate	303

1.87	xyz2grd	306
2	Supplementary modules	311
2.1	gshhg	311
2.2	img2grd	313
2.3	pscoupe	317
2.4	psmeca	322
2.5	pspolar	326
2.6	psvelo	329
2.7	mgd77convert	334
2.8	mgd77info	336
2.9	mgd77list	338
2.10	mgd77magref	347
2.11	mgd77manage	350
2.12	mgd77path	356
2.13	mgd77sniffer	357
2.14	mgd77track	364
2.15	dimfilter	367
2.16	gmtgravmag3d	371
2.17	gravfft	373
2.18	grdgravmag3d	378
2.19	grdredpol	380
2.20	grdseamount	381
2.21	pssegy	383
2.22	pssegyz	386
2.23	segy2grd	388
2.24	backtracker	390
2.25	grdpmodeler	393
2.26	grdrotater	395
2.27	grdspotter	398
2.28	hotspotter	401
2.29	originator	404
2.30	rotconverter	406
2.31	x2sys_binlist	408
2.32	x2sys_cross	410
2.33	x2sys_datalist	413
2.34	x2sys_get	416
2.35	x2sys_init	417
2.36	x2sys_list	422
2.37	x2sys_merge	425
2.38	x2sys_put	425
2.39	x2sys_report	427
2.40	x2sys_solve	429

Index	431
--------------	------------

Core modules

1.1 blockmean

blockmean - Block average (x,y,z) data tables by L2 norm

1.1.1 Synopsis

blockmean [*table*] **-I***inc*[*unit*][=|+][/*yinc*[*unit*][=|+]] **-R**[*unit*]*xmin/xmax/ymin/ymax*[**r**] [**-C**] [**-E**[**p**]] [**-S**[*m*|*n*|*s*|*w*]] [**-V**[*level*]] [**-W**[**ilo**]] [**-b**[**ilo**][*ncol*][**type**][*w*][+**L**|+**B**]] [**-f**[**ilo**]*colinfo*] [**-h**[**ilo**][*n*][+**c**][+**d**][+**r***remark*][+**r***title*]] [**-i***cols*[**I**][*sscale*][*offset*][,...]] [**-o***cols*[,...]] [**-r**] [**-:**[**ilo**]]

Note: No space is allowed between the option flag and the associated arguments.

1.1.2 Description

blockmean reads arbitrarily located (x,y,z) triples [or optionally weighted quadruples (x,y,z,w)] from standard input [or *table*] and writes to standard output a mean position and value for every non-empty block in a grid region defined by the **-R** and **-I** arguments. Either **blockmean**, **blockmedian**, or **blockmode** should be used as a pre-processor before running *surface* to avoid aliasing short wavelengths. These routines are also generally useful for decimating or averaging (x,y,z) data. You can modify the precision of the output format by editing the *FORMAT_FLOAT_OUT* parameter in your *gmt.conf* file, or you may choose binary input and/or output to avoid loss of precision.

1.1.3 Required Arguments

-I*inc*[*unit*][=|+][/*yinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain.

The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

1.1.4 Optional Arguments

table 3 [or 4, see **-W**] column ASCII data table file(s) [or binary, see **-bi**] holding $(x,y,z[,w])$ data values. [w] is an optional weight for the data. If no file is specified, **blockmean** will read from standard input.

-C Use the center of the block as the output location [Default uses the mean location].

-E[p] Provide Extended report which includes **s** (the standard deviation about the mean), **l**, the lowest value, and **h**, the high value for each block. Output order becomes $x,y,z,s,l,h[,w]$. [Default outputs $x,y,z[,w]$. See **-W** for w output. If **-Ep** is used we assume weights are $1/(\text{sigma squared})$ and s becomes the propagated error of the mean.

-S[mnlslw] Use **-Sn** to report the number of points inside each block, **-Ss** to report the sum of all z -values inside a block, **-Sw** to report the sum of weights [Default (or **-Sm** reports mean value)].

-V[level] (*more ...*) Select verbosity level [c].

-W[ilo] Weighted modifier[s]. Unweighted input and output has 3 columns x,y,z ; Weighted i/o has 4 columns x,y,z,w . Weights can be used in input to construct weighted mean values in blocks. Weight sums can be reported in output for later combining several runs, etc. Use **-W** for weighted i/o, **-Wi** for weighted input only, **-Wo** for weighted output only. [Default uses unweighted i/o].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 3 (or 4 if **-Wi** is set)].

-bo[ncols][type] (*more ...*) Select binary output. [Default is 3 (or 4 if **-Wo** is set)]. **-E** adds 3 additional columns.

-f[ilo]colinfo (*more ...*) Specify data types of input and/or output columns.

-h[ilo][n][+c][+d][+rremark][+rtitle] (*more ...*) Skip or produce header record(s).

-icols[l][sscale][offset][,...] (*more ...*) Select input columns (0 is first column).

-ocols[,...] (*more ...*) Select output columns (0 is first column).

-r (*more ...*) Set pixel node registration [gridline]. Each block is the locus of points nearest the grid value location. For example, with **-R10/15/10/15** and **-I1**: with the **-r** option $10 \leq (x,y) < 11$ is one of 25 blocks; without it $9.5 \leq (x,y) < 10.5$ is one of 36 blocks.

-:[ilo] (*more ...*) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.1.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.1.6 Examples

To find 5 by 5 minute block mean values from the ASCII data in hawaii.xyg, run

```
gmt blockmean hawaii.xyg -R198/208/18/25 -I5m > hawaii_5x5.xyg
```

1.1.7 See Also

blockmedian, *blockmode*, *gmt*, *gmt.conf*, *greenspline*, *nearneighbor*, *sphtriangulate*, *surface*, *triangulate*

1.2 blockmedian

blockmedian - Block average (x,y,z) data tables by L1 norm

1.2.1 Synopsis

```
blockmedian [ table ] -Iinc[unit][=|+][/yinc[unit][=|+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -C ] [ -E[b] ] [ -E[r][-] ] [ -Q ] [ -Tquantile ] [ -V[level] ] [ -W[ilo] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,...] ] [ -ocols[,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.2.2 Description

blockmedian reads arbitrarily located (x,y,z) triples [or optionally weighted quadruples (x,y,z,w)] from standard input [or *table*] and writes to standard output a median position and value for every non-empty block in a grid region defined by the **-R** and **-I** arguments. Either *blockmean*, **blockmedian**, or *blockmode* should be used as a pre-processor before running *surface* to avoid aliasing short wavelengths. These routines are also generally useful for decimating or averaging (x,y,z) data. You can modify the precision of the output format by editing the *FORMAT_FLOAT_OUT* parameter in your *gmt.conf* file, or you may choose binary input and/or output to avoid loss of precision.

1.2.3 Required Arguments

-I*inc*[*unit*][*=|+*][*/yinc*[*unit*][*=|+*]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment

is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *ly_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

1.2.4 Optional Arguments

table 3 [or 4, see **-W**] column ASCII data table] column ASCII file(s) [or binary, see **-bi**] holding (*x,y,z[,w]*) data values. [*w*] is an optional weight for the data. If no file is specified, **blockmedian** will read from standard input.

-C Use the center of the block as the output location [Default uses the median *x* and median *y* as location (but see **-Q**)].

-E[b] Provide Extended report which includes **s** (the L1 scale of the median), **l**, the lowest value, and **h**, the high value for each block. Output order becomes *x,y,z,s,l,h[,w]*. [Default outputs *x,y,z[,w]*. For box-and-whisker calculation, use **-Eb** which will output *x,y,z,l,q25,q75,h[,*w*]*, where *q25* and *q75* are the 25% and 75% quantiles, respectively. See **-W** for *w* output.

-Ers[-] Provide source id **s** or record number **r** output, i.e., append the source id or record number associated with the median value. If tied then report the record number of the higher of the two values; append - to instead report the record number of the lower value. Note that both **-E[b]** and **-Er[-]** may be specified. For **-Es** we expect input records of the form *x,y,z[,w],sid*, where *sid* is an unsigned integer source id.

-Q (Quicker) Finds median *z* and (*x,y*) at that the median *z* [Default finds median *x*, median *y* independent of *z*]. Also see **-C**.

-Tquantile Sets the *quantile* of the distribution to be returned [Default is 0.5 which returns the median *z*]. Here, $0 < \text{quantile} < 1$.

-V[level] (*more ...*) Select verbosity level [*c*].

-W[i|o] Weighted modifier[s]. Unweighted input and output has 3 columns *x,y,z*; Weighted i/o has 4 columns *x,y,z,w*. Weights can be used in input to construct weighted mean values in blocks. Weight sums can be reported in output for later combining several runs, etc. Use **-W** for weighted i/o, **-Wi** for weighted input only, **-Wo** for weighted output only. [Default uses unweighted i/o].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 3 (or 4 if **-Wi** is set)].

-bo[ncols][type] (*more ...*) Select binary output. [Default is 3 (or 4 if **-Wo** is set)]. **-E** adds 3 additional columns.

-f[i|o]colinfo (*more ...*) Specify data types of input and/or output columns.

-h[i|o][n][+c][+d][+rremark][+rtitle] (*more ...*) Skip or produce header record(s).

-icols[l][sscale][offset][,...] (*more ...*) Select input columns (0 is first column).

- ocols[,...]** (*more ...*) Select output columns (0 is first column).
- r** (*more ...*) Set pixel node registration [gridline]. Each block is the locus of points nearest the grid value location. For example, with **-R10/15/10/15** and **-I1**: with the **-r** option $10 \leq (x,y) < 11$ is one of 25 blocks; without it $9.5 \leq (x,y) < 10.5$ is one of 36 blocks.
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.2.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.2.6 Examples

To find 5 by 5 minute block medians from the double precision binary data in *hawaii_b.xy* and output an ASCII table, run

```
gmt blockmedian hawaii_b.xy -R198/208/18/25 -I5m -bi3d > hawaii_5x5.xy
```

To compute the shape of a data distribution per bin via a box-and-whisker diagram we need the 0%, 25%, 50%, 75%, and 100% quantiles. To do so on a global 5 by 5 degree basis from the ASCII table *depths.xyz* and send output to an ASCII table, run

```
gmt blockmedian depths.xyz -Rg -I5 -Eb -r > depths_5x5.txt
```

1.2.7 See Also

blockmean, *blockmode*, *gmt*, *gmt.conf*, *greenspline*, *nearneighbor*, *surface*, *sphtriangulate*, *triangulate*

1.3 blockmode

blockmode - Block average (*x,y,z*) data tables by mode estimation

1.3.1 Synopsis

```
blockmode [ table ] -Iinc[unit][=+][yinc[unit][=+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -C ] [ -D[width][+c][+a]+l+h ] [ -Er[s-] ] [ -Q ] [ -V[level] ] [ -W[ilo] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,...] ] [ -ocols[,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.3.2 Description

blockmode reads arbitrarily located (x,y,z) triples [or optionally weighted quadruples (x,y,z,w)] from standard input [or *table*] and writes to standard output mode estimates of position and value for every non-empty block in a grid region defined by the **-R** and **-I** arguments. Either *blockmean*, *blockmedian*, or **blockmode** should be used as a pre-processor before running *surface* to avoid aliasing short wavelengths. These routines are also generally useful for decimating or averaging (x,y,z) data. You can modify the precision of the output format by editing the *FORMAT_FLOAT_OUT* parameter in your *gmt.conf* file, or you may choose binary input and/or output to avoid loss of precision.

1.3.3 Required Arguments

-I*inc*[*unit*][=+][*yinc*[*unit*][=+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest.

1.3.4 Optional Arguments

table 3 [or 4, see **-W**] column ASCII data table file(s) [or binary, see **-bi**] holding $(x,y,z[,w])$ data values. [*w*] is an optional weight for the data. If no file is specified, **blockmode** will read from standard input.

-C Use the center of the block as the output location [Default uses the modal xy location (but see **-Q**)]. **-C** overrides **-Q**.

-D[*width*][+*c*][+*a*]+*l*+*h*] Perform unweighted mode calculation via histogram binning, using the specified histogram *width*. Append +**c** to center bins so that their mid point is a multiple of *width* [uncentered]. If multiple modes are found for a block we return the average mode [+**a**]. Append +**l** or +**h** to return the low of high mode instead, respectively. If *width* is not given it will default

to 1 provided your data set only contains integers. Also, for integer data and integer bin *width* we enforce bin centering (**+c**) and select the lowest mode (**+l**) if there are multiples. [Default mode is normally the Least Median of Squares (LMS) statistic].

- E** Provide Extended report which includes **s** (the L1 scale of the mode), **l**, the lowest value, and **h**, the high value for each block. Output order becomes x,y,z,s,l,h,w . [Default outputs x,y,z,w]. See **-W** for w output.
- Erls[-]** Provide source id **s** or record number **r** output, i.e., append the source id or record number associated with the modal value. If tied then report the record number of the higher of the two values; append **-** to instead report the record number of the lower value. Note that both **-E** and **-Er[-]** may be specified. For **-Es** we expect input records of the form x,y,z,w,sid , where *sid* is an unsigned integer source id.
- Q** (Quicker) Finds mode z and mean (x,y) [Default finds mode x , mode y , mode z].
- V[level] (more ...)** Select verbosity level [c].
- W[i/o]** Weighted modifier[s]. Unweighted input and output has 3 columns x,y,z ; Weighted i/o has 4 columns x,y,z,w . Weights can be used in input to construct weighted mean values in blocks. Weight sums can be reported in output for later combining several runs, etc. Use **-W** for weighted i/o, **-Wi** for weighted input only, **-Wo** for weighted output only. [Default uses unweighted i/o].
- bi[ncols][type] (more ...)** Select binary input. [Default is 3 (or 4 if **-Wi** is set)].
- bo[ncols][type] (more ...)** Select binary output. [Default is 3 (or 4 if **-Wo** is set)]. **-E** adds 3 additional columns.
- f[i/o]colinfo (more ...)** Specify data types of input and/or output columns.
- h[i/o][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- ocols[,...] (more ...)** Select output columns (0 is first column).
- r (more ...)** Set pixel node registration [gridline]. Each block is the locus of points nearest the grid value location. For example, with **-R10/15/10/15** and **-I1**: with the **-r** option $10 \leq (x,y) < 11$ is one of 25 blocks; without it $9.5 \leq (x,y) < 10.5$ is one of 36 blocks.
- :[i/o] (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.3.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in

the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.3.6 Examples

To find 5 by 5 minute block mode estimates from the double precision binary data in hawaii_b.xyg and output an ASCII table, run:

```
gmt blockmode hawaii_b.xyg -R198/208/18/25 -I5m -bi3d > hawaii_5x5.xyg
```

To determine the most frequently occurring values per 5x5 block using histogram binning, with data representing integer counts, try

```
gmt blockmode data.txt -R0/100/0/100 -I5 -r -C -D
```

1.3.7 See Also

blockmean, *blockmedian*, *gmt*, *gmt.conf*, *greenspline*, *nearneighbor*, *sphtriangulate*, *surface*, *triangulate*

1.4 filter1d

filter1d - Do time domain filtering of 1-D data tables

1.4.1 Synopsis

```
filter1d [ table ] -Ftype<width>[mode] [ -Dincrement ] [ -E ] [ -Iignore_val ] [ -Llack_width ] [ -Nt_col ] [ -Qq_factor ] [ -Ssymmetry_factor ] [ -Tt_min/t_max/t_inc[+] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -ocols[,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.4.2 Description

filter1d is a general time domain filter for multiple column time series data. The user specifies which column is the time (i.e., the independent variable). (See **-N** option below). The fastest operation occurs when the input time series are equally spaced and have no gaps or outliers and the special options are not needed. **filter1d** has options **-L**, **-Q**, and **-S** for unevenly sampled data with gaps.

1.4.3 Required Arguments

-F*type*<*width*>[*mode*] Sets the filter *type*. Choose among convolution and non-convolution filters. Append the filter code followed by the full filter <*width*> in same units as time column. Available convolution filters are:

(b) Boxcar: All weights are equal.

- (c) Cosine Arch: Weights follow a cosine arch curve.
- (g) Gaussian: Weights are given by the Gaussian function.
- (f) Custom: Instead of *width* give name of a one-column file with your own weight coefficients.

Non-convolution filters are:

- (m) Median: Returns median value.
- (p) Maximum likelihood probability (a mode estimator): Return modal value. If more than one mode is found we return their average value. Append - or + to the filter width if you rather want to return the smallest or largest of the modal values.
- (l) Lower: Return the minimum of all values.
- (L) Lower: Return minimum of all positive values only.
- (u) Upper: Return maximum of all values.
- (U) Upper: Return maximum or all negative values only.

Upper case type **B, C, G, M, P, F** will use robust filter versions: i.e., replace outliers (2.5 L1 scale off median) with median during filtering.

In the case of **L|U** it is possible that no data passes the initial sign test; in that case the filter will return 0.0.

1.4.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Dincrement *increment* is used when series is NOT equidistantly sampled. Then *increment* will be the abscissae resolution, i.e., all abscissae will be rounded off to a multiple of *increment*. Alternatively, resample data with *sample1d*.

-E Include Ends of time series in output. Default loses half the filter-width of data at each end.

-ignore_val To ignore values; If an input value equals *ignore_val* it will be set to NaN.

-Llack_width Checks for Lack of data condition. If input data has a gap exceeding *width* then no output will be given at that point [Default does not check Lack].

-Nt_col Indicates which column contains the independent variable (time). The left-most column is # 0, the right-most is # (*n_cols* - 1). [Default is 0].

-Qq_factor Assess Quality of output value by checking mean weight in convolution. Enter *q_factor* between 0 and 1. If mean weight < *q_factor*, output is suppressed at this point [Default does not check Quality].

-Ssymmetry_factor Checks symmetry of data about window center. Enter a factor between 0 and 1. If $(\text{abs}(n_{\text{left}} - n_{\text{right}}) / (n_{\text{left}} + n_{\text{right}})) > \text{factor}$, then no output will be given at this point [Default does not check Symmetry].

-Tt_min/t_max/t_inc[+] Make evenly spaced time-steps from *t_min* to *t_max* by *t_inc* [Default uses input times]. Append + to *t_inc* if you are specifying the number of equidistant points instead.

-V[level] (more ...) Select verbosity level [c].

-bi[ncols][type] (more ...) Select binary input.

- bo**[*ncols*][*type*] (*more ...*) Select binary output. [Default is same as input].
- f**[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- g**[*a*]*xlyldlXlYlDl*[*col*]*z*[+|-]*gap*[*u*] (*more ...*) Determine data gaps and line breaks.
- h**[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- i***cols*[*l*][*sscale*][*ooffset*][,...] (*more ...*) Select input columns (0 is first column).
- o***cols*[,...] (*more ...*) Select output columns (0 is first column).
- :**[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.4.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.4.6 Examples

To filter the data set in the file *cruise.gmtd* containing evenly spaced gravity, magnetics, topography, and distance (in m) with a 10 km Gaussian filter, removing outliers, and output a filtered value every 2 km between 0 and 100 km:

```
gmt filter1d cruise.gmtd -T0/1.0e5/2000 -FG10000 -N3 -V > filtered_cruise.gmtd
```

Data along track often have uneven sampling and gaps which we do not want to interpolate using *sample1d*. To find the median depth in a 50 km window every 25 km along the track of cruise v3312, stored in *v3312.dt*, checking for gaps of 10km and asymmetry of 0.3:

```
gmt filter1d v3312.dt -FM50 -T0/100000/25 -L10 -S0.3 > v3312_filt.dt
```

1.4.7 See Also

gmt, *sample1d*, *splitxyz*

1.5 fitcircle

fitcircle - find mean position and pole of best-fit great [or small] circle to points on a sphere.

1.5.1 Synopsis

```
fitcircle [ table ] -Lnorm [ -S[lat] ] [ -V[level] ] [ -bi[ncols][type][w][+L+B] ] [ -f[i]o]colinfo ] [ -g[a]x|y|d|X|Y|D|[col]z[+|-]gap[u] ] [ -h[i]o][n][+c][+d][+rremark][+rtitle] ] [ -i]cols[I][sscale][ooffset][,...] ] [ -o]cols[,...] ] [ -:[i]o ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.5.2 Description

fitcircle reads lon,lat [or lat,lon] values from the first two columns on standard input [or *xyfile*]. These are converted to Cartesian three-vectors on the unit sphere. Then two locations are found: the mean of the input positions, and the pole to the great circle which best fits the input positions. The user may choose one or both of two possible solutions to this problem. The first is called **-L1** and the second is called **-L2**. When the data are closely grouped along a great circle both solutions are similar. If the data have large dispersion, the pole to the great circle will be less well determined than the mean. Compare both solutions as a qualitative check.

The **-L1** solution is so called because it approximates the minimization of the sum of absolute values of cosines of angular distances. This solution finds the mean position as the Fisher average of the data, and the pole position as the Fisher average of the cross-products between the mean and the data. Averaging cross-products gives weight to points in proportion to their distance from the mean, analogous to the “leverage” of distant points in linear regression in the plane.

The **-L2** solution is so called because it approximates the minimization of the sum of squares of cosines of angular distances. It creates a 3 by 3 matrix of sums of squares of components of the data vectors. The eigenvectors of this matrix give the mean and pole locations. This method may be more subject to roundoff errors when there are thousands of data. The pole is given by the eigenvector corresponding to the smallest eigenvalue; it is the least-well represented factor in the data and is not easily estimated by either method.

1.5.3 Required Arguments

-Lnorm Specify the desired *norm* as 1 or 2, or use **-L** or **-L3** to see both solutions.

1.5.4 Optional Arguments

table One or more ASCII [or binary, see **-bi**] files containing lon,lat [or lat,lon; see **-:[i]o**] values in the first 2 columns. If no file is specified, **fitcircle** will read from standard input.

-S[lat] Attempt to fit a small circle instead of a great circle. The pole will be constrained to lie on the great circle connecting the pole of the best-fit great circle and the mean location of the data. Optionally append the desired fixed latitude of the small circle [Default will determine the latitude].

-V[level] (more ...) Select verbosity level [c].

-bi[ncols][type] (more ...) Select binary input. [Default is 2 input columns].

- f[ilo]colinfo** (*more ...*) Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|D|col|z[+|-]gap[u]** (*more ...*) Determine data gaps and line breaks.
- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
- icols[l][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
- ocols[,...]** (*more ...*) Select output columns (0 is first column).
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.5.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.5.6 Examples

Suppose you have lon,lat,grav data along a twisty ship track in the file *ship.xyg*. You want to project this data onto a great circle and resample it in distance, in order to filter it or check its spectrum. Do the following:

```
gmt fitcircle ship.xyg -L2
gmt project ship.xyg -Cox/oy -Tpx/py -S -Fpz | sample1d -S-100 -I1 > output.pg
```

Here, *ox/oy* is the lon/lat of the mean from **fitcircle**, and *px/py* is the lon/lat of the pole. The file *output.pg* has distance, gravity data sampled every 1 km along the great circle which best fits *ship.xyg*

1.5.7 See Also

gmt, gmtvector, project, mapproject, sample1d

1.6 gmt_shell_functions.sh

gmt_shell_functions.sh - Practical functions to be used in GMT bourne shell scripts

1.6.1 Synopsis

gmt_init_tmpdir

gmt_remove_tmpdir

gmt_clean_up [*prefix*]

gmt_message *message*

gmt_abort *message*

gmt_get_nrecords *file(s)*

gmt_get_ndatarecords *file(s)*

gmt_get_nfields *string*

gmt_get_field *string*

gmt_get_region *file(s)* [*options*]

gmt_get_gridregion *file* [*options*]

gmt_get_map_width -R -J

gmt_get_map_height -R -J

gmt_set_psfile *file*

gmt_set_pdffile *file*

gmt_set_framename *prefix framenumbers*

gmt_set_framenext *framenumbers*

1.6.2 Description

gmt_shell_functions.sh provides a set of functions to Bourne (again) shell scripts in support of GMT. The calling shell script should include the following line, before the functions can be used:

```
. gmt_shell_functions.sh
```

Once included in a shell script, **gmt_shell_functions.sh** allows GMT users to do some scripting more easily than otherwise. The functions made available are:

gmt_init_tmpdir Creates a temporary directory in **/tmp** or (when defined) in the directory specified by the environment variable **TMPDIR**. The name of the temporary directory is returned as environment variable **GMT_TMPDIR**. This function also causes GMT to run in ‘isolation mode’, i.e., all temporary files will be created in **GMT_TMPDIR** and the *gmt.conf* file will not be adjusted.

gmt_remove_tmpdir Removes the temporary directory and unsets the **GMT_TMPDIR** environment variable.

gmt_cleanup Remove all files and directories in which the current process number is part of the file name. If the optional *prefix* is given then we also delete all files and directories that begins with the given prefix.

gmt_message Send a message to standard error.

gmt_abort Send a message to standard error and exit the shell.

gmt_get_nrecords Returns the total number of lines in *file(s)*

- gmt_get_ndatarecords** Returns the total number of data records in *file(s)*, i.e., not counting headers.
- gmt_get_nfields** Returns the number of fields or words in *string*
- gmt_get_field** Returns the given *field* in a *string*. Must pass *string* between double quotes to preserve it as one item.
- gmt_get_region** Returns the region in the form *w/e/s/n* based on the data in table *file(s)*. Optionally add *-I*dx*/dy* to round off the answer.
- gmt_get_gridregion** Returns the region in the form *w/e/s/n* based on the header of a grid *file*. Optionally add *-I*dx*/dy* to round off the answer.
- gmt_get_map_width** Expects the user to give the desired **-R -J** settings and returns the map width in the current measurement unit.
- gmt_get_map_height** Expects the user to give the desired **-R -J** settings and returns the map height in the current measurement unit.
- gmt_set_psfile** Create the output PostScript file name based on the base name of a given file (usually the script name \$0).
- gmt_set_framename** Returns a lexically ordered filename stem (i.e., no extension) given the file prefix and the current frame number, using a width of 6 for the integer including leading zeros. Useful when creating animations and lexically sorted filenames are required.
- gmt_set_framenext** Accepts the current frame integer counter and returns the next integer counter.

1.6.3 Notes

1. These functions only work in the Bourne shell (**sh**) and their derivatives (like **ash**, **bash**, **ksh** and **zsh**). These functions do not work in the C shell (**cs**) or their derivatives (like **tcsh**), and cannot be used in DOS batch scripts either.
2. **gmt_shell_functions.sh** were first introduced in GMT version 4.2.2 and have since been regularly expanded with other practical scripting short-cuts. If you want to suggest other functions, please do so by adding a New Issue request on gmt.soest.hawaii.edu.

1.6.4 See Also

gmt , *gmt.conf* , *gmtinfo* , *grdinfo*

1.7 gmt.conf

gmt.conf - Configuration for GMT

1.7.1 Description

The following is a list of the parameters that are user-definable in GMT. The parameter names are always given in UPPER CASE. The parameter values are case-insensitive unless otherwise noted. The system defaults are given in brackets [for SI (and US)]. Those marked * can be set on the command line as well (the corresponding option is given in parentheses). Note that default distances and lengths below are given in both cm or inch; the chosen default depends on your choice of default unit (see

PROJ_LENGTH_UNIT). You can explicitly specify the unit used for distances and lengths by appending **c** (cm), **i** (inch), or **p** (points). When no unit is indicated the value will be assumed to be in the unit set by *PROJ_LENGTH_UNIT*. Several parameters take only **true** or **false**. Finally, most of these parameters can be changed on-the-fly via the **--PARAMETER=VALUE** option to any GMT program. However, a few are static and are only read via the **gmt.conf** file; these are labeled (static).

1.7.2 Common Specifications

The full explanation for how to specify pens, pattern fills, colors, and fonts can be found in the *gmt* man page.

THEMATIC SUB-SECTIONS	<i>prefix</i>
COLOR Parameters	COLOR_
DIR Parameters	DIR_
FONT Parameters	FONT_
FORMAT Parameters	FORMAT_
GMT Miscellaneous Parameters	GMT_
I/O Parameters	IO_
MAP Parameters	MAP_
Projection Parameters	PROJ_
PostScript Parameters	PS_
Calendar/Time Parameters	TIME_

COLOR_BACKGROUND Color used for the background of images (i.e., when $z <$ lowest color table entry) [black].

COLOR_FOREGROUND Color used for the foreground of images (i.e., when $z >$ highest color table entry) [white].

COLOR_HSV_MAX_S Maximum saturation (0-1) assigned for most positive intensity value [0.1].

COLOR_HSV_MIN_S Minimum saturation (0-1) assigned for most negative intensity value [1.0].

COLOR_HSV_MAX_V Maximum value (0-1) assigned for most positive intensity value [1.0].

COLOR_HSV_MIN_V Minimum value (0-1) assigned for most negative intensity value [0.3].

COLOR_MODEL Selects in which color space a color palette should be interpolated. By default, color interpolation takes place directly on the RGB values which can produce some unexpected hues, whereas interpolation directly on the HSV values better preserves those hues. The choices are: **none** (default: use whatever the **COLOR_MODEL** setting in the color palette file demands), **rgb** (force interpolation in RGB), **hsv** (force interpolation in HSV), **cmymk** (assumes colors are in CMYK but interpolates in RGB).

COLOR_NAN Color used for the non-defined areas of images (i.e., where $z == \text{NaN}$) [127.5].

DIR_DATA Session data dir. Overrides the value of the environment variable **\$GMT_DATADIR** (see *Directory parameters* in the Cookbook).

DIR_DCW Path to optional Digital Chart of the World polygon files.

DIR_GSHHG Path to GSHHG files. Defaults to **\$GMT_SHAREDIR/coast** if empty.

FONT Sets the default for all fonts, except **FONT_LOGO**. This setting is not included in the **gmt.conf** file.

FONT_ANNOT_PRIMARY Font used for primary annotations, etc. [12p,Helvetica,black]. When **+** is prepended, scale fonts, offsets and tick-lengths relative to **FONT_ANNOT_PRIMARY**.

FONT_ANNOT_SECONDARY Font to use for time axis secondary annotations [14p,Helvetica,black].

FONT_LABEL Font to use when plotting labels below axes [16p,Helvetica,black].

FONT_LOGO Font to use for text plotted as part of the GMT time logo [8p,Helvetica,black].

FONT_TITLE Font to use when plotting titles over graphs [24p,Helvetica,black].

FORMAT_CLOCK_IN Formatting template that indicates how an input clock string is formatted. This template is then used to guide the reading of clock strings in data fields. To properly decode 12-hour clocks, append am or pm (or upper case) to match your data records. As examples, try hh:mm, hh:mm:ssAM, etc. [hh:mm:ss].

FORMAT_CLOCK_MAP Formatting template that indicates how an output clock string is to be plotted. This template is then used to guide the formatting of clock strings in plot annotations. See **FORMAT_CLOCK_OUT** for details. [hh:mm:ss].

FORMAT_CLOCK_OUT Formatting template that indicates how an output clock string is to be formatted. This template is then used to guide the writing of clock strings in data fields. To use a floating point format for the smallest unit (e.g., seconds), append .xxx, where the number of x indicates the desired precision. If no floating point is indicated then the smallest specified unit will be rounded off to nearest integer. For 12-hour clocks, append am, AM, a.m., or A.M. (GMT will replace aA with pP for pm). If your template starts with a leading hyphen (-) then each integer item (y,m,d) will be printed without leading zeros (default uses fixed width formats). As examples, try hh:mm, hh.mm.ss, hh:mm:ss.xxxx, hha.m., etc. [hh:mm:ss]. If the format is simply - then no clock is output and the ISO T divider between date and clock is omitted.

FORMAT_DATE_IN Formatting template that indicates how an input date string is formatted. This template is then used to guide the reading of date strings in data fields. You may specify either Gregorian calendar format or ISO week calendar format. Gregorian calendar: Use any combination of yyyy (or yy for 2-digit years; if so see **TIME_Y2K_OFFSET_YEAR**), mm (or o for abbreviated month name in the current time language), and dd, with or without delimiters. For day-of-year data, use jjj instead of mm and/or dd. Examples can be ddmmyyyy, yy-mm-dd, dd-o-yyyy, yyyy/dd/mm, yyyy-jjj, etc. ISO Calendar: Expected template is yyyy[-]W[-]ww[-]d, where ww is ISO week and d is ISO week day. Either template must be consistent, e.g., you cannot specify months if you do not specify years. Examples are yyyyWwwd, yyyy-Www, etc. [yyyy-mm-dd].

FORMAT_DATE_MAP Formatting template that indicates how an output date string is to be plotted. This template is then used to guide the plotting of date strings in data fields. See **FORMAT_DATE_OUT** for details. In addition, you may use a single o instead of mm (to plot month name) and u instead of W[-]ww to plot “Week ##”. Both of these text strings will be affected by the **TIME_LANGUAGE**, **FORMAT_TIME_PRIMARY_MAP** and **FORMAT_TIME_SECONDARY_MAP** setting. [yyyy-mm-dd].

FORMAT_DATE_OUT Formatting template that indicates how an output date string is to be formatted. This template is then used to guide the writing of date strings in data fields. You may specify either Gregorian calendar format or ISO week calendar format. Gregorian calendar: Use any combination of yyyy (or yy for 2-digit years; if so see **TIME_Y2K_OFFSET_YEAR**), mm (or o for abbreviated month name in the current time language), and dd, with or without delimiters. For day-of-year data, use jjj instead of mm and/or dd. As examples, try yy/mm/dd, yyyy=jjj, dd-o-yyyy, dd-mm-yy, yy-mm, etc. ISO Calendar: Expected template is yyyy[-]W[-]ww[-]d, where ww is ISO week and d is ISO week day. Either template must be consistent, e.g., you cannot specify months if you do not specify years. As examples, try yyyyWww, yy-W-ww-d, etc. If your template starts with a leading hyphen (-) then each integer item (y,m,d) will be printed without

leading zeros (default uses fixed width formats) [yyyy-mm-dd]. If the format is simply - then no date is output and the ISO T divider between date and clock is omitted.

FORMAT_GEO_MAP Formatting template that indicates how an output geographical coordinate is to be plotted. This template is then used to guide the plotting of geographical coordinates in data fields. See **FORMAT_GEO_OUT** for details. In addition, you can append A which plots the absolute value of the coordinate. The default is ddd:mm:ss. Not all items may be plotted as this depends on the annotation interval.

FORMAT_GEO_OUT Formatting template that indicates how an output geographical coordinate is to be formatted. This template is then used to guide the writing of geographical coordinates in data fields. The template is in general of the form [+|-]D or [+|-]ddd[:mm[:ss]][.xxx][F]. By default, longitudes will be reported in the range [-180,180]. The various terms have the following purpose:

+D Output longitude in the range [0,360]

-D Output longitude in the range [-360,0]

D Use **FORMAT_FLOAT_OUT** for floating point degrees.

ddd Fixed format integer degrees

: delimiter used

mm Fixed format integer arc minutes

ss Fixed format integer arc seconds

.xxx Floating fraction of previous integer field, fixed width.

F Encode sign using WESN suffix

G Same as F but with a leading space before suffix

The default is D.

FORMAT_FLOAT_MAP Format (C language printf syntax) to be used when plotting double precision floating point numbers along plot frames and contours. For geographic coordinates, see **FORMAT_GEO_MAP**. [% .12lg].

FORMAT_FLOAT_OUT Format (C language printf syntax) to be used when printing double precision floating point numbers to output files. For geographic coordinates, see **FORMAT_GEO_OUT**. [% .12lg]. To give some columns a separate format, supply one or more comma-separated *cols:format* specifications, where *cols* can be specific columns (e.g., 5 for 6th since 0 is the first) or a range of columns (e.g., 3-7). The last specification without column information will override the format for all other columns.

FORMAT_TIME_PRIMARY_MAP Controls how primary month-, week-, and weekday-names are formatted. Choose among **full**, **abbreviated**, and **character**. If the leading **f**, **a**, or **c** are replaced with **F**, **A**, and **C** the entire annotation will be in upper case [full].

FORMAT_TIME_SECONDARY_MAP Controls how secondary month-, week-, and weekday-names are formatted. Choose among **full**, **abbreviated**, and **character**. If the leading **f**, **a**, or **c** are replaced with **F**, **A**, and **C** the entire annotation will be in upper case [full].

FORMAT_TIME_STAMP Defines the format of the time information in the UNIX time stamp. This format is parsed by the C function **strftime**, so that virtually any text can be used (even not containing any time information) [%Y %b %d %H:%M:%S].

GMT_COMPATIBILITY Determines if this GMT version should be able to parse command-line options for a prior major release. Specify either 4 or 5. If 4 is set we will parse obsolete GMT 4 options and issue warnings; if 5 is set then parsing GMT 4 only syntax will result in errors [4].

GMT_EXTRAPOLATE_VAL Determines what to do if extrapolating beyond the data domain. Choose among 'NaN', 'extrap' or 'extrapval,val' (without quotes). In the first case return NaN for any element of x that is outside range [Default]. Second case lets the selected algorithm compute the extrapolation values. Third case sets the extrapolation values to the constant value passed in 'val' (this value must of course be numeric).

GMT_CUSTOM_LIBS Comma-separated list of GMT-compliant shared libraries that extend the capability of GMT with additional custom modules [none]. Alternatively, provide a directory name, that **MUST** end with a slash (or back slash), to use all shared libraries in that directory. On Windows, if the dir name is made up only of a single slash ('/') search inside a subdirectory called 'gmt_plugins' of the directory that contains the 'gmt' executable. See the API documentation for how to build your own shared modules.

GMT_FFT Determines which Fast Fourier Transform (FFT) should be used among those that have been configured during installation. Choose from **auto** (pick the most suitable for the task among available algorithms), **fftw**[,*planner_flag*] (The Fastest Fourier Transform in the West), **accelerate** (Use the Accelerate Framework under OS X; Note, that the number of samples to be processed must be a base 2 exponent), **kiss**, (Kiss FFT), **brenner** Brenner Legacy FFT [auto]. FFTW can “learn” how to optimally compute Fourier transforms on the current hardware and OS by computing several FFTs and measuring their execution time. This so gained “Wisdom” will be stored in and reloaded from the file `fftw_wisdom_<hostname>` in `$GMT_USERDIR` or, if `$GMT_USERDIR` is not writable, in the current directory. To use this feature append *planner_flag*, which can be one of *measure*, *patient*, and *exhaustive*; see FFTW reference for details. The default FFTW planner flag is *estimate*, i.e., pick a (probably sub-optimal) plan quickly. Note: if you need a single transform of a given size only, the one-time cost of the smart planner becomes significant. In that case, stick to the default planner, *estimate*, based on heuristics.

GMT_HISTORY Passes the history of past common command options via the `gmt.history` file. The different values for this setting are: **true**, **readonly**, **false**, to either read and write to the `gmt.history` file, only read, or not use the file at all [true].

GMT_INTERPOLANT Determines if linear (linear), Akima's spline (akima), natural cubic spline (cubic) or no interpolation (none) should be used for 1-D interpolations in various programs [akima].

GMT_TRIANGULATE Determines if we use the **Watson** [Default] or **Shewchuk** algorithm (if configured during installation) for triangulation. Note that Shewchuk is required for operations involving Voronoi constructions.

GMT_VERBOSE (-V) Determines the level of verbosity used by GMT programs. Choose among 6 levels; each level adds to the verbosity of the lower levels: **quiet**, **nnormal** (errors and warnings), compatibility warnings, **verbose** progress reports, **long verbose** progress reports, **debugging** messages [c].

IO_COL_SEPARATOR This setting determines what character will separate ASCII output data columns written by GMT. Choose from tab, space, comma, and none [tab].

IO_GRIDFILE_FORMAT Default file format for grids, with optional scale, offset and invalid value, written as *ff/scale/offset/invalid*. The 2-letter format indicator can be one of [**abcegnrs**][**bsifd**]. See *gridreformat* and Section 4.20 of the GMT Technical Reference and Cookbook for more information. The *scale* and *offset* modifiers may be left empty to select default values (scale = 1, offset = 0), or you may specify *a* for auto-adjusting the scale and/or offset of packed integer grids (=id/a

is a shorthand for `=id/a/a`). When *invalid* is omitted the appropriate value for the given format is used (NaN or largest negative). [nf].

IO_GRIDFILE_SHORTHAND If true, all grid file names are examined to see if they use the file extension shorthand discussed in Section 4.17 of the GMT Technical Reference and Cookbook. If false, no filename expansion is done [false].

IO_HEADER (-h) Specifies whether input/output ASCII files have header record(s) or not [false].

IO_LONLAT_TOGGLE (-:) Set if the first two columns of input and output files contain (latitude,longitude) or (y,x) rather than the expected (longitude,latitude) or (x,y). false means we have (x,y) both on input and output. true means both input and output should be (y,x). IN means only input has (y,x), while OUT means only output should be (y,x). [false].

IO_N_HEADER_RECS Specifies how many header records to expect if **-h** is used [0]. Note: This will skip the specified number of records regardless of what they are. Since any records starting with # is automatically considered a header you will only specify a non-zero number in order to skip headers that do not conform to that convention.

IO_NAN_RECORDS Determines what happens when input records containing NaNs for *x* or *y* (and in some cases *z*) are read. Choose between **skip**, which will simply report how many bad records were skipped, and **pass** [Default], which will pass these records on to the calling programs. For most programs this will result in output records with NaNs as well, but some will interpret these NaN records to indicate gaps in a series; programs may then use that information to detect segmentation (if applicable).

IO_NC4_CHUNK_SIZE Sets the default chunk size for the **lat** and **lon** dimension of the **z** variable. Very large chunk sizes and sizes smaller than 128 should be avoided because they can lead to unexpectedly bad performance. Note that a chunk of a single precision floating point variable of size 2896x2896 completely fills the chunk cache of 32MiB. Specify the chunk size for each dimension separated by a comma, or **auto** for optimally chosen chunk sizes in the range [128,256]. Setting **IO_NC4_CHUNK_SIZE** will produce netCDF version 4 files, which can only be read with the netCDF 4 library, unless all dimensions are less than 128 or **classic** is specified for classic netCDF. [auto]

IO_NC4_DEFLATION_LEVEL Sets the compression level for netCDF4 files upon output. Values allowed are integers from 0 (no compression) to 9 (maximum compression). Enabling a low compression level can dramatically improve performance and reduce the size of certain data. While higher compression levels further reduce the data size, they do so at the cost of extra processing time. This parameter does not apply to classic netCDF files. [3]

IO_SEGMENT_MARKER This holds the character we expect to indicate a segment header in an incoming ASCII data or text table [>]. If this marker should be different for output then append another character for the output segment marker. The two characters must be separated by a comma. Two marker characters have special meaning: B means “blank line” and will treat blank lines as initiating a new segment, whereas N means “NaN record” and will treat records with all NaNs as initiating a new segment. If you choose B or N for the output marker then the normal GMT segment header is replaced by a blank or NaN record, respectively, and no segment header information is written. To use B or N as regular segment markers you must escape them with a leading backslash.

MAP_ANNOT_MIN_ANGLE If the angle between the map boundary and the annotation baseline is less than this minimum value (in degrees), the annotation is not plotted (this may occur for certain oblique projections.) Give a value in the range [0,90]. [20]

MAP_ANNOT_MIN_SPACING If an annotation would be plotted less than this minimum distance

from its closest neighbor, the annotation is not plotted (this may occur for certain oblique projections.) [0p]

MAP_ANNOT_OBLIQUE This integer is a sum of 6 bit flags (most of which only are relevant for oblique projections): If bit 1 is set (1), annotations will occur wherever a gridline crosses the map boundaries, else longitudes will be annotated on the lower and upper boundaries only, and latitudes will be annotated on the left and right boundaries only. If bit 2 is set (2), then longitude annotations will be plotted horizontally. If bit 3 is set (4), then latitude annotations will be plotted horizontally. If bit 4 is set (8), then oblique tick-marks are extended to give a projection equal to the specified tick length. If bit 5 is set (16), tick-marks will be drawn normal to the border regardless of gridline angle. If bit 6 is set (32), then latitude annotations will be plotted parallel to the border. To set a combination of these, add up the values in parentheses. [1].

MAP_ANNOT_OFFSET_PRIMARY Distance from end of tick-mark to start of annotation [5p].

MAP_ANNOT_OFFSET_SECONDARY Distance from base of primary annotation to the top of the secondary annotation [5p] (Only applies to time axes with both primary and secondary annotations).

MAP_ANNOT_ORTHO Determines which axes will get their annotations (for linear projections) plotted orthogonally to the axes. Combine any **w, e, s, n, z** (uppercase allowed as well). [we].

MAP_DEFAULT_PEN Sets the default of all pens related to **-W** options. Prepend **+** to overrule the color of the parameters **MAP_GRID_PEN_PRIMARY**, **MAP_GRID_PEN_SECONDARY**, **MAP_FRAME_PEN**, **MAP_TICK_PEN_PRIMARY**, and **MAP_TICK_PEN_SECONDARY** by the color of **MAP_DEFAULT_PEN** [default,black].

MAP_DEGREE_SYMBOL Determines what symbol is used to plot the degree symbol on geographic map annotations. Choose between ring, degree, colon, or none [ring].

MAP_FRAME_AXES Sets which axes to draw and annotate. Combine any uppercase **W, E, S, N, Z** to draw and annotate west, east, south, north and/or vertical (perspective view only) axis. Use lower case to draw the axis only, but not annotate. Add an optional **+** to draw a cube of axes in perspective view. [WESN].

MAP_FRAME_PEN Pen attributes used to draw plain map frame [thicker,black].

MAP_FRAME_TYPE Choose between **inside**, **plain** and **fancy** (thick boundary, alternating black/white frame; append **+** for rounded corners) [fancy]. For some map projections (e.g., Oblique Mercator), plain is the only option even if fancy is set as default. In general, fancy only applies to situations where the projected x and y directions parallel the longitude and latitude directions (e.g., rectangular projections, polar projections). For situations where all boundary ticks and annotations must be inside the maps (e.g., for preparing geotiffs), chose **inside**.

MAP_FRAME_WIDTH Width (> 0) of map borders for fancy map frame [5p].

MAP_GRID_CROSS_SIZE_PRIMARY Size (>= 0) of grid cross at lon-lat intersections. 0 means draw continuous gridlines instead [0p].

MAP_GRID_CROSS_SIZE_SECONDARY Size (>= 0) of grid cross at secondary lon-lat intersections. 0 means draw continuous gridlines instead [0p].

MAP_GRID_PEN_PRIMARY Pen attributes used to draw primary grid lines in dpi units or points (append p) [default,black].

MAP_GRID_PEN_SECONDARY Pen attributes used to draw secondary grid lines in dpi units or points (append p) [thinner,black].

MAP_LABEL_OFFSET Distance from base of axis annotations to the top of the axis label [8p].

- MAP_LINE_STEP** Determines the maximum length (> 0) of individual straight line-segments when drawing arcuate lines [0.75p]
- MAP_LOGO** (-U) Specifies if a GMT logo with system timestamp should be plotted at the lower left corner of the plot [false].
- MAP_LOGO_POS** (-U) Sets the justification and the position of the logo/timestamp box relative to the current plots lower left corner of the plot [BL/-54p/-54p].
- MAP_ORIGIN_X** (-X) Sets the x-coordinate of the origin on the paper for a new plot [1i]. For an overlay, the default offset is 0.
- MAP_ORIGIN_Y** (-Y) Sets the y-coordinate of the origin on the paper for a new plot [1i]. For an overlay, the default offset is 0.
- MAP_POLAR_CAP** Controls the appearance of gridlines near the poles for all azimuthal projections and a few others in which the geographic poles are plotted as points (Lambert Conic, Hammer, Mollweide, Sinusoidal, and van der Grinten). Specify either none (in which case there is no special handling) or *pc_lat/pc_dlon*. In that case, normal gridlines are only drawn between the latitudes *-pc_lat/+*pc_lat**, and above those latitudes the gridlines are spaced at the (presumably coarser) *pc_dlon* interval; the two domains are separated by a small circle drawn at the *pc_lat* latitude [85/90]. Note for r-theta (polar) projection where $r = 0$ is at the center of the plot the meaning of the cap is reversed, i.e., the default 85/90 will draw a $r = 5$ radius circle at the center of the map with less frequent radial lines there.
- MAP_SCALE_HEIGHT** Sets the height (> 0) on the map of the map scale bars drawn by various programs [5p].
- MAP_TICK_LENGTH_PRIMARY** The length of a primary major/minor tick-marks [5p/2.5p]. If only the first value is set, the second is assumed to be 50% of the first.
- MAP_TICK_LENGTH_SECONDARY** The length of a secondary major/minor tick-marks [15p/3.75p]. If only the first value is set, the second is assumed to be 25% of the first.
- MAP_TICK_PEN_PRIMARY** Pen attributes to be used for primary tick-marks in dpi units or points (append p) [thinner,black].
- MAP_TICK_PEN_SECONDARY** Pen attributes to be used for secondary tick-marks in dpi units or points (append p) [thinner,black].
- MAP_TITLE_OFFSET** Distance from top of axis annotations (or axis label, if present) to base of plot title [14p].
- MAP_VECTOR_SHAPE** Determines the shape of the head of a vector. Normally (i.e., for *vector_shape = 0*), the head will be triangular, but can be changed to an arrow (1) or an open V (2). Intermediate settings give something in between. Negative values (up to -2) are allowed as well [0].
- PROJ_AUX_LATITUDE** Only applies when geodesics are approximated by great circle distances on an equivalent sphere. Select from authalic, geocentric, conformal, meridional, parametric, or none [authalic]. When not none we convert any latitude used in the great circle calculation to the chosen auxiliary latitude before doing the distance calculation. See also **PROJ_MEAN_RADIUS**.
- PROJ_ELLIPSOID** The (case sensitive) name of the ellipsoid used for the map projections [WGS-84]. Choose among:

Airy: Applies to Great Britain (1830)
Airy-Ireland: Applies to Ireland in 1965 (1830)
Andrae: Applies to Denmark and Iceland (1876)
APL4.9: Appl. Physics (1965)
ATS77: Average Terrestrial System, Canada Maritime provinces (1977)
Australian: Applies to Australia (1965)
Bessel: Applies to Central Europe, Chile, Indonesia (1841)
Bessel-Namibia: Same as Bessel-Schwazeck (1841)
Bessel-NGO1948: Modified Bessel for NGO 1948 (1841)
Bessel-Schwazeck: Applies to Namibia (1841)
Clarke-1858: Clarke's early ellipsoid (1858)
Clarke-1866: Applies to North America, the Philippines (1866)
Clarke-1866-Michigan: Modified Clarke-1866 for Michigan (1866)
Clarke-1880: Applies to most of Africa, France (1880)
Clarke-1880-Arc1950: Modified Clarke-1880 for Arc 1950 (1880)
Clarke-1880-IGN: Modified Clarke-1880 for IGN (1880)
Clarke-1880-Jamaica: Modified Clarke-1880 for Jamaica (1880)
Clarke-1880-Merchich: Modified Clarke-1880 for Merchich (1880)
Clarke-1880-Palestine: Modified Clarke-1880 for Palestine (1880)
CPM: Comm. des Poids et Mesures, France (1799)
Delambre: Applies to Belgium (1810)
Engelis: Goddard Earth Models (1985)
Everest-1830: India, Burma, Pakistan, Afghanistan, Thailand (1830)
Everest-1830-Kalianpur: Modified Everest for Kalianpur (1956) (1830)
Everest-1830-Kertau: Modified Everest for Kertau, Malaysia & Singapore (1830)
Everest-1830-Pakistan: Modified Everest for Pakistan (1830)
Everest-1830-Timbalai: Modified Everest for Timbalai, Sabah Sarawak (1830)
Fischer-1960: Used by NASA for Mercury program (1960)
Fischer-1960-SouthAsia: Same as Modified-Fischer-1960 (1960)
Fischer-1968: Used by NASA for Mercury program (1968)
FlatEarth: As Sphere, but implies fast "Flat Earth" distance calculations (1984)
GRS-67: International Geodetic Reference System (1967)
GRS-80: International Geodetic Reference System (1980)
Hayford-1909: Same as the International 1924 (1909)
Helmert-1906: Applies to Egypt (1906)
Hough: Applies to the Marshall Islands (1960)
Hughes-1980: Hughes Aircraft Company for DMSP SSM/I grid products (1980)
IAG-75: International Association of Geodesy (1975)
Indonesian: Applies to Indonesia (1974)
International-1924: Worldwide use (1924)
International-1967: Worldwide use (1967)
Kaula: From satellite tracking (1961)
Krassovsky: Used in the (now former) Soviet Union (1940)
Lerch: For geoid modelling (1979)
Maupertius: Really old ellipsoid used in France (1738)

Mercury-1960: Same as Fischer-1960 (1960)
MERIT-83: United States Naval Observatory (1983)
Modified-Airy: Same as Airy-Ireland (1830)
Modified-Fischer-1960: Applies to Singapore (1960)
Modified-Mercury-1968: Same as Fischer-1968 (1968)
NWL-10D: Naval Weapons Lab (Same as WGS-72) (1972)
NWL-9D: Naval Weapons Lab (Same as WGS-66) (1966)
OSU86F: Ohio State University (1986)
OSU91A: Ohio State University (1991)
Plessis: Old ellipsoid used in France (1817)
SGS-85: Soviet Geodetic System (1985)
South-American: Applies to South America (1969)
Sphere: The mean radius in WGS-84 (for spherical/plate tectonics applications) (1984)
Struve: Friedrich Georg Wilhelm Struve (1860)
TOPEX: Used commonly for altimetry (1990)
Walbeck: First least squares solution by Finnish astronomer (1819)
War-Office: Developed by G. T. McCaw (1926)
WGS-60: World Geodetic System (1960)
WGS-66: World Geodetic System (1966)
WGS-72: World Geodetic System (1972)
WGS-84: World Geodetic System [Default] (1984)
Moon: Moon (IAU2000) (2000)
Mercury: Mercury (IAU2000) (2000)
Venus: Venus (IAU2000) (2000)
Mars: Mars (IAU2000) (2000)
Jupiter: Jupiter (IAU2000) (2000)
Saturn: Saturn (IAU2000) (2000)
Uranus: Uranus (IAU2000) (2000)
Neptune: Neptune (IAU2000) (2000)
Pluto: Pluto (IAU2000) (2000)

Note that for some global projections, GMT may use a spherical approximation of the ellipsoid chosen, setting the flattening to zero, and using a mean radius. A warning will be given when this happens. If a different ellipsoid name than those mentioned here is given, GMT will attempt to parse the name to extract the semi-major axis (a in m) and the flattening. Formats allowed are:

a implies a zero flattening

a,inv_f where inv_f is the inverse flattening

$a,b=b$ where b is the semi-minor axis (in m)

$a,f=f$ where f is the flattening

This way a custom ellipsoid (e.g., those used for other planets) may be used. Further note that coordinate transformations in **mapproject** can also specify specific datums; see the [mapproject](#) man page for further details and how to view ellipsoid and datum parameters.

PROJ_LENGTH_UNIT Sets the unit length. Choose between **cm**, **inch**, or **point** [c (or i)]. Note that, in GMT, one point is defined as 1/72 inch (the PostScript definition), while it is often defined as 1/72.27 inch in the typesetting industry. There is no universal definition.

PROJ_MEAN_RADIUS Applies when geodesics are approximated by great circle distances on an equivalent sphere or when surface areas are computed. Select from mean (R_1), authalic (R_2), volumetric (R_3), meridional, or quadratic [authalic]. See also PROJ_MEAN_RADIUS.

PROJ_SCALE_FACTOR Changes the default map scale factor used for the Polar Stereographic [0.9996], UTM [0.9996], and Transverse Mercator [1] projections in order to minimize areal distortion. Provide a new scale-factor or leave as default.

PS_CHAR_ENCODING (static) Names the eight bit character set being used for text in files and in command line parameters. This allows GMT to ensure that the PostScript output generates the correct characters on the plot.. Choose from Standard, Standard+, ISOLatin1, ISOLatin1+, and ISO-8859-x (where x is in the ranges [1,10] or [13,15]). See Appendix F for details [ISOLatin1+ (or Standard+)].

PS_COLOR_MODEL Determines whether PostScript output should use RGB, HSV, CMYK, or GRAY when specifying color [rgb]. Note if HSV is selected it does not apply to images which in that case uses RGB. When selecting GRAY, all colors will be converted to gray scale using YIQ (television) conversion.

PS_COMMENTS (static) If true we will issue comments in the PostScript file that explain the logic of operations. These are useful if you need to edit the file and make changes; otherwise you can set it to false which yields a somewhat slimmer PostScript file [false].

PS_IMAGE_COMPRESS Determines if PostScript images are compressed using the Run-Length Encoding scheme (rle), Lempel-Ziv-Welch compression (lzw), DEFLATE compression (deflate[,level]), or not at all (none) [deflate,5]. When specifying deflate, the compression level (1–9) may optionally be appended.

PS_LINE_CAP Determines how the ends of a line segment will be drawn. Choose among a *butt* cap (default) where there is no projection beyond the end of the path, a *round* cap where a semicircular arc with diameter equal to the line-width is drawn around the end points, and *square* cap where a half square of size equal to the line-width extends beyond the end of the path [butt].

PS_LINE_JOIN Determines what happens at kinks in line segments. Choose among a *miter* join where the outer edges of the strokes for the two segments are extended until they meet at an angle (as in a picture frame; if the angle is too acute, a bevel join is used instead, with threshold set by **PS_MITER_LIMIT**), *round* join where a circular arc is used to fill in the cracks at the kinks, and *bevel* join which is a miter join that is cut off so kinks are triangular in shape [miter].

PS_MEDIA Sets the physical format of the current plot paper [a4 (or letter)]. The following formats (and their widths and heights in points) are recognized (Additional site-specific formats may be specified in the gmt_custom_media.conf file in **\$GMT_SHAREDIR/conf** or **~/gmt**; see that file for details):

Media width height

- A0 2380 3368
- A1 1684 2380
- A2 1190 1684
- A3 842 1190
- A4 595 842

- A5 421 595
- A6 297 421
- A7 210 297
- A8 148 210
- A9 105 148
- A10 74 105
- B0 2836 4008
- B1 2004 2836
- B2 1418 2004
- B3 1002 1418
- B4 709 1002
- B5 501 709
- archA 648 864
- archB 864 1296
- archC 1296 1728
- archD 1728 2592
- archE 2592 3456
- flsa 612 936
- halfletter 396 612
- statement 396 612
- note 540 720
- letter 612 792
- legal 612 1008
- 11x17 792 1224
- tabloid 792 1224
- ledger 1224 792

For a completely custom format (e.g., for large format plotters) you may also specify WxH, where W and H are in points unless you append a unit to each dimension (**c**, **i**, **m** or **p** [Default]).

PS_MITER_LIMIT Sets the threshold angle in degrees (integer in range [0,180]) used for mitered joins only. When the angle between joining line segments is smaller than the threshold the corner will be bevelled instead of mitered. The default threshold is 35 degrees. Setting the threshold angle to 0 implies the PostScript default of about 11 degrees. Setting the threshold angle to 180 causes all joins to be beveled.

PS_PAGE_COLOR Sets the color of the imaging background, i.e., the paper [white].

PS_PAGE_ORIENTATION (* **-P**) Sets the orientation of the page. Choose portrait or landscape [landscape].

PS_SCALE_X Global x-scale (> 0) to apply to plot-coordinates before plotting. Normally used to shrink the entire output down to fit a specific height/width [1.0].

PS_SCALE_Y Global y-scale (> 0) to apply to plot-coordinates before plotting. Normally used to shrink the entire output down to fit a specific height/width [1.0].

PS_TRANSPARENCY Sets the transparency mode to use when preparing PS for rendering to PDF. Choose from Color, ColorBurn, ColorDodge, Darken, Difference, Exclusion, HardLight, Hue, Lighten, Luminosity, Multiply, Normal, Overlay, Saturation, SoftLight, and Screen [Normal].

TIME_EPOCH Specifies the value of the calendar and clock at the origin (zero point) of relative time units (see **TIME_UNIT**). It is a string of the form yyyy-mm-ddT[hh:mm:ss] (Gregorian) or yyyy-Www-ddT[hh:mm:ss] (ISO) Default is 1970-01-01T00:00:00, the origin of the UNIX time epoch.

TIME_INTERVAL_FRACTION Determines if partial intervals at the start and end of an axis should be annotated. If the range of the partial interval exceeds the specified fraction of the normal interval stride we will place the annotation centered on the partial interval [0.5].

TIME_IS_INTERVAL Used when input calendar data should be truncated and adjusted to the middle of the relevant interval. In the following discussion, the unit **u** can be one of these time units: (**y** year, **o** month, **u** ISO week, **d** day, **h** hour, **m** minute, and **s** second). **TIME_IS_INTERVAL** can have any of the following three values: (1) OFF [Default]. No adjustment, time is decoded as given. (2) **+nu**. Activate interval adjustment for input by truncate to previous whole number of *n* units and then center time on the following interval. (3) **-nu**. Same, but center time on the previous interval. For example, with **TIME_IS_INTERVAL** = +1o, an input data string like 1999-12 will be interpreted to mean 1999-12-15T12:00:00.0 (exactly middle of December), while if **TIME_IS_INTERVAL** = off then that date is interpreted to mean 1999-12-01T00:00:00.0 (start of December) [off].

TIME_LANGUAGE Language to use when plotting calendar items such as months and days. Select from:

- BR Brazilian Portuguese
- CN1 Simplified Chinese
- CN2 Traditional Chinese
- DE German
- DK Danish
- EH Basque
- ES Spanish
- FI Finnish
- FR French
- GR Greek
- HI Hawaiian
- HU Hungarian
- IE Irish
- IL Hebrew
- IS Icelandic

- IT Italian
- JP Japanese
- NL Dutch
- NO Norwegian
- PL Polish
- PT Portuguese
- RU Russian
- SE Swedish
- SG Scottish Gaelic
- TO Tongan
- TR Turkish
- UK British English
- US US English

If your language is not supported, please examine the `$GMT_SHAREDIR/time/us.d` file and make a similar file. Please submit it to the GMT Developers for official inclusion. Custom language files can be placed in directories `$GMT_SHAREDIR/time` or `~/gmt`. Note: Some of these languages may require you to also change the `PS_CHAR_ENCODING` setting.

TIME_SYSTEM Shorthand for a combination of **TIME_EPOCH** and **TIME_UNIT**, specifying which time epoch the relative time refers to and what the units are. Choose from one of the preset systems below (epoch and units are indicated):

JD -4713-11-25T12:00:00 d (Julian Date)

MJD 1858-11-17T00:00:00 d (Modified Julian Date)

J2000 2000-01-01T12:00:00 d (Astronomical time)

S1985 1985-01-01T00:00:00 s (Altimetric time)

UNIX 1970-01-01T00:00:00 s (UNIX time)

RD0001 0001-01-01T00:00:00 s

RATA 0000-12-31T00:00:00 d

This parameter is not stored in the `gmt.conf` file but is translated to the respective values of **TIME_EPOCH** and **TIME_UNIT**.

TIME_UNIT Specifies the units of relative time data since epoch (see **TIME_EPOCH**). Choose y (year - assumes all years are 365.2425 days), o (month - assumes all months are of equal length y/12), d (day), h (hour), m (minute), or s (second) [s].

TIME_WEEK_START When weeks are indicated on time axes, this parameter determines the first day of the week for Gregorian calendars. (The ISO weekly calendar always begins weeks with Monday.) [Monday (or Sunday)].

TIME_Y2K_OFFSET_YEAR When 2-digit years are used to represent 4-digit years (see various **FORMAT_DATES**), **TIME_Y2K_OFFSET_YEAR** gives the first year in a 100-year sequence. For example, if **TIME_Y2K_OFFSET_YEAR** is 1729, then numbers 29 through 99 correspond to 1729 through 1799, while numbers 00 through 28 correspond to 1800 through 1828. [1950].

1.7.3 See Also

gmt , *gmtdefaults* , *gmtcolors* , *gmtget* , *gmtset*

1.8 gmt

gmt - The Generic Mapping Tools data processing and display software package

1.8.1 Introduction

GMT is a collection of public-domain Unix tools that allows you to manipulate x,y and x,y,z data sets (filtering, trend fitting, gridding, projecting, etc.) and produce PostScript illustrations ranging from simple x-y plots, via contour maps, to artificially illuminated surfaces and 3-D perspective views in black/white or full color. Linear, log10, and power scaling is supported in addition to over 30 common map projections. The processing and display routines within GMT are completely general and will handle any (x,y) or (x,y,z) data as input.

1.8.2 Synopsis

gmt is the main program that can start any of the modules:

gmt module module-options

where module is the name of a GMT module and the options are those that pertain to that particular module. If no module is given then several other options are available:

--help List and description of GMT modules.

--version Print version and exit.

--show-datadir Show data directory and exit.

--show-bindir Show directory of executables and exit.

= Check if that module exist and if so the program will exit with status of 0; otherwise the status of exit will be non-zero.

1.8.3 Command-line completion

GMT provides basic command-line completion (tab completion) for bash. The completion rules are either installed in `/etc/bash_completion.d/gmt` or `<prefix>/share/tools/gmt_completion.bash`. Depending on the distribution, you may still need to source the gmt completion file from `~/.bash_completion` or `~/.bashrc`. For more information see Section *command-line-completion* in the CookBook.

1.8.4 GMT Overview

The following is a summary of all the programs supplied with GMT and a very short description of their purpose. Detailed information about each program can be found in the separate manual pages.

Continued on next page

Table 1.1 – continued from previous page

blockmean blockmedian blockmode filter1d fitcircle gmt2kml gmtconnect gmtconvert gmtdefaults gmtget gmtinfo gmtmath gmtselect gmtset gmtspatial gmsimplify gmtvector gmtwhich grd2cpt grd2rgb grd2xyz grdblend grdclip grdcontour grdcut grdedit grdffft grdfilter grdgradient grdhisteq grdimage grdinfo grdlandmask grdmask grdmath grdpaste grdproject grdreformat grdraster grdsample grdtrack grdtrend grdvector grdview grdvolume greenspline kml2gmt makecpt mapproject	
Continued on next page	

Table 1.1 – continued from previous page

<p> nearneighbor project ps2raster psbasemap psclip pscoast pscontour pshistogram psimage pslegend psmask psrose psscale pstext pswiggle psxy psxyz sample1d spectrum1d splitxyz surface trend1d trend2d triangulate xyz2grd </p>	
gshhg	Supplement <i>gshhg</i> :
img2grd	Supplement <i>img</i> :
pscoupe psmeca pspolar psvelo	Supplement <i>meca</i> :
mgd77convert mgd77info mgd77list mgd77magref mgd77manage mgd77path mgd77sniffer mgd77track	Supplement <i>mgd77</i> :
gmtgravmag3d gravfft grdgravmag3d redpol	Supplement <i>potential</i> :
	Supplement <i>segy</i> :
Continued on next page	

Table 1.1 – continued from previous page

<p>pssegy pssegyz segy2grd</p>	Supplement <i>sph</i> :
<p>sphdistance sphinterpolate sphtriangulate</p>	Supplement <i>spotter</i> :
<p>backtracker grdpmodeler grdrotater grdspotter hotspotter originator rotconverter</p>	Supplement <i>x2sys</i> :
<p>x2sys_binlist x2sys_cross x2sys_datalist x2sys_get x2sys_init x2sys_list x2sys_merge x2sys_put x2sys_report x2sys_solve</p>	

1.8.5 Custom Modules

The **gmt** program can also load custom modules from shared libraries built as specified in the GMT API documentation. This way your modules can benefit from the GMT infrastructure and extend GMT in specific ways.

1.8.6 The Common GMT Options

-B[pls]parameters -Jparameters -Jz/Zparameters -K -O -P -Rwest/east/south/north[/zmin/zmax][r]
-U[just/dx/dy][c]label -V[level] -X[alclfr][x-shift[u]] -Y[alclfr][y-shift[u]] -
acol=name[...] -b[ilo][ncol][type][w][+L+B] -f[ilo]colinfo -g[a]x|y|d|X|Y|D|col]z[+|
]gap[u] -h[ilo][n][+c][+d][+rremark][+rtitle] -i[cols][l][sscale][offset][,...] -o[cols][,...] -
p[x|y|z]azim|elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] -r -s[cols][alr] -t[transp] -:[ilo]

1.8.7 Description

These are all the common GMT options that remain the same for all GMT programs. No space between the option flag and the associated arguments.

-B[pls]parameters Set map Frame and Axes parameters. The Frame parameters are specified by

-B[axes][+b][+gfill][+olon/lat][+ttitle]

where *axes* selects which axes to plot. By default, all 4 map boundaries (or plot axes) are plotted (named **W**, **E**, **S**, **N**). To customize, append the codes for those you want (e.g., **WSn**). Upper case means plot and annotate while lower case just plots the specified axes. If a 3-D basemap is selected with **-p** and **-Jz**, append **Z** or **z** to control the appearance of the vertical axis. By default a single vertical axes will be plotted at the most suitable map corner. Override the default by appending any combination of corner ids **1234**, where **1** represents the lower left corner and the order goes counter-clockwise. Append **+b** to draw the outline of the 3-D cube defined by **-R**; this modifier is also needed to display gridlines in the x-z, y-z planes. Note that for 3-D views the title, if given, will be suppressed. You can paint the interior of the canvas with **+gfill**. Optionally append **+oplon/plat** to draw oblique gridlines about specified pole [regular gridlines]. Ignored if gridlines are not requested (below) and disallowed for the oblique Mercator projection. To add a plot title (**+ttitle**). The Frame setting is optional but can be invoked once to override the above defaults.

The Axes parameters are specified by

-B[pls][x|y|z]intervals[+llabel][+pprefix][+uunit]

but you may also split this into two separate invocations for clarity, i.e.,

- **-B[pls][x|y|z][+llabel][+pprefix][+uunit]**
- **-B[pls][x|y|z]intervals**

The first optional flag following **-B** selects **p** (rimary) [Default] or **s** (econdary) axes information (mostly used for time axes annotations). The **[x|x|z]** flags specify which axes you are providing information for. If none are given then we default to **xy**. If you wish to give different annotation intervals or labels for the various axes then you must repeat the **B** option for each axis (If a 3-D basemap is selected with **-p** and **-Jz**, use **-Bz** to give settings for the vertical axis.). To add a label to an axis, just append **+llabel** (Cartesian projections only). If the axis annotation should have a leading text prefix (e.g., dollar sign for those plots of your net worth) you can append **+pprefix**. For geographic maps the addition of degree symbols, etc. is automatic (and controlled by the GMT default setting *FORMAT_GEO_MAP*). However, for other plots you can add specific units by adding **+uunit**. If any of these text strings contain spaces or special UNIX characters you will need to enclose them in quotes. The *intervals* specification is a concatenated string made up of substrings of the form

[a|f|g]stride[+-phase][u].

The leading **a** is used to specify the annotation and major tick spacing [Default], **f** for minor tick spacing, and **g** for gridline spacing. *stride* is the desired stride interval. The optional *phase* shifts the annotation interval by that amount (positive or negative). The optional *unit* indicates the unit of the *stride* and can be any of

- **Y** (year, plot with 4 digits)
- **y** (year, plot with 2 digits)
- **O** (month, plot using *FORMAT_DATE_MAP*)
- **o** (month, plot with 2 digits)
- **U** (ISO week, plot using *FORMAT_DATE_MAP*)
- **u** (ISO week, plot using 2 digits)
- **r** (Gregorian week, 7-day stride from start of week *TIME_WEEK_START*)
- **K** (ISO weekday, plot name of day)
- **D** (date, plot using *FORMAT_DATE_MAP*)

- **d** (day, plot day of month 0-31 or year 1-366, via *FORMAT_DATE_MAP*)
- **R** (day, same as **d**, aligned with *TIME_WEEK_START*)
- **H** (hour, plot using *FORMAT_CLOCK_MAP*)
- **h** (hour, plot with 2 digits)
- **M** (minute, plot using *FORMAT_CLOCK_MAP*)
- **m** (minute, plot with 2 digits)
- **S** (second, plot using *FORMAT_CLOCK_MAP*)
- **s** (second, plot with 2 digits).

Note for geographic axes **m** and **s** instead mean arc minutes and arc seconds. All entities that are language-specific are under control by *TIME_LANGUAGE*. Alternatively, for linear maps, we can omit *stride*, thus setting *xinfo*, *yinfo*, or *zinfo* to **a** plots annotations at automatically determined intervals,

- **ag** plots both annotations and grid lines with the same spacing,
- **afg** adds suitable minor tick intervals,
- **g** plots grid lines with the same interval as if **-Bf** was used.

For custom annotations and intervals, let *intervals* be given as *cintfile*, where *intfile* contains any number of records with *coord type [label]*. Here, *type* is one or more letters from **ali**, **f**, and **g**. For **ali** you must supply a *label* that will be plotted at the *coord* location. For non-geographical projections: Give negative scale (in **-Jx**) or axis length (in **-JX**) to change the direction of increasing coordinates (i.e., to make the y-axis positive down). For log10 axes: Annotations can be specified in one of three ways:

1. *stride* can be 1, 2, 3, or *-n*. Annotations will then occur at 1, 1-2-5, or 1-2-3-4-...-9, respectively; for *-n* we annotate every *n*^t magnitude. This option can also be used for the frame and grid intervals.
2. An **l** is appended to the *tickinfo* string. Then, log10 of the tick value is plotted at every integer log10 value.
3. A **p** is appended to the *tickinfo* string. Then, annotations appear as 10 raised to log10 of the tick value.

For power axes: Annotations can be specified in one of two ways:

1. *stride* sets the regular annotation interval.
2. A **p** is appended to the *tickinfo* string. Then, the annotation interval is expected to be in transformed units, but the annotation value will be plotted as untransformed units. E.g., if *stride* = 1 and *power* = 0.5 (i.e., sqrt), then equidistant annotations labeled 1-4-9... will appear.

These GMT parameters can affect the appearance of the map boundary: *MAP_ANNOT_MIN_ANGLE*, *MAP_ANNOT_MIN_SPACING*, *FONT_ANNOT_PRIMARY*, *FONT_ANNOT_SECONDARY*, *MAP_ANNOT_OFFSET_PRIMARY*, *MAP_ANNOT_OFFSET_SECONDARY*, *MAP_ANNOT_ORTHO*, *MAP_FRAME_AXES*, *MAP_DEFAULT_PEN*, *MAP_FRAME_TYPE*, *FORMAT_GEO_MAP*, *MAP_FRAME_PEN*, *MAP_FRAME_WIDTH*, *MAP_GRID_CROSS_SIZE_PRIMARY*, *MAP_GRID_PEN_PRIMARY*, *MAP_GRID_CROSS_SIZE_SECONDARY*, *MAP_GRID_PEN_SECONDARY*, *FONT_TITLE*,

FONT_LABEL, *MAP_LINE_STEP*, *MAP_ANNOT_OBLIQUE*, *FORMAT_CLOCK_MAP*, *FORMAT_DATE_MAP*, *FORMAT_TIME_PRIMARY_MAP*, *FORMAT_TIME_SECONDARY_MAP*, *TIME_LANGUAGE*, *TIME_WEEK_START*, *MAP_TICK_LENGTH_PRIMARY*, and *MAP_TICK_PEN_PRIMARY*; see the *gmt.conf* man page for details.

-J*parameters*

Select map projection. The following character determines the projection. If the character is upper case then the argument(s) supplied as *scale*(s) is interpreted to be the map width (or axis lengths), else the *scale* argument(s) is the map scale (see its definition for each projection). UNIT is cm, inch, or point, depending on the *PROJ_LENGTH_UNIT* setting in **gmt.conf**, but this can be overridden on the command line by appending **c**, **i**, or **p** to the *scale* or *width* values. Append **h**, **+**, or **-** to the given *width* if you instead want to set map height, the maximum dimension, or the minimum dimension, respectively [Default is **w** for width]. In case the central meridian is an optional parameter and it is being omitted, then the center of the longitude range given by the **-R** option is used. The default standard parallel is the equator. The ellipsoid used in the map projections is user-definable by editing the **gmt.conf** file in your home directory. 73 commonly used ellipsoids and spheroids are currently supported, and users may also specify their own custom ellipsoid parameters [Default is WGS-84]. Several GMT parameters can affect the projection: *PROJ_ELLIPSOID*, *GMT_INTERPOLANT*, *PROJ_SCALE_FACTOR*, and *PROJ_LENGTH_UNIT*; see the *gmt.conf* man page for details. Choose one of the following projections (The **E** or **C** after projection names stands for Equal-Area and Conformal, respectively):

CYLINDRICAL PROJECTIONS:

-Jclon0/lat0/scale or **-JC***lon0/lat0/width* (Cassini).

Give projection center *lon0/lat0* and *scale* (**1:xxxx** or UNIT/degree).

-Jcyl_stere*[/lon0[/lat0/]]scale* or **-JCyl_stere***[/lon0[/lat0/]]width* (Cylindrical Stereographic).

Give central meridian *lon0* (optional), standard parallel *lat0* (optional), and *scale* along parallel (**1:xxxx** or UNIT/degree). The standard parallel is typically one of these (but can be any value):

- 66.159467 - Miller's modified Gall
- 55 - Kamenetskiy's First
- 45 - Gall's Stereographic
- 30 - Bolshoi Sovietskii Atlas Mira or Kamenetskiy's Second
- 0 - Braun's Cylindrical

-Jj*[/lon0/]scale* or **-JJ***[/lon0/]width* (Miller Cylindrical Projection).

Give the central meridian *lon0* (optional) and *scale* (**1:xxxx** or UNIT/degree).

-Jm*[/lon0[/lat0/]]scale* or **-JM***[/lon0[/lat0/]]width* (Mercator [**C**])

Give central meridian *lon0* (optional), standard parallel *lat0* (optional), and *scale* along parallel (**1:xxxx** or UNIT/degree).

-Jop*parameters* (Oblique Mercator [**C**]).

Typically used with **-RLLx/LLy/URx/URy**r or with projected coordinates. Specify one of:

-Jo*[a]lon0/lat0/azimuth/scale* or **-JO***[a]lon0/lat0/azimuth/width* Set projection center *lon0/lat0*, *azimuth* of oblique equator, and *scale*.

-Jo[b]lon0/lat0/lon1/lat1/scale or **-JO[b]lon0/lat0/lon1/lat1/scale** Set projection center *lon0/lat0*, another point on the oblique equator *lon1/lat1*, and *scale*.

-Joclon0/lat0/lonp/latp/scale or **-JOclon0/lat0/lonp/latp/scale** Set projection center *lon0/lat0*, pole of oblique projection *lonp/latp*, and *scale*. Give *scale* along oblique equator (**1:xxxx** or UNIT/degree).

-Jq[lon0/[lat0/]]scale or **-JQ[lon0/[lat0/]]width** (Cylindrical Equidistant).

Give the central meridian *lon0* (optional), standard parallel *lat0* (optional), and *scale* (**1:xxxx** or UNIT/degree). The standard parallel is typically one of these (but can be any value):

- 61.7 - Grafarend and Niermann, minimum linear distortion
- 50.5 - Ronald Miller Equirectangular
- 43.5 - Ronald Miller, minimum continental distortion
- 42 - Grafarend and Niermann
- 37.5 - Ronald Miller, minimum overall distortion
- 0 - Plate Carree, Simple Cylindrical, Plain/Plane Chart

-Jtlon0/[lat0/]scale or **-JTlon0/[lat0/]width** (Transverse Mercator [**C**])

Give the central meridian *lon0*, central parallel *lat0* (optional), and *scale* (**1:xxxx** or UNIT/degree).

-Juzone/scale or **-JUzone/width** (UTM - Universal Transverse Mercator [**C**]).

Give the UTM zone (A,B,1-60[C-X],Y,Z) and *scale* (**1:xxxx** or UNIT/degree). Zones: If C-X not given, prepend - or + to enforce southern or northern hemisphere conventions [northern if south > 0].

-Jy[lon0/[lat0/]]scale or **-JY[lon0/[lat0/]]width** (Cylindrical Equal-Area [**E**]).

Give the central meridian *lon0* (optional), standard parallel *lat0* (optional), and *scale* (**1:xxxx** or UNIT/degree). The standard parallel is typically one of these (but can be any value):

- 50 - Balthasart
- 45 - Gall-Peters
- 37.0666 - Caster
- 37.4 - Trystan Edwards
- 37.5 - Hobo-Dyer
- 30 - Behrman
- 0 - Lambert (default)

CONIC PROJECTIONS:

-Jblon0/lat0/lat1/lat2/scale or **-JBlon0/lat0/lat1/lat2/width** (Albers [**E**]). Give projection center *lon0/lat0*, two standard parallels *lat1/lat2*, and *scale* (**1:xxxx** or UNIT/degree).

- Jdlon0/lat0/lat1/lat2/scale** or **-JDlon0/lat0/lat1/lat2/width** (**Conic Equidistant**) Give projection center *lon0/lat0*, two standard parallels *lat1/lat2*, and *scale* (**1:xxxx** or UNIT/degree).
- Jllon0/lat0/lat1/lat2/scale** or **-JLlon0/lat0/lat1/lat2/width** (**Lambert [C]**) Give origin *lon0/lat0*, two standard parallels *lat1/lat2*, and *scale* along these (**1:xxxx** or UNIT/degree).
- Jpoly/[lon0/[lat0/]]scale** or **-JPoly/[lon0/[lat0/]]width** (**(American) Polyconic**). Give the central meridian *lon0* (optional), reference parallel *lat0* (optional, default = equator), and *scale* along central meridian (**1:xxxx** or UNIT/degree).

AZIMUTHAL PROJECTIONS:

Except for polar aspects, **-Rw/e/s/n** will be reset to **-Rg**. Use **-R<...>r** for smaller regions.

- Jalon0/lat0/[horizon]/scale** or **-JAlon0/lat0/[horizon]/width** (**Lambert [E]**). *lon0/lat0* specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, <= 180, default 90). Give *scale* as **1:xxxx** or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.
- Jelon0/lat0/[horizon]/scale** or **-JElon0/lat0/[horizon]/width** (**Azimuthal Equidistant**). *lon0/lat0* specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, <= 180, default 180). Give *scale* as **1:xxxx** or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.
- Jflon0/lat0/[horizon]/scale** or **-JFlon0/lat0/[horizon]/width** (**Gnomonic**). *lon0/lat0* specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, < 90, default 60). Give *scale* as **1:xxxx** or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.
- Jglon0/lat0/[horizon]/scale** or **-JGlon0/lat0/[horizon]/width** (**Orthographic**). *lon0/lat0* specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, <= 90, default 90). Give *scale* as **1:xxxx** or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.
- Jglon0/lat0/altitude/azimuth/tilt/twist/Width/Height/scale** or **-JGlon0/lat0/altitude/azimuth/tilt/twist/Width/Height/scale** (**Oblique Mercator**). *lon0/lat0* specifies the projection center. *altitude* is the height (in km) of the viewpoint above local sea level. If *altitude* is less than 10, then it is the distance from the center of the earth to the viewpoint in earth radii. If *altitude* has a suffix **r** then it is the radius from the center of the earth in kilometers. *azimuth* is measured to the east of north of view. *tilt* is the upward tilt of the plane of projection. If *tilt* is negative, then the viewpoint is centered on the horizon. Further, specify the clockwise *twist*, *Width*, and *Height* of the viewpoint in degrees. Give *scale* as **1:xxxx** or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.
- Jslon0/lat0/[horizon]/scale** or **-JSlon0/lat0/[horizon]/width** (**General Stereographic [C]**). *lon0/lat0* specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, < 180, default 90). Give *scale* as **1:xxxx** (true at pole) or *lat0/1:xxxx* (true at standard parallel *lat*) or *radius/lat* (*radius* in UNIT from origin to the oblique latitude *lat*). Note if **1:xxxx** is used then to specify *horizon* you must also specify the *lat* as +-90 to avoid ambiguity.

MISCELLANEOUS PROJECTIONS:

- Jh[lon0]/scale** or **-JH[lon0]/width** (**Hammer [E]**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).

- Ji**[*lon0*]/*scale* or -**Jl**[*lon0*]/*width* (**Sinusoidal [E]**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jkf**[*lon0*]/*scale* or -**JKf**[*lon0*]/*width* (**Eckert IV [E]**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jk[s]**[*lon0*]/*scale* or -**JK[s]**[*lon0*]/*width* (**Eckert VI [E]**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jn**[*lon0*]/*scale* or -**JN**[*lon0*]/*width* (**Robinson**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jr**[*lon0*]/*scale* -**JR**[*lon0*]/*width* (**Winkel Tripel**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jv**[*lon0*]/*scale* or -**JV**[*lon0*]/*width* (**Van der Grinten**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).
- Jw**[*lon0*]/*scale* or -**JW**[*lon0*]/*width* (**Mollweide [E]**). Give the central meridian *lon0* (optional) and *scale* along equator (**1:xxxx** or UNIT/degree).

NON-GEOGRAPHICAL PROJECTIONS:

-**Jp**[*a*]*scale*[/*origin*][*r|z*] or -**JP**[*a*]*width*[/*origin*][*r|z*] (Polar coordinates (theta,r))

Optionally insert **a** after -**Jp** [or -**JP**] for azimuths CW from North instead of directions CCW from East [Default]. Optionally append *origin* in degrees to indicate an angular offset [0]). Finally, append **r** if *r* is elevations in degrees (requires *s* >= 0 and *n* <= 90) or **z** if you want to annotate depth rather than radius [Default]. Give *scale* in UNIT/*r*-unit.

-**Jxx**-*scale*[/*y-scale*] or -**JX***width*[/*height*] (Linear, log, and power scaling)

Give *x-scale* (**1:xxxx** or UNIT/*x*-unit) and/or *y-scale* (**1:xxxx** or UNIT/*y*-unit); or specify *width* and/or *height* in UNIT. *y-scale*=*x-scale* if not specified separately and using **1:xxxx** implies that *x*-unit and *y*-unit are in meters. Use negative scale(s) to reverse the direction of an axis (e.g., to have *y* be positive down). Set *height* or *width* to 0 to have it recomputed based on the implied scale of the other axis. Optionally, append to *x-scale*, *y-scale*, *width* or *height* one of the following:

d Data are geographical coordinates (in degrees).

l Take log10 of values before scaling.

ppower Raise values to *power* before scaling.

t Input coordinates are time relative to *TIME_EPOCH*.

T Input coordinates are absolute time.

Default axis lengths (see **gmt.conf**) can be invoked using -**JXh** (for landscape); -**JXv** (for portrait) will swap the *x*- and *y*-axis lengths. The default unit for this installation is either cm or inch, as defined in the file **share/gmt.conf**. However, you may change this by editing your **gmt.conf** file(s).

When -**J** is used without any further arguments, or just with the projection type, the arguments of the last used -**J**, or the last used -**J** with that projection type, will be used.

-**Jz**/*Zparameters* Set *z*-axis scaling; same syntax as -**Jx**.

-**K** More PostScript code will be appended later [Default terminates the plot system]. Required for all but the last plot command when building multi-layer plots.

- O** Selects Overlay plot mode [Default initializes a new plot system]. Required for all but the first plot command when building multi-layer plots.
- P** Select “Portrait” plot orientation [Default is “Landscape”; see **gmt.conf** or **gmtset** to change the **PS_PAGE_ORIENTATION** parameter, or supply **--PS_PAGE_ORIENTATION=orientation** on the command line].
- R**[*unit*]**xmin/xmax/ymin/ymax**[**r**] *xmin*, *xmax*, *ymin*, and *ymax* specify the region of interest. For geographic regions, these limits correspond to *west*, *east*, *south*, and *north* and you may specify them in decimal degrees or in [+]*-dd:mm[:ss.xxx]*[**W**|**E**|**S**|**N**] format. Append **r** if lower left and upper right map coordinates are given instead of *west/east/south/north*. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. When **-R** is used without any further arguments, the values from the last use of **-R** in a previous GMT command will be used. For calendar time coordinates you may either give (a) relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **-JX|x**), or (b) absolute time of the form [*date*]**T**[*clock*] (append **T** to **-JX|x**). At least one of *date* and *clock* must be present; the **T** is always required. The *date* string must be of the form [-]*yyyy[-mm[-dd]]* (Gregorian calendar) or *yyyy[-Www[-d]]* (ISO week calendar), while the *clock* string must be of the form *hh:mm:ss[.xxx]*. The use of delimiters and their type and positions must be exactly as indicated (however, input, output and plot formats are customizable; see **gmt.conf**). You can also use Cartesian projected coordinates compatible with the chosen projection. Give them with a leading length *unit*, (e.g., **-Rk-200/200/-300/300** for a 400 by 600 km rectangular area centered on the projection center (0, 0). These coordinates are internally converted to the corresponding geographic (longitude, latitude) coordinates for the lower left and upper right corners. This form is convenient when you want to specify a region directly in the projected units (e.g., UTM meters).

In case of perspective view **p**, a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **Jz** option, not when using only the **p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

- U**[*just/dx/dy/*][*clabel*] Draw Unix System time stamp on plot. By adding *just/dx/dy/*, the user may specify the justification of the stamp and where the stamp should fall on the page relative to lower left corner of the plot. For example, **BL/0/0** will align the lower left corner of the time stamp with the lower left corner of the plot. Optionally, append a *label*, or **c** (which will plot the command string.). The GMT parameters **MAP_LOGO**, **MAP_LOGO_POS**, and **FORMAT_TIME_STAMP** can affect the appearance; see the **gmt.conf** man page for details. The time string will be in the locale set by the environment variable **TZ** (generally local time).
- V**[*level*] Select verbose mode, which will send progress reports to *stderr*. Choose among 6 levels of verbosity; each level adds mode messages: **q** - Complete silence, not even fatal error messages are produced. **n** - Normal verbosity: produce only fatal error messages. **c** - Produce also compatibility warnings (same as when **-V** is omitted). **v** - Produce also warnings and progress messages (same as **-V** only). **l** - Produce also detailed progress messages. **d** - Produce also debugging messages.
- X**[**alcflr**][*x-shift*]**[u]**
- Y**[**alcflr**][*y-shift*]**[u]** Shift plot origin relative to the current origin by (*x-shift*,*y-shift*) and optionally append the length unit (**c**, **i**, or **p**). You can prepend **a** to shift the origin back to the original position after plotting, prepend **c** to center the plot on the center of the paper (optionally add shift), prepend **f** to shift the origin relative to the fixed lower left corner of the page, or prepend **r** [Default] to move the origin relative to its current location. If **-O** is used then the default (*x-shift*,*y-shift*) is (r0), otherwise it is (r1i). When **-X** or **-Y** are used without any further arguments, the values from the last use of that option in a previous GMT command will be used.

- a[*col=*]*name*[...]** Control how aspatial data are handled in GMT during input and output. Reading OGR/GMT-formatted files. To assign certain aspatial data items to GMT data columns, give one or more comma-separated associations *col=name*, where *name* is the name of an aspatial attribute field in a OGR/GMT file and whose value we wish to as data input for column *col*. In addition, to assign an aspatial value to non-column data, you may specify *col* as **D** for *distance*, **G** for *fill*, **I** for *ID*, **L** for *label*, **T** for *text*, **W** for *pen*, and **Z** for *value* [e.g., used to look-up color via a CPT]. If you skip the leading “*col=*” in the argument then we supply (and automatically increment) a column value starting at 2. Write OGR/GMT-formatted files. Give one or more comma-separated associations *col=name[:type]*, To write OGR/GMT-formatted files, give one or more comma-separated associations *col=name[:type]*, with an optional data type from DOUBLE, FLOAT, INTEGER, CHAR, STRING, DATETIME, or LOGICAL [DOUBLE]. To extract information from GMT multisegment headers encoded in the **-Ddistance**, **-Gfill**, **-IID**, **-Llabel**, **-Ttext**, **-Wpen**, or **-Zvalue** settings, specify *COL* as **D**, **G**, **I**, **L**, **T**, **W** or **Z**, respectively; type will be set automatically. Finally, you *must* append **+ggeometry**, where *geometry* is either POINT, LINE, or POLY. Optionally, prepend **M** for multi-versions of these geometries. To force the clipping of features crossing the Dateline, use upper-case **+G** instead. See GMT Appendix Q for details of the OGR/GMT file format.
- bi[*ncols*][*type*]** Select binary input. Append one or more comma-separated combinations of *ncols/type*, where *ncols* is the actual number of data columns and *type* must be one of **c** (int8_t, aka char), **u** (uint8_t, aka unsigned char), **h** (int16_t, 2-byte signed int), **H** (uint16_t, 2-byte unsigned int), **i** (int32_t, 4-byte signed int), **I** ((capital i) uint32_t, 4-byte unsigned int), **l** ((lower case el) int64_t, 8-byte signed int), **L** (uint64_t, 8-byte unsigned int), **f** (4-byte single-precision float), and **d** (8-byte double-precision float). In addition, use **x** to skip *ncols* bytes anywhere in the record. Append **w** to any item to force byte-swapping. Alternatively, append **+LIB** to indicate that the entire data file should be read as little- or big-endian, respectively. The total number of *ncols* may exceed the columns actually needed by the program. If no *ncols* is specified we assume that *type* applies to all columns and that *ncols* is implied by the expectation of the program. If the input file is netCDF, no **-b** is needed; simply append *?var1/var2/...* to the filename to specify the variables to be read.
- bo[*ncols*][*type*]** Select binary output. Append one or more comma-separated combinations of *ncols/type*, where *ncols* is the actual number of data columns and *type* must be one of **c**, **u**, **h**, **H**, **i**, **I** (capital i), **l** (lower case elle), **L**, **f**, and **d** (see **-bi**). Append **w** or **+LIB** for byte-swapping. *ncols* is the number of each item in the binary file. If no *ncols* is specified we assume that *type* applies to all columns and that *ncols* is implied by the default output of the program. NetCDF file output is not supported.
- ccopies** Specify number of plot copies. [Default is 1]. When used without argument, use the same number of copies and specified in the last **-c** in a previous GMT command.
- :[i]lo** Swap 1st and 2nd column on input and/or output [Default is no swapping]. Append **i** to select input only or **o** to select output only. [Default affects both]. This option is typically used to handle (latitude, longitude) files; see also **-i[cols][l][sscale][offset][,...]**.
- f[i]lo***colinfo* Specify the data types of input and/or output columns (time or geographical data). Specify **i** or **o** to make this apply only to input or output [Default applies to both]. Give one or more columns (or column ranges) separated by commas, or use **-f** multiple times. Append **T** (absolute calendar time), **t** (relative time in chosen *TIME_UNIT* since *TIME_EPOCH*), **x** (longitude), **y** (latitude), **p[*unit*]** (projected x,y map coordinates in given unit [meter]) or **f** (floating point) to each column or column range item. Shorthand **-f[i]lg** means **-f[i]l0x,1y** (geographic coordinates).
- g[a]x|y|d|X|Y|D|[*col*]z[+|-]*gap*[*u*]** Examine the spacing between consecutive data points in order to impose breaks in the line. Append **x|X** or **y|Y** to define a gap when there is a large enough change in the x or y coordinates, respectively, or **d|D** for distance gaps; use upper case to calculate gaps

from projected coordinates. For gap-testing on other columns use *[col]z*; if *col* is not prepended the it defaults to 2 (i.e., 3rd column). Append *[-+]*gap** and optionally a unit *u*. Regarding optional signs: -ve means previous minus current column value must exceed *gap* to be a gap, +ve means current minus previous column value must exceed *gap*, and no sign means the absolute value of the difference must exceed *gap*. For geographic data (*x|y|d*), the unit *u* may be arc **d**egree, **m**inute, or second, or meter [Default], **f**oot, **k**ilometer, **M**ile, **n**autical mile, or **s**urvey foot. For projected data (*X|Y|D*), choose from **i**nch, **c**entimeter, or **p**oint [Default unit set by *PROJ_LENGTH_UNIT*]. Note: For *x|y|z* with time data the unit is instead controlled by *TIME_UNIT*. Repeat the option to specify multiple criteria, of which any can be met to produce a line break. Issue an additional **-ga** to indicate that all criteria must be met instead.

-h*[ilo][n][+c][+d][+rremark][+ttitle]* Input file(s) has header record(s). If used, the default number of header records is *IO_N_HEADER_RECS* [1]. Use **-hi** if only input data should have header records [Default will write out header records if the input data have them]. Blank lines and lines starting with # are always skipped. For output you may request additional headers to be written via the option modifiers, and use **+d** to remove existing header records. Append **+c** to issue a header comment with column names to the output [none]. Append **+r** to add a *remark* comment to the output [none]. Append **+t** to add a *title* comment to the output [none]. These optional strings may contain *n* to indicate line-breaks). If used with native binary data we interpret *n* to instead mean the number of bytes to skip on input or pad on output.

-i*cols***[l][sscale][offset][,...]** Select specific data columns for input, in arbitrary order. Columns not listed will be skipped. Give columns (or column ranges) separated by commas [Default reads all columns in order, starting with the first column (0)]. To each column, optionally add any of the following, in this order: **l** takes the **log10** of the input values first; *sscale*, subsequently multiplies by a given scale factor; *offset*, finally adds a given offset.

-n**[blc|lln][+a][+bBC][+c][+tthreshold]** Select grid interpolation mode by adding **b** for B-spline smoothing, **c** for bicubic interpolation, **l** for bilinear interpolation, or **n** for nearest-neighbor value (for example to plot categorical data). Optionally, append **+a** to switch off antialiasing (where supported). Append **+bBC** to override the boundary conditions used, adding **g** for geographic, **p** for periodic, or **n** for natural boundary conditions. For the latter two you may append **x** or **y** to specify just one direction, otherwise both are assumed. Append **+c** to clip the interpolated grid to input *z*-min/max [Default may exceed limits]. Add append **+tthreshold** to control how close to nodes with NaNs the interpolation will go. A *threshold* of 1.0 requires all (4 or 16) nodes involved in interpolation to be non-NaN. 0.5 will interpolate about half way from a non-NaN value; 0.1 will go about 90% of the way, etc. [Default is bicubic interpolation with antialiasing and a threshold of 0.5, using geographic (if grid is known to be geographic) or natural boundary conditions].

-o*cols***[,...]** Select specific data columns for output, in arbitrary order. Columns not listed will be skipped. Give columns (or column ranges) separated by commas [Default writes all columns in order].

-p*[x|y|z]azim/elev[*zlevel*][+wlon0/lat0/*z0*][+vx0/y0]* Selects perspective view and sets the azimuth and elevation of the viewpoint [180/90]. When **-p** is used in consort with **-Jz** or **-JZ**, a third value can be appended which indicates at which *z*-level all 2D material, like the plot frame, is plotted (in perspective). [Default is at the bottom of the *z*-axis]. Use **-px** or **-py** to plot against the “wall” *x* = level or *y* = level (default is on the horizontal plane, which is the same as using **-pz**). For frames used for animation, you may want to append **+** to fix the center of your data domain (or specify a particular world coordinate point with **+wlon0/lat0/*z***) which will project to the center of your page size (or specify the coordinates of the projected view point with **+vx0/y0**). When **-p** is used without any further arguments, the values from the last use of **-p** in a previous GMT command will be used.

-r Force pixel node registration [Default is gridline registration]. (Node registrations are defined in Sec-

tion *grid-registration* of the GMT Technical Reference and Cookbook.)

- s[cols][alr]** Suppress output for records whose *z*-value equals NaN [Default outputs all records]. Append **a** to skip records where at least one field equal NaN. Append **r** to reverse the suppression, i.e., only output the records whose *z*-value equals NaN. Alternatively, indicate a comma-separated list of all columns or column ranges to consider for this NaN test.
- t[transp]** Set PDF transparency level for an overlay, in 0-100 percent range. [Default is 0, i.e., opaque].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

Specifying Color

color The *color* of lines, areas and patterns can be specified by a valid color name; by a gray shade (in the range 0-255); by a decimal color code (r/g/b, each in range 0-255; h-s-v, ranges 0-360, 0-1, 0-1; or c/m/y/k, each in range 0-1); or by a hexadecimal color code (#rrggbb, as used in HTML). For PDF transparency, append *@transparency* in the 0-100 percent range [0 or opaque]. See the **gmtcolors** manpage for more information and a full list of color names. See GMT Cookbook & Technical Reference Chapter 4 for more information.

Specifying Fill

fill The attribute *fill* specifies the solid shade or solid *color* (see [Specifying Color](#) above) or the pattern used for filling polygons. Patterns are specified as **pdpi/pattern**, where *pattern* gives the number of the built-in pattern (1-90) *or* the name of a Sun 1-, 8-, or 24-bit raster file. The *dpi* sets the resolution of the image. For 1-bit rasters: use **Pdpi/pattern** for inverse video, or append **:Fcolor[B[color]]** to specify fore- and background colors (use *color* = - for transparency). See GMT Cookbook & Technical Reference Appendix E for information on individual patterns.

Specifying Fonts

font The attributes of text fonts as defined by *font* is a comma delimited list of *size*, *fonttype* and *fill*, each of which is optional. *size* is the font size (usually in points) but **c** or **i** can be added to indicate other units. *fonttype* is the name (case sensitive!) of the font or its equivalent numerical ID (e.g., Helvetica-Bold or 1). *fill* specifies the gray shade, color or pattern of the text (see [Specifying Fill](#) above). Optionally, you may append *=pen* to the *fill* value in order to draw the text outline; if used you may optionally skip the text fill by setting it to -. If any of the attributes is omitted their default or previous setting will be retained.

The 35 available fonts are:

0. Helvetica

1. Helvetica-Bold
2. Helvetica-Oblique
3. Helvetica-BoldOblique
4. Times-Roman
5. Times-Bold
6. Times-Italic
7. Times-BoldItalic
8. Courier
9. Courier-Bold
10. Courier-Oblique
11. Courier-BoldOblique
12. Symbol
13. AvantGarde-Book
14. AvantGarde-BookOblique
15. AvantGarde-Demi
16. AvantGarde-DemiOblique
17. Bookman-Demi
18. Bookman-DemiItalic
19. Bookman-Light
20. Bookman-LightItalic
21. Helvetica-Narrow
22. Helvetica-Narrow-Bold
23. Helvetica-Narrow-Oblique
24. Helvetica-Narrow-BoldOblique
25. NewCenturySchlbk-Roman
26. NewCenturySchlbk-Italic
27. NewCenturySchlbk-Bold
28. NewCenturySchlbk-BoldItalic
29. Palatino-Roman
30. Palatino-Italic
31. Palatino-Bold
32. Palatino-BoldItalic
33. ZapfChancery-MediumItalic
34. ZapfDingbats

Specifying Pens

pen The attributes of lines and symbol outlines as defined by *pen* is a comma-delimited list of *width*, *color* and *style*, each of which is optional. *width* can be indicated as a measure (in **points** (this is the default), centimeters, or inches) or as **faint**, **default**, **thin**[*ner*|*nest*], **thick**[*er*|*lest*], **fat**[*ter*|*test*], or **obese**. *color* specifies a gray shade or color (see [Specifying Color](#) above). *style* is a combination of dashes '-' and dots '.'. If any of the attributes is omitted their default or previous setting will be retained. See GMT Cookbook & Technical Reference Chapter 4 for more information.

1.8.8 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.8.9 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.8.10 See Also

Look up the individual man pages for more details and full syntax. Run `gmt --help` to list all GMT programs and to show all installation directories. For an explanation of the various GMT settings in this man page (like *FORMAT_FLOAT_OUT*), see the man page of the GMT configuration file *gmt.conf*. Information is also available on the GMT home page <http://gmt.soest.hawaii.edu/>

1.9 gmt2kml

gmt2kml - Convert GMT data tables to KML files for Google Earth

1.9.1 Synopsis

```
gmt2kml [ table ] [ -Aa|g|s[alt|xscale] ] [ -Ccpt ] [ -Ddescriptfile ] [ -E[altitude] ] [ -Fels|t|l|p ] [ -Gfln-|fill ] [ -Iicon ] [ -K ] [ -Lcol1:name1,col2:name2,... ] [ -N[+|*name_template*|name] ] [ -O ] [ -R|w|e|s|n ] [ -Sc|nscale ] [ -Ttitle[foldername] ] [ -V[level] ] [ -W[-|+|pen ] [ -Zargs ] [ -bi[ncols][type][w][+L+B] ] [ -f[i|o]colinfo ] [ -h[i|o][n][+c][+d][+rremark][+rtitle] ] [ -i|cols|I][s|scale][o|offset][,...] ] [ -:[i|o] ] [ > output.kml ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.9.2 Description

gmt2kml reads one or more GMT table file and converts them to a single output file using Google Earth's KML format. Data may represent points, lines, or polygons, and you may specify additional attributes such as title, altitude mode, colors, pen widths, transparency, regions, and data descriptions. You may also extend the feature down to ground level (assuming it is above it) and use custom icons for point symbols.

The input files should contain the following columns:

```
lon lat [ alt ] [ timestart [ timestop ] ]
```

where *lon* and *lat* are required for all features, *alt* is optional for all features (see also **-A** and **-C**), and *timestart* and *timestop* apply to events and timespan features.

1.9.3 Required Arguments

None.

1.9.4 Optional Arguments

table One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Aa|g|s[alt|xscale] Select one of three altitude modes recognized by Google Earth that determines the altitude (in m) of the feature: **a** absolute altitude, **g** altitude relative to sea surface or ground, **s** altitude relative to seafloor or ground. To plot the features at a fixed altitude, append an altitude *alt* (in m). Use 0 to clamp the features to the chosen reference surface. Append *xscale* to scale the altitude from the input file by that factor. If no value is appended, the altitude (in m) is read from the 3rd column of the input file. [By default the features are clamped to the sea surface or ground].

-Ccpt Use color palette for assigning colors to the symbol, event, or timespan icons, based on the value in the 3rd column of the input file. For lines or polygons we examine the segment header for **-Z<value>** statements and obtain the color via the cpt lookup. Note only discrete colors are possible.

-Ddescriptfile File with HTML snippets that will be included as part of the main description content for the KML file [no description]. See SEGMENT INFORMATION below for feature-specific descriptions.

-E[altitude] Extrude feature down to ground level [no extrusion].

- Felsfillp** Sets the feature type. Choose from points (**event**, **symbol**, or **timespan**), **line**, or **polygon** [symbol]. The first two columns of the input file should contain (*lon*, *lat*). When altitude or value is required (i.e., no *altitude* value was given with **-A**, or **-C** is set), the third column needs to contain the *altitude* (in m) or *value*. The event (**-Fe**) is a symbol that should only be active at a particular *time*, given in the next column. Timespan (**-Ft**) is a symbol that should only be active during a particular time period indicated by the next two columns (*timestart*, *timestop*). Use NaN to indicate unbounded time limits. If used, times should be in ISO format yyyy-mm-ddThh:mm:ss[.xxx] or in GMT relative time format (see **-f**).
- GfInfill** Set fill color for symbols, extrusions and polygons (**-Gf**) [Default is light orange at 75% transparency] or text labels (**-Gn**) [Default is white]. Optionally, use **-Gf-** to turn off polygon fill, and **-Gn-** to disable labels.
- Iicon** Specify the URL to an alternative icon that should be used for the symbol [Default is a Google Earth circle]. If the URL starts with + then we will prepend <http://maps.google.com/mapfiles/kml/> to the name. To turn off icons entirely (e.g., when just wanting a text label), use **-I-**. [Default is a local icon with no directory path].
- K** Allow more KML code to be appended to the output later [finalize the KML file].
- Lname1,name2,...** Extended data given. Append one or more column names separated by commas. We will expect the listed data columns to exist in the input immediately following the data coordinates and they will be encoded in the KML file as Extended Data sets, whose attributes will be available in the Google Earth balloon when the item is selected. This option is not available unless input is an ASCII file.
- N[-|+|name_template|name]** By default, if segment headers contain a **-L** “label string” then we use that for the name of the KML feature (polygon, line segment or set of symbols). Default names for these segments are “Line %d” and “Point Set %d”, depending on the feature, where %d is a sequence number of line segments within a file. Each point within a line segment will be named after the line segment plus a sequence number. Default is simply “Point %d”. Alternatively, select one of these options: (1) append **-** to supply individual symbol labels (single word) via the field immediately following the data coordinates, (2) append **+** to supply individual symbol labels as the rest the end of the data record following the data coordinates, (3) append a string that may include %d or a similar integer format to assign unique name IDs for each feature, with the segment number (for lines and polygons) or point number (symbols) appearing where %d is placed, (4) give no arguments to turn symbol labeling off; line segments will still be named. Note: if **-N-** is used with **-L** then the label must appear before the extended data columns. Also note that options (1) and (2) are not available unless input is an ASCII file.
- O** Appended KML code to an existing KML file [initialize a new KML file].
- Ralw/e/s/n** Issue a single Region tag. Append *w/e/s/n* to set a particular region (will ignore points outside the region), or append **a** to determine and use the actual domain of the data (single file only) [no region tags issued].
- ScInscale]** Scale icons or labels. Here, **-Sc** sets a scale for the symbol icon, whereas **-Sn** sets a scale for the name labels [1 for both].
- Ttitle[/foldername]** Sets the document title [default is unset]. Optionally, append */FolderName*; this allows you, with **-O**, **-K**, to group features into folders within the KML document. [The default folder name is “Name Features”, where *Name* is Point, Event, Timespan, Line, or Polygon].
- V[level] (*more ...*)** Select verbosity level [c].
- W[-|+|pen]** Set pen attributes for lines or polygon outlines. Append pen attributes to use [Defaults: width = default, color = black, style = solid]. If **-C** is given you may optionally use **-W-** to apply

the cpt color to the polygon outline only (fill determined by **-G**) or **-W+** to use the cpt color for both polygon fill and outline. Note that for KML the pen width is given as integer pixel widths so you must specify pen width as *np*, where *n* is an integer.

- Zargs** Set one or more attributes of the Document and Region tags. Append **+aalt_min/alt_max** to specify limits on visibility based on altitude. Append **+lod_min/lod_max** to specify limits on visibility based on Level Of Detail, where *lod_max* == -1 means it is visible to infinite size. Append **+ffade_min/fade_max** to fade in and out over a ramp [abrupt]. Append **+v** to make a feature *not* visible when loaded [visible]. Append **+o** to open a folder or document in the sidebar when loaded [closed].
- bi[ncols][type] (more ...)** Select binary input. [Default is 2 input columns].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|ID|[col]z[+|-]gap[u] (more ...)** Determine data gaps and line breaks.
- h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- :[ilo] (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.9.5 Examples

To convert a file with point locations (lon, lat) into a KML file with red circle symbols, try

```
gmt2kml mypoints.txt -Gfred -Fs > mypoints.kml
```

To convert a multisegment file with lines (lon, lat) separated by segment headers that contain a **-Llabelstring** with the feature name, selecting a thick white pen, and title the document, try

```
gmt2kml mylines.txt -Wthick,white -F1 -T"Lines from here to there" > mylines.kml
```

To convert a multisegment file with polygons (lon, lat) separated by segment headers that contain a **-Llabelstring** with the feature name, selecting a thick black pen and semi-transparent yellow fill, giving a title to the document, and prescribing a particular region limit, try

```
gmt gmt2kml mypolygons.txt -Gfyellow@50 -Fp -T"My polygons" -R30/90/-20/40 > mypolygons.kml
```

To convert a file with point locations (lon, lat, time) into a KML file with green circle symbols that will go active at the specified time and stay active going forward, try

```
awk '{print $1, $2, $3, "NaN"}' mypoints.txt | gmt gmt2kml -Gfgreen -Ft > mytimepoints.kml
```

To extract contours and labels every 10 units from the grid temp.nc and plot them in KML, using red lines at 75% transparency and red labels (no transparency), try

```
gmt grdcontour temp.nc -Jx1id -A10+tlabel.txt -C10 -Dcontours.txt
gmt gmt2kml contours.txt -F1 -Wlp,red@75 -K > contours.kml
gmt gmt2kml -O -N+ -Fs -Sn2 -Gnred@0 label.txt -I- >> contours.kml
```

To instead plot the contours as lines with colors taken from the cpt file contours.cpt, try

```
gmt gmt2kml contours.txt -F1 -Ccontours.cpt > contours.kml
```

1.9.6 Limitations

Google Earth has trouble displaying filled polygons across the Dateline. For now you must manually break any polygon crossing the dateline into a west and east polygon and plot them separately.

1.9.7 Making Kmz Files

Using the KMZ format is preferred as it takes less space. KMZ is simply a KML file and any data files, icons, or images referenced by the KML, contained in a zip archive. One way to organize large data sets is to split them into groups called Folders. A Document can contain any number of folders. Using scripts you can create a composite KML file using the **-K**, **-O** options just like you do with GMT plots. See **-T** for switching between folders and documents.

1.9.8 Kml Hierarchy

GMT stores the different features in hierarchical folders by feature type (when using **-O**, **-K** or **-T/foldername**), by input file (if not standard input), and by line segment (using the name from the segment header, or **-N**). This makes it more easy in Google Earth to switch on or off parts of the contents of the Document. The following is a crude example:

```
[ KML header information; not present if -O was used ]
<Document><name>GMT Data Document</name>
<Folder><name>Point Features</name>
<!--This level of folder is inserted only when using -O, -K>
<Folder><name>file1.dat</name>
<!--One folder for each input file (not when standard input)>
<Folder><name>Point Set 0</name>
<!--One folder per line segment>
<!--Points from the first line segment in file file1.dat go here>
<Folder><name>Point Set 1</name>
<!--Points from the second line segment in file file1.dat go here>
</Folder>
</Folder>
<Folder><name>Line Features</name>
<Folder><name>file1.dat</name>
```

<!--One folder for each input file (not when standard input)>

<Placemark><name>Line 0</name>

<!--Here goes the first line segment>

</Placemark>

<Placemark><name>Line 1</name>

<!--Here goes the second line segment>

</Placemark>

</Folder>

<Folder>

</Document>

[KML trailer information; not present if **-K** was used]

1.9.9 Segment Information

gmt2kml will scan the segment headers for substrings of the form **-L**“*some label*” [also see **-N** discussion] and **-T**“*some text description*”. If present, these are parsed to supply name and description tags, respectively, for the current feature.

1.9.10 See Also

gmt, *gmt.conf*, *img2google*, *kml2gmt*, *ps2raster*

1.10 gmt5syntax

gmt5syntax - Convert old GMT script to use new ‘gmt <module>’ syntax

1.10.1 Synopsis

gmt5syntax old_script > new_script

1.10.2 Description

gmt5syntax is a perl script that converts old-style GMT commands in, e.g., shell scripts, to the new `gmt <module>-syntax`. This utility is located in the `tools` subdirectory of the `data` directory. `gmt --show-datadir` will show the path to the latter.

1.10.3 See Also

gmt

1.11 gmtcolors

gmtcolors - Explanation of color codes in GMT

1.11.1 Description

Colors can be specified in GMT as arguments to commands, generally as part of the **-G** or **-W** options to select polygon fill or outline pen. Colors are also used in color palette tables (cpt files) that help convert numerical values to colors.

GMT allows several ways to represent a color:

Colorname Specify one of the named colors below. All names are case-insensitive.

R/G/B Specify **R**ed, **G**reen, and **B**lue levels. Each value is separated by a slash and is in the range from 0 (dark) to 255 (light). This representation is used to color monitors.

#RRGGBB Specify **R**ed, **G**reen, and **B**lue levels in the way that it is done in HTML. Use two characters for each color channel, ranging from 00 (dark) to FF (light). Upper and lower case are allowed.

Graylevel For shades of gray, $R = G = B$, and only one number needs to be used. This representation is popular with black and white printers.

H-S-V Specify **H**ue in the range 0 to 360 (degrees), **S**aturation between 0 (not saturated) and 1 (fully saturated), and value **V** between 0 (dark) and 1 (light). Number are separated by hyphens. This representation can be helpful when hue varies a lot.

C/M/Y/K Specify **C**yan, **M**agenta, **Y**ellow, and **blacBD(K)**. Each number is in the range from 0 (no paint) to 1 (maximum paint). This representation is used by most color printers.

1.11.2 List Of Colors

The following list contains the named colors that can be used in GMT and their equivalent color codes.

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|----------------|
| 255 | 250 | 250 | snow |
| 248 | 248 | 255 | ghostwhite |
| 255 | 250 | 240 | floralwhite |
| 255 | 245 | 238 | seashell |
| 253 | 245 | 230 | oldlace |
| 250 | 240 | 230 | linen |
| 250 | 235 | 215 | antiquewhite |
| 255 | 239 | 213 | papayawhip |
| 255 | 235 | 205 | blanchedalmond |
| 255 | 228 | 196 | bisque |
| 255 | 218 | 185 | peachpuff |
| 255 | 222 | 173 | navajowhite |
| 255 | 228 | 181 | moccasin |
| 255 | 250 | 205 | lemonchiffon |
| 255 | 248 | 220 | cornsilk |
| 255 | 255 | 240 | ivory |
| 240 | 255 | 240 | honeydew |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|----------|----------|----------|------------------------|
| 245 | 255 | 250 | mintcream |
| 240 | 255 | 255 | azure |
| 240 | 248 | 255 | aliceblue |
| 230 | 230 | 250 | lavender |
| 255 | 240 | 245 | lavenderblush |
| 255 | 228 | 225 | mistyrose |
| 100 | 149 | 237 | cornflowerblue |
| 106 | 90 | 205 | slateblue |
| 123 | 104 | 238 | mediumslateblue |
| 132 | 112 | 255 | lightslateblue |
| 135 | 206 | 235 | skyblue |
| 135 | 206 | 250 | lightskyblue |
| 119 | 136 | 153 | lightslategray |
| 119 | 136 | 153 | lightslategrey |
| 112 | 128 | 144 | slategray |
| 112 | 128 | 144 | slategrey |
| 176 | 196 | 222 | lightsteelblue |
| 173 | 216 | 230 | lightblue |
| 224 | 255 | 255 | lightcyan |
| 176 | 224 | 230 | powderblue |
| 175 | 238 | 238 | paleturquoise |
| 102 | 205 | 170 | mediumaquamarine |
| 127 | 255 | 212 | aquamarine |
| 144 | 238 | 144 | lightgreen |
| 143 | 188 | 143 | darkseagreen |
| 152 | 251 | 152 | palegreen |
| 124 | 252 | 0 | lawngreen |
| 127 | 255 | 0 | chartreuse |
| 173 | 255 | 47 | greenyellow |
| 154 | 205 | 50 | yellowgreen |
| 107 | 142 | 35 | olivedrab |
| 189 | 183 | 107 | darkkhaki |
| 240 | 230 | 140 | khaki |
| 238 | 232 | 170 | palegoldenrod |
| 238 | 221 | 130 | lightgoldenrod |
| 255 | 255 | 224 | lightyellow |
| 250 | 250 | 210 | lightgoldenrodyellow |
| 255 | 255 | 0 | yellow |
| 128 | 128 | 0 | darkyellow |
| 255 | 215 | 0 | gold |
| 218 | 165 | 32 | goldenrod |
| 184 | 134 | 11 | darkgoldenrod |
| 188 | 143 | 143 | rosybrown |
| 205 | 92 | 92 | indianred |
| 139 | 69 | 19 | saddlebrown |
| 160 | 82 | 45 | sienna |
| 205 | 133 | 63 | peru |
| 222 | 184 | 135 | burlywood |
| | | | Continued on next page |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|-----------------|
| 245 | 245 | 220 | beige |
| 245 | 222 | 179 | wheat |
| 244 | 164 | 96 | sandybrown |
| 210 | 180 | 140 | tan |
| 210 | 105 | 30 | chocolate |
| 178 | 34 | 34 | firebrick |
| 235 | 190 | 85 | lightbrown |
| 165 | 42 | 42 | brown |
| 120 | 60 | 30 | darkbrown |
| 233 | 150 | 122 | darksalmon |
| 250 | 128 | 114 | salmon |
| 255 | 160 | 122 | lightsalmon |
| 255 | 192 | 128 | lightorange |
| 255 | 165 | 0 | orange |
| 255 | 140 | 0 | darkorange |
| 255 | 127 | 80 | coral |
| 240 | 128 | 128 | lightcoral |
| 255 | 99 | 71 | tomato |
| 255 | 69 | 0 | orangered |
| 255 | 0 | 0 | red |
| 255 | 128 | 128 | lightred |
| 139 | 0 | 0 | darkred |
| 255 | 20 | 147 | deeppink |
| 255 | 105 | 180 | hotpink |
| 255 | 192 | 203 | pink |
| 255 | 182 | 193 | lightpink |
| 219 | 112 | 147 | palevioletred |
| 176 | 48 | 96 | maroon |
| 199 | 21 | 133 | mediumvioletred |
| 208 | 32 | 144 | violetred |
| 139 | 0 | 139 | darkmagenta |
| 255 | 0 | 255 | magenta |
| 255 | 128 | 255 | lightmagenta |
| 238 | 130 | 238 | violet |
| 218 | 112 | 214 | orchid |
| 221 | 160 | 221 | plum |
| 186 | 85 | 211 | mediumorchid |
| 153 | 50 | 204 | darkorchid |
| 148 | 0 | 211 | darkviolet |
| 138 | 43 | 226 | blueviolet |
| 160 | 32 | 240 | purple |
| 147 | 112 | 219 | mediumpurple |
| 216 | 191 | 216 | thistle |
| 105 | 105 | 105 | dimgray |
| 105 | 105 | 105 | dimgrey |
| 169 | 169 | 169 | darkgray |
| 169 | 169 | 169 | darkgrey |
| 190 | 190 | 190 | gray |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|----------------|
| 190 | 190 | 190 | grey |
| 211 | 211 | 211 | lightgrey |
| 211 | 211 | 211 | lightgray |
| 220 | 220 | 220 | gainsboro |
| 245 | 245 | 245 | whitesmoke |
| 255 | 255 | 255 | white |
| 255 | 250 | 250 | snow1 |
| 238 | 233 | 233 | snow2 |
| 205 | 201 | 201 | snow3 |
| 139 | 137 | 137 | snow4 |
| 255 | 245 | 238 | seashell1 |
| 238 | 229 | 222 | seashell2 |
| 205 | 197 | 191 | seashell3 |
| 139 | 134 | 130 | seashell4 |
| 255 | 239 | 219 | antiquewhite1 |
| 238 | 223 | 204 | antiquewhite2 |
| 205 | 192 | 176 | antiquewhite3 |
| 139 | 131 | 120 | antiquewhite4 |
| 255 | 228 | 196 | bisque1 |
| 238 | 213 | 183 | bisque2 |
| 205 | 183 | 158 | bisque3 |
| 139 | 125 | 107 | bisque4 |
| 255 | 218 | 185 | peachpuff1 |
| 238 | 203 | 173 | peachpuff2 |
| 205 | 175 | 149 | peachpuff3 |
| 139 | 119 | 101 | peachpuff4 |
| 255 | 222 | 173 | navajowhite1 |
| 238 | 207 | 161 | navajowhite2 |
| 205 | 179 | 139 | navajowhite3 |
| 139 | 121 | 94 | navajowhite4 |
| 255 | 250 | 205 | lemonchiffon1 |
| 238 | 233 | 191 | lemonchiffon2 |
| 205 | 201 | 165 | lemonchiffon3 |
| 139 | 137 | 112 | lemonchiffon4 |
| 255 | 248 | 220 | cornsilk1 |
| 238 | 232 | 205 | cornsilk2 |
| 205 | 200 | 177 | cornsilk3 |
| 139 | 136 | 120 | cornsilk4 |
| 255 | 255 | 240 | ivory1 |
| 238 | 238 | 224 | ivory2 |
| 205 | 205 | 193 | ivory3 |
| 139 | 139 | 131 | ivory4 |
| 240 | 255 | 240 | honeydew1 |
| 224 | 238 | 224 | honeydew2 |
| 193 | 205 | 193 | honeydew3 |
| 131 | 139 | 131 | honeydew4 |
| 255 | 240 | 245 | lavenderblush1 |
| 238 | 224 | 229 | lavenderblush2 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|-----------------|
| 205 | 193 | 197 | lavenderblush3 |
| 139 | 131 | 134 | lavenderblush4 |
| 255 | 228 | 225 | mistyrose1 |
| 238 | 213 | 210 | mistyrose2 |
| 205 | 183 | 181 | mistyrose3 |
| 139 | 125 | 123 | mistyrose4 |
| 240 | 255 | 255 | azure1 |
| 224 | 238 | 238 | azure2 |
| 193 | 205 | 205 | azure3 |
| 131 | 139 | 139 | azure4 |
| 131 | 111 | 255 | slateblue1 |
| 122 | 103 | 238 | slateblue2 |
| 105 | 89 | 205 | slateblue3 |
| 135 | 206 | 255 | skyblue1 |
| 126 | 192 | 238 | skyblue2 |
| 108 | 166 | 205 | skyblue3 |
| 176 | 226 | 255 | lightskyblue1 |
| 164 | 211 | 238 | lightskyblue2 |
| 141 | 182 | 205 | lightskyblue3 |
| 198 | 226 | 255 | slategray1 |
| 185 | 211 | 238 | slategray2 |
| 159 | 182 | 205 | slategray3 |
| 108 | 123 | 139 | slategray4 |
| 202 | 225 | 255 | lightsteelblue1 |
| 188 | 210 | 238 | lightsteelblue2 |
| 162 | 181 | 205 | lightsteelblue3 |
| 110 | 123 | 139 | lightsteelblue4 |
| 191 | 239 | 255 | lightblue1 |
| 178 | 223 | 238 | lightblue2 |
| 154 | 192 | 205 | lightblue3 |
| 104 | 131 | 139 | lightblue4 |
| 224 | 255 | 255 | lightcyan1 |
| 209 | 238 | 238 | lightcyan2 |
| 180 | 205 | 205 | lightcyan3 |
| 122 | 139 | 139 | lightcyan4 |
| 187 | 255 | 255 | paleturquoise1 |
| 174 | 238 | 238 | paleturquoise2 |
| 150 | 205 | 205 | paleturquoise3 |
| 102 | 139 | 139 | paleturquoise4 |
| 152 | 245 | 255 | cadetblue1 |
| 142 | 229 | 238 | cadetblue2 |
| 122 | 197 | 205 | cadetblue3 |
| 151 | 255 | 255 | darkslategray1 |
| 141 | 238 | 238 | darkslategray2 |
| 121 | 205 | 205 | darkslategray3 |
| 127 | 255 | 212 | aquamarine1 |
| 118 | 238 | 198 | aquamarine2 |
| 102 | 205 | 170 | aquamarine3 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|-----------------|
| 193 | 255 | 193 | darkseagreen1 |
| 180 | 238 | 180 | darkseagreen2 |
| 155 | 205 | 155 | darkseagreen3 |
| 105 | 139 | 105 | darkseagreen4 |
| 154 | 255 | 154 | palegreen1 |
| 144 | 238 | 144 | palegreen2 |
| 124 | 205 | 124 | palegreen3 |
| 127 | 255 | 0 | chartreuse1 |
| 118 | 238 | 0 | chartreuse2 |
| 102 | 205 | 0 | chartreuse3 |
| 192 | 255 | 62 | olivedrab1 |
| 179 | 238 | 58 | olivedrab2 |
| 154 | 205 | 50 | olivedrab3 |
| 105 | 139 | 34 | olivedrab4 |
| 202 | 255 | 112 | darkolivegreen1 |
| 188 | 238 | 104 | darkolivegreen2 |
| 162 | 205 | 90 | darkolivegreen3 |
| 110 | 139 | 61 | darkolivegreen4 |
| 255 | 246 | 143 | khaki1 |
| 238 | 230 | 133 | khaki2 |
| 205 | 198 | 115 | khaki3 |
| 139 | 134 | 78 | khaki4 |
| 255 | 236 | 139 | lightgoldenrod1 |
| 238 | 220 | 130 | lightgoldenrod2 |
| 205 | 190 | 112 | lightgoldenrod3 |
| 139 | 129 | 76 | lightgoldenrod4 |
| 255 | 255 | 224 | lightyellow1 |
| 238 | 238 | 209 | lightyellow2 |
| 205 | 205 | 180 | lightyellow3 |
| 139 | 139 | 122 | lightyellow4 |
| 255 | 255 | 0 | yellow1 |
| 238 | 238 | 0 | yellow2 |
| 205 | 205 | 0 | yellow3 |
| 139 | 139 | 0 | yellow4 |
| 255 | 215 | 0 | gold1 |
| 238 | 201 | 0 | gold2 |
| 205 | 173 | 0 | gold3 |
| 139 | 117 | 0 | gold4 |
| 255 | 193 | 37 | goldenrod1 |
| 238 | 180 | 34 | goldenrod2 |
| 205 | 155 | 29 | goldenrod3 |
| 139 | 105 | 20 | goldenrod4 |
| 255 | 185 | 15 | darkgoldenrod1 |
| 238 | 173 | 14 | darkgoldenrod2 |
| 205 | 149 | 12 | darkgoldenrod3 |
| 139 | 101 | 8 | darkgoldenrod4 |
| 255 | 193 | 193 | rosybrown1 |
| 238 | 180 | 180 | rosybrown2 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|--------------|
| 205 | 155 | 155 | rosybrown3 |
| 139 | 105 | 105 | rosybrown4 |
| 255 | 106 | 106 | indianred1 |
| 238 | 99 | 99 | indianred2 |
| 205 | 85 | 85 | indianred3 |
| 139 | 58 | 58 | indianred4 |
| 255 | 130 | 71 | sienna1 |
| 238 | 121 | 66 | sienna2 |
| 205 | 104 | 57 | sienna3 |
| 139 | 71 | 38 | sienna4 |
| 255 | 211 | 155 | burlywood1 |
| 238 | 197 | 145 | burlywood2 |
| 205 | 170 | 125 | burlywood3 |
| 139 | 115 | 85 | burlywood4 |
| 255 | 231 | 186 | wheat1 |
| 238 | 216 | 174 | wheat2 |
| 205 | 186 | 150 | wheat3 |
| 139 | 126 | 102 | wheat4 |
| 255 | 165 | 79 | tan1 |
| 238 | 154 | 73 | tan2 |
| 205 | 133 | 63 | tan3 |
| 139 | 90 | 43 | tan4 |
| 255 | 127 | 36 | chocolate1 |
| 238 | 118 | 33 | chocolate2 |
| 205 | 102 | 29 | chocolate3 |
| 139 | 69 | 19 | chocolate4 |
| 255 | 48 | 48 | firebrick1 |
| 238 | 44 | 44 | firebrick2 |
| 205 | 38 | 38 | firebrick3 |
| 139 | 26 | 26 | firebrick4 |
| 255 | 64 | 64 | brown1 |
| 238 | 59 | 59 | brown2 |
| 205 | 51 | 51 | brown3 |
| 139 | 35 | 35 | brown4 |
| 255 | 140 | 105 | salmon1 |
| 238 | 130 | 98 | salmon2 |
| 205 | 112 | 84 | salmon3 |
| 139 | 76 | 57 | salmon4 |
| 255 | 160 | 122 | lightsalmon1 |
| 238 | 149 | 114 | lightsalmon2 |
| 205 | 129 | 98 | lightsalmon3 |
| 139 | 87 | 66 | lightsalmon4 |
| 255 | 165 | 0 | orange1 |
| 238 | 154 | 0 | orange2 |
| 205 | 133 | 0 | orange3 |
| 139 | 90 | 0 | orange4 |
| 255 | 127 | 0 | darkorange1 |
| 238 | 118 | 0 | darkorange2 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|----------------|
| 205 | 102 | 0 | darkorange3 |
| 139 | 69 | 0 | darkorange4 |
| 255 | 114 | 86 | coral1 |
| 238 | 106 | 80 | coral2 |
| 205 | 91 | 69 | coral3 |
| 139 | 62 | 47 | coral4 |
| 255 | 99 | 71 | tomato1 |
| 238 | 92 | 66 | tomato2 |
| 205 | 79 | 57 | tomato3 |
| 139 | 54 | 38 | tomato4 |
| 255 | 69 | 0 | orangered1 |
| 238 | 64 | 0 | orangered2 |
| 205 | 55 | 0 | orangered3 |
| 139 | 37 | 0 | orangered4 |
| 255 | 0 | 0 | red1 |
| 238 | 0 | 0 | red2 |
| 205 | 0 | 0 | red3 |
| 139 | 0 | 0 | red4 |
| 255 | 20 | 147 | deeppink1 |
| 238 | 18 | 137 | deeppink2 |
| 205 | 16 | 118 | deeppink3 |
| 139 | 10 | 80 | deeppink4 |
| 255 | 110 | 180 | hotpink1 |
| 238 | 106 | 167 | hotpink2 |
| 205 | 96 | 144 | hotpink3 |
| 139 | 58 | 98 | hotpink4 |
| 255 | 181 | 197 | pink1 |
| 238 | 169 | 184 | pink2 |
| 205 | 145 | 158 | pink3 |
| 139 | 99 | 108 | pink4 |
| 255 | 174 | 185 | lightpink1 |
| 238 | 162 | 173 | lightpink2 |
| 205 | 140 | 149 | lightpink3 |
| 139 | 95 | 101 | lightpink4 |
| 255 | 130 | 171 | palevioletred1 |
| 238 | 121 | 159 | palevioletred2 |
| 205 | 104 | 137 | palevioletred3 |
| 139 | 71 | 93 | palevioletred4 |
| 255 | 52 | 179 | maroon1 |
| 238 | 48 | 167 | maroon2 |
| 205 | 41 | 144 | maroon3 |
| 139 | 28 | 98 | maroon4 |
| 255 | 62 | 150 | violetred1 |
| 238 | 58 | 140 | violetred2 |
| 205 | 50 | 120 | violetred3 |
| 139 | 34 | 82 | violetred4 |
| 255 | 0 | 255 | magenta1 |
| 238 | 0 | 238 | magenta2 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|---------------|
| 205 | 0 | 205 | magenta3 |
| 139 | 0 | 139 | magenta4 |
| 255 | 131 | 250 | orchid1 |
| 238 | 122 | 233 | orchid2 |
| 205 | 105 | 201 | orchid3 |
| 139 | 71 | 137 | orchid4 |
| 255 | 187 | 255 | plum1 |
| 238 | 174 | 238 | plum2 |
| 205 | 150 | 205 | plum3 |
| 139 | 102 | 139 | plum4 |
| 224 | 102 | 255 | mediumorchid1 |
| 209 | 95 | 238 | mediumorchid2 |
| 180 | 82 | 205 | mediumorchid3 |
| 122 | 55 | 139 | mediumorchid4 |
| 191 | 62 | 255 | darkorchid1 |
| 178 | 58 | 238 | darkorchid2 |
| 154 | 50 | 205 | darkorchid3 |
| 104 | 34 | 139 | darkorchid4 |
| 155 | 48 | 255 | purple1 |
| 145 | 44 | 238 | purple2 |
| 125 | 38 | 205 | purple3 |
| 171 | 130 | 255 | mediumpurple1 |
| 159 | 121 | 238 | mediumpurple2 |
| 137 | 104 | 205 | mediumpurple3 |
| 255 | 225 | 255 | thistle1 |
| 238 | 210 | 238 | thistle2 |
| 205 | 181 | 205 | thistle3 |
| 139 | 123 | 139 | thistle4 |
| 102 | 102 | 102 | gray40 |
| 102 | 102 | 102 | grey40 |
| 105 | 105 | 105 | gray41 |
| 105 | 105 | 105 | grey41 |
| 107 | 107 | 107 | gray42 |
| 107 | 107 | 107 | grey42 |
| 110 | 110 | 110 | gray43 |
| 110 | 110 | 110 | grey43 |
| 112 | 112 | 112 | gray44 |
| 112 | 112 | 112 | grey44 |
| 115 | 115 | 115 | gray45 |
| 115 | 115 | 115 | grey45 |
| 117 | 117 | 117 | gray46 |
| 117 | 117 | 117 | grey46 |
| 120 | 120 | 120 | gray47 |
| 120 | 120 | 120 | grey47 |
| 122 | 122 | 122 | gray48 |
| 122 | 122 | 122 | grey48 |
| 125 | 125 | 125 | gray49 |
| 125 | 125 | 125 | grey49 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|-------------|
| 127 | 127 | 127 | gray50 |
| 127 | 127 | 127 | grey50 |
| 130 | 130 | 130 | gray51 |
| 130 | 130 | 130 | grey51 |
| 133 | 133 | 133 | gray52 |
| 133 | 133 | 133 | grey52 |
| 135 | 135 | 135 | gray53 |
| 135 | 135 | 135 | grey53 |
| 138 | 138 | 138 | gray54 |
| 138 | 138 | 138 | grey54 |
| 140 | 140 | 140 | gray55 |
| 140 | 140 | 140 | grey55 |
| 143 | 143 | 143 | gray56 |
| 143 | 143 | 143 | grey56 |
| 145 | 145 | 145 | gray57 |
| 145 | 145 | 145 | grey57 |
| 148 | 148 | 148 | gray58 |
| 148 | 148 | 148 | grey58 |
| 150 | 150 | 150 | gray59 |
| 150 | 150 | 150 | grey59 |
| 153 | 153 | 153 | gray60 |
| 153 | 153 | 153 | grey60 |
| 156 | 156 | 156 | gray61 |
| 156 | 156 | 156 | grey61 |
| 158 | 158 | 158 | gray62 |
| 158 | 158 | 158 | grey62 |
| 161 | 161 | 161 | gray63 |
| 161 | 161 | 161 | grey63 |
| 163 | 163 | 163 | gray64 |
| 163 | 163 | 163 | grey64 |
| 166 | 166 | 166 | gray65 |
| 166 | 166 | 166 | grey65 |
| 168 | 168 | 168 | gray66 |
| 168 | 168 | 168 | grey66 |
| 171 | 171 | 171 | gray67 |
| 171 | 171 | 171 | grey67 |
| 173 | 173 | 173 | gray68 |
| 173 | 173 | 173 | grey68 |
| 176 | 176 | 176 | gray69 |
| 176 | 176 | 176 | grey69 |
| 179 | 179 | 179 | gray70 |
| 179 | 179 | 179 | grey70 |
| 181 | 181 | 181 | gray71 |
| 181 | 181 | 181 | grey71 |
| 184 | 184 | 184 | gray72 |
| 184 | 184 | 184 | grey72 |
| 186 | 186 | 186 | gray73 |
| 186 | 186 | 186 | grey73 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|------------------------|----------|----------|-------------|
| 189 | 189 | 189 | gray74 |
| 189 | 189 | 189 | grey74 |
| 191 | 191 | 191 | gray75 |
| 191 | 191 | 191 | grey75 |
| 194 | 194 | 194 | gray76 |
| 194 | 194 | 194 | grey76 |
| 196 | 196 | 196 | gray77 |
| 196 | 196 | 196 | grey77 |
| 199 | 199 | 199 | gray78 |
| 199 | 199 | 199 | grey78 |
| 201 | 201 | 201 | gray79 |
| 201 | 201 | 201 | grey79 |
| 204 | 204 | 204 | gray80 |
| 204 | 204 | 204 | grey80 |
| 207 | 207 | 207 | gray81 |
| 207 | 207 | 207 | grey81 |
| 209 | 209 | 209 | gray82 |
| 209 | 209 | 209 | grey82 |
| 212 | 212 | 212 | gray83 |
| 212 | 212 | 212 | grey83 |
| 214 | 214 | 214 | gray84 |
| 214 | 214 | 214 | grey84 |
| 217 | 217 | 217 | gray85 |
| 217 | 217 | 217 | grey85 |
| 219 | 219 | 219 | gray86 |
| 219 | 219 | 219 | grey86 |
| 222 | 222 | 222 | gray87 |
| 222 | 222 | 222 | grey87 |
| 224 | 224 | 224 | gray88 |
| 224 | 224 | 224 | grey88 |
| 227 | 227 | 227 | gray89 |
| 227 | 227 | 227 | grey89 |
| 229 | 229 | 229 | gray90 |
| 229 | 229 | 229 | grey90 |
| 232 | 232 | 232 | gray91 |
| 232 | 232 | 232 | grey91 |
| 235 | 235 | 235 | gray92 |
| 235 | 235 | 235 | grey92 |
| 237 | 237 | 237 | gray93 |
| 237 | 237 | 237 | grey93 |
| 240 | 240 | 240 | gray94 |
| 240 | 240 | 240 | grey94 |
| 242 | 242 | 242 | gray95 |
| 242 | 242 | 242 | grey95 |
| 245 | 245 | 245 | gray96 |
| 245 | 245 | 245 | grey96 |
| 247 | 247 | 247 | gray97 |
| 247 | 247 | 247 | grey97 |
| Continued on next page | | | |

Table 1.2 – continued from previous page

| <i>R</i> | <i>G</i> | <i>B</i> | <i>Name</i> |
|----------|----------|----------|-------------|
| 250 | 250 | 250 | gray98 |
| 250 | 250 | 250 | grey98 |
| 252 | 252 | 252 | gray99 |
| 252 | 252 | 252 | grey99 |
| 255 | 255 | 255 | gray100 |
| 255 | 255 | 255 | grey100 |

1.11.3 Further Information

For more information on the use of color, read Appendix I of the **GMT TECHNICAL REFERENCE AND COOKBOOK**.

1.11.4 See Also

gmt.conf , *gmtlogo* , *grdcontour* , *grdvector* , *grdview* *psbasemap* , *pscoast* , *pscontour* , *pshistogram* , *psimage* , *pslegend* , *psmask* , *psrose* , *pstext* , *pswiggle* , *psxy* , *psxyz*

1.12 gmtconnect

gmtconnect - Connect individual lines whose end points match within tolerance

1.12.1 Synopsis

```
gmtconnect [ table ] [ -C[closed] ] [ -D[template] ] [ -L[linkfile] ] [ -Q[template] ]
[ -Tcutoff[unit]/[nn_dist] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L|+B] ] [ -f[ilo]colinfo
] [ -g[a]xly|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -
icols[I][sscale][offset][...] ] [ -ocols[...] ] [ -:ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.12.2 Description

gmtconnect reads standard input or one or more data files, which may be multisegment files, and examines the coordinates of the end points of all line segments. If a pair of end points are identical or closer to each other than the specified separation tolerance then the two line segments are joined into a single segment. The process repeats until all the remaining endpoints no longer pass the tolerance test; the resulting segments are then written out to standard output or specified output file. If it is not clear what the separation tolerance should be then use **-L** to get a list of all separation distances and analyze them to determine a suitable cutoff.

1.12.3 Required Arguments

None.

1.12.4 Optional Arguments

- table** One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.
- C**[*closed*] Write all the closed polygons to *closed* [gmtconnect_closed.txt] and all other segments as they are to stdout. No connection takes place. Use **-T***cutoff* to set a minimum separation [0], and if *cutoff* is > 0 then we also explicitly close the polygons on output.
- D**[*template*] For multiple segment data, dump each segment to a separate output file [Default writes a single multiple segment file]. Append a format template for the individual file names; this template **must** contain a C format specifier that can format an integer argument (the segment number); this is usually %d but could be %08d which gives leading zeros, etc. Optionally, it may also contain the format %c before the integer; this will then be replaced by C (closed) or O (open) to indicate segment type. [Default is gmtconnect_segment_%d.txt]. Note that segment headers will be written in either case. For composite segments, a generic segment header will be written and the segment headers of individual pieces will be written out as comments to make it possible to identify where the connected pieces came from.
- L**[*linkfile*] Writes the link information to the specified file [gmtconnect_link.txt]. For each segment we write the original segment id, and for the beginning and end point of the segment we report the id of the closest segment, whether it is the beginning (B) or end (E) point that is closest, and the distance between those points in units determined by **-T**.
- Q**[*template*] Used with **-D** to a list file with the names of the individual output files. Optionally, append a filename template for the individual file names; this template **may** contain a C format specifier that can format an character (C or O for closed or open, respectively). [Default is gmtconnect_list.txt].
- T***cutoff*[*unit*][*nn_dist*] Specifies the separation tolerance in the data coordinate units [0]; append distance unit (see UNITS). If two lines has end-points that are closer than this cutoff they will be joined. Optionally, append *nn_dist* which adds the requirement that a link will only be made if the second closest connection exceeds the *nn_dist*. The latter distance must be given in the same units as *cutoff*.
- V**[*level*] (*more ...*) Select verbosity level [c].
- bi**[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- bo**[*ncols*][*type*] (*more ...*) Select binary output. [Default is same as input].
- f**[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- g**[*a*]*xly|d|X|Y|D*[*col*]*z*[+|-]*gap*[*u*] (*more ...*) Determine data gaps and line breaks.
- h**[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- i***cols*[1][*sscale*][*offset*][,...] (*more ...*) Select input columns (0 is first column).
- o***cols*[,...] (*more ...*) Select output columns (0 is first column).
- :**[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.12.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.12.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.12.7 Examples

To combine the digitized segment lines *segment_*.txt* (whose coordinates are in cm) into as few complete lines as possible, assuming the end points slop could be up to 0.1 mm, run

```
gmt connect segment_*.txt -Tf0.1 > new_segments.txt
```

To combine the digitized segments in the multisegment file *my_lines.txt* (whose coordinates are in lon,lat) into as few complete lines as possible, assuming the end points slop could be up to 150 m, and write the complete segments to separate files called *Map_segment_0001.dat*, *Map_segment_0002.dat*, etc., run

```
gmt connect my_lines.txt -T150e -DMap_segment_%04d.dat
```

1.12.8 Bugs

The line connection does not work if a line only has a single point. However, *gmtconnect* will correctly add the point to the nearest segment. Running *gmtconnect* again on the new set of lines will eventually connect all close lines.

1.12.9 See Also

gmt, *gmt.conf*, *gmtsimplify*, *gmtspatial*, *mapproject*

1.13 gmtconvert

gmtconvert - Convert, Paste, and/or Extract columns from data tables

1.13.1 Synopsis

```
gmtconvert [ table ] [ -A ] [ -D[template] ] [ -E[fillmstride] ] [ -L ] [ -I[tsr] ] [ -Q[~]*selection* ] [ -S[~]"search string" | -S[~]/regexp/[i] ] [ -T ] [ -V[level] ] [ -acol=name[...] ] [ -b[ilo][ncol][type][w][+L]+B ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D[col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -i[cols][l][sscale][offset][,...] ] [ -o[cols][,...] ] [ -s[cols][alr] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.13.2 Description

gmtconvert reads its standard input [or input files] and writes out the desired information to standard output. It can do a combination of seven tasks: (1) convert between binary and ASCII data tables, (2) paste corresponding records from multiple files horizontally into a single file, (3) extract a subset of the available columns, (4) only extract segments whose header record matches a text pattern search, (5) only list segment headers and no data records, (6) extract first and/or last data record for each segment, and (7) reverse the order of items on output. Input (and hence output) may have multiple sub-headers, and ASCII tables may have regular headers as well.

1.13.3 Required Arguments

None

1.13.4 Optional Arguments

table One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-A The records from the input files should be pasted horizontally, not appended vertically [Default]. All files must have the same number of segments and number of rows per segment. Note for binary input, all the files you want to paste must have the same number of columns (as set with **-bi**); ascii tables can have different number of columns.

-D[template] For multiple segment data, dump each segment to a separate output file [Default writes a multiple segment file to stdout]. Append a format template for the individual file names; this template **must** contain a C format specifier that can format an integer argument (the running segment number across all tables); this is usually `%d` but could be `%08d` which gives leading zeros, etc. [Default is `gmtconvert_segment_%d.{txt|bin}`, depending on **-bo**]. Alternatively, give a template with two C format specifiers and we will supply the table number and the segment number within the table to build the file name.

-E[fillmstride] Only extract the first and last record for each segment of interest [Default extracts all records]. Optionally, append **f** or **l** to only extract the first or last record of each segment, respectively. Alternatively, append **mstride** to extract only one out of *stride* records.

- I** Invert the order of items, i.e., output the items in reverse order, starting with the last and ending up with the first item [Default keeps original order]. Append up to three items that should be reversed: **t** will reverse the order of tables, **s** will reverse the order of segments within each table, and **r** will reverse the order of records within each segment [Default].
- L** Only output a listing of all segment header records and no data records (requires ASCII data).
- Q[~]*selection*** Only write segments whose number is included in *selection* and skip all others. Cannot be used with **-S**. The *selection* syntax is *range*[,*range,...]* where each *range* of items is either a single segment *number*, a range of segment numbers *start-stop*, or a range with stepped increments given via *start:step:stop*. A leading **~** will invert the selection and write all segments but the ones listed. Instead of a list of ranges, use **+file** to supply a file list with one *range* per line.
- S[~]"search string" or -S[~]/regexp/[i]** Only output those segments whose header record contains the specified text string. To reverse the search, i.e., to output segments whose headers do *not* contain the specified pattern, use **-S~**. Should your pattern happen to start with **~** you need to escape this character with a backslash [Default output all segments]. Cannot be used with **-Q**. For matching segments based on aspatial values (via OGR/GMT format), give the search string as *varname=value* and we will compare *value* against the value of *varname* for each segment. Note: If the features are polygons then a match of a particular polygon perimeter also means that any associated polygon holes will also be matched. For matching segment headers against extended regular expressions enclose the expression in slashes. Append **i** for caseless matching. For a list of such patterns, give **+file** with one pattern per line. To give a single pattern starting with **+f**, escape it with a backslash.
- T** Suppress the writing of segment headers on output.
- V[level] (more ...)** Select verbosity level [c].
- acol=name[...]** (more ...) Set aspatial column associations *col=name*.
- bi[ncols][type] (more ...)** Select binary input.
- bo[ncols][type] (more ...)** Select binary output. [Default is same as input].
- f[i]lo[colinfo] (more ...)** Specify data types of input and/or output columns.
- g[a]xly|d|X|Y|ID|[col]z[+|-]gap[u] (more ...)** Determine data gaps and line breaks.
- h[i]lo|[n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][ooffset][,...] (more ...)** Select input columns (0 is first column).
- ocols[,...] (more ...)** Select output columns (0 is first column).
- s[cols][alr] (more ...)** Set handling of NaN records.
- :[i]lo (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.13.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (*-bo* if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.13.6 Examples

To convert the binary file *test.b* (single precision) with 4 columns to ASCII:

```
gmt gmtconvert test.b -bi4f > test.dat
```

To convert the multiple segment ASCII table *test.d* to a double precision binary file:

```
gmt gmtconvert test.d -bo > test.b
```

You have an ASCII table with 6 columns and you want to plot column 5 versus column 0. Try

```
gmt gmtconvert table.d -o5,0 | psxy ...
```

If the file instead is the binary file *results.b* which has 9 single-precision values per record, we extract the last column and columns 4-6 and write ASCII with the command

```
gmt gmtconvert results.b -o8,4-6 -bi9s | psxy ...
```

You want to plot the 2nd column of a 2-column file *left.d* versus the first column of a file *right.d*:

```
gmt gmtconvert left.d right.d -A -o1,2 | psxy ...
```

To extract all segments in the file *big_file.d* whose headers contain the string “RIDGE AXIS”, try

```
gmt gmtconvert big_file.d -S"RIDGE AXIS" > subset.d
```

To invert the selection of segments whose headers begin with “profile ” followed by an integer number and any letter between “g” and “l”, try

```
gmt gmtconvert -S~/^profile [0-9]+[g-l]$/
```

To reverse the order of segments in a file without reversing the order of records within each segment, try

```
gmt gmtconvert lots_of_segments.txt -Is > last_segment_first.txt
```

To extract segments 20 to 40 in steps of 2, plus segment 0 in a file, try

```
gmt gmtconvert lots_of_segments.txt -Q0,20:2:40 > my_segments.txt
```

To extract the attribute ELEVATION from an ogr gmt file like this

```
# @VGMT1.0 @GPOINT
...
# @NELEVATION|DISPX|DISPY
# @Tdouble|double|double
# FEATURE_DATA
# @D4.945000|-106500.00000000|-32700.00000000
-9.36890245902635 39.367156766570389
```

do

```
gmt gmtconvert file.gmt -a2=ELEVATION > xyz.dat
```

or just

```
gmt gmtconvert file.gmt -aELEVATION > xyz.dat
```

1.13.7 See Also

gmt, gmtinfo, gmtselect

1.14 gmtdefaults

gmtdefaults - List current GMT default parameters

1.14.1 Synopsis

gmtdefaults [**-D**[**uls**]]

Note: No space is allowed between the option flag and the associated arguments.

1.14.2 Description

gmtdefaults lists the GMT parameter defaults if the option **-D** is used. There are three ways to change some of the settings: (1) Use the command *gmtset*, (2) use any texteditor to edit the file *gmt.conf* in your home, *~/gmt* or current directory (if you do not have this file, run *gmtset -D* to get one with the system default settings), or (3) override any parameter by specifying one or more **--PARAMETER=VALUE** statements on the commandline of any GMT command (**PARAMETER** and **VALUE** are any combination listed below). The first two options are permanent changes until explicitly changed back, while the last option is ephemeral and only applies to the single GMT command that received the override. GMT can provide default values in US or SI units. This choice is determined at compile time.

1.14.3 Required Arguments

None.

1.14.4 Optional Arguments

- D** Print the system GMT defaults to standard output. Append **u** for US defaults or **s** for SI defaults. [**-D** alone gives the version selected at compile time; If **-D** is omitted, the user's currently active defaults are printed.]
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

Your currently active defaults come from the *gmt.conf* file in the current working directory, if present; else from the *gmt.conf* file in your home directory, if present; else from the file *~/gmt/gmt.conf* if present; else from the system defaults set at the time GMT was compiled.

1.14.5 GMT PARAMETERS

Read the *gmt.conf* man page for a full list of the parameters that are user-definable in GMT.

1.14.6 Examples

To get a copy of the GMT parameter defaults in your home directory, run

```
gmt gmtdefaults -D > ~/gmt.conf
```

You may now change the settings by editing this file using a text editor of your choice, or use **gmtset** to change specified parameters on the command line.

1.14.7 Bugs

If you have typographical errors in your *gmt.conf* file(s), a warning message will be issued, and the GMT defaults for the affected parameters will be used.

1.14.8 See Also

gmt, *gmt.conf*, *gmtcolors*, *gmtget*, *gmtset*

1.15 gmtget

gmtget - Get individual GMT default parameters

1.15.1 Synopsis

```
gmtget [ -Gdefaultsfile ] [ -L ] PARAMETER1 [ PARAMETER2 PARAMETER3 ... ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.15.2 Description

gmtget will list the value of one or more GMT default parameters.

1.15.3 Required Arguments

PARAMETER Provide one or several parameters of interest. The current value of those parameters will be written to *stdout*. For a complete listing of available parameters and their meaning, see the *gmt.conf* man page.

1.15.4 Optional Arguments

- Gdefaultsfile** Name of specific *gmt.conf* file to read [Default looks first in current directory, then in your home directory, then in *~/.gmt* and finally in the system defaults].
- L** Return the values of the parameters on separate lines [Default returns all selected parameter values on one line separated by spaces]

1.15.5 Example

To list the value of the parameter PS_COMMENTS:

```
gmt get PS_COMMENTS
```

To get both the values of the parameter MAP_GRID_CROSS_SIZE_PRIMARY and MAP_GRID_CROSS_SIZE_SECONDARY on one line, try

```
gmt get MAP_GRID_CROSS_SIZE_PRIMARY MAP_GRID_CROSS_SIZE_SECONDARY
```

1.15.6 See Also

gmt, *gmt.conf*, *gmtdefaults*, *gmtset*

1.16 gmtinfo

gmtinfo - Get information about data tables

1.16.1 Synopsis

```
gmtinfo [ table ] [ -Aa[fls] ] [ -C ] [ -D[dx/dy] ] [ -ELlHh[hcol] ] [ -I[p/fls]dx/dy/dz...] [ -S[x][y] ] [ -Tdz/col] [ -V[level] ] [ -bi[ncols][type][w][+L][+B] ] [ -f[ilo]colinfo] [ -g[a]x/y/d/X/Y/D][col]z[+]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -ocols[,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.16.2 Description

gmtinfo reads its standard input [or from files] and finds the extreme values in each of the columns. It recognizes NaNs and will print warnings if the number of columns vary from record to record. As an option, **gmtinfo** will find the extent of the first *n* columns rounded up and down to the nearest multiple of the supplied increments. By default, this output will be in the form **-Rw/e/s/n** which can be used directly in the command line for other programs (hence only *dx* and *dy* are needed), or the output will be in

column form for as many columns as there are increments provided. A similar option (**-T**) will provide a **-T $z_{min}/z_{max}/dz$** string for makecpt.

1.16.3 Required Arguments

None.

1.16.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Aafls Specify how the range should be reported. Choose **-Aa** for the range of all files combined, **-Af** to report the range for each file separately, and **-As** to report the range for each segment (in multisegment files) separately. [Default is **-Aa**].

-C Report the min/max values per column in separate columns [Default uses <min/max> format]. When used, users may also use **-o** to limit which output columns should be reported [all].

-D Modifies results obtained by **-I** by shifting the region to better align with the center of the data. Optionally, append granularity for this shift [Default performs an exact shift].

-EL||H|hcol Returns the record whose column *col* contains the minimum (**l**) or maximum (**h**) value. Upper case (**L|H**) works on absolute value of the data. In case of multiple matches, only the first record is returned. If *col* is not specified we default to the last column in the data.

-I[*plfls*]**dx**[*dy*][*dz*...] Report the min/max of the first *n* columns to the nearest multiple of the provided increments (separate the *n* increments by slashes), and output results in the form **-R $w/e/s/n$** (unless **-C** is set). If only one increment is given we also use it for the second column (for backwards compatibility). To override this behavior, use **-I pdx** . If the input *x*- and *y*-coordinates all have the same phase shift relative to the *dx* and *dy* increments then we use those phase shifts in determining the region, and you may use **-r** to switch from gridline-registration to pixel-registration. For irregular data both phase shifts are set to 0 and the **-r** is ignored. Use **-I fdx [/dy]** to report an extended region optimized to give grid dimensions for fastest results in programs using FFTs. Use **-I sdx [/dy]** to report an extended region optimized to give grid dimensions for fastest results in programs like surface. If *dx* is given as - then the actual min/max of the input is given in the **-R** string.

-S[*x*][*y*] Add extra space for error bars. Useful together with **-I** option and when later plotting with **psxy** **-E**. **-Sx** leaves space for horizontal error bars using the values in third (2) column. **-Sy** leaves space for vertical error bars using the values in third (2) column. **-S** or **-Sxy** leaves space for both error bars using the values in third and fourth (2 and 3) columns.

-Tdz[*col*] Report the min/max of the first (0'th) column to the nearest multiple of *dz* and output this in the form **-T $z_{min}/z_{max}/dz$** . To use another column, append *col*. Only works when **-I** is selected.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].

-f[*ilo*]**colinfo** (*more ...*) Specify data types of input and/or output columns.

-g[*a*]*x***y**[*d*]**X****Y**[*D*][*col*]**z**[*+*]**gap**[*u*] (*more ...*) Determine data gaps and line breaks.

-h[*ilo*][*n*][*+***c**][*+***d**][*+***rremark**][*+***rtitle**] (*more ...*) Skip or produce header record(s).

-icols[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).

- r** (*more ...*) Set pixel node registration [gridline].
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.16.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.16.6 Examples

To find the extreme values in the file *ship_gravity.xygd*:

```
gmt info ship_gravity.xygd
```

Output should look like

```
ship_gravity.xygd: N = 6992 <326.125/334.684> <-28.0711/-8.6837> <-47.7/177.6> <0.6/3544.9>
```

To find the extreme values in the file *track.xy* to the nearest 5 units but shifted to within 1 unit of the data center, and use this region to draw a line using *psxy*, run

```
gmt psxy `gmt info -I5 -D1 track.xy` track.xy -Jx1 -B5 -P > track.ps
```

To find the min and max values for each of the first 4 columns, but rounded to integers, and return the result individually for each data file, use

```
gmt info profile_*.txt -C -I1/1/1/1
```

1.16.7 Bugs

The **-I** option does not yet work properly with time series data (e.g., **-f0T**). Thus, such variable intervals as months and years are not calculated. Instead, specify your interval in the same units as the current setting of *TIME_UNIT*.

1.16.8 See Also

gmt, *gmtconvert*

1.17 gmtlogo

gmtlogo - Adding a GMT graphics logo overlay to an illustration

1.17.1 Synopsis

```
gmtlogo dx dy [ -Gfill ] [ -W[pen] ] >> plot.ps
```

Note: No space is allowed between the option flag and the associated arguments.

1.17.2 Description

This script appends the GMT logo as an overlay to an “open” PostScript file. The logo is 2 inches wide and 1 inch high and will be positioned relative to the current plot origin.

1.17.3 Required Arguments

dx,dy Sets the lower-left corner of the logo relative to current plot origin.

1.17.4 Optional Arguments

-Gfill Select color or pattern for filling the underlying box [Default is no fill].

-W[pen] Set pen attributes for the outline of the box [Default is no outline].

1.17.5 See Also

gmt, gmtcolors, psimage

1.18 gmtmath

gmtmath - Reverse Polish Notation (RPN) calculator for data tables

1.18.1 Synopsis

```
gmtmath [ -At_f(t).d[+slr] ] [ -Ccols ] [ -Eeigen ] [ -I ] [ -Nn_col[lt_col] ] [ -Q ]
[ -S[fil] ] [ -Tt_min/t_max/t_inc[+]tfile ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -
f[ilo]colinfo ] [ -g[a]xly|d|X|Y|D|[col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -
icols[I][sscale][offset][,...] ] [ -ocols[,...] ] [ -s[cols][alr] ] operand [ operand ] OPERATOR [ operand
] OPERATOR ... = [ outfile ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.18.2 Description

gmtmath will perform operations like add, subtract, multiply, and divide on one or more table data files or constants using Reverse Polish Notation (RPN) syntax (e.g., Hewlett-Packard calculator-style). Arbitrarily complicated expressions may therefore be evaluated; the final result is written to an output file [or standard output]. Data operations are element-by-element, not matrix manipulations (except where noted). Some operators only require one operand (see below). If no data tables are used in the expression then options **-T**, **-N** can be set (and optionally **-bo** to indicate the data type for binary tables). If STDIN is given, the standard input will be read and placed on the stack as if a file with that content had been given on the command line. By default, all columns except the “time” column are operated on, but this can be changed (see **-C**). Complicated or frequently occurring expressions may be coded as a macro for future use or stored and recalled via named memory locations.

1.18.3 Required Arguments

operand If *operand* can be opened as a file it will be read as an ASCII (or binary, see **-bi**) table data file. If not a file, it is interpreted as a numerical constant or a special symbol (see below). The special argument STDIN means that *stdin* will be read and placed on the stack; STDIN can appear more than once if necessary.

outfile The name of a table data file that will hold the final result. If not given then the output is sent to stdout.

1.18.4 Optional Arguments

-At *f(t).d* Requires **-N** and will partially initialize a table with values from the given file containing *t* and *f(t)* only. The *t* is placed in column *t_col* while *f(t)* goes into column *n_col* - 1 (see **-N**). If used with operators LSQFIT and SVDFIT you can optionally append the modifiers **+s** or **+r** which will write out the model solution or the residuals, respectively [Default writes one column with model coefficients].

-Ccols Select the columns that will be operated on until next occurrence of **-C**. List columns separated by commas; ranges like 1,3-5,7 are allowed. **-C** (no arguments) resets the default action of using all columns except time column (see **-N**). **-Ca** selects all columns, including time column, while **-Cr** reverses (toggles) the current choices. When **-C** is in effect it also controls which columns from a file will be placed on the stack.

-Eeigen Sets the minimum eigenvalue used by operators LSQFIT and SVDFIT [1e-7]. Smaller eigenvalues are set to zero and will not be considered in the solution.

-I Reverses the output row sequence from ascending time to descending [ascending].

-Nn_col[/*t_col*] Select the number of columns and optionally the column number that contains the “time” variable [0]. Columns are numbered starting at 0 [2/0]. If input files are specified then **-N** will add any missing columns.

-Q Quick mode for scalar calculation. Shorthand for **-Ca -N1/0 -T0/0/1**.

-S[fil] Only report the first or last row of the results [Default is all rows]. This is useful if you have computed a statistic (say the **MODE**) and only want to report a single number instead of numerous records with identical values. Append **l** to get the last row and **f** to get the first row only [Default].

-Tt_min/*t_max*/*t_inc*[+]*tfile* Required when no input files are given. Sets the t-coordinates of the first and last point and the equidistant sampling interval for the “time” column (see **-N**). Append **+** if

you are specifying the number of equidistant points instead. If there is no time column (only data columns), give **-T** with no arguments; this also implies **-Ca**. Alternatively, give the name of a file whose first column contains the desired t-coordinates which may be irregular.

- V[level]** (*more ...*) Select verbosity level [c].
- bi[ncols][type]** (*more ...*) Select binary input.
- bo[ncols][type]** (*more ...*) Select binary output. [Default is same as input, but see **-o**]
- f[ilo]colinfo** (*more ...*) Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|D|[col]z[+|-]gap[u]** (*more ...*) Determine data gaps and line breaks.
- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
- icols[l][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
- ocols[,,...]** (*more ...*) Select output columns (0 is first column).
- s[cols][alr]** (*more ...*) Set handling of NaN records.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.18.5 Operators

Choose among the following 143 operators. “args” are the number of input and output arguments.

| Operator | args | Returns |
|----------------|------|--------------------------------------|
| ABS | 1 1 | abs (A) |
| ACOS | 1 1 | acos (A) |
| ACOSH | 1 1 | acosh (A) |
| ACSC | 1 1 | acsc (A) |
| ACOT | 1 1 | acot (A) |
| ADD | 2 1 | A + B |
| AND | 2 1 | B if A == NaN, else A |
| ASEC | 1 1 | asec (A) |
| ASIN | 1 1 | asin (A) |
| ASINH | 1 1 | asinh (A) |
| ATAN | 1 1 | atan (A) |
| ATAN2 | 2 1 | atan2 (A, B) |
| ATANH | 1 1 | atanh (A) |
| BEI | 1 1 | bei (A) |
| BER | 1 1 | ber (A) |
| BITAND | 2 1 | A & B (bitwise AND operator) |
| BITLEFT | 2 1 | A << B (bitwise left-shift operator) |

Continued on next page

Table 1.3 – continued from previous page

| | | |
|------------------|-----|--|
| BITNOT | 1 1 | $\sim A$ (bitwise NOT operator, i.e., return two's complement) |
| BITOR | 2 1 | $A \mid B$ (bitwise OR operator) |
| BITRIGHT | 2 1 | $A \gg B$ (bitwise right-shift operator) |
| BITTEST | 2 1 | 1 if bit B of A is set, else 0 (bitwise TEST operator) |
| BITXOR | 2 1 | $A \wedge B$ (bitwise XOR operator) |
| CEIL | 1 1 | ceil (A) (smallest integer $\geq A$) |
| CHICRIT | 2 1 | Critical value for chi-squared-distribution, with alpha = A and n = B |
| CHIDIST | 2 1 | chi-squared-distribution $P(\chi^2, n)$, with $\chi^2 = A$ and $n = B$ |
| COL | 1 1 | Places column A on the stack |
| CORRCOEFF | 2 1 | Correlation coefficient $r(A, B)$ |
| COS | 1 1 | cos (A) (A in radians) |
| COSD | 1 1 | cos (A) (A in degrees) |
| COSH | 1 1 | cosh (A) |
| COT | 1 1 | cot (A) (A in radians) |
| COTD | 1 1 | cot (A) (A in degrees) |
| CSC | 1 1 | csc (A) (A in radians) |
| CSCD | 1 1 | csc (A) (A in degrees) |
| CPOISS | 2 1 | Cumulative Poisson distribution $F(x, \lambda)$, with $x = A$ and $\lambda = B$ |
| DDT | 1 1 | $d(A)/dt$ Central 1st derivative |
| D2DT2 | 1 1 | $d^2(A)/dt^2$ 2nd derivative |
| D2R | 1 1 | Converts Degrees to Radians |
| DILOG | 1 1 | dilog (A) |
| DIFF | 1 1 | Difference between adjacent elements of A ($A[1]-A[0]$, $A[2]-A[1]$, ..., 0) |
| DIV | 2 1 | A / B |
| DUP | 1 2 | Places duplicate of A on the stack |
| ERF | 1 1 | Error function erf (A) |
| ERFC | 1 1 | Complementary Error function erfc (A) |
| ERFINV | 1 1 | Inverse error function of A |
| EQ | 2 1 | 1 if $A == B$, else 0 |
| EXCH | 2 2 | Exchanges A and B on the stack |
| EXP | 1 1 | exp (A) |
| FACT | 1 1 | $A!$ (A factorial) |
| FCRIT | 3 1 | Critical value for F-distribution, with alpha = A, $n_1 = B$, and $n_2 = C$ |
| FDIST | 3 1 | F-distribution $Q(F, n_1, n_2)$, with $F = A$, $n_1 = B$, and $n_2 = C$ |
| FLIPUD | 1 1 | Reverse order of each column |
| FLOOR | 1 1 | floor (A) (greatest integer $\leq A$) |
| FMOD | 2 1 | $A \% B$ (remainder after truncated division) |
| GE | 2 1 | 1 if $A \geq B$, else 0 |
| GT | 2 1 | 1 if $A > B$, else 0 |
| HYPOT | 2 1 | hypot (A, B) = $\sqrt{A^2 + B^2}$ |
| I0 | 1 1 | Modified Bessel function of A (1st kind, order 0) |
| I1 | 1 1 | Modified Bessel function of A (1st kind, order 1) |
| IFELSE | 3 1 | B if $A \neq 0$, else C |
| IN | 2 1 | Modified Bessel function of A (1st kind, order B) |
| INRANGE | 3 1 | 1 if $B \leq A \leq C$, else 0 |
| INT | 1 1 | Numerically integrate A |
| INV | 1 1 | $1 / A$ |
| ISFINITE | 1 1 | 1 if A is finite, else 0 |
| ISNAN | 1 1 | 1 if $A == \text{NaN}$, else 0 |

Continued on next page

Table 1.3 – continued from previous page

| | | |
|---------------|-----|--|
| J0 | 1 1 | Bessel function of A (1st kind, order 0) |
| J1 | 1 1 | Bessel function of A (1st kind, order 1) |
| JN | 2 1 | Bessel function of A (1st kind, order B) |
| K0 | 1 1 | Modified Kelvin function of A (2nd kind, order 0) |
| K1 | 1 1 | Modified Bessel function of A (2nd kind, order 1) |
| KN | 2 1 | Modified Bessel function of A (2nd kind, order B) |
| KEI | 1 1 | kei (A) |
| KER | 1 1 | ker (A) |
| KURT | 1 1 | Kurtosis of A |
| LE | 2 1 | 1 if A <= B, else 0 |
| LMSSCL | 1 1 | LMS scale estimate (LMS STD) of A |
| LOG | 1 1 | log (A) (natural log) |
| LOG10 | 1 1 | log10 (A) (base 10) |
| LOG1P | 1 1 | log (1+A) (accurate for small A) |
| LOG2 | 1 1 | log2 (A) (base 2) |
| LOWER | 1 1 | The lowest (minimum) value of A |
| LRAND | 2 1 | Laplace random noise with mean A and std. deviation B |
| LSQFIT | 1 0 | Let current table be [A b] return least squares solution $x = A \setminus b$ |
| LT | 2 1 | 1 if A < B, else 0 |
| MAD | 1 1 | Median Absolute Deviation (L1 STD) of A |
| MAX | 2 1 | Maximum of A and B |
| MEAN | 1 1 | Mean value of A |
| MED | 1 1 | Median value of A |
| MIN | 2 1 | Minimum of A and B |
| MOD | 2 1 | A mod B (remainder after floored division) |
| MODE | 1 1 | Mode value (Least Median of Squares) of A |
| MUL | 2 1 | A * B |
| NAN | 2 1 | NaN if A == B, else A |
| NEG | 1 1 | -A |
| NEQ | 2 1 | 1 if A != B, else 0 |
| NORM | 1 1 | Normalize (A) so max(A)-min(A) = 1 |
| NOT | 1 1 | NaN if A == NaN, 1 if A == 0, else 0 |
| NRAND | 2 1 | Normal, random values with mean A and std. deviation B |
| OR | 2 1 | NaN if B == NaN, else A |
| PLM | 3 1 | Associated Legendre polynomial P(A) degree B order C |
| PLMg | 3 1 | Normalized associated Legendre polynomial P(A) degree B order C (geophysical convention) |
| POP | 1 0 | Delete top element from the stack |
| POW | 2 1 | A ^ B |
| PQUANT | 2 1 | The B'th Quantile (0-100%) of A |
| PSI | 1 1 | Psi (or Digamma) of A |
| PV | 3 1 | Legendre function Pv(A) of degree v = real(B) + imag(C) |
| QV | 3 1 | Legendre function Qv(A) of degree v = real(B) + imag(C) |
| R2 | 2 1 | R2 = A^2 + B^2 |
| R2D | 1 1 | Convert Radians to Degrees |
| RAND | 2 1 | Uniform random values between A and B |
| RINT | 1 1 | rint (A) (round to integral value nearest to A) |
| ROTT | 2 1 | Rotate A by the (constant) shift B in the t-direction |
| SEC | 1 1 | sec (A) (A in radians) |
| SECD | 1 1 | sec (A) (A in degrees) |

Continued on next page

Table 1.3 – continued from previous page

| | | |
|--------------|-----|---|
| SIGN | 1 1 | sign (+1 or -1) of A |
| SIN | 1 1 | sin (A) (A in radians) |
| SINC | 1 1 | sinc (A) (sin (pi*A)/(pi*A)) |
| SIND | 1 1 | sin (A) (A in degrees) |
| SINH | 1 1 | sinh (A) |
| SKEW | 1 1 | Skewness of A |
| SQR | 1 1 | A^2 |
| SQRT | 1 1 | sqrt (A) |
| STD | 1 1 | Standard deviation of A |
| STEP | 1 1 | Heaviside step function H(A) |
| STEPT | 1 1 | Heaviside step function H(t-A) |
| SUB | 2 1 | A - B |
| SUM | 1 1 | Cumulative sum of A |
| TAN | 1 1 | tan (A) (A in radians) |
| TAND | 1 1 | tan (A) (A in degrees) |
| TANH | 1 1 | tanh (A) |
| TAPER | 1 1 | Unit weights cosine-tapered to zero within A of end margins |
| TN | 2 1 | Chebyshev polynomial Tn(-1<A<+1) of degree B |
| TCRIT | 2 1 | Critical value for Student's t-distribution, with alpha = A and n = B |
| TDIST | 2 1 | Student's t-distribution A(t,n), with t = A, and n = B |
| UPPER | 1 1 | The highest (maximum) value of A |
| XOR | 2 1 | B if A == NaN, else A |
| Y0 | 1 1 | Bessel function of A (2nd kind, order 0) |
| Y1 | 1 1 | Bessel function of A (2nd kind, order 1) |
| YN | 2 1 | Bessel function of A (2nd kind, order B) |
| ZCRIT | 1 1 | Critical value for the normal-distribution, with alpha = A |
| ZDIST | 1 1 | Cumulative normal-distribution C(x), with x = A |
| ROOTS | 2 1 | Treats col A as f(t) = 0 and returns its roots |

1.18.6 Symbols

The following symbols have special meaning:

| | |
|--------------|--------------------------|
| PI | 3.1415926... |
| E | 2.7182818... |
| EULER | 0.5772156... |
| TMIN | Minimum t value |
| TMAX | Maximum t value |
| TINC | t increment |
| N | The number of records |
| T | Table with t-coordinates |

1.18.7 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with

enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.18.8 Notes On Operators

1. The operators **PLM** and **PLMg** calculate the associated Legendre polynomial of degree L and order M in x which must satisfy $-1 \leq x \leq +1$ and $0 \leq M \leq L$. x , L , and M are the three arguments preceding the operator. **PLM** is not normalized and includes the Condon-Shortley phase $(-1)^M$. **PLMg** is normalized in the way that is most commonly used in geophysics. The C-S phase can be added by using $-M$ as argument. **PLM** will overflow at higher degrees, whereas **PLMg** is stable until ultra high degrees (at least 3000).
2. Files that have the same names as some operators, e.g., **ADD**, **SIGN**, **=**, etc. should be identified by prepending the current directory (i.e., *./*).
3. The stack depth limit is hard-wired to 100.
4. All functions expecting a positive radius (e.g., **LOG**, **KEI**, etc.) are passed the absolute value of their argument.
5. The **DDT** and **D2DT2** functions only work on regularly spaced data.
6. All derivatives are based on central finite differences, with natural boundary conditions.
7. **ROOTS** must be the last operator on the stack, only followed by **=**.

1.18.9 STORE, RECALL and CLEAR

You may store intermediate calculations to a named variable that you may recall and place on the stack at a later time. This is useful if you need access to a computed quantity many times in your expression as it will shorten the overall expression and improve readability. To save a result you use the special operator **STO@label**, where *label* is the name you choose to give the quantity. To recall the stored result to the stack at a later time, use **[RCL]@label**, i.e., **RCL** is optional. To clear memory you may use **CLR@label**. Note that **STO** and **CLR** leave the stack unchanged.

8. The bitwise operators (**BITAND**, **BITLEFT**, **BITNOT**, **BITOR**, **BITRIGHT**, **BITTEST**, and **BITXOR**) convert a table's double precision values to unsigned 64-bit ints to perform the bitwise operations. Consequently, the largest whole integer value that can be stored in a double precision value is 2^{53} or 9,007,199,254,740,992. Any higher result will be masked to fit in the lower 54 bits. Thus, bit operations are effectively limited to 54 bits. All bitwise operators return NaN if given NaN arguments or bit-settings ≤ 0 .
9. **TAPER** will interpret its argument to be a width in the same units as the time-axis, but if no time is provided (i.e., plain data tables) then the width is taken to be given in number of rows.

1.18.10 Macros

Users may save their favorite operator combinations as macros via the file *gmtmath.macros* in their current or user directory. The file may contain any number of macros (one per record); comment lines starting with **#** are skipped. The format for the macros is **name = arg1 arg2 ... arg2 [: comment]** where **name** is how the macro will be used. When this operator appears on the command line we simply replace it with the listed argument list. No macro may call another macro. As an example, the following

macro expects that the time-column contains seafloor ages in Myr and computes the predicted half-space bathymetry:

DEPTH = SQRT 350 MUL 2500 ADD NEG : *usage: DEPTH to return half-space seafloor depths*

Note: Because geographic or time constants may be present in a macro, it is required that the optional comment flag (:) must be followed by a space. As another example, we show a macro **GPSWEEK** which determines which GPS week a timestamp belongs to:

GPSWEEK = 1980-01-06T00:00:00 SUB 86400 DIV 7 DIV FLOOR : GPS week without rollover

1.18.11 Examples

To take the square root of the content of the second data column being piped through **gmtmath** by process1 and pipe it through a 3rd process, use

```
process1 | gmt gmtmath STDIN SQRT = | process3
```

To take log10 of the average of 2 data files, use

```
gmt gmtmath file1.d file2.d ADD 0.5 MUL LOG10 = file3.d
```

Given the file samples.d, which holds seafloor ages in m.y. and seafloor depth in m, use the relation $\text{depth(in m)} = 2500 + 350 * \text{sqrt}(\text{age})$ to print the depth anomalies:

```
gmt gmtmath samples.d T SQRT 350 MUL 2500 ADD SUB = | lpr
```

To take the average of columns 1 and 4-6 in the three data sets sizes.1, sizes.2, and sizes.3, use

```
gmt gmtmath -C1,4-6 sizes.1 sizes.2 ADD sizes.3 ADD 3 DIV = ave.d
```

To take the 1-column data set ages.d and calculate the modal value and assign it to a variable, try

```
gmt set mode_age = `gmt gmtmath -S -T ages.d MODE =`
```

To evaluate the dilog(x) function for coordinates given in the file t.d:

```
gmt gmtmath -Tt.d T DILOG = dilog.d
```

To demonstrate the use of stored variables, consider this sum of the first 3 cosine harmonics where we store and repeatedly recall the trigonometric argument ($2*\pi*T/360$):

```
gmt gmtmath -T0/360/1 2 PI MUL 360 DIV T MUL STO@kT COS @kT 2 MUL COS ADD \  
@kT 3 MUL COS ADD = harmonics.d
```

To use **gmtmath** as a RPN Hewlett-Packard calculator on scalars (i.e., no input files) and calculate arbitrary expressions, use the **-Q** option. As an example, we will calculate the value of Kei $((1 + 1.75)/2.2) + \cos(60)$ and store the result in the shell variable z:

```
set z = `gmt gmtmath -Q 1 1.75 ADD 2.2 DIV 60 COSD ADD KEI =`
```

To use **gmtmath** as a general least squares equation solver, imagine that the current table is the augmented matrix $[A | b]$ and you want the least squares solution x to the matrix equation $A * x = b$. The operator **LSQFIT** does this; it is your job to populate the matrix correctly first. The **-A** option will facilitate this. Suppose you have a 2-column file ty.d with t and $b(t)$ and you would like to fit a the model $y(t) = a + b*t + c*H(t-t_0)$, where H is the Heaviside step function for a given $t_0 = 1.55$. Then, you need a 4-column augmented table loaded with t in column 1 and your observed $y(t)$ in column 3. The calculation becomes

```
gmt gmtmath -N4/1 -Aty.d -C0 1 ADD -C2 1.55 STEPT ADD -Ca LSQFIT = solution.d
```

Note we use the **-C** option to select which columns we are working on, then make active all the columns we need (here all of them, with **-Ca**) before calling **LSQFIT**. The second and fourth columns (col numbers 1 and 3) are preloaded with t and $y(t)$, respectively, the other columns are zero. If you already have a pre-calculated table with the augmented matrix $[A | b]$ in a file (say `lsqsys.d`), the least squares solution is simply

```
gmt gmtmath -T lsqsys.d LSQFIT = solution.d
```

Users must be aware that when **-C** controls which columns are to be active the control extends to placing columns from files as well. Contrast the different result obtained by these very similar commands:

```
echo 1 2 3 4 | gmt math STDIN -C3 1 ADD =
1 2 3 5
```

versus

```
echo 1 2 3 4 | gmt math -C3 STDIN 1 ADD =
0 0 0 5
```

1.18.12 References

Abramowitz, M., and I. A. Stegun, 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, vol. 55, Dover, New York.

Holmes, S. A., and W. E. Featherstone, 2002, A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalized associated Legendre functions. *Journal of Geodesy*, 76, 279-299.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992, *Numerical Recipes*, 2nd edition, Cambridge Univ., New York.

Spanier, J., and K. B. Oldman, 1987, *An Atlas of Functions*, Hemisphere Publishing Corp.

1.18.13 See Also

gmt, *grdmath*

1.19 gmtselect

gmtselect - Select data table subsets based on multiple spatial criteria

1.19.1 Synopsis

```
gmtselect [ table ] [ -Amin_area[min_level/max_level][+as][+r][ppercent] ] [ -Cdist[unit]/ptfile ] [ -Dresolution[+] ] [ -E[fn] ] [ -Fpolygonfile ] [ -I[cfirsz] ] [ -Jparameters ] [ -L[p]dist[unit]/linefile ] [ -Nmaskvalues ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Zmin[lmax][+ccol] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L][B] ] [ -f[ilo]colinfo ] [ -g[a]xlyd[X]Y[D][col]z[+][-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -ocols[,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.19.2 Description

gmtselect is a filter that reads (longitude, latitude) positions from the first 2 columns of *infile*s [or standard input] and uses a combination of 1-6 criteria to pass or reject the records. Records can be selected based on whether or not they are 1) inside a rectangular region (**-R** [and **-J**]), 2) within *dist* km of any point in *ptfile*, 3) within *dist* km of any line in *linefile*, 4) inside one of the polygons in the *polygonfile*, 5) inside geographical features (based on coastlines), or 6) has z-values within a given range. The sense of the tests can be reversed for each of these 6 criteria by using the **-I** option. See option **-:** on how to read (latitude,longitude) files.

1.19.3 Required Arguments

None

1.19.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Amin_area[*/min_level/max_level*][**+as**][**+rll**][**+ppercent**] Features with an area smaller than *min_area* in km² or of hierarchical level that is lower than *min_level* or higher than *max_level* will not be plotted [Default is 0/0/4 (all features)]. Level 2 (lakes) contains regular lakes and wide river bodies which we normally include as lakes; append **+r** to just get river-lakes or **+l** to just get regular lakes. For expert users who wish to print their own Antarctica coastline and islands via *psxy* you can use **+as** to skip all GSHHG features below 60S. Finally, append **+ppercent** to exclude polygons whose percentage area of the corresponding full-resolution feature is less than *percent*. See GSHHG INFORMATION below for more details. Ignored unless **-N** is set.

-Cdist[*unit*]/*ptfile* Pass all records whose location is within *dist* of any of the points in the ASCII file *ptfile*. If *dist* is zero then the 3rd column of *ptfile* must have each point's individual radius of influence. Distances are Cartesian and in user units; specify **-fg** to indicate spherical distances and append a distance unit (see UNITS). Alternatively, if **-R** and **-J** are used then geographic coordinates are projected to map coordinates (in cm, inch, or points, as determined by *PROJ_LENGTH_UNIT*) before Cartesian distances are compared to *dist*.

-Dresolution[**+**] Ignored unless **-N** is set. Selects the resolution of the coastline data set to use ((**f**ull, (**h**igh, (**i**ntermediate, (**l**ow, or (**c**rude)). The resolution drops off by ~80% between data sets. [Default is **l**]. Append (**+**) to automatically select a lower resolution should the one requested not be available [abort if not found]. Note that because the coastlines differ in details it is not guaranteed that a point will remain inside [or outside] when a different resolution is selected.

-E[**fn**] Specify how points exactly on a polygon boundary should be considered. By default, such points are considered to be inside the polygon. Append **n** and/or **f** to change this behavior for the **-F** and **-N** options, respectively, so that boundary points are considered to be outside.

-Fpolygonfile Pass all records whose location is within one of the closed polygons in the multiple-segment file *polygonfile*. For spherical polygons (lon, lat), make sure no consecutive points are separated by 180 degrees or more in longitude. Note that *polygonfile* must be in ASCII regardless of whether **-bi** is used.

-I[**cfirsz**] Reverses the sense of the test for each of the criteria specified:

c select records NOT inside any point's circle of influence.

- f** select records NOT inside any of the polygons.
- l** select records NOT within the specified distance of any line.
- r** select records NOT inside the specified rectangular region.
- s** select records NOT considered inside as specified by **-N** (and **-A**, **-D**).
- z** select records NOT within the range specified by **-Z**.

-Jparameters (more ...) Select map projection.

-L[p]dist[unit]/linefile Pass all records whose location is within *dist* of any of the line segments in the ASCII multiple-segment file *linefile*. If *dist* is zero then we will scan each sub-header in the *pfile* for an embedded **-Ddist** setting that sets each line's individual distance value. Distances are Cartesian and in user units; specify **-fg** to indicate spherical distances append a distance unit (see UNITS). Alternatively, if **-R** and **-J** are used then geographic coordinates are projected to map coordinates (in cm, inch, m, or points, as determined by *PROJ_LENGTH_UNIT*) before Cartesian distances are compared to *dist*. Use **-Lp** to ensure only points whose orthogonal projections onto the nearest line-segment fall within the segments endpoints [Default considers points "beyond" the line's endpoints].

-Nmaskvalues Pass all records whose location is inside specified geographical features. Specify if records should be skipped (s) or kept (k) using 1 of 2 formats:

-Nwet/dry.

-Noccean/land/lake/island/pond.

[Default is s/k/s/k/s (i.e., s/k), which passes all points on dry land].

-R[unit]xmin/xmax/ymin/ymax[r] (more ...) Specify the region of interest. If no map projection is supplied we implicitly set **-Jx1**.

-V[level] (more ...) Select verbosity level [c].

-Zmin[/max][+ccol] Pass all records whose 3rd column (*z*; *col* = 2) lies within the given range or is NaN (use **-s** to skip NaN records). If *max* is omitted then we test if *z* equals *min* instead. Input file must have at least three columns. To indicate no limit on min or max, specify a hyphen (-). If your 3rd column is absolute time then remember to supply **-f2T**. To specify another column, append **+ccol**, and to specify several tests just repeat the **Z** option as many times as you have columns to test. Note: when more than one **Z** option is given then the **Iz** option cannot be used.

-bi[ncols][type] (more ...) Select binary input. [Default is 2 input columns].

-bo[ncols][type] (more ...) Select binary output. [Default is same as input].

-f[ilo]colinfo (more ...) Specify data types of input and/or output columns.

-g[a]x|y|d|X|Y|ID|[col]z[+|-]gap[u] (more ...) Determine data gaps and line breaks.

-h[ilo][n][+c][+d][+rremark][+rtitle] (more ...) Skip or produce header record(s).

-icols[I][sscale][offset][,...] (more ...) Select input columns (0 is first column).

-ocols[,...] (more ...) Select output columns (0 is first column).

-s[cols][alr] (more ...) Set handling of NaN records.

-:[ilo] (more ...) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.19.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.19.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

This note applies to ASCII output only in combination with binary or netCDF input or the **-:** option. See also the note below.

1.19.7 Note On Processing Ascii Input Records

Unless you are using the **-:** option, selected ASCII input records are copied verbatim to output. That means that options like **-foT** and settings like *FORMAT_FLOAT_OUT* and *FORMAT_GEO_OUT* will not have any effect on the output. On the other hand, it allows selecting records with diverse content, including character strings, quoted or not, comments, and other non-numerical content.

1.19.8 Note On Distances

If options **-C** or **-L** are selected then distances are Cartesian and in user units; use **-fg** to imply spherical distances in km and geographical (lon, lat) coordinates. Alternatively, specify **-R** and **-J** to measure projected Cartesian distances in map units (cm, inch, or points, as determined by *PROJ_LENGTH_UNIT*).

This program has evolved over the years. Originally, the **-R** and **-J** were mandatory in order to handle geographic data, but now there is full support for spherical calculations. Thus, **-J** should only be used if you want the tests to be applied on projected data and not the original coordinates. If **-J** is used the distances given via **-C** and **-L** are projected distances.

1.19.9 Note On Segments

Segment headers in the input files are copied to output if one or more records from a segment passes the test. Selection is always done point by point, not by segment.

1.19.10 Examples

To extract the subset of data set that is within 300 km of any of the points in pts.d but more than 100 km away from the lines in lines.d, run

```
gmt gmtselect lonlatfile -fg -C300k/pts.d -L100/lines.d -I1 > subset
```

Here, you must specify **-fg** so the program knows you are processing geographical data.

To keep all points in data.d within the specified region, except the points on land (as determined by the high-resolution coastlines), use

```
gmt gmtselect data.d -R120/121/22/24 -Dh -Nk/s > subset
```

To return all points in quakes.d that are inside or on the spherical polygon lonlatpath.d, try

```
gmt gmtselect quakes.d -Flonlatpath.d -fg > subset1
```

To return all points in stations.d that are within 5 cm of the point in origin.d for a certain projection, try

```
gmt gmtselect stations.d -C5/origin.d -R20/50/-10/20 -JM20c \
--PROJ_LENGTH_UNIT=cm > subset2
```

1.19.11 Gshhs Information

The coastline database is GSHHG (formerly GSHHS) which is compiled from two sources: World Vector Shorelines (WVS) and CIA World Data Bank II (WDBII). In particular, all level-1 polygons (ocean-land boundary) are derived from the more accurate WVS while all higher level polygons (level 2-4, representing land/lake, lake/island-in-lake, and island-in-lake/lake-in-island-in-lake boundaries) are taken from WDBII. Much processing has taken place to convert WVS and WDBII data into usable form for GMT: assembling closed polygons from line segments, checking for duplicates, and correcting for crossings between polygons. The area of each polygon has been determined so that the user may choose not to draw features smaller than a minimum area (see **-A**); one may also limit the highest hierarchical level of polygons to be included (4 is the maximum). The 4 lower-resolution databases were derived from the full resolution database using the Douglas-Peucker line-simplification algorithm. The classification of rivers and borders follow that of the WDBII. See the GMT Cookbook and Technical Reference Appendix K for further details.

1.19.12 See Also

gmt, *gmt.conf*, *gmtconvert*, *gmtsimplify*, *gmtspatial*, *grdlandmask*, *pscoast*

1.20 gmtset

gmtset - Change individual GMT default parameters

1.20.1 Synopsis

gmtset [**-C** | **-D**[*slu*] | **-G***defaultsfile*] [**-[BJRXxYycp]***value*] PARAMETER1 [=] *value1* PARAMETER2 [=] *value2* PARAMETER3 [=] *value3* ...

Note: No space is allowed between the option flag and the associated arguments.

1.20.2 Description

gmtset will adjust individual GMT defaults settings in the current directory's *gmt.conf* file. If no such file exists one will be created. The main purpose of **gmtset** is temporarily to change certain parameters inside a shell script, e.g., set the dots-per-inch to 72, run the script, and reset to 1200 dpi. Optionally, you can specify one or more temporary changes directly on any GMT command line with the syntax **--PARAMETER=VALUE**; such changes are only in effect for that command and do not permanently change the default settings on disk.

1.20.3 Required Arguments

PARAMETER value Provide one or several pairs of parameter/value combinations that you want to modify. For a complete listing of available parameters and their meaning, see the *gmt.conf* man page.

1.20.4 Optional Arguments

-C Convert a *.gmtdefaults4* file created by GMT4 to a *gmt.conf* file used by GMT5. The original file is retained.

-D[*slu*] Modify the GMT defaults based on the system settings. Append **u** for US defaults or **s** for SI defaults. [**-D** alone gives the version selected at compile time]

-G*defaultsfile* Name of specific *gmt.conf* file to read and modify [Default looks first in current directory, then in your home directory, then in *~/gmt* and finally in the system defaults].

-[BJRXxYycp]*value* Set the expansion of any of these shorthand options.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.20.5 Examples

To change annotation font to 12-point Helvetica, select grid-crosses of size 0.1 inch, and set annotation offset to 0.2 cm:

```
gmt set FONT_ANNOT_PRIMARY 12p,Helvetica \
      MAP_GRID_CROSS_SIZE_PRIMARY 0.1i MAP_ANNOT_OFFSET_PRIMARY 0.2c
```

1.20.6 See Also

gmt, *gmt.conf*, *gmtdefaults*, *gmtget*

1.21 gmtsimplify

gmtsimplify - Line reduction using the Douglas-Peucker algorithm

1.21.1 Synopsis

```
gmtsimplify [ table ] -Ttolerance[unit] [ -V[level] ] [ -b[ilo][ncol][type][w][+L][B] ] [ -f[ilo]colinfo ] [ -g[a]xyd|X|Y|D][col]z[+]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][ooffset][,...] ] [ -ocols[,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.21.2 Description

gmtsimplify reads one or more data files and apply the Douglas-Peucker line simplification algorithm. The method recursively subdivides a polygon until a run of points can be replaced by a straight line segment, with no point in that run deviating from the straight line by more than the tolerance. Have a look at this site to get a visual insight on how the algorithm works (http://geometryalgorithms.com/Archive/algorithm_0205/algorithm_0205.htm)

1.21.3 Required Arguments

-T*tolerance*[*unit*] Specifies the maximum mismatch tolerance in the user units. If the data is not Cartesian then append the distance unit (see UNITS).

1.21.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].

-bo[*ncols*][*type*] (*more ...*) Select binary output. [Default is same as input].

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-g[*a*]*x**y**d*|*X*|*Y*|*D*][*col*]*z*[**+**]*gap*[*u*] (*more ...*) Determine data gaps and line breaks.

-h[*ilo*][*n*][**+c**][**+d**][**+rremark**][**+rtitle**] (*more ...*) Skip or produce header record(s).

-i*cols*[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).

- ocols[,...]** (*more ...*) Select output columns (0 is first column).
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.21.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.21.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.21.7 Examples

To reduce the geographic line segment.d using a tolerance of 2 km, run

```
gmt gmtsimplify segment.d -T2k > new_segment.d
```

To reduce the Cartesian lines xylines.d using a tolerance of 0.45 and write the reduced lines to file new_xylines.d, run

```
gmt gmtsimplify xylines.d -T0.45 > new_xylines.d
```

1.21.8 Notes

There is a slight difference in how *gmtsimplify* processes lines versus closed polygons. Segments that are explicitly closed will be considered polygons, otherwise we treat them as line segments. Hence, segments recognized as polygons may reduce to a 3-point polygon with no area; these are suppressed from the output.

1.21.9 Bugs

One known issue with the Douglas-Peucker has to do with crossovers. Specifically, it cannot be guaranteed that the reduced line does not cross itself. Depending on how many lines you are considering it is also possible that reduced lines may intersect other reduced lines. Finally, the current implementation only does Flat Earth calculations even if you specify spherical; **gmtsimplify** will issue a warning and reset the calculation mode to Flat Earth.

1.21.10 References

Douglas, D. H., and T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line of its caricature, *Can. Cartogr.*, **10**, 112-122, 1973.

This implementation of the algorithm has been kindly provided by Dr. Gary J. Robinson, Department of Meteorology, University of Reading, Reading, UK; his subroutine forms the basis for this program.

1.21.11 See Also

gmt, *gmt.conf*, *gmtconnect*, *gmtconvert*, *gmtselect*

1.22 gmtspatial

gmtspatial - Do geospatial operations on lines and polygons

1.22.1 Synopsis

```
gmtspatial [ table ] [ -A[amin_dist][unit] ] [ -C ] [ -D[+ffile][+aamax][+ddmax][+cCcmax][+sfact] ]
[ -E+/- ] [ -I[eli] ] [ -Npfile[+a][+pstart][+r][+z] ] [ -Q[+[unit]] ] [ -R[unit]xmin/xmax/ymin/ymax[r]
] [ -Silslj ] [ -T[clippolygon] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L]B] ] [ -f[ilo]colinfo ]
[ -g[a]x/y/dX/YD][col]z[+/-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][...] ] [ -ocols[...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.22.2 Description

gmtspatial reads one or more data files (which may be multisegment files) that contains closed polygons and operates of these polygons in the specified way. Operations include area calculation, handedness reversals, and polygon intersections.

1.22.3 Required Arguments

None.

1.22.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-A[*amin_dist*][*unit*] Perform spatial nearest neighbor (NN) analysis: Determine the nearest neighbor of each point and report the NN distances and the point IDs involved in each pair (IDs are the input record numbers starting at 0). Use **-Aa** to decimate a data set so that no NN distance is lower than the threshold *min_dist*. In this case we write out the (possibly averaged) coordinates and the updated NN distances and point IDs. A negative point number means the original point was replaced by a weighted average (the absolute ID value gives the ID of the first original point ID to be included in the average.)

-C Clips polygons to the map region, including map boundary to the polygon as needed. The result is a closed polygon (see **-T** for truncation instead). Requires **-R**.

-D[*+f*file][*+a*amax][*+d*dmax][*+c*Ccmax][*+s*fact] Check for duplicates among the input lines or polygons, or, if *file* is given via **+f**, check if the input features already exist among the features in *file*. We consider the cases of exact (same number and coordinates) and approximate matches (average distance between nearest points of two features is less than a threshold). We also consider that some features may have been reversed. Features are considered approximate matches if their minimum distance is less than *dmax* [0] (see UNITS) and their closeness (defined as the ratio between the average distance between the features divided by their average length) is less than *cmax* [0.01]. For each duplicate found, the output record begins with the single letter Y (exact match) or ~ (approximate match). If the two matching segments differ in length by more than a factor of 2 then we consider the duplicate to be either a subset (-) or a superset (+). For polygons we also consider the fractional difference in areas; duplicates must differ by less than *amax* [0.01]. By default, we compute the mean line separation. Use **-+Ccmin** to instead compute the median line separation and therefore a robust closeness value. Also by default we consider all distances between points on one line and another. Append **-+p** to limit the comparison to points that project perpendicularly to points on the other line (and not its extension).

-E+|-] Reset the handedness of all polygons to match the given + (counter-clockwise) or - (clockwise). Implies **-Q+**.

-I[*eli*] Determine the intersection locations between all pairs of polygons. Append **i** to only compute internal (i.e., self-intersecting polygons) crossovers or **e** to only compute external (i.e., between pairs of polygons) crossovers [Default is both].

-N*pfile*[*+a*][*+p*start][*+r*][*+z*] Determine if one (or all, with **+a**) points of each feature in the input data are inside any of the polygons given in the *pfile*. If inside, then report which polygon it is; the polygon ID is either taken from the *aspatial* value assigned to Z, the segment header (first **-Z**, then **-L** are scanned), or it is assigned the running number that is initialized to *start* [0]. By default the input segment that are found to be inside a polygon are written to stdout with the polygon ID encoded in the segment header as **-ZID**. Alternatively, append **+r** to just report which polygon contains a feature or **+z** to have the IDs added as an extra data column on output. Segments that fail to be inside a polygon are not written out. If more than one polygon contains the same segment we skip the second (and further) scenario.

-Q[*+unit*] Measure the area of all polygons or length of line segments. Use **-Q+** to append the area to each polygons segment header [Default simply writes the area to stdout]. For polygons we also compute the centroid location while for line data we compute the mid-point (half-length) position. Append a distance unit to select the unit used (see UNITS). Note that the area will depend on the current setting of *PROJ_ELLIPSOID*; this should be a recent ellipsoid to get accurate results.

- R[*unit*]***west/east/south/north*[*zmin/zmax*][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. Clips polygons to the map region, including map boundary to the polygon as needed. The result is a closed polygon.
- Siljlsu** Spatial processing of polygons. Choose from **-Si** which returns the intersection of polygons (closed), **-Su** which returns the union of polygons (closed), **-Ss** which will split polygons that straddle the Dateline, and **-Sj** which will join polygons that were split by the Dateline. Note: Only **-Ss** has been implemented.
- T[*clippolygon*]** Truncate polygons against the specified polygon given, possibly resulting in open polygons. If no argument is given to **-T** we create a clipping polygon from **-R** which then is required. Note that when the **-R** clipping is in effect we will also look for polygons of length 4 or 5 that exactly match the **-R** clipping polygon.
- V[*level*]** (*more ...*) Select verbosity level [*c*].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. [Default is 2 input columns].
- bo[*ncols*][*type*]** (*more ...*) Select binary output. [Default is same as input].
- f[*ilo*]*colinfo*** (*more ...*) Specify data types of input and/or output columns.
- g[*a*]*xlyldlXlYlD*[*col*]*z*[+]*-*[*gap*][*u*]** (*more ...*) Determine data gaps and line breaks.
- h[*ilo*][*n*][+]*c*][+]*d*][+]*rremark*][+]*rtitle*** (*more ...*) Skip or produce header record(s).
- i[*cols*][1][*sscale*][*ooffset*][,]*...*]** (*more ...*) Select input columns (0 is first column).
- o[*cols*][,]*...*]** (*more ...*) Select output columns (0 is first column).
- :[*ilo*]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.22.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to

a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend + to perform exact geodesic calculations (slower but more accurate).

1.22.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.22.7 Example

To compute the area of all geographic polygons in the multisegment file *polygons.d*, run

```
gmt gmtspatial polygons.d -Q > areas.d
```

Same data, but now orient all polygons to go counter-clockwise and write their areas to the segment headers, run

```
gmt gmtspatial polygons.d -Q+ -E+ > areas.d
```

To determine the intersections between the polygons *A.d* and *B.d*, run

```
gmt gmtspatial A.d B.d -Ce > crossovers.d
```

1.22.8 See Also

gmt, *gmtconvert*, *gmtselect*, *gmtsimplify*

1.23 gmtswitch

gmtswitch - Switching between different GMT versions

1.23.1 Synopsis

```
gmtswitch [ D | version ]
```

1.23.2 Introduction

gmtswitch helps you modify your environment to allow for the switching back and forth between several installed GMT versions, in particular GMT 5 and versions from the GMT 4 series. It works by maintaining a list of directories to GMT installations in a file in your home directory, then manipulates a symbolic link to point to the GMT directory whose executables we wish to use [The Windows version works a bit differently; see WINDOWS below].

1.23.3 Required Arguments

None. If no arguments are given you are presented with a menu of installed GMT versions from 1 to n and you specify which one you wish to switch to.

1.23.4 Optional Arguments

D Select the default GMT version. This is the first entry in the `~/gmtversions` file

version Search for a unique match in the `~/gmtversions` file. If one match is found we switch to that entry; otherwise an error is generated. where module is the name of a GMT program and the options are those that pertain to that particular program.

1.23.5 Setup

If you have official versions installed then running `gmtswitch` the very first time will examine your hard disk starting at `/` and look for directories with GMT4 or GMT5 in the name. This will fail to find the subversion directories and possibly others you have placed elsewhere. The fastest way to get up and running is this:

1. **Edit/Create `~/gmtversions` and add the paths to all GMT installations** you have or care to consider. Each path goes on separate lines and points to the top dir of each distribution, e.g., `/Users/pwessel/UH/RESEARCH/PROJECTS/GMTdev/GMT4.5.7`
2. **In your `.bashrc` or `.[t]cshrc` or wherever you are maintaining your `PATH` or path variable**, remove any directories you have added that contain GMT, and add the new path `$HOME/this_gmt/bin` (might be `$home` for `csh` users). Make sure this path appears before any others that might contain a GMT installation, such as those used by package managers (e.g., `/sw/bin` for `fink`, `/opt/local/bin` for `Macports`, etc.).
3. Make the new path take effect (quit/restart terminal, logout/login, etc).
4. **cd to the most recent GMT directory where a `gmtswitch` version lives**, and run `gmtswitch` with no argument. Select one of the version from the menu.
5. If in `csh` you may have to say `rehash` afterwards.
6. **Type “`psxy -`” and the synopsis should tell you that you got the** correct version. You can now run `gmtswitch` from anywhere; try it out and make sure that you can switch between the versions.

1.23.6 Examples

To switch to GMT version 4.5.7 (assuming it was installed as such and not via a package manager), try

```
gmtswitch GMT4.5.7
```

To switch to the default (your top choice), do

```
gmtswitch D
```

Finally, to select from the menu, just run

```
gmtswitch
```

and pick the one you want.

1.23.7 Beware

GMT remembers where it was installed the first time and uses that dir to find the default GMT share directory. If you move entire GMT installation after compilation then you may have to set GMT_SHARED_DIR to point to the top dir in order for things to work. It is best not to move things after installation.

1.23.8 Windows

Under Windows use gmtswitch.bat which is a batch script that changes the Windows PATH variable so that the BIN directory of the preferred version always comes first. To do that the batch works in two alternative modes:

1 - Permanent mode

2 - Temporary mode

The permanent mode makes use of the free executable program “EditPath” to change the user path in the registry. It’s called permanent because the changes remains until ... next change. See

<http://www.softpedia.com/get/Tweak/Registry-Tweak/EditPath.shtml>

Of course the editpath.exe binary must be in your system’s path as well. WARNING: The path change will not be visible on the shell cmd where it was executed. For the change to be active you will need to open a new cmd window.

The second mode is temporary because the path to the selected GMT binary dir is prepended to the previous path via a shell command line. This modification disappears when the shell cmd window where it was executes is deleted.

It is the user responsibility to set the contents of the G32_32 to G5_64 below to valid paths where the binaries of the different GMT versions are installed Note that it is not mandatory to have all four of them in you computer. For the ones you do not have just let them pointing to nothing e.g.,

```
set G4_64=
```

The permanent mode is the default one (but this can be changed. See edit section) To run in the temporary mode just give a second argument (doesn’t matter what)

Example usage to set a GMT5 64 bits permanent

```
gmtswitch g5_64
```

To temporary set a GMT4 32 bits do

```
gmtswitch g4_32 1
```

Run without arguments to get a “Usage” (for permanent mode)

1.24 gmtvector

gmtvector - Basic manipulation of Cartesian vectors

1.24.1 Synopsis

```
gmtvector [ table ] [ -Am[conf]|vector ] [ -C[i]o ] [ -E ] [ -N ] [ -Svector ] [ -Tald|D|paz|r[arg|R]s|x ]
[ -V[level] ] [ -b[i]o][ncol][type][w][+L+B] ] [ -f[i]o]colinfo ] [ -g[a]x|y|d|X|Y|D|col]z[+-]gap[u] ] [
-h[i]o][n][+c][+d][+rremark][+rtile] ] [ -i[cols]l][sscale][offset][,...] ] [ -o[cols][,...] ] [ -:[i]o ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.24.2 Description

gmtvector reads either (x, y), (x, y, z), (r, theta) or (lon, lat) [or (lat,lon); see **-:**] coordinates from the first 2-3 columns on standard input [or *infile*s]. If **-fg** is selected and only two items are read (i.e., lon, lat) then these coordinates are converted to Cartesian three-vectors on the unit sphere. Otherwise we expect (r, theta) unless **-Ci** is in effect. If no file is found we expect a single vector to be given as argument to **-A**; this argument will also be interpreted as an x/y/[z], lon/lat, or r/theta vector. The input vectors (or the one provided via **-A**) are denoted the prime vector(s). Several standard vector operations (angle between vectors, cross products, vector sums, and vector rotations) can be selected; most require a single second vector, provided via **-S**. The output vectors will be converted back to (lon, lat) or (r, theta) unless **-Co** is set which requests (x, y[, z]) Cartesian coordinates.

1.24.3 Required Arguments

None.

1.24.4 Optional Arguments

table One or more ASCII [or binary, see **-bi**] file containing lon,lat [lat,lon if **-:**] values in the first 2 columns (if **-fg** is given) or (r, theta), or perhaps (x, y[, z]) if **-Ci** is given). If no file is specified, **gmtvector**, will read from standard input.

-Am[conf]|vector Specify a single, primary vector instead of reading *infile*s; see *infile*s for possible vector formats. Alternatively, append **m** to read *infile*s and set the single, primary vector to be the mean resultant vector first. We also compute the confidence ellipse for the mean vector (azimuth of major axis, major axis, and minor axis; for geographic data the axes will be reported in km). You may optionally append the confidence level in percent [95]. These three parameters are reported in the final three output columns.

-C[i]o Select Cartesian coordinates on input and output. Append **i** for input only or **o** for output only; otherwise both input and output will be assumed to be Cartesian [Default is polar r/theta for 2-D data and geographic lon/lat for 3-D].

-E Convert input geographic coordinates from geodetic to geocentric and output geographic coordinates from geocentric to geodetic. Ignored unless **-fg** is in effect, and is bypassed if **-C** is selected.

-N Normalize the resultant vectors prior to reporting the output [No normalization]. This only has an effect if **-Co** is selected.

-S[vector] Specify a single, secondary vector in the same format as the first vector. Required by operations in **-T** that need two vectors (average, bisector, dot product, cross product, and sum).

-Tald|D|paz|s|r[arg|R]x Specify the vector transformation of interest. Append **a** for average, **b** for the pole of the two points bisector, **d** for dot product (use **D** to get angle in degrees between the two vectors), **paz** for the pole to the great circle specified by input vector and the circle's *az* (no

second vector used), **s** for vector sum, **rpar** for vector rotation (here, *par* is a single angle for 2-D Cartesian data and *lon/lat/angle* for a 3-D rotation pole and angle), **R** will instead rotate the fixed secondary vector by the rotations implied by the input records, and **x** for cross-product. If **-T** is not given then no transformation takes place; the output is determined by other options such as **-A**, **-C**, **-E**, and **-N**.

- V[level] (more ...)** Select verbosity level [*c*].
- bi[ncols][type] (more ...)** Select binary input. [Default is 2 or 3 input columns].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|D|[col]z[+|-]gap[u] (more ...)** Determine data gaps and line breaks.
- h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- ocols[,...] (more ...)** Select output columns (0 is first column).
- :[ilo] (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.24.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.24.6 Examples

Suppose you have a file with lon, lat called points.txt. You want to compute the spherical angle between each of these points and the location 133/34. Try

```
gmt gmtvector points.txt -S133/34 -TD -fg > angles.txt
```

To rotate the same points 35 degrees around a pole at 133/34, and output Cartesian 3-D vectors, use

```
gmt gmtvector points.txt -Tr133/34/35 -Co -fg > reconstructed.txt
```

To rotate the point 65/33 by all rotations given in file rots.txt, use

```
gmt gmtvector rots.txt -TR -S64/33 -fg > reconstructed.txt
```

To compute the cross-product between the two Cartesian vectors 0.5/1/2 and 1/0/0.4, and normalizing the result, try

```
gmt gmtvector -A0.5/1/2 -Tx -S1/0/0.4 -N -C > cross.txt
```

To rotate the 2-D vector, given in polar form as $r = 2$ and $\theta = 35$, by an angle of 120, try

```
gmt gmtvector -A2/35 -Tr120 > rotated.txt
```

To find the mid-point along the great circle connecting the points 123/35 and -155/-30, use

```
gmt gmtvector -A123/35 -S-155/-30 -Ta -fg > midpoint.txt
```

To find the mean location of the geographical points listed in points.txt, with its 99% confidence ellipse, use

```
gmt gmtvector points.txt -Am99 -fg > centroid.txt
```

To find the pole corresponding to the great circle that goes through the point -30/60 at an azimuth of 105 degrees, use

```
gmt gmtvector -A-30/60 -Tp105 -fg > pole.txt
```

1.24.7 Rotations

For more advanced 3-D rotations as used in plate tectonic reconstructions, see the GMT “spotter” supplement.

1.24.8 See Also

gmt, *project*, *mapproject*

1.25 gmtwhich

gmtwhich - Find full path to specified files

1.25.1 Synopsis

```
gmtwhich files [ -A ] [ -C ] [ -D ] [ -V[level] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.25.2 Description

gmtwhich reports the full paths to the files given on the command line. We look for the file in (1) the current directory, (2) in \$GMT_USERDIR (if defined), (3) in \$GMT_DATADIR (if defined). If found we print the full path name to the file, just the directory (see **-D**), or a confirmation (see **-C**). The \$GMT_USERDIR and \$GMT_DATADIR environment variables can be colon-separated list of directories, and we search recursively down any directory that ends with / (i.e., /export/data is a single directory whereas /export/data/ will be searched recursively.)

1.25.3 Required Arguments

files One or more file names of any data type (grids, tables, etc.).

1.25.4 Optional Arguments

- A Only consider files that the user has permission to read [Default consider all files found].
- C Instead of reporting the paths, print the confirmation Y if the file is found and N if it is not.
- D Instead of reporting the paths, print the directories that contains the files.
- V[*level*] (*more ...*) Select verbosity level [c].
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

1.25.5 See Also

gmt

1.26 grd2cpt

grd2cpt - Make linear or histogram-equalized color palette table from grid

1.26.1 Synopsis

```
grd2cpt grid [ -A[+]transparency ] [ -Ccptmaster ] [ -D[ilo] ] [ -Enlevels ] [ -F[R|r|l|c ] [ -Gzlo/zhi ] [ -I ] [ -Lminlimit/maxlimit ] [ -M ] [ -N ] [ -Q[ilo] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Szstart/zstop/zinc ] [ -T-|+|_|= ] [ -V[level] ] [ -W ] [ -Z ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.26.2 Description

grd2cpt reads one or more grid files and writes a color palette (cpt) file to standard output. The cpt file is based on an existing master cpt file of your choice, and the mapping from data value to colors is through the data's cumulative distribution function (CDF), so that the colors are histogram equalized. Thus if the grid(s) and the resulting cpt file are used in **grdimage** with a linear projection, the colors will be uniformly distributed in area on the plot. Let z be the data values in the grid. Define $CDF(Z) = (\# \text{ of } z < Z) / (\# \text{ of } z \text{ in grid})$. (NaNs are ignored). These z -values are then normalized to the master cpt file and colors are sampled at the desired intervals.

The color palette includes three additional colors beyond the range of z-values. These are the background color (B) assigned to values lower than the lowest z-value, the foreground color (F) assigned to values higher than the highest z-value, and the NaN color (N) painted wherever values are undefined. For color tables beyond the standard GMT offerings, visit cpt-city: <http://soliton.vm.bytemark.co.uk/pub/cpt-city/>.

If the master cpt file includes B, F, and N entries, these will be copied into the new master file. If not, the parameters **COLOR_BACKGROUND**, **COLOR_FOREGROUND**, and **COLOR_NAN** from the *gmt.conf* file or the command line will be used. This default behavior can be overruled using the options **-D**, **-M** or **-N**.

The color model (RGB, HSV or CMYK) of the palette created by **makecpt** will be the same as specified in the header of the master cpt file. When there is no **COLOR_MODEL** entry in the master cpt file, the **COLOR_MODEL** specified in the *gmt.conf* file or on the command line will be used.

1.26.3 Required Arguments

grid Names of one or more grid files used to derive the color palette table. All grids need to have the same size and dimensions. (See GRID FILE FORMATS below).

1.26.4 Optional Arguments

- A[+]*transparency*** Sets a constant level of transparency (0-100) for all color slices. Prepend **+** to also affect the fore-, back-, and nan-colors [Default is no transparency, i.e., 0 (opaque)].
- C*cptmaster*** Selects the master color table to use in the interpolation. Choose among the built-in tables (type **grd2cpt** to see the list) or give the name of an existing cpt file [Default gives a rainbow cpt file].
- D[ilo]** Select the back- and foreground colors to match the colors for lowest and highest z-values in the output cpt file [Default uses the colors specified in the master file, or those defined by the parameters **COLOR_BACKGROUND**, **COLOR_FOREGROUND**, and **COLOR_NAN**]. Append **i** to match the colors for the lowest and highest values in the input (instead of the output) cpt file.
- E*nlevels*** Create a linear color table by dividing the grid z-range into *nlevels* equidistant slices.
- F[*Rr|lhlc*]** Force output cpt file to written with r/g/b codes, gray-scale values or color name (**R**, default) or r/g/b codes only (**r**), or h-s-v codes (**h**), or c/m/y/k codes (**c**).
- G*zlo/zhi*** Truncate the incoming CPT so that the lowest and highest z-levels are to *zlo* and *zhi*. If one of these equal NaN then we leave that end of the CPT alone. The truncation takes place before any resampling.
- I** Reverses the sense of color progression in the master cpt file. Also exchanges the foreground and background colors, including those specified by the parameters **COLOR_BACKGROUND** and **COLOR_FOREGROUND**.
- L*minlimit/maxlimit*** Limit range of cpt file to *minlimit/maxlimit*, and don't count data outside this range when estimating CDF(Z). [Default uses min and max of data.]
- M** Overrule background, foreground, and NaN colors specified in the master cpt file with the values of the parameters **COLOR_BACKGROUND**, **COLOR_FOREGROUND**, and **COLOR_NAN** specified in the *gmt.conf* file or on the command line. When combined with **-D**, only **COLOR_NAN** is considered.
- N** Do not write out the background, foreground, and NaN-color fields [Default will write them].

- Q**[*ilo*] Selects a logarithmic interpolation scheme [Default is linear]. **-Qi** expects input *z*-values to be $\log_{10}(z)$, assigns colors, and writes out *z* [Default]. **-Qo** takes $\log_{10}(z)$ first, assigns colors, and writes out *z*.
- R**[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.
- Szstart/zstop/zinc** or **-Sn** Set steps in cpt file. Calculate entries in cpt file from *zstart* to *zstop* in steps of (*zinc*). Default chooses arbitrary values by a crazy scheme based on equidistant values for a Gaussian CDF. Use **-Sn** to select *n* points from such a cumulative normal distribution [11].
- T-|+|_|=** Force the color table to be symmetric about zero (from -R to +R). Append flag to set the range R: - for R = |zmin|, + for R = |zmax|, _ for R = min(|zmin|, |zmax|), or = for R = max(|zmin|, |zmax|).
- V** Verbose operation. This will write CDF(Z) estimates to stderr. [Default is silent.]
- W** Do not interpolate the input color table but pick the output colors starting at the beginning of the map. This is particularly useful in combination with a categorical color table. Cannot be used in combination with **-Z**.
- Z** Will create a continuous color palette. [Default is discontinuous, i.e., constant color intervals]
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.26.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.26.6 Examples

Sometimes you don't want to make a `cpt` file (yet) but would find it helpful to know that 90% of your data lie between `z1` and `z2`, something you cannot learn from `grdinfo`. So you can do this to see some points on the CDF(Z) curve (use `-V` option to see more):

```
gmt grd2cpt mydata.nc -V > /dev/null
```

To make a `cpt` file with entries from 0 to 200 in steps of 20, and ignore data below zero in computing CDF(Z), and use the built-in master `cpt` file `relief`, run

```
gmt grd2cpt mydata.nc -Crelief -L0/10000 -S0/200/20 > mydata.cpt
```

1.26.7 See Also

`gmt`, `gmt.conf`, `grdhisteq`, `grdinfo`, `makecpt`

1.27 `grd2rgb`

`grd2rgb` - Write r/g/b grid files from a grid file, a raw RGB file, or SUN rasterfile

1.27.1 Synopsis

```
grd2rgb infile -Gtemplate [ -Ccptfile ] [ -Ixinc[m|s][/yinc[m|s]] ] [ -Llayer ] -R[unit]xmin/xmax/ymin/ymax[r] -V[level] [ -Wwidth/height[n_bytes] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.27.2 Description

`grd2rgb` reads one of three types of input files: (1) A Sun 8-, 24-, or 32-bit raster file; we then write out the red, green, and blue components (0-255 range) to separate grid files. Since the raster file header is limited you may use the **-R**, **-I**, **-r** options to set a complete header record [Default is simply based on the number of rows and columns]. (2) A binary 2-D grid file; we then convert the `z`-values to red, green, blue via the provided `cpt` file. Optionally, only write out one of the `r`, `g`, `b`, layers. (3) A RGB or RGBA raw raster file. Since raw rasterfiles have no header, you have to give the image dimensions via the **-W** option.

1.27.3 Required Arguments

infile The (1) Sun raster file, (2) 2-D binary grid file, or (3) raw raster file to be converted.

-Gtemplate Provide an output name template for the three output grids. The template should be a regular grid file name except it must contain the string `%c` which on output will be replaced by `r`, `g`, or `b`.

1.27.4 Optional Arguments

- Ccptfile* name of the color palette table (for 2-D binary input grid only).
- Ixinc*[*unit*][=|+][*Iyinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *Iy_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.
- Llayer* Output only the specified layer (r, g, or b). [Default outputs all 3 layers].
- R**[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- Wwidth/height**[*n_bytes*] Sets the size of the raw raster file. By default an RGB file (which has 3 bytes/pixel) is assumed. For RGBA files use *n_bytes* = 4. Use **-W** for guessing the image size of a RGB raw file, and **-W=**/*n*/*m* if the raw image is of the RGBA type. Notice that this might be a bit slow because the guessing algorithm makes uses of FFTs.
- r** (*more ...*) Set pixel node registration [gridline].
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.27.5 Examples

To use the color palette *topo.cpt* to create r, g, b component grids from *hawaii_grv.nc* file, use

```
gmt grd2rgb hawaii_grv.nc -Ctopo.cpt -Ghawaii_grv_%c.nc
```

To output the red component from the Sun raster *radiation.ras* file, use

```
gmt grd2rgb radiation.ras -Lr -Gcomp_%c.nc
```

1.27.6 See Also

gmt, *gmt.conf*, *grdedit*, *grdimage*, *grdmath*, *grdview*

1.28 grd2xyz

grd2xyz - Convert grid file to data table

1.28.1 Synopsis

```
grd2xyz grid [ -C[fli] ] [ -N[i]nodata ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -W[weight] ]
[ -Z[flags] ] [ -bo[ncols][type][w][+L|+B] ] [ -f[io]colinfo ] [ -ho[n] ] [ -ocols[...] ] [ -s[cols][alr] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.28.2 Description

grd2xyz reads one or more binary 2-D grid files and writes out xyz-triplets in ASCII [or binary] format to standard output. Modify the precision of the ASCII output format by editing the *FORMAT_FLOAT_OUT* parameter in your *gmt.conf* file or use **--D_FORMAT=***format* on the command line, or choose binary output using single or double precision storage. As an option you may output z-values without the (x,y) coordinates; see **-Z** below.

1.28.3 Required Arguments

grid Names of 2-D binary grid files to be converted. (See GRID FILE FORMATS below.)

1.28.4 Optional Arguments

- C**[*fli*] Replace the x- and y-coordinates on output with the corresponding column and row numbers. These start at 0 (C-style counting); append **f** to start at 1 (Fortran-style counting). Alternatively, append **i** to write just the two columns *index* and *z*, where *index* is the 1-D indexing that GMT uses when referring to grid nodes.
- N**[*i*]*nodata* Output this z-value where the latter equals NaN [Default writes NaN]. Alternatively prepend **i** to do the inverse. That is, to replace the *nodata* values in grid with NaN. Useful to use with the **-s** option.
- R**[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest. Using the **-R** option will select a subsection of the grid. If this subsection exceeds the boundaries of the grid, only the common region will be output.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- W**[*weight*] Write out x,y,z,w, where w is the supplied *weight* (or 1 if not supplied) [Default writes x,y,z only].
- Z**[*flags*] Write a 1-column ASCII [or binary] table. Output will be organized according to the specified ordering convention contained in *flags*. If data should be written by rows, make *flags* start with **T** (op) if first row is y = ymax or **B** (ottom) if first row is y = ymin. Then, append **L** or **R** to indicate

that first element should start at left or right end of row. Likewise for column formats: start with **L** or **R** to position first column, and then append **T** or **B** to position first element in a row. For gridline registered grids: If grid is periodic in x but the outcoming data should not contain the (redundant) column at $x = x_{max}$, append **x**. For grid periodic in y, skip writing the redundant row at $y = y_{max}$ by appending **y**. If the byte-order needs to be swapped, append **w**. Select one of several data types (all binary except **a**):

- **a** ASCII representation of a single item per record
- **c** int8_t, signed 1-byte character
- **u** uint8_t, unsigned 1-byte character
- **h** int16_t, short 2-byte integer
- **H** uint16_t, unsigned short 2-byte integer
- **i** int32_t, 4-byte integer
- **I** uint32_t, unsigned 4-byte integer
- **l** int64_t, long (8-byte) integer
- **L** uint64_t, unsigned long (8-byte) integer
- **f** 4-byte floating point single precision
- **d** 8-byte floating point double precision

Default format is scanline orientation of ASCII numbers: **-ZTLa**. Note that **-Z** only applies to 1-column output.

-bo[ncols][type] (*more ...*) Select binary output. [Default is 3]. This option only applies to xyz output; see **-Z** for z table output.

-f[ilo]colinfo (*more ...*) Specify data types of input and/or output columns. See also **TIME COORDINATES** below. **-h** Output 1 header record based on information in the first grid file header. Ignored if binary output is selected. [Default is no header].

-ocols[,...] (*more ...*) Select output columns (0 is first column).

-s[cols][alr] (*more ...*) Set handling of NaN records.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.28.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in

the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.28.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix *=id[/scale/offset[/nan]]*, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append *?varname* to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of *?* in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The *?varname* suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.28.7 Time Coordinates

Time coordinates in netCDF grids, be it the x, y, or z coordinate, will be recognized as such. The variable’s **unit** attribute is parsed to determine the unit and epoch of the time coordinate in the grid. Values are then converted to the internal time system specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. The default output is relative time in that time system, or absolute time when using the option **-f0T**, **-f1T**, or **-f2T** for x, y, or z coordinate, respectively.

1.28.8 Examples

To edit individual values in the 5’ by 5’ hawaii_grv.nc file, dump the .nc to ASCII:

```
gmt grd2xyz hawaii_grv.nc > hawaii_grv.xyz
```

To write a single precision binary file without the x,y positions from the file raw_data.nc file, using scanline orientation, run

```
gmt grd2xyz raw_data.nc -ZTLf > hawaii_grv.b
```

1.28.9 See Also

gmt.conf, *gmt*, *grdedit*, *grdreformat*, *xyz2grd*

1.29 grdblend

grdblend - Blend several partially over-lapping grids into one large grid

1.29.1 Synopsis

```
grdblend [ blendfile | grid1 grid2 ... ] -Goutgrid [ -Ixinc[unit][=|+][/yinc[unit][=|+]] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Cfillolu ] [ -Nnodata ] [ -Q ] [ -Zscale ] [ -V[level] ] [ -W ] [ -f[ilo]colinfo ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.29.2 Description

grdblend reads a listing of grid files and blend parameters and creates a binary grid file by blending the other grids using cosine-taper weights. **grdblend** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the weighted average value. Any input grid that does not share the final output grid's node registration and grid spacing will automatically be resampled via calls to `grdsample`. Note: Due to the row-by-row i/o nature of operations in `grdblend` we only support the netCDF and native binary grid formats for both input and output.

1.29.3 Required Arguments

-G*outgrid* *outgrid* is the name of the binary output grid file. (See GRID FILE FORMATS below). Only netCDF and native binary grid formats are can be written directly. Other output format choices will be handled by reformatting the output once blending is complete.

-I*xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest.

1.29.4 Optional Arguments

blendfile ASCII file with one record per grid file to include in the blend. Each record may contain up to three items, separated by spaces or tabs: the gridfile name (required), the **-R**-setting for the interior region (optional), and the relative weight *wr* (optional). In the combined weighting scheme, this

grid will be given zero weight outside its domain, weight = *wr* inside the interior region, and a 2-D cosine-tapered weight between those end-members in the boundary strip. However, if a negative *wr* is given then the sense of tapering is inverted (i.e., zero weight inside its domain). If the inner region should instead exactly match the grid region then specify a - instead of the **-R**-setting, or leave it off entirely. Likewise, if a weight *wr* is not specified we default to a weight of 1. If the ASCII *blendfile* file is not given **grdblend** will read standard input. Alternatively, if you have more than one grid file to blend and you wish (a) all input grids to have the same weight (1) and (b) all grids should use their actual region as the interior region, then you may simply list all the grids on the command line instead of providing a *blendfile*. You must specify at least 2 input grids for this mechanism to work. Any grid that is not co-registered with the desired output layout implied by **-R**, **-I** (and **-r**) will first be resampled via **grdsample**. Also, grids that are not in netCDF or native binary format will first be reformatted via **grdreformat**.

- C** Clobber mode: Instead of blending, simply pick the value of one of the grids that covers a node. Select from the following modes: **f** for the first grid to visit a node; **o** for the last grid to visit a node; **l** for the grid with the lowest value, and **u** for the grid with the uppermost value. For modes **f** and **o** the ordering of grids in the *blendfile* will dictate which grid contributes to the final result. Weights and cosine tapering are not considered when clobber mode is active.
- Nnodata** No data. Set nodes with no input grid to this value [Default is NaN].
- Q** Create a header-less grid file suitable for use with **grdraster**. Requires that the output grid file is a native format (i.e., not netCDF).
- V[level] (*more ...*)** Select verbosity level [*c*].
- W** Do not blend, just output the weights used for each node. This option is valid when only one input grid is provided [Default makes the blend].
- Zscale** Scale output values by *scale* before writing to file. [*1*].
- f[ilo]colinfo (*more ...*)** Specify data types of input and/or output columns.
- r (*more ...*)** Set pixel node registration [gridline].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.29.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When writing a netCDF file, the grid is stored by default with the variable name “z”. To specify another variable name *varname*, append *?varname* to the file name. Note that you may need to escape the special meaning of *?* in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes.

1.29.6 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x** **-f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.29.7 Tapering

While the weights computed are tapered from 1 to 0, we are computing weighted averages, so if there is only a single grid given then the weighted output will be identical to the input. If you are looking for a way to taper your data grid, see *grdmath*’s TAPER operator.

1.29.8 Examples

To create a grid file from the four grid files *piece_?.nc*, giving them each the different weights, make the blendfile like this

```
piece_1.nc -R<subregion_1> 1
piece_2.nc -R<subregion_2> 1.5
piece_3.nc -R<subregion_3> 0.9
piece_4.nc -R<subregion_4> 1
```

Then run

```
gmt grdblend blend.job -Gblend.nc -R<full_region> -I<dx/dy> -V
```

To blend all the grids called *MB_*.nc* given them all equal weight, try

```
gmt grdblend MB_*.nc -Gblend.nc -R<full_region> -I<dx/dy> -V
```

1.29.9 Warning on large file sets

While *grdblend* can process any number of files, it works by keeping those files open that are being blended, and close files as soon as they are finished. Depending on your session, many files may remain open at the same time. Some operating systems set fairly modest default limits on how many concurrent files can be open, e.g., 256. If you run into this problem then you can change this limit; see your operating system documentation for how to change system limits.

1.29.10 See Also

gmt, *grd2xyz*, *grdedit* *grdraster*

1.30 grdclip

grdclip - Clip the range of grids

1.30.1 Synopsis

grdclip *ingrid* **-G***outgrid* [**-R**[*unit*]*xmin/xmax/ymin/ymax*[**r**]] [**-S***ahigh/above*] [**-S***blow/below*] [**-S***ilow/high/between*] [**-S***rold/new*] [**-V**[*level*]]

Note: No space is allowed between the option flag and the associated arguments.

1.30.2 Description

grdclip will set values $< low$ to *below* and/or values $> high$ to *above*. You can also specify one or more intervals where all values should be set to IT(between), or replace individual values. Such operations are useful when you want all of a continent or an ocean to fall into one color or gray shade in image processing, when clipping of the range of data values is required, or for reclassification of data values. *above*, *below*, *between*, *old* and *new* can be any number or even NaN (Not a Number). You must choose at least one of the **-S** options. Use **-R** to only extract a subset of the *ingrid* file.

1.30.3 Required Arguments

ingrid The input 2-D binary grid file.

-G*outgrid* *outgrid* is the modified output grid file.

1.30.4 Optional Arguments

-R[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest. Using the **-R** option will select a subsection of *ingrid* grid. If this subsection exceeds the boundaries of the grid, only the common region will be extracted.

-S*ahigh/above* Set all data[i] $> high$ to *above*.

-S*blow/below* Set all data[i] $< low$ to *below*.

-S*ilow/high/between* Set all data[i] $\geq low$ and $\leq high$ to *between*. Repeat the option for as many intervals as are needed.

-S*rold/new* Set all data[i] $== old$ to *new*. This is mostly useful when your data are known to be integer values. Repeat the option for as many replacements as are needed.

-V[*level*] (*more ...*) Select verbosity level [c].

-^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.30.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.30.6 Examples

To set all values > 70 to NaN and all values < 0 to 0 in file `data.nc`:

```
gmt grdclip data.nc -Gnew_data.nc -Sa70/NaN -Sb0/0 -V
```

To reclassify all values in the 25-30 range to 99, those in 35-39 to 55, exchange 17 for 11 and all values < 10 to 0 in file `classes.nc`, try

```
gmt grdclip classes.nc -Gnew_classes.nc -Si25/30/99 -Si35/39/55 -Sr17/11 -Sb10/0 -V
```

1.30.7 See Also

gmt, *grdlandmask*, *grdmask*, *grdmath*, *grd2xyz*, *xyz2grd*

1.31 *grdcontour*

grdcontour - Make contour map using a grid

1.31.1 Synopsis

```
grdcontour grid -C[+]cont_intlcpt -Jparameters [ -A[-|+]annot_int][labelinfo] [ -B[pls]parameters ] [ -D<template> ] [ -F[lr] ] [ -G[d|f|n|l|L|x|X]params ] [ -Jz/Zparameters ] [ -K ] [ -Llow/high
```

**[-O] [-P] [-Qcut] [-Rwest/east/south/north[/zmin/zmax][r]] [-Ssmoothfactor] [-T[+/-]
][gap/length][:labels]] [-U[just/dx/dy/][clabel]] [-V[level]] [-W[+][type]pen] [-X[alcfir][x-
 shift[u]] [-Y[alcfir][y-shift[u]] [-Z[factor[/shift]][p]] [-bo[ncols][type][w][+L|+B]] [-ccopies
] [-ho[n]] [-p[x|y|z]azim/lev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0]] [-t[transp]]**

Note: No space is allowed between the option flag and the associated arguments.

1.31.2 Description

grdcontour reads a 2-D grid file and produces a contour map by tracing each contour through the grid. PostScript code is generated and sent to standard output. Various options that affect the plotting are available. Alternatively, the x/y/z positions of the contour lines may be saved to one or more output files (or stdout) and no plot is produced.

1.31.3 Required Arguments

grid 2-D gridded data set to be contoured. (See GRID FILE FORMATS below).

-C[+]*cont_int* The contours to be drawn may be specified in one of three possible ways:

1. If *cont_int* has the suffix ".cpt" and can be opened as a file, it is assumed to be a color palette table. The color boundaries are then used as contour levels. If the cpt-file has annotation flags in the last column then those contours will be annotated. By default all contours are labeled; use **-A-** to disable all annotations.
2. If *cont_int* is a file but not a cpt-file, it is expected to contain contour levels in column 1 and a C(ontour) OR A(nnotate) in col 2. The levels marked C (or c) are contoured, the levels marked A (or a) are contoured and annotated. Optionally, a third column may be present and contain the fixed annotation angle for this contour level.
3. If no file is found, then *cont_int* is interpreted as a constant contour interval. However, if prepended with the + sign the *cont_int* is taken as meaning draw that single contour. The **-A** option offers the same possibility so they may be used together to plot only one annotated and one non-annotated contour. If **-A** is set and **-C** is not, then the contour interval is set equal to the specified annotation interval.

If a file is given and **-T** is set, then only contours marked with upper case C or A will have tickmarks. In all cases the contour values have the same units as the grid.

-Jparameters (more ...) Select map projection.

1.31.4 Optional Arguments

-A[-][+]*annot_int*[[*labelinfo*] *annot_int* is annotation interval in data units; it is ignored if contour levels are given in a file. [Default is no annotations]. Append - to disable all annotations implied by **-C**. Alternatively prepend + to the annotation interval to plot that as a single contour. The optional *labelinfo* controls the specifics of the label formatting and consists of a concatenated string made up of any of the following control arguments:

+aangle For annotations at a fixed angle, **+an** for contour-normal, or **+ap** for contour-parallel [Default]. For **+ap**, you may optionally append **u** for up-hill and **d** for down-hill cartographic annotations.

- +cdx[/dy]** Sets the clearance between label and optional text box. Append **clip** to specify the unit or % to indicate a percentage of the label font size [15%].
 - +d** Turns on debug which will draw helper points and lines to illustrate the workings of the contour line setup.
 - +e** Delay the plotting of the text. This is used to build a clip path based on the text, then lay down other overlays while that clip path is in effect, then turning of clipping with **psclip -Cs** which finally plots the original text.
 - +ffont** Sets the desired font [Default **FONT_ANNOT_PRIMARY** with its size changed to 9p].
 - +g[*color*]** Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PS_PAGE_COLOR**].
 - +jjust** Sets label justification [Default is MC].
 - +ndx[/dy]** Nudges the placement of labels by the specified amount (append **clip** to specify the units). Increments are considered in the coordinate system defined by the orientation of the contour; use **+N** to force increments in the plot x/y coordinates system [no nudging]. Not allowed with **+v**.
 - +o** Selects rounded rectangular text box [Default is rectangular]. Not applicable for curved text (**+v**) and only makes sense for opaque text boxes.
 - +p[*pen*]** Draws the outline of text boxes [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, style = solid].
 - +rmin_rad** Will not place labels where the contours's radius of curvature is less than *min_rad* [Default is 0].
 - +t[*file*]** Saves contour label x, y, and text to *file* [Contour_labels.txt]. Use **+T** to save x, y, angle, text instead.
 - +uunit** Appends *unit* to all contour labels. [Default is no unit]. If **z** is appended we use the z-unit from the grdf file.
 - +v** Specifies curved labels following the contour [Default is straight labels].
 - +w** Specifies how many (x,y) points will be used to estimate label angles [Default is 10].
 - +=*prefix*** Prepends *prefix* to all contour labels. [Default is no prefix].
- B[*pls*]parameters (*more ...*)** Set map boundary intervals.
- D<*template*>** Dump contours as data line segments; no plotting takes place. Append filename template which may contain C-format specifiers. If no filename template is given we write all lines to stdout. If filename has no specifiers then we write all lines to a single file. If a float format (e.g., %6.2f) is found we substitute the contour z-value. If an integer format (e.g., %06d) is found we substitute a running segment count. If an char format (%c) is found we substitute C or O for closed and open contours. The 1-3 specifiers may be combined and appear in any order to produce the the desired number of output files (e.g., just %c gives two files, just %f would. separate segments into one file per contour level, and %d would write all segments. to individual files; see manual page for more examples.
- F[*l|r*]** Force dumped contours to be oriented so that higher z-values are to the left (**-Fl** [Default]) or right (**-Fr**) as we move along the contour [Default is arbitrary orientation]. Requires **-D**.
- G[*d|fn|l|L|x|X*]params**

The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

ddist[clilp] or **Ddist[dle|flklm|Minls]** For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **f** (foot), **k** (km), **M** (mile), **n** (nautical mile) or **u** (US survey foot), and **d** (arc degree), **m** (arc minute), or **s** (arc second). [Default is 10**c** or 4**i**]. As an option, you can append */fraction* which is used to place the very first label for each contour when the cumulative along-contour distance equals *fraction * dist* [0.25].

ffile.d Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact matches and points outside the region are skipped.

l|Lline1[,line2,...] Give *start* and *stop* coordinates for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the quoted lines. The format of each *line* specification is *start/stop*, where *start* and *stop* are either a specified point *lon/lat* or a 2-character **XY** key that uses the justification format employed in **pstext** to indicate a point on the map, given as [LCR][BMT]. In addition, you can use **Z-**, **Z+** to mean the global minimum and maximum locations in the grid. **L** will interpret the point pairs as defining great circles [Default is straight line].

nn_label Specifies the number of equidistant labels for quoted lines line [1]. Upper case **N** starts labeling exactly at the start of the line [Default centers them along the line]. **N-1** places one justified label at start, while **N+1** places one justified label at the end of quoted lines. Optionally, append */min_dist[clilp]* to enforce that a minimum distance separation between successive labels is enforced.

x|Xxfile.d Reads the multisegment file *xfile.d* and places labels at the intersections between the quoted lines and the lines in *xfile.d*. **X** will resample the lines first along great-circle arcs.

In addition, you may optionally append **+rradius[clilp]** to set a minimum label separation in the x-y plane [no limitation].

-Jz/Zparameters (more ...) Set z-axis scaling; same syntax as **-Jx**.

-K (more ...) Do not finalize the PostScript plot.

-Llow/high Limit range: Do not draw contours for data values below *low* or above *high*.

-O (more ...) Append to existing PostScript plot.

-P (more ...) Select "Portrait" plot orientation.

-Qcut Do not draw contours with less than *cut* number of points [Draw all contours].

-R[unit]xmin/xmax/ymin/ymax[r] (more ...) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax. (more ...)* [Default is region defined in the grid file].

-Ssmoothfactor Used to resample the contour lines at roughly every (*gridbox_size/smoothfactor*) interval.

-T[+|-][gap/length][:[labels]] Will draw tickmarks pointing in the downward direction every *gap* along the innermost closed contours. Append *gap* and tickmark length (append units as **c**, **i**, or **p**) or use

defaults [15p/3p]. User may choose to tick only local highs or local lows by specifying **-T+** or **-T-**, respectively. Append **:labels** to annotate the centers of closed innermost contours (i.e, the local lows and highs). If no *labels* is appended we use - and + as the labels. Appending two characters, **:LH**, will plot the two characters (here, L and H) as labels. For more elaborate labels, separate the two label strings by a comma (e.g., **:lo,hi**). If a file is given by **-C** and **-T** is set, then only contours marked with upper case C or A will have tickmarks [and annotation].

- U[just/dx/dy/][clabel] (more ...)** Draw GMT time stamp logo on plot.
- V[level] (more ...)** Select verbosity level [c].
- W[+][type]pen (more ...)** *type*, if present, can be **a** for annotated contours or **c** for regular contours [Default]. *pen* sets the attributes for the particular line. Default pen for annotated contours: 0.75p,black. Regular contours use pen 0.25p,black. If the **+** flag is prepended then the color of the contour lines are taken from the cpt file (see **-C**). If the **-** flag is prepended then the color from the cpt file is applied both to the contours and the contour annotations.
- X[alclflr][x-shift[u]]**
- Y[alclflr][y-shift[u]] (more ...)** Shift plot origin.
- Z[factor[/shift]][p]** Use to subtract *shift* from the data and multiply the results by *factor* before contouring starts [1/0]. (Numbers in **-A**, **-C**, **-L** refer to values after this scaling has occurred.) Append **p** to indicate that this grid file contains z-values that are periodic in 360 degrees (e.g., phase data, angular distributions) and that special precautions must be taken when determining 0-contours.
- bo[ncols][type] (more ...)** Select binary output.
- ccopies (more ...)** Specify number of plot copies [Default is 1].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- p[xly/z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (more ...)** Select perspective view.
- t[transp] (more ...)** Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.31.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.31.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.31.7 Examples

To contour the file `hawaii_grav.nc` every 25 mGal on a Mercator map at 0.5 inch/degree, annotate every 50 mGal (using `fontsize = 10p`), using 1 degree tickmarks, and draw 30 minute gridlines:

```
gmt grdcontour hawaii_grav.nc -Jm0.5i -C25 -A50+f10p -B1g30m > hawaii_grav.ps
```

To contour the file `image.nc` using the levels in the file `cont.d` on a linear projection at 0.1 cm/x-unit and 50 cm/y-unit, using 20 (x) and 0.1 (y) tickmarks, smooth the contours a bit, use “RMS Misfit” as plot-title, use a thick red pen for annotated contours, and a thin, dashed, blue pen for the rest, and send the output to the default printer:

```
gmt grdcontour image.nc -Jx0.1c/50.0c -Ccont.d -S4 -B20/0.1:."RMS \
  Misfit":-Wathick,red -Wcthinnest,blue,- | lp
```

The labeling of local highs and lows may plot outside the innermost contour since only the mean value of the contour coordinates is used to position the label.

To save the smoothed 100-m contour lines in `topo.nc` and separate them into two multisegment files: `contours_C.txt` for closed and `contours_O.txt` for open contours, try

```
gmt grdcontour topo.nc -C100 -S4 -Dcontours_%c.txt
```

1.31.8 See Also

gmt, *gmt.conf*, *gmtcolors*, *psbasemap*, *grdimage*, *grdview*, *pscontour*

1.32 grdcut

`grdcut` - Extract subregion from a grid

1.32.1 Synopsis

```
grdcut ingrid -Goutgrid -R[unit]xmin/xmax/ymin/ymax[r] [ -N[nodata] ] [ -S[n]lon/lat/radius[unit] ] [ -V[level] ] [ -Z[nr]min/max ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.32.2 Description

grdcut will produce a new *outgrid* file which is a subregion of *ingrid*. The subregion is specified with **-R** as in other programs; the specified range must not exceed the range of *ingrid* (but see **-N**). If in doubt, run **grdinfo** to check range. Alternatively, define the subregion indirectly via a range check on the node values or via distances from a given point. Complementary to **grdcut** there is **grdpaste**, which will join together two grid files along a common edge.

1.32.3 Required Arguments

ingrid This is the input grid file.

-G*outgrid* This is the output grid file.

1.32.4 Optional Arguments

-N[*nodata*] Allow grid to be extended if new **-R** exceeds existing boundaries. Append *nodata* value to initialize nodes outside current region [Default is NaN].

-R[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest. This defines the subregion to be cut out.

-S[**n**]*lon/lat/radius*[*unit*] Specify an origin and radius; append a distance unit (see UNITS) and we determine the corresponding rectangular region so that all grid nodes on or inside the circle are contained in the subset. If **-Sn** is used we set all nodes outside the circle to NaN.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-Z[**n****r**]*min/max* Determine the new rectangular region so that all nodes *outside* this region are also outside the given *z*-range [-inf/+inf]. To indicate no limit on min or max, specify a hyphen (-). Normally, any NaNs encountered are simply skipped and not considered in the decision. Use **-Zn** to consider a NaN to be outside the *z*-range. This means the new subset will be NaN-free. Alternatively, use **-Zr** to consider NaNs to be within the data range. In this case we stop shrinking the boundaries once a NaN is found [Default simply skips NaNs when making the range decision].

-f[**ilo**]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.32.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.32.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.32.7 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x** **-f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.32.8 Examples

Suppose you have used **surface** to grid ship gravity in the region between 148E - 162E and 8N - 32N, and you do not trust the gridding near the edges, so you want to keep only the area between 150E - 160E and 10N - 30N, then:

```
gmt grdcut grav_148_162_8_32.nc -Ggrav_150_160_10_30.nc -R150/160/10/30 -V
```

To return the subregion of a grid such that any boundary strips where all values are entirely above 0 are excluded, try

```
gmt grdcut bathy.nc -Gtrimmed_bathy.nc -Z-/0 -V
```

To return the subregion of a grid that contains all nodes within a distance of 500 km from the point 45,30 try

```
gmt grdcut bathy.nc -Gsubset_bathy.nc -S45/30/500k -V
```

1.32.9 See Also

gmt, grdclip, grdpaste, grdinfo

1.33 grdedit

grdedit - Modify header or content of a grid

1.33.1 Synopsis

```
grdedit grid [ -A ] [ -Dxname/yname/zname/scale/offset/invalid/title/remark ] [ -E ] [ -Ntable ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ -T ] [ -V[level] ] [ -bi[ncols][type][w][+L|+B] ] [ -f[ilo]colinfo ] [ -i[cols][l][sscale][offset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.33.2 Description

grdedit reads the header information in a binary 2-D grid file and replaces the information with values provided on the command line [if any]. As an option, global, geographical grids (with 360 degrees longitude range) can be rotated in the east-west direction, and individual nodal values can be replaced from a table of *x, y, z* values. **grdedit** only operates on files containing a gridheader. Note: If it is important to retain the original data you should work on a copy of that file.

1.33.3 Required Arguments

grid Name of the 2-D grid file to modify. (See GRID FILE FORMATS below).

1.33.4 Optional Arguments

-A If necessary, adjust the file's *x_inc, y_inc* to be compatible with its domain (or a new domain set with **-R**). Older grid files (i.e., created prior to GMT 3.1) often had excessive slop in *x_inc, y_inc* and an adjustment is necessary. Newer files are created correctly.

-D*xname/yname/zname/scale/offset/invalid/title/remark* Give values for *xname, yname, zname* (give the names of those variables and in square bracket their units, e.g., "distance [km]"), *scale* (to multiply grid values after read [normally 1]), *offset* (to add to grid after scaling [normally 0]), *invalid* (a value to represent missing data [NaN]), *title* (anything you like), and *remark* (anything you like). To leave some of these values untouched, leave field blank. Empty fields in the end may be skipped. Alternatively, to allow "/" to be part of one of the values, use any non-alphanumeric

character (and not the equal sign) as separator by both starting and ending with it. For example: **-D**:*xname:yname:zname:scale:offset:invalid:title:remark*: Use quotes to group texts with more than one word. Note that for geographic grids (**-fg**) *xname* and *yname* are set automatically.

- E** Transpose the grid and exchange the *x* and *y* information. Incompatible with the other options.
- Ntable** Read the ASCII (or binary; see **-bi**) file *table* and replace the corresponding nodal values in the grid with these *x,y,z* values.
- R[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest. The new *w/e/s/n* values will replace those in the grid, and the *x_inc*, *y_inc* values are adjusted, if necessary.**
- S** For global, geographical grids only. Grid values will be shifted longitudinally according to the new borders given in **-R**.
- T** Make necessary changes in the header to convert a gridline-registered grid to a pixel-registered grid, or vice-versa. Basically, gridline-registered grids will have their domain extended by half the *x*- and *y*-increments whereas pixel-registered grids will have their domain shrunk by the same amount.
- V[*level*] (*more ...*)** Select verbosity level [*c*].
- bi[*ncols*][*type*] (*more ...*)** Select binary input. [Default is 3 input columns].
- f[*ilo*]*colinfo* (*more ...*)** Specify data types of input and/or output columns.
- h[*ilo*][*n*][**+c**][**+d**][**+rremark**][**+rtitle**] (*more ...*)** Skip or produce header record(s).
- i[*cols*][*sscale*][*ooffset*][*,...*] (*more ...*)** Select input columns (0 is first column).
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +- or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.33.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front

of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.33.6 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the `-f` or `-R` options. For example, both `-f0x` `-f1t` and `-R90w/90e/0t/3t` will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by `TIME_UNIT` and `TIME_EPOCH` in the `gmt.conf` file or on the command line. In addition, the `unit` attribute of the time variable will indicate both this unit and epoch.

1.33.7 Examples

Let us assume the file `data.nc` covers the area 300/310/10/30. We want to change the boundaries from geodetic longitudes to geographic and put a new title in the header. We accomplish this by

```
gmt grdedit data.nc -R-60/-50/10/30 -D////////"Gravity Anomalies"
```

The grid `world.nc` has the limits 0/360/-72/72. To shift the data so that the limits would be -180/180/-72/72, use

```
gmt grdedit world.nc -R-180/180/-72/72 -S
```

The file `junk.nc` was created prior to GMT 3.1 with incompatible `-R` and `-I` arguments. To reset the x- and y-increments we run

```
gmt grdedit junk.nc -A
```

The file `junk.nc` was created prior to GMT 4.1.3 and does not contain the required information to indicate that the grid is geographic. To add this information, run

```
gmt grdedit junk.nc -fg
```

1.33.8 See Also

gmt, *grd2xyz*, *grdinfo xyz2grd*

1.34 grdfft

`grdfft` - Do mathematical operations on grids in the wavenumber (or frequency) domain

1.34.1 Synopsis

```
grdfft ingrid [ ingrid2 ] -Goutfile [ -Aazimuth ] [ -Czlevel ] [ -D[scale]g ] [ -E[rx|y][w]k ] [ -F[rx|y]params ] [ -I[scale]g ] [ -N[flq|s]nx/ny][+adlhl][+enlm][+twidth][+w[suffix]][+z[p]] [ -Sscale ] [ -V[level] ] [ -fg ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.34.2 Description

grdfft will take the 2-D forward Fast Fourier Transform and perform one or more mathematical operations in the frequency domain before transforming back to the space domain. An option is provided to scale the data before writing the new values to an output file. The horizontal dimensions of the grid are assumed to be in meters. Geographical grids may be used by specifying the **-fg** option that scales degrees to meters. If you have grids with dimensions in km, you could change this to meters using *grdedit* or scale the output with *grdmath*.

1.34.3 Required Arguments

ingrid 2-D binary grid file to be operated on. (See GRID FILE FORMATS below). For cross-spectral operations, also give the second grid file *ingrd2*.

-Goutfile Specify the name of the output grid file or the 1-D spectrum table (see **-E**). (See GRID FILE FORMATS below).

1.34.4 Optional Arguments

-Aazimuth Take the directional derivative in the *azimuth* direction measured in degrees CW from north.

-Czlevel Upward (for *zlevel* > 0) or downward (for *zlevel* < 0) continue the field *zlevel* meters.

-D[scale]g Differentiate the field, i.e., take $d(\text{field})/dz$. This is equivalent to multiplying by *kr* in the frequency domain (*kr* is radial wave number). Append a scale to multiply by ($kr * \text{scale}$) instead. Alternatively, append **g** to indicate that your data are geoid heights in meters and output should be gravity anomalies in mGal. [Default is no scale].

-E[r|xy][w[k]] Estimate power spectrum in the radial direction [**r**]. Place **x** or **y** immediately after **-E** to compute the spectrum in the x or y direction instead. No grid file is created. If one grid is given then *f* (i.e., frequency or wave number), *power[f]*, and 1 standard deviation in *power[f]* are written to the file set by **-G** [stdout]. If two grids are given we write *f* and 8 quantities: *Xpower[f]*, *Ypower[f]*, *coherent power[f]*, *noise power[f]*, *phase[f]*, *admittance[f]*, *gain[f]*, *coherency[f]*. Each quantity is followed by its own 1-std dev error estimate, hence the output is 17 columns wide. Append **w** to write wavelength instead of frequency. If your grid is geographic you may further append **k** to scale wavelengths from meter [Default] to km.

-F[r|xy]params Filter the data. Place **x** or **y** immediately after **-F** to filter *x* or *y* direction only; default is isotropic [**r**]. Choose between a cosine-tapered band-pass, a Gaussian band-pass filter, or a Butterworth band-pass filter.

Cosine-taper: Specify four wavelengths *lc/lp/hp/hc* in correct units (see **-fg**) to design a bandpass filter: wavelengths greater than *lc* or less than *hc* will be cut, wavelengths greater than *lp* and less than *hp* will be passed, and wavelengths in between will be cosine-tapered. E.g., **-F1000000/250000/50000/10000 -fg** will bandpass, cutting wavelengths > 1000 km and < 10 km, passing wavelengths between 250 km and 50 km. To make a highpass or lowpass filter, give hyphens (-) for *hp/hc* or *lc/lp*. E.g., **-Fx-/-/50/10** will lowpass *x*, passing wavelengths > 50 and rejecting wavelengths < 10. **-Fy1000/250/-/-** will highpass *y*, passing wavelengths < 250 and rejecting wavelengths > 1000.

Gaussian band-pass: Append *lolhi*, the two wavelengths in correct units (see **-fg**) to design a bandpass filter. At the given wavelengths the Gaussian filter weights will be 0.5. To make a highpass or lowpass filter, give a hyphen (-) for the *hi* or *lo* wavelength, respectively. E.g.,

-F/*30* will lowpass the data using a Gaussian filter with half-weight at 30, while **-F400**/*-* will highpass the data.

Butterworth band-pass: Append *lo*/*hi*/*order*, the two wavelengths in correct units (see **-fg**) and the filter order (an integer) to design a bandpass filter. At the given wavelengths the Butterworth filter weights will be 0.5. To make a highpass or lowpass filter, give a hyphen (-) for the *hi* or *lo* wavelength, respectively. E.g., **-F**/*30*/*2* will lowpass the data using a 2nd-order Butterworth filter, with half-weight at 30, while **-F400**/*-*/*2* will highpass the data.

-I[*scale*g] Integrate the field, i.e., compute `integral_over_z (field * dz)`. This is equivalent to divide by *kr* in the frequency domain (*kr* is radial wave number). Append a *scale* to divide by (*kr* * *scale*) instead. Alternatively, append **g** to indicate that your data set is gravity anomalies in mGal and output should be geoid heights in meters. [Default is no scale].

-N[*flq*|*snx*/*ny*][+a**][**+d**|*hl*l][**+e**|*nl*m][**+t***width*][**+w**[*suffix*]][**+z**[*p*]]** Choose or inquire about suitable grid dimensions for FFT and set optional parameters. Control the FFT dimension:

-Nf will force the FFT to use the actual dimensions of the data.

-Nq will inQUIRE about more suitable dimensions, report those, then continue.

-Ns will present a list of optional dimensions, then exit.

-N*nx*/*ny* will do FFT on array size *nx/ny* (must be \geq grid file size). Default chooses dimensions \geq data which optimize speed and accuracy of FFT. If FFT dimensions $>$ grid file dimensions, data are extended and tapered to zero.

Control detrending of data: Append modifiers for removing a linear trend:

+d: Detrend data, i.e. remove best-fitting linear trend [Default].

+a: Only remove mean value.

+h: Only remove mid value, i.e. $0.5 * (\max + \min)$.

+l: Leave data alone.

Control extension and tapering of data: Use modifiers to control how the extension and tapering are to be performed:

+e extends the grid by imposing edge-point symmetry [Default],

+m extends the grid by imposing edge mirror symmetry

+n turns off data extension.

Tapering is performed from the data edge to the FFT grid edge [100%]. Change this percentage via **+twidth**. When **+n** is in effect, the tapering is applied instead to the data margins as no extension is available [0%].

Control writing of temporary results: For detailed investigation you can write the intermediate grid being passed to the forward FFT; this is likely to have been detrended, extended by point-symmetry along all edges, and tapered. Append **+w**[*suffix*] from which output file name(s) will be created (i.e., *ingrid_prefix.ext*) [tapered], where *ext* is your file extension. Finally, you may save the complex grid produced by the forward FFT by appending **+z**. By default we write the real and imaginary components to *ingrid_*real* ext* and *ingrid_imag.ext*. Append **p** to save instead the polar form of magnitude and phase to files *ingrid_mag.ext* and *ingrid_phase.ext*.

-Scale Multiply each element by *scale* in the space domain (after the frequency domain operations). [Default is 1.0].

- V[*level*] (*more ...*) Select verbosity level [c].
- fg Geographic grids (dimensions of longitude, latitude) will be converted to meters via a “Flat Earth” approximation using the current ellipsoid parameters.
- ^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

1.34.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.34.6 Grid Distance Units

If the grid does not have meter as the horizontal unit, append `+uunit` to the input file name to convert from the specified unit to meter. If your grid is geographic, convert distances to meters by supplying `-fg` instead.

1.34.7 Considerations

netCDF COARDS grids will automatically be recognized as geographic. For other grids geographical grids were you want to convert degrees into meters, select `-fg`. If the data are close to either pole, you should consider projecting the grid file onto a rectangular coordinate system using *grdproject*

1.34.8 Examples

To upward continue the sea-level magnetic anomalies in the file `mag_0.nc` to a level 800 m above sealevel:

```
gmt grdfft mag_0.nc -C800 -V -Gmag_800.nc
```

To transform geoid heights in m (`geoid.nc`) on a geographical grid to free-air gravity anomalies in mGal:

```
gmt grdfft geoid.nc -Dg -V -Ggrav.nc
```

To transform gravity anomalies in mGal (`faa.nc`) to deflections of the vertical (in micro-radians) in the 038 direction, we must first integrate gravity to get geoid, then take the directional derivative, and finally scale radians to micro-radians:

```
gmt grdfft faa.nc -Ig -A38 -S1e6 -V -Gdefl_38.nc
```

Second vertical derivatives of gravity anomalies are related to the curvature of the field. We can compute these as mGal/m² by differentiating twice:

```
gmt grdfft gravity.nc -D -D -V -Ggrav_2nd_derivative.nc
```

To compute cross-spectral estimates for co-registered bathymetry and gravity grids, and report result as functions of wavelengths in km, try

```
gmt grdfft bathymetry.nc gravity.grd -Ewk -fg -V > cross_spectra.txt
```

To examine the pre-FFT grid after detrending, point-symmetry reflection, and tapering has been applied, as well as saving the real and imaginary components of the raw spectrum of the data in `topo.nc`, try

```
gmt grdfft topo.nc -N+w+z -fg -V
```

You can now make plots of the data in `topo_taper.nc`, `topo_real.nc`, and `topo_imag.nc`.

1.34.9 See Also

gmt, *grdedit*, *grdfilter*, *grdmath*, *grdproject*

1.35 *grdfilter*

grdfilter - Filter a grid in the space (or time) domain

1.35.1 Synopsis

```
grdfilter ingrid -Ddistance_flag -Fxwidth[/width2][mode][+qquantile] -Goutgrid [ -  
Ixinc[unit][=|+][/yinc[unit][=|+] ] [ -Nlpr ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -T ] [ -V[level] ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.35.2 Description

grdfilter will filter a *.nc* file in the time domain using one of the selected convolution or non-convolution isotropic or rectangular filters and compute distances using Cartesian or Spherical geometries. The output *.nc* file can optionally be generated as a sub-region of the input (via **-R**) and/or with new increment (via **-I**) or registration (via **-T**). In this way, one may have “extra space” in the input data so that the edges will not be used and the output can be within one-half- width of the input edges. If the filter is low-pass, then the output may be less frequently sampled than the input.

1.35.3 Required Arguments

ingrid The grid file of points to be filtered. (See GRID FILE FORMATS below).

-Ddistance_flag Distance *flag* tells how grid (x,y) relates to filter *width* as follows:

flag = p: grid (px,py) with *width* an odd number of pixels; Cartesian distances.

flag = 0: grid (x,y) same units as *width*, Cartesian distances.

flag = 1: grid (x,y) in degrees, *width* in kilometers, Cartesian distances.

flag = 2: grid (x,y) in degrees, *width* in km, dx scaled by cos(middle y), Cartesian distances.

The above options are fastest because they allow weight matrix to be computed only once. The next three options are slower because they recompute weights for each latitude.

flag = 3: grid (x,y) in degrees, *width* in km, dx scaled by cosine(y), Cartesian distance calculation.

flag = 4: grid (x,y) in degrees, *width* in km, Spherical distance calculation.

flag = 5: grid (x,y) in Mercator **-Jm1** img units, *width* in km, Spherical distance calculation.

-Fwidth[/width2][mode] Sets the filter type. Choose among convolution and non-convolution filters. Use any filter code **x** (listed below) followed by the full diameter *width*. This gives an isotropic filter; append */width2* for a rectangular filter (requires **-Dp** or **-D0**).

Convolution filters (and their codes) are:

(b) Boxcar: All weights are equal.

(c) Cosine Arch: Weights follow a cosine arch curve.

(g) Gaussian: Weights are given by the Gaussian function, where *width* is 6 times the conventional Gaussian sigma.

(f) Custom: Weights are given by the precomputed values in the filter weight grid file *weight*, which must have odd dimensions; also requires **-D0** and output spacing must match input spacing or be integer multiples.

(o) Operator: Weights are given by the precomputed values in the filter weight grid file *weight*, which must have odd dimensions; also requires **-D0** and output spacing must match input spacing or be integer multiples. Weights are assumed to sum to zero so no accumulation of weight sums and normalization will be done.

Non-convolution filters (and their codes) are:

(m) Median: Returns median value. To select another quantile append **+qquantile** in the 0-1 range [Default is 0.5, i.e., median].

(**p**) Maximum likelihood probability (a mode estimator): Return modal value. If more than one mode is found we return their average value. Append - or + to the filter width if you rather want to return the smallest or largest of the modal values. By mode we mean the least median of squares approximation to the mode.

(**l**) Lower: Return the minimum of all values.

(**L**) Lower: Return minimum of all positive values only.

(**u**) Upper: Return maximum of all values.

(**U**) Upper: Return maximum or all negative values only.

In the case of **L|U** it is possible that no data passes the initial sign test; in that case the filter will return NaN.

-Goutgrid *outgrid* is the output grid file of the filter. (See GRID FILE FORMATS below).

1.35.4 Optional Arguments

-Ixinc[unit][=|+][yinc[unit][=|+]] *x_inc* [and optionally *y_inc*] is the output Increment. Append **m** to indicate arc minutes, or **s** to indicate arc seconds. If the new *x_inc*, *y_inc* are NOT integer multiples of the old ones (in the input data), filtering will be considerably slower. [Default: Same as input.]

-Nilplr Determine how NaN-values in the input grid affects the filtered output: Append **i** to ignore all NaNs in the calculation of filtered value [Default], **r** is same as **i** except if the input node was NaN then the output node will be set to NaN (only applies if both grids are co-registered), and **p** which will force the filtered value to be NaN if any grid-nodes with NaN-values are found inside the filter circle.

-R *west*, *east*, *south*, and *north* defines the Region of the output points. [Default: Same as input.]

-T Toggle the node registration for the output grid so as to become the opposite of the input grid [Default gives the same registration as the input grid].

-V[level] (*more ...*) Select verbosity level [*c*].

-f[ilo]colinfo (*more ...*) Specify data types of input and/or output columns.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.35.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify

the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.35.6 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the `-f` or `-R` options. For example, both `-f0x-f1t` and `-R90w/90e/0t/3t` will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by `TIME_UNIT` and `TIME_EPOCH` in the `gmt.conf` file or on the command line. In addition, the `unit` attribute of the time variable will indicate both this unit and epoch.

1.35.7 Examples

Suppose that `north_pacific_etopo5.nc` is a file of 5 minute bathymetry from 140E to 260E and 0N to 50N, and you want to find the medians of values within a 300km radius (600km full width) of the output points, which you choose to be from 150E to 250E and 10N to 40N, and you want the output values every 0.5 degree. Using spherical distance calculations, you need:

```
gmt grdfilter north_pacific_etopo5.nc -Gfiltered_pacific.nc -Fm600 \
-D4 -R150/250/10/40 -I0.5 -V
```

If we instead wanted a high-pass result then one can perform the corresponding low-pass filter using a coarse grid interval as `grdfilter` will resample the result to the same resolution as the input grid so we can compute the residuals, e.g.,

```
gmt grdfilter north_pacific_etopo5.nc -Gresidual_pacific.nc -Fm-600 \
-D4 -R150/250/10/40 -I0.5 -V
```

Here, the `residual_pacific.nc` grid will have the same 5 minute resolution as the original.

To filter the dataset in `ripples.nc` using a custom anisotropic Gaussian filter $\exp(-0.5*r^2)$ whose distances *r* from the center is given by $(2x^2 + y^2 - 2xy)/6$, with major axis at an angle of 63 degrees with the horizontal, try

```
gmt grdmath -R-10/10/-10/10 -I1 X 2 POW 2 MUL Y 2 POW ADD X Y MUL 2 MUL \
SUB 6 DIV NEG 2 DIV EXP DUP SUM DIV = gfilter.nc
gmt grdfilter ripples.nc -Ffgfilter.nc -D0 -Gsmooth.nc -V
```

1.35.8 Limitations

To use the **-D5** option the input Mercator grid must be created by `img2mercgrd` using the **-C** option so the origin of the y-values is the Equator (i.e., $x = y = 0$ correspond to $\text{lon} = \text{lat} = 0$).

1.35.9 See Also

`gmt`, `grdfft` `img2grd`

1.36 `grdgradient`

`grdgradient` - Compute directional derivative or gradient from a grid

1.36.1 Synopsis

```
grdgradient in_grdfile -Gout_grdfile [ -Aazim[/azim2] ] [ -D[c][o][n] ] [ -E[slp]azim/elev[/ambient/diffuse/specular/shine] ] [ -Lflag ] [ -N[e][t][amp]/[sigma/offset] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Sslopefile ] [ -V[level] ] [ -fg] [ -n[b][c][l][n][+a][+bBC][+tthreshold] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.36.2 Description

grdgradient may be used to compute the directional derivative in a given direction (**-A**), or the direction (**-S**) [and the magnitude (**-D**)] of the vector gradient of the data.

Estimated values in the first/last row/column of output depend on boundary conditions (see **-L**).

1.36.3 Required Arguments

in_grdfile 2-D grid file from which to compute directional derivative. (See GRID FILE FORMATS below).

-Gout_grdfile Name of the output grid file for the directional derivative. (See GRID FILE FORMATS below).

1.36.4 Optional Arguments

-A*azim*[/*azim2*] Azimuthal direction for a directional derivative; *azim* is the angle in the x,y plane measured in degrees positive clockwise from north (the +y direction) toward east (the +x direction). The negative of the directional derivative, $-[dz/dx*\sin(azim) + dz/dy*\cos(azim)]$, is found; negation yields positive values when the slope of $z(x,y)$ is downhill in the *azim* direction, the correct sense for shading the illumination of an image (see `grdimage` and `grdview`) by a light source above the x,y plane shining from the *azim* direction. Optionally, supply two azimuths, **-A***azim/azim2*, in which case the gradients in each of these directions are calculated and the one larger in magnitude is retained; this is useful for illuminating data with two directions of lineated structures, e.g., **-A0/270** illuminates from the north (top) and west (left).

- D[c][o][n]** Find the direction of the gradient of the data. By default, the directions are measured clockwise from north, as *azim* in **-A** above. Append **c** to use conventional Cartesian angles measured counterclockwise from the positive x (east) direction. Append **o** to report orientations (0-180) rather than directions (0-360). Append **n** to add 90 degrees to all angles (e.g., to give orientation of lineated features). Unlike **-A**, here we return the up-slope direction of the gradient.
- E[s|p]azim/elev[ambient/diffuse/specular/shine]** Compute Lambertian radiance appropriate to use with *grdimage* and *grdview*. The Lambertian Reflection assumes an ideal surface that reflects all the light that strikes it and the surface appears equally bright from all viewing directions. *azim* and *elev* are the azimuth and elevation of light vector. Optionally, supply *ambient* *diffuse* *specular* *shine* which are parameters that control the reflectance properties of the surface. Default values are: 0.55/0.6/0.4/10 To leave some of the values untouched, specify = as the new value. For example **-E60/30/=0.5** sets the *azim* *elev* and *diffuse* to 60, 30 and 0.5 and leaves the other reflectance parameters untouched. Append **s** to use a simpler Lambertian algorithm. Note that with this form you only have to provide the azimuth and elevation parameters. Append **p** to use the Peucker piecewise linear approximation (simpler but faster algorithm; in this case the *azim* and *elev* are hardwired to 315 and 45 degrees. This means that even if you provide other values they will be ignored.)
- Lflag** Boundary condition *flag* may be *x* or *y* or *xy* indicating data is periodic in range of *x* or *y* or both, or *flag* may be *g* indicating geographical conditions (*x* and *y* are lon and lat). [Default uses “natural” conditions (second partial derivative normal to edge is zero).]
- N[e][t][amp][sigma][offset]** Normalization. [Default: no normalization.] The actual gradients *g* are offset and scaled to produce normalized gradients *gn* with a maximum output magnitude of *amp*. If *amp* is not given, default *amp* = 1. If *offset* is not given, it is set to the average of *g*. **-N** yields $gn = amp * (g - offset) / \max(\text{abs}(g - offset))$. **-Ne** normalizes using a cumulative Laplace distribution yielding $gn = amp * (1.0 - \exp(\text{sqrt}(2) * (g - offset) / sigma))$ where *sigma* is estimated using the L1 norm of (*g* - *offset*) if it is not given. **-Nt** normalizes using a cumulative Cauchy distribution yielding $gn = (2 * amp / PI) * \text{atan}((g - offset) / sigma)$ where *sigma* is estimated using the L2 norm of (*g* - *offset*) if it is not given.
- R[unit]xmin/xmax/ymin/ymax[r] (more ...)** Specify the region of interest. Using the **-R** option will select a subsection of *in_grdfile* grid. If this subsection exceeds the boundaries of the grid, only the common region will be extracted.
- Sslopefile** Name of output grid file with scalar magnitudes of gradient vectors. Requires **-D** but makes **-G** optional.
- V[level] (more ...)** Select verbosity level [*c*].
- fg** Geographic grids (dimensions of longitude, latitude) will be converted to meters via a “Flat Earth” approximation using the current ellipsoid parameters.
- n[blc|lln][+a][+bBC][+c][+threshold] (more ...)** Select interpolation mode for grids.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.36.5 Grid Distance Units

If the grid does not have meter as the horizontal unit, append **+uunit** to the input file name to convert from the specified unit to meter. If your grid is geographic, convert distances to meters by supplying **-fg** instead.

1.36.6 Hints

If you don't know what **-N** options to use to make an intensity file for *grdimage* or *grdview*, a good first try is **-Ne0.6**.

Usually 255 shades are more than enough for visualization purposes. You can save 75% disk space by appending **=nb/a** to the output filename *out_grdfile*.

If you want to make several illuminated maps of subregions of a large data set, and you need the illumination effects to be consistent across all the maps, use the **-N** option and supply the same value of *sigma* and *offset* to **grdgradient** for each map. A good guess is *offset* = 0 and *sigma* found by **grdinfo -L2** or **-L1** applied to an unnormalized gradient grid.

If you simply need the *x*- or *y*-derivatives of the grid, use *grdmath*.

1.36.7 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix **=id[/scale/offset[/nan]]**, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append **?varname** to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of **?** in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The **?varname** suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.36.8 Examples

To make a file for illuminating the data in *geoid.nc* using exp-normalized gradients in the range [-0.6,0.6] imitating light sources in the north and west directions:

```
gmt grdgradient geoid.nc -A0/270 -Ggradients.nc=nb/a -Ne0.6 -V
```

To find the azimuth orientations of seafloor fabric in the file *topo.nc*:

```
gmt grdgradient topo.nc -Dno -Gazimuths.nc -V
```

1.36.9 References

Horn, B.K.P., Hill-Shading and the Reflectance Map, Proceedings of the IEEE, Vol. 69, No. 1, January 1981, pp. 14-47. (<http://people.csail.mit.edu/bkph/papers/Hill-Shading.pdf>)

1.36.10 See Also

gmt, *gmt.conf* *grdhisteq*, *grdmath*, *grdimage*, *grdview*, *grdvector*

1.37 grdhisteq

grdhisteq - Perform histogram equalization for a grid

1.37.1 Synopsis

```
grdhisteq in_grdfile [ -Gout_grdfile ] [ -Cn_cells ] [ -D[file] ] [ -N[norm] ] [ -Q ] -  
R[unit]xmin/xmax/ymin/ymax[r] -V[level]
```

Note: No space is allowed between the option flag and the associated arguments.

1.37.2 Description

grdhisteq allows the user to find the data values which divide a given grid file into patches of equal area. One common use of **grdhisteq** is in a kind of histogram equalization of an image. In this application, the user might have a grid of flat topography with a mountain in the middle. Ordinary gray shading of this file (using *grdimage*/*grdview*) with a linear mapping from topography to graytone will result in most of the image being very dark gray, with the mountain being almost white. One could use **grdhisteq** to write to stdout or file an ASCII list of those data values which divide the range of the data into *n_cells* segments, each of which has an equal area in the image. Using **awk** or *makecpt* one can take this output and build a cpt file; using the cptfile with *grdimage* will result in an image with all levels of gray occurring equally. Alternatively, see *grd2cpt*.

The second common use of **grdhisteq** is in writing a grid with statistics based on some kind of cumulative distribution function. In this application, the output has relative highs and lows in the same (x,y) locations as the input file, but the values are changed to reflect their place in some cumulative distribution. One example would be to find the lowest 10% of the data: Take a grid, run **grdhisteq** and make a grid using *n_cells* = 10, and then contour the result to trace the 1 contour. This will enclose the lowest 10% of the data, regardless of their original values. Another example is in equalizing the output of *grdgradient*. For shading purposes it is desired that the data have a smooth distribution, such as a Gaussian. If you run **grdhisteq** on output from *grdgradient* and make a grid file output with the Gaussian option, you will have a grid whose values are distributed according to a Gaussian distribution with zero mean and unit variance. The locations of these values will correspond to the locations of the input; that is, the most negative output value will be in the (x,y) location of the most negative input value, and so on.

1.37.3 Required Arguments

in_grdfile 2-D grid file to be equalized. (See GRID FILE FORMATS below).

1.37.4 Optional Arguments

-Cn_cells Sets how many cells (or divisions) of data range to make [16].

-D Dump level information to *file*, or standard output if no file is provided.

-Gout_grdfile Name of output 2-D grid file. Used with **-N** only. (See GRID FILE FORMATS below).

-N[norm] Gaussian output. Use with **-G** to make an output grid with standard normal scores. Append *norm* to force the scores to fall in the <-1,+1> range [Default is standard normal scores].

-Q Use quadratic intensity scaling. [Default is linear].

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest. Using the **-R** option will select a subsection of *in_grdfile* grid. If this subsection exceeds the boundaries of the grid, only the common region will be extracted.

-V[level] (*more ...*) Select verbosity level [c].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.37.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.37.6 Examples

To find the height intervals that divide the file `heights.nc` into 16 divisions of equal area:

```
gmt grdhisteq heights.nc -C16 -D > levels.d
```

To make the poorly distributed intensities in the file `raw_intens.nc` suitable for use with `grdimage` or `grdview`, run

```
gmt grdhisteq raw_intens.nc -Gsmooth_intens.nc -N -V
```

1.37.7 Restrictions

If you use `grdhisteq` to make a Gaussian output for gradient shading in `grdimage` or `grdview`, you should be aware of the following: the output will be in the range $[-x, x]$, where x is based on the number of data in the input grid ($n_x * n_y$) and the cumulative Gaussian distribution function $F(x)$. That is, let $N = n_x * n_y$. Then x will be adjusted so that $F(x) = (N - 1 + 0.5)/N$. Since about 68% of the values from a standard normal distribution fall within ± 1 , this will be true of the output grid. But if N is very large, it is possible for x to be greater than 4. Therefore, with the `grdview` program clipping gradients to the range $[-1, 1]$, you will get correct shading of 68% of your data, while 16% of them will be clipped to -1 and 16% of them clipped to $+1$. If this makes too much of the image too light or too dark, you should take the output of `grdhisteq` and rescale it using `grdmath` and multiplying by something less than 1.0, to shrink the range of the values, thus bringing more than 68% of the image into the range $[-1, 1]$. Alternatively, supply a normalization factor with `-N`.

1.37.8 See Also

`gmt`, `gmt.conf`, `grd2cpt`, `grdgradient`, `grdimage`, `grdmath`, `grdview`, `makecpt`

1.38 `grdimage`

`grdimage` - Project grids or images and plot them on maps

1.38.1 Synopsis

```
grdimage grd_z | grd_r grd_g grd_b [ -Aout_img=driver ] [ -Ccptfile ] [ -D[r] ] -Jparameters [ -U[just/dx/dy]/[clabel] ] [ -G[fb]color ] [ -Iintensfile|intensity ] [ -Jz/Zparameters ] [ -K ] [ -M ] [ -N ] [ -O ] [ -P ] [ -Q ] [ -Rwest/east/south/north[zmin/zmax]][r] ] [ -T ] [ -U[just/dx/dy]/[clabel] ] [ -V[level] ] [ -X[alcflr]][x-shift[u]] ] [ -Y[alcflr]][y-shift[u]] ] [ -ccopies ] [ -f[ilo]colinfo ] [ -n[blc|lln]][+a][+bBC][+tthreshold] ] [ -p[x|y|z]azim/elev[zlevel]][+wlon0/lat0][z0]][+vx0/y0] ] [ -tr ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.38.2 Description

`grdimage` reads one 2-D grid file and produces a gray-shaded (or colored) map by plotting rectangles centered on each grid node and assigning them a gray-shade (or color) based on the z -value. Alternatively, `grdimage` reads three 2-D grid files with the red, green, and blue components directly (all must be in the 0-255 range). Optionally, illumination may be added by providing a file with intensities in the

(-1,+1) range. Values outside this range will be clipped. Such intensity files can be created from the grid using *grdgradient* and, optionally, modified by *grdmath* or *grdhisteq*. Yet as a third alternative available when GMT is build with GDAL support the *grd_z* file can be an image referenced or not (than see **-Dr**). In this case the images can be illuminated with the file provided via the **-I** option. Here if image has no coordinates those of the intensity file will be used.

When using map projections, the grid is first resampled on a new rectangular grid with the same dimensions. Higher resolution images can be obtained by using the **-E** option. To obtain the resampled value (and hence shade or color) of each map pixel, its location is inversely projected back onto the input grid after which a value is interpolated between the surrounding input grid values. By default bi-cubic interpolation is used. Aliasing is avoided by also forward projecting the input grid nodes. If two or more nodes are projected onto the same pixel, their average will dominate in the calculation of the pixel value. Interpolation and aliasing is controlled with the **-n** option.

The **-R** option can be used to select a map region larger or smaller than that implied by the extent of the grid.

A (color) PostScript file is output.

1.38.3 Required Arguments

grd_z | *grd_r* *grd_g* *grd_b* 2-D gridded data set (or red, green, blue grids) to be imaged (See GRID FILE FORMATS below.)

-Jparameters (*more ...*) Select map projection.

1.38.4 Optional Arguments

-Aout_img=driver With GDAL aware versions: save image in a raster format instead of PostScript. Append *out_img=driver* to select the file name and image format. The *driver* is the driver code name used by GDAL. For example, **-Aimg.tif=GTiff** will write a GeoTiff image if the subset of GMT syntax projections that is currently possible to translate into the PROJ4 syntax allows it, or a plain tiff file otherwise. Note: any vector elements are lost.

-B[pls]parameters (*more ...*) Set map boundary intervals.

-Ccptfile Name of the color palette table (for *grd_z* only). Alternatively, supply the name of a GMT color master CPT [rainbow] and let **grdimage** automatically determine a 16-level continuous CPT from the grid's z-range.

-D[r] Specifies that the grid supplied is an image file to be read via GDAL. Obviously this option will work only with GMT versions built with GDAL support. The image can be indexed or true color (RGB) and can be an URL of a remotely located file. That is **-D** <http://www.somewhere.com/image.jpg> is a valid file syntax. Note, however, that to use it this way you must not be blocked by a proxy. If you are, chances are good that it can work by setting the environmental variable *http_proxy* with the value 'your_proxy:port' Append **r** to use the region specified by **-R** to apply to the image. For example, if you have used **-Rd** then the image will be assigned the limits of a global domain. The interest of this mode is that you can project a raw image (an image without referencing coordinates).

-Ei[*dpi*] Sets the resolution of the projected grid that will be created if a map projection other than Linear or Mercator was selected [100]. By default, the projected grid will be of the same size (rows and columns) as the input file. Specify **i** to use the PostScript image operator to interpolate the image at the device resolution.

- G[flb]color** This option only applies when the resulting image otherwise would consist of only two colors: black (0) and white (255). If so, this option will instead use the image as a transparent mask and paint the mask (or its inverse, with **-Gb**) with the given color combination.
 - lintensfile/intensity** Gives the name of a grid file with intensities in the (-1,+1) range, or a constant intensity to apply everywhere. [Default is no illumination].
 - Jz/Zparameters (more ...)** Set z-axis scaling; same syntax as **-Jx**.
 - K (more ...)** Do not finalize the PostScript plot.
 - M** Force conversion to monochrome image using the (television) YIQ transformation. Cannot be used with **-Q**.
 - N** Do not clip the image at the map boundary (only relevant for non-rectangular maps).
 - O (more ...)** Append to existing PostScript plot.
 - P (more ...)** Select "Portrait" plot orientation.
 - Q** Make grid nodes with z = NaN transparent, using the colormasking feature in PostScript Level 3 (the PS device must support PS Level 3).
 - R[unit]xmin/xmax/ymin/ymax[r] (more ...)** Specify the region of interest.
- For perspective view **p**, optionally append */zmin/zmax. (more ...)* You may ask for a larger *w/e/s/n* region to have more room between the image and the axes. A smaller region than specified in the grid file will result in a subset of the grid [Default is the region given by the grid file].
- U[just/dx/dy/][clabel] (more ...)** Draw GMT time stamp logo on plot.
 - V[level] (more ...)** Select verbosity level [c].
 - X[alclfr][x-shift[u]]**
 - Y[alclfr][y-shift[u]] (more ...)** Shift plot origin.
 - ccopies (more ...)** Specify number of plot copies [Default is 1].
 - f[i]o[colinfo] (more ...)** Specify data types of input and/or output columns.
 - n[blclln][+a][+bBC][+c][+threshold] (more ...)** Select interpolation mode for grids.
 - p[x|y|z]azim/elev[/zlevel][+wlon0/lat0/z0][+vx0/y0] (more ...)** Select perspective view.
 - t[transp] (more ...)** Set PDF transparency level in percent.
 - ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
 - + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
 - ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
 - version** Print GMT version and exit.
 - show-datadir** Print full path to GMT share directory and exit.

1.38.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.38.6 Imaging Grids With Nans

Be aware that if your input grid contains patches of NaNs, these patches can become larger as a consequence of the resampling that must take place with most map projections. Because **grdimage** uses the PostScript colorimage operator, for most non-linear projections we must resample your grid onto an equidistant rectangular lattice. If you find that the NaN areas are not treated adequately, consider (a) use a linear projection, or (b) use *grdview -Ts* instead.

1.38.7 Consequences of grid resampling

Except for Cartesian cases, we need to resample your geographic grid onto an equidistant projected grid. In doing so various algorithms come into play that projects data from one lattice to another while avoiding anti-aliasing, leading to possible distortions. One expected effect of resampling with splines is the tendency for the new resampled grid to slightly exceed the global min/max limits of the original grid. If this is coupled with tight CPT file limits you may find that some map areas may show up with fore- or background color due to the resampling. In that case you have two options: (1) Modify your CPT file to fit the resampled extrema (reported with **-V**) or (2) Impose clipping of resampled values so they do not exceed the input min/max values (add **+c** to your **-n** option).

1.38.8 Examples

For a quick-and-dirty color map of the data in the file `stuff.nc`, with the maximum map dimension limited to be 6 inches, try

```
gmt grdimage stuff.nc -JX6i+ > quick.ps
```

To gray-shade the file `hawaii_grav.nc` with shades given in `shades.cpt` on a Lambert map at 1.5 cm/degree along the standard parallels 18 and 24, and using 1 degree tickmarks:

```
gmt grdimage hawaii_grav.nc -Jl18/24/1.5c -Cshades.cpt -B1 > hawaii_grav_image.ps
```

To create an illuminated color PostScript plot of the gridded data set `image.nc`, using the intensities provided by the file `intens.nc`, and color levels in the file `colors.cpt`, with linear scaling at 10 inch/x-unit, tickmarks every 5 units:

```
gmt grdimage image.nc -Jx10i -Ccolors.cpt -Iintens.nc -B5 > image.ps
```

To create an false color PostScript plot from the three grid files `red.nc`, `green.nc`, and `blue.nc`, with linear scaling at 10 inch/x-unit, tickmarks every 5 units:

```
gmt grdimage red.nc green.nc blue.nc -Jx10i -B5 > rgbimage.ps
```

When GDAL support is built in: To create a sinusoidal projection of a remotely located Jessica Rabbit

```
gmt grdimage -JI15c -Rd -Dr
http://larryfire.files.wordpress.com/2009/07/untooned_jessicarabbit.jpg \
-P > jess.ps
```

1.38.9 See Also

gmt, *gmt.conf*, *grd2rgb*, *grdcontour*, *grdview*, *grdgradient*, *grdhisteq*

1.39 grdinfo

`grdinfo` - Extract information from grids

1.39.1 Synopsis

```
grdinfo grdfiles [ -C ] [ -F ] [ -I[dx/dy]|-lb ] [ -L[0|1|2] ] [ -M ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -T[s]dz ] [ -V[level] ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.39.2 Description

grdinfo reads a 2-D binary grid file and reports metadata and various statistics for the (*x,y,z*) data in the grid file(s). The output information contains the minimum/maximum values for *x*, *y*, and *z*, where the min/max of *z* occur, the *x*- and *y*-increments, and the number of *x* and *y* nodes, and [optionally] the mean, standard deviation, and/or the median, L1 scale of *z*, and number of nodes set to NaN. We also report if the grid is pixel- or gridline-registered and if it is a Cartesian or Geographic data set (based on metadata in the file).

1.39.3 Required Arguments

grdfile The name of one or several 2-D grid files. (See GRID FILE FORMATS below.)

1.39.4 Optional Arguments

- C** Formats the report using tab-separated fields on a single line. The output is *w e s n z0 z1 dx dy nx ny [x0 y0 x1 y1] [med scale] [mean std rms] [n_nan]*. The data in brackets are output only if the corresponding options **-M**, **-L1**, **-L2**, and **-M** are used, respectively. If the **-I** option is used, the output format is instead *NF w e s n z0 z1*, where *NF* is the total number of grids read and *w e s n* are rounded off (see **-I**).
- F** Report grid domain and x/y-increments in world mapping format [Default is generic]. Does not apply to the **-C** option.
- I**[*dx/dy*]-**lb**] Report the min/max of the region to the nearest multiple of *dx* and *dy*, and output this in the form **-R***w/e/s/n* (unless **-C** is set). To report the actual grid region, select **-I**. If no argument is given then we report the grid increment in the form **-I***inc/ync*. If **-lb** is given we write each grid's bounding box polygon instead.
- L**[**0** | **1** | **2**]
 - L0** Report range of *z* after actually scanning the data, not just reporting what the header says.
 - L1** Report median and L1 scale of *z* (L1 scale = 1.4826 * Median Absolute Deviation (MAD)).
 - L2** Report mean, standard deviation, and root-mean-square (rms) of *z*.
- M** Find and report the location of min/max *z*-values, and count and report the number of nodes set to NaN, if any.
- R**[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest. Using the **-R** option will select a subsection of the input grid(s). If this subsection exceeds the boundaries of the grid, only the common region will be extracted.
- Tdz** Determine min and max *z*-value, round off to multiples of *dz*, and report as the text string **-T***zmin/zmax/dz* for use by **makecpt**. To get a symmetrical range about zero, using the max absolute multiple of *dz*, use **-Tsdz** instead.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- f**[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.39.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix *=id*[*scale*/*offset*[/*nan*]], where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset

to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append *?varname* to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of *?* in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The *?varname* suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.39.6 Examples

To obtain all the information about the data set in file *hawaii_topo.nc*:

```
gmt grdinfo -L1 -L2 -M hawaii_topo.nc
```

1.39.7 See Also

gmt, *grd2cpt*, *grd2xyz*, *grdedit*

1.40 *grdlandmask*

grdlandmask - Create a “wet-dry” mask grid from shoreline data base

1.40.1 Synopsis

```
grdlandmask -Gmask_grd_file -Lxinc[unit][=|+][/yinc[unit][=|+]] -Rwest/east/south/north[r] [ -Amin_area[/min_level/max_level][+as][+rll][pppercent] ] [ -Dresolution[+] ] [ -Nmaskvalues[o] ] [ -V[level] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.40.2 Description

grdlandmask reads the selected shoreline database and uses that information to decide which nodes in the specified grid are over land or over water. The nodes defined by the selected region and lattice spacing will be set according to one of two criteria: (1) land vs water, or (2) the more detailed (hierarchical) ocean vs land vs lake vs island vs pond. The resulting mask may be used in subsequent operations involving *grdmath* to mask out data from land [or water] areas.

1.40.3 Required Arguments

-**G***mask_grd_file*] Name of resulting output mask grid file. (See GRID FILE FORMATS below).

-I x_{inc} [unit][=|+][y_{inc} [unit][=|+]] x_{inc} [and optionally y_{inc}] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If y_{inc} is given but set to 0 it will be reset equal to x_{inc} ; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max x (*east*) or y (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]west/east/south/north[z_{min}/z_{max}][r**]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+ $-$]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append z_{min}/z_{max} . In case of perspective view (**-p**), a z-range (z_{min} , z_{max}) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

1.40.4 Optional Arguments

-Amin_area[/min_level/max_level][+as][+r|l][+ppercent] Features with an area smaller than min_area in km² or of hierarchical level that is lower than min_level or higher than max_level will not be plotted [Default is 0/0/4 (all features)]. Level 2 (lakes) contains regular lakes and wide river bodies which we normally include as lakes; append **+r** to just get river-lakes or **+l** to just get regular lakes. For expert users who wish to print their own Antarctica coastline and islands via *psxy* you can use **+as** to skip all GSHHG features below 60S. Finally, append **+ppercent** to exclude polygons whose percentage area of the corresponding full-resolution feature is less than *percent*. See GSHHG INFORMATION below for more details.

-Dresolution[+] Selects the resolution of the data set to use ((**f**)ull, (**h**)igh, (**i**)ntermediate, (**l**)ow, or (**c**)rude). The resolution drops off by ~80% between data sets. [Default is **l**]. Append **+** to automatically select a lower resolution should the one requested not be available [abort if not found]. Note that because the coastlines differ in details a node in a mask file using one resolution is not guaranteed to remain inside [or outside] when a different resolution is selected.

-Nmaskvalues[o] Sets the values that will be assigned to nodes. Values can be any number, including the textstring NaN. Append **o** to let nodes exactly on feature boundaries be considered outside [Default is inside]. Specify this information using 1 of 2 formats:

-Nwet/dry.

-Nocean/land/lake/island/pond.

[Default is 0/1/0/1/0 (i.e., 0/1)].

- V[*level*]** (*more ...*) Select verbosity level [*c*].
- r** (*more ...*) Set pixel node registration [*gridline*].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.40.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When writing a netCDF file, the grid is stored by default with the variable name “z”. To specify another variable name *varname*, append `?varname` to the file name. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes.

1.40.6 Examples

To set all nodes on land to NaN, and nodes over water to 1, using the high resolution data set, do

```
gmt grdlandmask -R-60/-40/-40/-30 -Dh -I5m -N1/NaN -Gland_mask.nc -V
```

To make a 1x1 degree global grid with the hierarchical levels of the nodes based on the low resolution data:

```
gmt grdlandmask -R0/360/-90/90 -Dl -I1 -N0/1/2/3/4 -Glevels.nc -V
```

1.40.7 Gshhs Information

The coastline database is GSHHG (formerly GSHHS) which is compiled from two sources: World Vector Shorelines (WVS) and CIA World Data Bank II (WDBII). In particular, all level-1 polygons (ocean-land boundary) are derived from the more accurate WVS while all higher level polygons (level 2-4, representing land/lake, lake/island-in-lake, and island-in-lake/lake-in-island-in-lake boundaries) are taken from WDBII. Much processing has taken place to convert WVS and WDBII data into usable form for GMT: assembling closed polygons from line segments, checking for duplicates, and correcting for crossings between polygons. The area of each polygon has been determined so that the user may choose not to draw features smaller than a minimum area (see **-A**); one may also limit the highest hierarchical level of polygons to be included (4 is the maximum). The 4 lower-resolution databases were derived

from the full resolution database using the Douglas-Peucker line-simplification algorithm. The classification of rivers and borders follow that of the WDBII. See the GMT Cookbook and Technical Reference Appendix K for further details.

1.40.8 See Also

gmt, grdmath, grdclip, psmask, psclip, pscoast

1.41 grdmask

grdmask - Create mask grid from polygons or point coverage

1.41.1 Synopsis

```
grdmask      pathfiles      -Gmask_grd_file]      -Ixinc[unit][=|+][yinc[unit][=|+]]      -
R[unit]xmin/xmax/ymin/ymax[r] [ -A[m|p] ] [ -N[z|Z|p|P]values ] [ -Ssearch_radius[unit] ] [
-V[level] ] [ -bi[ncols][type][w][+L|+B] ] [ -f[i|o]colinfo ] [ -g[a]x|y|d|X|Y|D|col|z[+|-]gap[u] ] [
-h[i|o][n][+c][+d][+rremark][+rtitle] ] [ -i[cols][l][sscale][offset][,...] ] [ -r ] [ -:i|o ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.41.2 Description

grdmask can operate in two different modes. 1. It reads one or more xy-files that each define a closed polygon. The nodes defined by the specified region and lattice spacing will be set equal to one of three possible values depending on whether the node is outside, on the polygon perimeter, or inside the polygon. The resulting mask may be used in subsequent operations involving **grdmath** to mask out data from polygonal areas. 2. The xy-files simply represent data point locations and the mask is set to the inside or outside value depending on whether a node is within a maximum distance from the nearest data point. If the distance specified is zero then only the nodes nearest each data point are considered “inside”.

1.41.3 Required Arguments

pathfiles The name of 1 or more ASCII [or binary, see **-bi**] files holding the polygon(s) or data points.

-Gmask_grd_file Name of resulting output mask grid file. (See GRID FILE FORMATS below).

-Ixinc[unit][=|+][yinc[unit][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain.

The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

1.41.4 Optional Arguments

-A[m|p] If the input data are geographic (as indicated by **-fi**) then the sides in the polygons will be approximated by great circle arcs. When using the **-A** sides will be regarded as straight lines. Alternatively, append **m** to have sides first follow meridians, then parallels. Or append **p** to first follow parallels, then meridians.

-N[z|Z|p|P]values Sets the *out/edge/in* that will be assigned to nodes that are *outside* the polygons, on the *edge*, or *inside*. Values can be any number, including the textstring NaN [Default is 0/0/1]. Optionally, use **Nz** to set polygon insides to the z-value obtained from the data (either segment header **-Zzval**, **-Lheader** or via **-aZ=name**); use **-NZ** to consider the polygon boundary as part of the inside. Alternatively, use **-Np** to use a running number as polygon ID; optionally append start of the sequence [0]. Here, **-NP** includes the polygon perimeter as inside. Note: **-Nz|Z|p|P** cannot be used in conjunction with **-S**; they also all optionally accept *out* [0].

-Ssearch_radius[unit] Set nodes to inside, on edge, or outside depending on their distance to the nearest data point. Nodes within *radius* [0] from the nearest data point are considered inside; append a distance unit (see UNITS). If *radius* is given as **z** then we instead read individual radii from the 3rd input column. If **-S** is not set then we consider the input data to define closed polygon(s) instead.

-V[level] (*more ...*) Select verbosity level [c].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 2 input columns].

-f[i|l]colinfo (*more ...*) Specify data types of input and/or output columns.

-g[a]x|y|d|X|Y|D[*col*]z[+|-]gap[*u*] (*more ...*) Determine data gaps and line breaks.

-h[i|l][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).

-i[cols][*l*][*sscale*][*ooffset*][,...] (*more ...*) Select input columns (0 is first column).

-r (*more ...*) Set pixel node registration [gridline].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.41.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot.

By default we compute such distances using a spherical approximation with great circles. Prepend `-` to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend `+` to perform exact geodesic calculations (slower but more accurate).

1.41.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When writing a netCDF file, the grid is stored by default with the variable name “z”. To specify another variable name *varname*, append `?varname` to the file name. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes.

1.41.7 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the `-f` or `-R` options. For example, both `-f0x` `-f1t` and `-R90w/90e/0t/3t` will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.41.8 Examples

To set all nodes inside and on the polygons coastline_*.xy to 0, and outside points to 1, do

```
gmt grdmask coastline_*.xy -R-60/-40/-40/-30 -I5m -N1/0/0 -Gland_mask.nc=nb -V
```

To set nodes within 50 km of data points to 1 and other nodes to NaN, do

```
gmt grdmask data.xyz -R-60/-40/-40/-30 -I5m -NNaN/1/1 -S50k -Gdata_mask.nc=nb -V
```

To assign polygon IDs to the gridnodes using the insides of the polygons in plates.gmt, based on the attribute POL_ID, do

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Nz -Gplate_IDs.nc -aZ=POL_ID -V
```

Same exercise, but instead compute running polygon IDs starting at 100, do

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Np100 -Gplate_IDs.nc -V
```

1.41.9 See Also

gmt, *grdlandmask*, *grdmath*, *grdclip*, *psmask*, *psclip*

1.42 grdmath

grdmath - Reverse Polish Notation (RPN) calculator for grids (element by element)

1.42.1 Synopsis

```
grdmath [ -Iinc[unit][=+][yinc[unit][=+]] ] [ -M ] [ -N ] [ -R[unit]xmin/xmax/ymin/ymax[r] ]
[ -V[level] ] [ -b[ncols][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -icols[l][sscale][ooffset][,...] ]
[ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][ooffset][,...] ] [ -n[blc|ln][+a][+bBC][+tthreshold] ] [ -r ] operand [ operand ] OPERATOR [ operand ] OPERATOR
... = outgrdfile
```

Note: No space is allowed between the option flag and the associated arguments.

1.42.2 Description

grdmath will perform operations like add, subtract, multiply, and divide on one or more grid files or constants using Reverse Polish Notation (RPN) syntax (e.g., Hewlett-Packard calculator-style). Arbitrarily complicated expressions may therefore be evaluated; the final result is written to an output grid file. Grid operations are element-by-element, not matrix manipulations. Some operators only require one operand (see below). If no grid files are used in the expression then options **-R**, **-I** must be set (and optionally **-r**). The expression = *outgrdfile* can occur as many times as the depth of the stack allows in order to save intermediate results. Complicated or frequently occurring expressions may be coded as a macro for future use or stored and recalled via named memory locations.

1.42.3 Required Arguments

operand If *operand* can be opened as a file it will be read as a grid file. If not a file, it is interpreted as a numerical constant or a special symbol (see below).

outgrdfile The name of a 2-D grid file that will hold the final result. (See GRID FILE FORMATS below).

1.42.4 Optional Arguments

-I*inc[unit][=+][yinc[unit][=+]]* *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.

- M By default any derivatives calculated are in z_units/ x(or y)_units. However, the user may choose this option to convert dx,dy in degrees of longitude,latitude into meters using a flat Earth approximation, so that gradients are in z_units/meter.
- N Turn off strict domain match checking when multiple grids are manipulated [Default will insist that each grid domain is within 1e-4 * grid_spacing of the domain of the first grid listed].
- R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.
- V[*level*] (*more ...*) Select verbosity level [*c*].
- bi[*ncols*][*type*] (*more ...*) Select binary input. The binary input option only applies to the data files needed by operators **LDIST**, **PDIST**, and **INSIDE**.
- f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- g[*a*]*x|y|d|X|Y|D*[*col*]*z*[+|-]*gap*[*u*] (*more ...*) Determine data gaps and line breaks.
- h[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- i[*cols*][*l*][*sscale*][*offset*][*,...*] (*more ...*) Select input columns (0 is first column).
- n[*blc|lln*][+*a*][+*bBC*][+*c*][+*tthreshold*] (*more ...*) Select interpolation mode for grids.
- r (*more ...*) Set pixel node registration [gridline]. Only used with **-R -I**.
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

1.42.5 Operators

Choose among the following 161 operators. “args” are the number of input and output arguments.

| Operator | args | Returns |
|--------------|------|-----------------------|
| ABS | 1 1 | abs (A) |
| ACOS | 1 1 | acos (A) |
| ACOSH | 1 1 | acosh (A) |
| ACOT | 1 1 | acot (A) |
| ACSC | 1 1 | acsc (A) |
| ADD | 2 1 | A + B |
| AND | 2 1 | B if A == NaN, else A |
| ASEC | 1 1 | asec (A) |
| ASIN | 1 1 | asin (A) |
| ASINH | 1 1 | asinh (A) |
| ATAN | 1 1 | atan (A) |
| ATAN2 | 2 1 | atan2 (A, B) |
| ATANH | 1 1 | atanh (A) |

Continued on next page

Table 1.4 – continued from previous page

| | | |
|------------------|-----|---|
| BEI | 1 1 | bei (A) |
| BER | 1 1 | ber (A) |
| BITAND | 2 1 | A & B (bitwise AND operator) |
| BITLEFT | 2 1 | A << B (bitwise left-shift operator) |
| BITNOT | 1 1 | ~A (bitwise NOT operator, i.e., return two's complement) |
| BITOR | 2 1 | A B (bitwise OR operator) |
| BITRIGHT | 2 1 | A >> B (bitwise right-shift operator) |
| BITTEST | 2 1 | 1 if bit B of A is set, else 0 (bitwise TEST operator) |
| BITXOR | 2 1 | A ^ B (bitwise XOR operator) |
| CAZ | 2 1 | Cartesian azimuth from grid nodes to stack x,y (i.e., A, B) |
| CBAZ | 2 1 | Cartesian backazimuth from grid nodes to stack x,y (i.e., A, B) |
| CDIST | 2 1 | Cartesian distance between grid nodes and stack x,y (i.e., A, B) |
| CEIL | 1 1 | ceil (A) (smallest integer >= A) |
| CHICRIT | 2 1 | Critical value for chi-squared-distribution, with alpha = A and n = B |
| CHIDIST | 2 1 | chi-squared-distribution P(chi2,n), with chi2 = A and n = B |
| CORRCOEFF | 2 1 | Correlation coefficient r(A, B) |
| COS | 1 1 | cos (A) (A in radians) |
| COSD | 1 1 | cos (A) (A in degrees) |
| COSH | 1 1 | cosh (A) |
| COT | 1 1 | cot (A) (A in radians) |
| COTD | 1 1 | cot (A) (A in degrees) |
| CPOISS | 2 1 | Cumulative Poisson distribution F(x,lambda), with x = A and lambda = B |
| CSC | 1 1 | csc (A) (A in radians) |
| CSCD | 1 1 | csc (A) (A in degrees) |
| CURV | 1 1 | Curvature of A (Laplacian) |
| D2DX2 | 1 1 | d^2(A)/dx^2 2nd derivative |
| D2DY2 | 1 1 | d^2(A)/dy^2 2nd derivative |
| D2DXY | 1 1 | d^2(A)/dxdy 2nd derivative |
| D2R | 1 1 | Converts Degrees to Radians |
| DDX | 1 1 | d(A)/dx Central 1st derivative |
| DDY | 1 1 | d(A)/dy Central 1st derivative |
| DEG2KM | 1 1 | Converts Spherical Degrees to Kilometers |
| DILOG | 1 1 | dilog (A) |
| DIV | 2 1 | A / B |
| DUP | 1 2 | Places duplicate of A on the stack |
| ERF | 1 1 | Error function erf (A) |
| ERFC | 1 1 | Complementary Error function erfc (A) |
| EQ | 2 1 | 1 if A == B, else 0 |
| ERFINV | 1 1 | Inverse error function of A |
| EXCH | 2 2 | Exchanges A and B on the stack |
| EXP | 1 1 | exp (A) |
| FACT | 1 1 | A! (A factorial) |
| EXTREMA | 1 1 | Local Extrema: +2/-2 is max/min, +1/-1 is saddle with max/min in x, 0 elsewhere |
| FCRIT | 3 1 | Critical value for F-distribution, with alpha = A, n1 = B, and n2 = C |
| FDIST | 3 1 | F-distribution Q(F,n1,n2), with F = A, n1 = B, and n2 = C |
| FLIPLR | 1 1 | Reverse order of values in each row |
| FLIPUD | 1 1 | Reverse order of values in each column |
| FLOOR | 1 1 | floor (A) (greatest integer <= A) |
| FMOD | 2 1 | A % B (remainder after truncated division) |

Continued on next page

Table 1.4 – continued from previous page

| | | |
|-----------------|-----|--|
| GE | 2 1 | 1 if A >= B, else 0 |
| GT | 2 1 | 1 if A > B, else 0 |
| HYPOT | 2 1 | hypot (A, B) = sqrt (A*A + B*B) |
| I0 | 1 1 | Modified Bessel function of A (1st kind, order 0) |
| I1 | 1 1 | Modified Bessel function of A (1st kind, order 1) |
| IFELSE | 3 1 | B if A != 0, else C |
| IN | 2 1 | Modified Bessel function of A (1st kind, order B) |
| INRANGE | 3 1 | 1 if B <= A <= C, else 0 |
| INSIDE | 1 1 | 1 when inside or on polygon(s) in A, else 0 |
| INV | 1 1 | 1 / A |
| ISFINITE | 1 1 | 1 if A is finite, else 0 |
| ISNAN | 1 1 | 1 if A == NaN, else 0 |
| J0 | 1 1 | Bessel function of A (1st kind, order 0) |
| J1 | 1 1 | Bessel function of A (1st kind, order 1) |
| JN | 2 1 | Bessel function of A (1st kind, order B) |
| K0 | 1 1 | Modified Kelvin function of A (2nd kind, order 0) |
| K1 | 1 1 | Modified Bessel function of A (2nd kind, order 1) |
| KEI | 1 1 | kei (A) |
| KER | 1 1 | ker (A) |
| KM2DEG | 1 1 | Converts Kilometers to Spherical Degrees |
| KN | 2 1 | Modified Bessel function of A (2nd kind, order B) |
| KURT | 1 1 | Kurtosis of A |
| LDIST | 1 1 | Compute minimum distance (in km if -fg) from lines in multi-segment ASCII file A |
| LDIST2 | 2 1 | As LDIST, from lines in ASCII file B but only to nodes where A != 0 |
| LE | 2 1 | 1 if A <= B, else 0 |
| LOG | 1 1 | log (A) (natural log) |
| LOG10 | 1 1 | log10 (A) (base 10) |
| LOG1P | 1 1 | log (1+A) (accurate for small A) |
| LOG2 | 1 1 | log2 (A) (base 2) |
| LMSSCL | 1 1 | LMS scale estimate (LMS STD) of A |
| LOWER | 1 1 | The lowest (minimum) value of A |
| LRAND | 2 1 | Laplace random noise with mean A and std. deviation B |
| LT | 2 1 | 1 if A < B, else 0 |
| MAD | 1 1 | Median Absolute Deviation (L1 STD) of A |
| MAX | 2 1 | Maximum of A and B |
| MEAN | 1 1 | Mean value of A |
| MED | 1 1 | Median value of A |
| MIN | 2 1 | Minimum of A and B |
| MOD | 2 1 | A mod B (remainder after floored division) |
| MODE | 1 1 | Mode value (Least Median of Squares) of A |
| MUL | 2 1 | A * B |
| NAN | 2 1 | NaN if A == B, else A |
| NEG | 1 1 | -A |
| NEQ | 2 1 | 1 if A != B, else 0 |
| NORM | 1 1 | Normalize (A) so max(A)-min(A) = 1 |
| NOT | 1 1 | NaN if A == NaN, 1 if A == 0, else 0 |
| NRAND | 2 1 | Normal, random values with mean A and std. deviation B |
| OR | 2 1 | NaN if B == NaN, else A |
| PDIST | 1 1 | Compute minimum distance (in km if -fg) from points in ASCII file A |

Continued on next page

Table 1.4 – continued from previous page

| | | |
|---------------|-----|--|
| PDIST2 | 2 1 | As PDIST, from points in ASCII file B but only to nodes where A != 0 |
| POP | 1 0 | Delete top element from the stack |
| PLM | 3 1 | Associated Legendre polynomial P(A) degree B order C |
| PLMg | 3 1 | Normalized associated Legendre polynomial P(A) degree B order C (geophysical convention) |
| POW | 2 1 | $A \wedge B$ |
| PQUANT | 2 1 | The B'th Quantile (0-100%) of A |
| PSI | 1 1 | Psi (or Digamma) of A |
| PV | 3 1 | Legendre function Pv(A) of degree v = real(B) + imag(C) |
| QV | 3 1 | Legendre function Qv(A) of degree v = real(B) + imag(C) |
| R2 | 2 1 | $R2 = A^2 + B^2$ |
| R2D | 1 1 | Convert Radians to Degrees |
| RAND | 2 1 | Uniform random values between A and B |
| RINT | 1 1 | rint (A) (round to integral value nearest to A) |
| ROTX | 2 1 | Rotate A by the (constant) shift B in x-direction |
| ROTY | 2 1 | Rotate A by the (constant) shift B in y-direction |
| SDIST | 2 1 | Spherical (Great circle/geodesic) distance (in km) between nodes and stack (A, B) (...) |
| SAZ | 2 1 | Spherical azimuth from grid nodes to stack lon, lat (i.e., A, B) |
| SBAZ | 2 1 | Spherical backazimuth from grid nodes to stack lon, lat (i.e., A, B) |
| SEC | 1 1 | sec (A) (A in radians) |
| SECD | 1 1 | sec (A) (A in degrees) |
| SIGN | 1 1 | sign (+1 or -1) of A |
| SIN | 1 1 | sin (A) (A in radians) |
| SINC | 1 1 | sinc (A) ($\sin(\pi * A) / (\pi * A)$) |
| SIND | 1 1 | sin (A) (A in degrees) |
| SINH | 1 1 | sinh (A) |
| SKEW | 1 1 | Skewness of A |
| SQR | 1 1 | A^2 |
| SQRT | 1 1 | sqrt (A) |
| STD | 1 1 | Standard deviation of A |
| STEP | 1 1 | Heaviside step function: H(A) |
| STEPX | 1 1 | Heaviside step function in x: H(x-A) |
| STEPLY | 1 1 | Heaviside step function in y: H(y-A) |
| SUB | 2 1 | A - B |
| SUM | 1 1 | Sum of all values in A |
| TAN | 1 1 | tan (A) (A in radians) |
| TAND | 1 1 | tan (A) (A in degrees) |
| TANH | 1 1 | tanh (A) |
| TAPER | 2 1 | Unit weights cosine-tapered to zero within A and B of x and y grid margins |
| TN | 2 1 | Chebyshev polynomial $T_n(-1 < t < +1, n)$, with $t = A$, and $n = B$ |
| TCRIT | 2 1 | Critical value for Student's t-distribution, with alpha = A and n = B |
| TDIST | 2 1 | Student's t-distribution $A(t, n)$, with $t = A$, and $n = B$ |
| UPPER | 1 1 | The highest (maximum) value of A |
| XOR | 2 1 | 0 if A == NaN and B == NaN, NaN if B == NaN, else A |
| Y0 | 1 1 | Bessel function of A (2nd kind, order 0) |
| Y1 | 1 1 | Bessel function of A (2nd kind, order 1) |
| YLM | 2 2 | Re and Im orthonormalized spherical harmonics degree A order B |
| YLMg | 2 2 | Cos and Sin normalized spherical harmonics degree A order B (geophysical convention) |
| YN | 2 1 | Bessel function of A (2nd kind, order B) |
| ZCRIT | 1 1 | Critical value for the normal-distribution, with alpha = A |

Continued on next page

Table 1.4 – continued from previous page

| | | |
|--------------|-----|--|
| ZDIST | 1 1 | Cumulative normal-distribution $C(x)$, with $x = A$ |
|--------------|-----|--|

1.42.6 Symbols

The following symbols have special meaning:

| | |
|--------------|---|
| PI | 3.1415926... |
| E | 2.7182818... |
| EULER | 0.5772156... |
| XMIN | Minimum x value |
| XMAX | Maximum x value |
| XINC | x increment |
| NX | The number of x nodes |
| YMIN | Minimum y value |
| YMAX | Maximum y value |
| YINC | y increment |
| NY | The number of y nodes |
| X | Grid with x-coordinates |
| Y | Grid with y-coordinates |
| Xn | Grid with normalized [-1 to +1] x-coordinates |
| Yn | Grid with normalized [-1 to +1] y-coordinates |

1.42.7 Notes On Operators

1. The operator **SDIST** calculates spherical distances in km between the (lon, lat) point on the stack and all node positions in the grid. The grid domain and the (lon, lat) point are expected to be in degrees. Similarly, the **SAZ** and **SBAZ** operators calculate spherical azimuth and back-azimuths in degrees, respectively. The operators **LDIST** and **PDIST** compute spherical distances in km if **-fg** is set or implied, else they return Cartesian distances. Note: If the current *PROJ_ELLIPSOID* is ellipsoidal then geodesics are used in calculations of distances, which can be slow. You can trade speed with accuracy by changing the algorithm used to compute the geodesic (see *PROJ_GEODESIC*).
2. The operator **PLM** calculates the associated Legendre polynomial of degree L and order M ($0 \leq M \leq L$), and its argument is the sine of the latitude. **PLM** is not normalized and includes the Condon-Shortley phase $(-1)^M$. **PLMg** is normalized in the way that is most commonly used in geophysics. The C-S phase can be added by using **-M** as argument. **PLM** will overflow at higher degrees, whereas **PLMg** is stable until ultra high degrees (at least 3000).
3. The operators **YLM** and **YLMg** calculate normalized spherical harmonics for degree L and order M ($0 \leq M \leq L$) for all positions in the grid, which is assumed to be in degrees. **YLM** and **YLMg** return two grids, the real (cosine) and imaginary (sine) component of the complex spherical harmonic. Use the **POP** operator (and **EXCH**) to get rid of one of them, or save both by giving two consecutive **= file.nc** calls.

The orthonormalized complex harmonics **YLM** are most commonly used in physics and seismology. The square of **YLM** integrates to 1 over a sphere. In geophysics, **YLMg** is normalized to produce unit power when averaging the cosine and sine terms (separately!) over a sphere (i.e., their squares each integrate to 4π). The Condon-Shortley phase $(-1)^M$ is not included in **YLM** or **YLMg**, but it can be added by using **-M** as argument.

4. All the derivatives are based on central finite differences, with natural boundary conditions.

5. Files that have the same names as some operators, e.g., **ADD**, **SIGN**, **=**, etc. should be identified by prepending the current directory (i.e., **./LOG**).
6. Piping of files is not allowed.
7. The stack depth limit is hard-wired to 100.
8. All functions expecting a positive radius (e.g., **LOG**, **KEI**, etc.) are passed the absolute value of their argument. (9) The bitwise operators (**BITAND**, **BITLEFT**, **BITNOT**, **BITOR**, **BITRIGHT**, **BITTEST**, and **BITXOR**) convert a grid's single precision values to unsigned 32-bit ints to perform the bitwise operations. Consequently, the largest whole integer value that can be stored in a float grid is 2^{24} or 16,777,216. Any higher result will be masked to fit in the lower 24 bits. Thus, bit operations are effectively limited to 24 bit. All bitwise operators return NaN if given NaN arguments or bit-settings ≤ 0 .

1.42.8 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.42.9 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.42.10 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x -f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in

the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.42.11 STORE, RECALL and CLEAR

You may store intermediate calculations to a named variable that you may recall and place on the stack at a later time. This is useful if you need access to a computed quantity many times in your expression as it will shorten the overall expression and improve readability. To save a result you use the special operator **STO@label**, where *label* is the name you choose to give the quantity. To recall the stored result to the stack at a later time, use **[RCL]@label**, i.e., **RCL** is optional. To clear memory you may use **CLR@label**. Note that **STO** and **CLR** leave the stack unchanged.

1.42.12 Macros

Users may save their favorite operator combinations as macros via the file *grdmath.macros* in their current or user directory. The file may contain any number of macros (one per record); comment lines starting with # are skipped. The format for the macros is **name = arg1 arg2 ... arg2 : comment** where **name** is how the macro will be used. When this operator appears on the command line we simply replace it with the listed argument list. No macro may call another macro. As an example, the following macro expects three arguments (radius x0 y0) and sets the modes that are inside the given circle to 1 and those outside to 0:

INCIRCLE = CDIST EXCH DIV 1 LE : usage: r x y INCIRCLE to return 1 inside circle

Note: Because geographic or time constants may be present in a macro, it is required that the optional comment flag (:) must be followed by a space.

1.42.13 Examples

To compute all distances to north pole:

```
gmt grdmath -Rg -I1 0 90 SDIST = dist_to_NP.nc
```

To take log10 of the average of 2 files, use

```
gmt grdmath file1.nc file2.nc ADD 0.5 MUL LOG10 = file3.nc
```

Given the file *ages.nc*, which holds seafloor ages in m.y., use the relation $\text{depth}(\text{in m}) = 2500 + 350 * \text{sqrt}(\text{age})$ to estimate normal seafloor depths:

```
gmt grdmath ages.nc SQRT 350 MUL 2500 ADD = depths.nc
```

To find the angle *a* (in degrees) of the largest principal stress from the stress tensor given by the three files *s_xx.nc* *s_yy.nc*, and *s_xy.nc* from the relation $\tan(2*a) = 2 * s_{xy} / (s_{xx} - s_{yy})$, use

```
gmt grdmath 2 s_xy.nc MUL s_xx.nc s_yy.nc SUB DIV ATAN 2 DIV = direction.nc
```

To calculate the fully normalized spherical harmonic of degree 8 and order 4 on a 1 by 1 degree world map, using the real amplitude 0.4 and the imaginary amplitude 1.1:

```
gmt grdmath -R0/360/-90/90 -I1 8 4 YML 1.1 MUL EXCH 0.4 MUL ADD = harm.nc
```

To extract the locations of local maxima that exceed 100 mGal in the file *faa.nc*:

```
gmt grdmath faa.nc DUP EXTREMA 2 EQ MUL DUP 100 GT MUL 0 NAN = z.nc
gmt grd2xyz z.nc -s > max.xyz
```

To demonstrate the use of named variables, consider this radial wave where we store and recall the normalized radial arguments in radians:

```
gmt grdmath -R0/10/0/10 -I0.25 5 5 CDIST 2 MUL PI MUL 5 DIV STO@r COS @r SIN MUL = wave.nc
```

1.42.14 References

Abramowitz, M., and I. A. Stegun, 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, vol. 55, Dover, New York.

Holmes, S. A., and W. E. Featherstone, 2002, A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions. *Journal of Geodesy*, 76, 279-299.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992, *Numerical Recipes*, 2nd edition, Cambridge Univ., New York.

Spanier, J., and K. B. Oldman, 1987, *An Atlas of Functions*, Hemisphere Publishing Corp.

1.42.15 See Also

gmt, *gmtmath*, *grd2xyz*, *grdedit*, *grdinfo*, *xyz2grd*

1.43 grdpaste

grdpaste - Join two grids along their common edge

1.43.1 Synopsis

```
grdpaste file_a.nc file_b.nc -Goutfile.nc [ -V[level] ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.43.2 Description

grdpaste will combine *file_a.nc* and *file_b.nc* into *outfile.nc* by pasting them together along their common edge. Files *file_a.nc* and *file_b.nc* must have the same dx, dy and have one edge in common. If in doubt, check with **grdinfo** and use **grdcut** and/or **grdsample** if necessary to prepare the edge joint. Note: For geographical grids, you may have to use **-f** to handle periodic longitudes unless the input grids are properly recognized as such via their meta-data.

1.43.3 Required Arguments

file_a.nc One of two files to be pasted together.

file_b.nc The other of two files to be pasted together.

-Goutfile.nc The name for the combined output.

1.43.4 Optional Arguments

-V[level] (*more ...*) Select verbosity level [c].

-f[ilo]colinfo (*more ...*) Specify data types of input and/or output columns.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

+- or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.43.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.43.6 Examples

Suppose `file_a.nc` is 150E - 180E and 0 - 30N, and `file_b.nc` is 150E - 180E, -30S - 0, then you can make `outfile.nc` which will be 150 - 180 and -30S - 30N by:

```
gmt grdpaste file_a.nc file_b.nc -Goutfile.nc -V -fg
```

1.43.7 See Also

gmt, *grdblend*, *grdclip*, *grdcut*, *grdinfo*, *grdsample*

1.44 grdproject

grdproject - Forward and inverse map transformation of grids

1.44.1 Synopsis

```
grdproject in_grdfile -Gout_grdfile -Jparameters [ -A[clilplelflklMlnlu] ] [ -C[dx/dy] ] [ -Dxinc[unit][=|+][yinc[unit][=|+]] ] [ -Edpi ] [ -I ] [ -Mclilp ] [ [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -n[blclln][+a][+bBC][+tthreshold] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.44.2 Description

grdproject will do one of two things depending whether **-I** has been set. If set, it will transform a gridded data set from a rectangular coordinate system onto a geographical system by resampling the surface at the new nodes. If not set, it will project a geographical gridded data set onto a rectangular grid. To obtain the value at each new node, its location is inversely projected back onto the input grid after which a value is interpolated between the surrounding input grid values. By default bi-cubic interpolation is used. Aliasing is avoided by also forward projecting the input grid nodes. If two or more nodes are projected onto the same new node, their average will dominate in the calculation of the new node value. Interpolation and aliasing is controlled with the **-n** option. The new node spacing may be determined in one of several ways by specifying the grid spacing, number of nodes, or resolution. Nodes not constrained by input data are set to NaN.

The **-R** option can be used to select a map region larger or smaller than that implied by the extent of the grid file.

1.44.3 Required Arguments

in_grdfile 2-D binary grid file to be transformed. (See GRID FILE FORMATS below.)

-Gout_grdfile Specify the name of the output grid file. (See GRID FILE FORMATS below.)

-Jparameters (*more ...*) Select map projection.

1.44.4 Optional Arguments

-A[*clilplelflklMlnlu*] Force 1:1 scaling, i.e., output (or input, see **-I**) data are in actual projected meters [**e**]. To specify other units, append **f** (foot), **k** (km), **M** (statute mile), **n** (nautical mile), **u** (US survey foot), **i** (inch), **c** (cm), or **p** (point). Without **-A**, the output (or input, see **-I**) are in the units specified by *PROJ_LENGTH_UNIT* (but see **-M**).

-C[*dx/dy*] Let projected coordinates be relative to projection center [Default is relative to lower left corner]. Optionally, add offsets in the projected units to be added (or subtracted when **-I** is set) to (from) the projected coordinates, such as false eastings and northings for particular projection zones [0/0].

-D*xinc[unit][=|+][yinc[unit][=|+]]* Set the grid spacing for the new grid. Append **m** for arc minute, **s** for arc second. If neither **-D** nor **-E** are set then we select the same number of output nodes as there are input nodes.

- Edpi** Set the resolution for the new grid in dots per inch.
- I** Do the Inverse transformation, from rectangular to geographical.
- Mclilp** Append **c**, **i**, or **p** to indicate that cm, inch, or point should be the projected measure unit [Default is set by *PROJ_LENGTH_UNIT* in *gmt.conf*]. Cannot be used with **-A**.
- R[unit]xmin/xmax/ymin/ymax[r] (more ...)** Specify the region of interest. You may ask to project only a subset of the grid by specifying a smaller input *w/e/s/n* region [Default is the region given by the grid file].
- V[level] (more ...)** Select verbosity level [c].
- n[blclln][+a][+bBC][+c][+threshold] (more ...)** Select interpolation mode for grids.
- r (more ...)** Set pixel node registration [gridline].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.44.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix *=id[/scale/offset[/nan]]*, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append *?varname* to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of **?** in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The *?varname* suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.44.6 Examples

To transform the geographical grid *dbdb5.nc* onto a pixel Mercator grid at 300 dpi, run

```
gmt grdproject dbdb5.nc -R20/50/12/25 -Jm0.25i -E300 -r -Gdbdb5_merc.nc
```

To inversely transform the file `topo_tm.nc` back onto a geographical grid, use

```
gmt grdproject topo_tm.nc -R-80/-70/20/40 -Jt-75/1:500000 -I -D5m -V -Gtopo.nc
```

This assumes, of course, that the coordinates in `topo_tm.nc` were created with the same projection parameters.

To inversely transform the file `topo_utm.nc` (which is in UTM meters) back to a geographical grid we specify a one-to-one mapping with meter as the measure unit:

```
gmt grdproject topo_utm.nc -R203/205/60/65 -Ju5/1:1 -I -Mm -Gtopo.nc -V
```

1.44.7 Restrictions

The boundaries of a projected (rectangular) data set will not necessarily give rectangular geographical boundaries (Mercator is one exception). In those cases some nodes may be unconstrained (set to NaN). To get a full grid back, your input grid may have to cover a larger area than you are interested in.

1.44.8 See Also

gmt, *gmt.conf*, *mapproject*

1.45 grdraster

grdraster - Extract subregion from a binary raster and save as a GMT grid

1.45.1 Synopsis

```
grdraster [ filename | “text pattern” ] -R[unit]xmin/xmax/ymin/ymax[r] [ -Ggrdfile ] [ -Iinc[unit][=|+][yinc[unit][=|+]] ] [ -Jparameters ] [ -V[level] ] [ -bo[ncols][type][w][+L|+B] ] [ -ocol[...] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.45.2 Description

grdraster reads a file called *grdraster.info* from the current working directory, the directories pointed to by the environment variables **\$GMT_USERDIR** and **\$GMT_DATADIR**, or in **\$GMT_SHAREDIR**/dbase (in that order). The file *grdraster.info* defines binary arrays of data stored in scan-line format in data files. Each file is given a *filename* in the info file. **grdraster** figures out how to load the raster data into a grid file spanning a region defined by **-R**. By default the grid spacing equals the raster spacing. The **-I** option may be used to sub-sample the raster data. No filtering or interpolating is done, however; the *x_inc* and *y_inc* of the grid must be multiples of the increments of the raster file and **grdraster** simply takes every *n*'th point. The output of **grdraster** is either grid or pixel registered depending on the registration of the raster used. It is up to the GMT system person to maintain the *grdraster.info* file in accordance with the available rasters at each site. Raster data sets are not supplied with GMT but can be obtained by anonymous ftp and on CD-ROM (see README page in

dbase directory). **grdraster** will list the available files if no arguments are given. Finally, **grdraster** will write xyz-triplets to stdout if no output gridfile name is given

1.45.3 Required Arguments

filename If an integer matching one of the files listed in the *grdraster.info* file is given we will use that data set, else we will match the given text pattern with the data set description in order to determine the data set.

-R[*unit*]west/east/south/north[/zmin/zmax][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. If *r* is appended, you may also specify a map projection to define the shape of your region. The output region will be rounded off to the nearest whole grid-step in both dimensions.

1.45.4 Optional Arguments

-Ggridfile Name of output grid file. If not set, the grid will be written as ASCII (or binary; see **-bo**) xyz-triplets to stdout instead.

-I x_{inc} [*unit*][=*+*][/*y_{inc}*[*unit*][=*+*]] *x_{inc}* [and optionally *y_{inc}*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If */y_{inc}* is given but set to 0 it will be reset equal to *x_{inc}*; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgridfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-Jparameters (more ...) Select map projection.

-V[level] (more ...) Select verbosity level [*c*].

-bo[ncols][type] (more ...) Select binary output.

-ocols[,...] (more ...) Select output columns (0 is first column).

This option applies only if no **-G** option has been set.

- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.45.5 Examples

To extract data from raster 1, taking one point every 30 minutes, in an area extended beyond 360 degrees to allow later filtering, run

```
gmt grdraster 1 -R-4/364/-62/62 -I30m -Gdata.nc
```

To obtain data for an oblique Mercator projection we need to extract more data that is actually used. This is necessary because the output of **grdraster** has edges defined by parallels and meridians, while the oblique map in general does not. Hence, to get all the data from the ETOPO2 data needed to make a contour map for the region defined by its lower left and upper right corners and the desired projection, use

```
gmt grdraster ETOPO2 -R160/20/220/30r -Joc190/25.5/292/69/1 -Gdata.nc
```

To extract data from the 2 min Geoware relief blend and write it as binary double precision xyz-triplets to standard output:

```
gmt grdraster "2 min Geoware" -R20/25/-10/5 -bo > triplets.b
```

1.45.6 See Also

gmtdefaults, gmt, grdsample, grdfilter

1.46 grdreformat

grdreformat - Convert between different grid formats

1.46.1 Synopsis

```
grdreformat ingrdfile[=id/scale/offset/NaNvalue]] outgrdfile[=id/scale/offset/NaNvalue]][:driver/datatype]]
[ -N ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.46.2 Description

grdreformat reads a grid file in one format and writes it out using another format. As an option the user may select a subset of the data to be written and to specify scaling, translation, and NaN-value.

1.46.3 Required Arguments

ingridfile The grid file to be read. Append format *=id* code if not a standard COARDS-compliant netCDF grid file. If *=id* is set (see below), you may optionally append *scale* and *offset*. These options will scale the data and then offset them with the specified amounts after reading. If *scale* and *offset* are supplied you may also append a value that represents ‘Not-a-Number’ (for floating-point grids this is unnecessary since the IEEE NaN is used; however integers need a value which means no data available). The *scale* and *offset* modifiers may be left empty to select default values (*scale* = 1, *offset* = 0).

outgridfile The grid file to be written. Append format *=id* code if not a standard COARDS-compliant netCDF grid file. If *=id* is set (see below), you may optionally append *scale* and *offset*. These options are particularly practical when storing the data as integers, first removing an offset and then scaling down the values. Since the *scale* and *offset* are applied in reverse order when reading, this does not affect the data values (except for round-offs).

If *scale* and *offset* are supplied you may also append a value that represents ‘Not-a-Number’ (for floating-point grids this is unnecessary since the IEEE NaN is used; however integers need a value which means no data available). The *scale* and *offset* modifiers may be left empty to select default values (*scale* = 1, *offset* = 0), or you may specify *a* for auto-adjusting the *scale* and/or *offset* of packed integer grids (*=id/a/a* is a shorthand for *=id/a/a*). When *id=gd*, the file will be saved using the GDAL library. Append the format *:driver* and optionally the output *datatype*. The driver names are those used by GDAL itself (e.g., netCDF, GTIFF, etc.), and the data type is one of *u8|u16|i16|u32|i32|float32*, where ‘i’ and ‘u’ denote signed and unsigned integers respectively. The default type is *float32*. Note also that both driver names and data types are case insensitive.

Consider setting *IO_NC4_DEFLATION_LEVEL* to reduce file size and to further increase read/write performance. Especially when working with subsets of global grids, masks, and grids with repeating grid values, the improvement is usually significant.

1.46.4 Optional Arguments

-N Suppress the writing of the GMT header structure. This is useful when you want to write a native grid to be used by **grdraster**. It only applies to native grids and is ignored for netCDF output.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

+- or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.46.5 Format Identifier

By default, grids will be written as floating point data stored in binary files using the netCDF format and meta-data structure. This format is conform the COARDS conventions. GMT versions prior to 4.1 produced netCDF files that did not conform to these conventions. Although these files are still supported, their use is deprecated. To write other than floating point COARDS-compliant netCDF files, append the *=id* suffix to the filename *outgrdfile*.

When reading files, **grdreformat** and other GMT programs will try to automatically recognize the type of the input grid file. If this fails you may append the *=id* suffix to the filename *ingridfile*.

| ID | Explanation |
|-----------|---|
| nb | GMT netCDF format (8-bit integer, COARDS, CF-1.5) |
| ns | GMT netCDF format (16-bit integer, COARDS, CF-1.5) |
| ni | GMT netCDF format (32-bit integer, COARDS, CF-1.5) |
| nf | GMT netCDF format (32-bit float, COARDS, CF-1.5) |
| nd | GMT netCDF format (64-bit float, COARDS, CF-1.5) |
| cb | GMT netCDF format (8-bit integer, deprecated) |
| cs | GMT netCDF format (16-bit integer, deprecated) |
| ci | GMT netCDF format (32-bit integer, deprecated) |
| cf | GMT netCDF format (32-bit float, deprecated) |
| cd | GMT netCDF format (64-bit float, deprecated) |
| bm | GMT native, C-binary format (bit-mask) |
| bb | GMT native, C-binary format (8-bit integer) |
| bs | GMT native, C-binary format (16-bit integer) |
| bi | GMT native, C-binary format (32-bit integer) |
| bf | GMT native, C-binary format (32-bit float) |
| bd | GMT native, C-binary format (64-bit float) |
| rb | SUN rasterfile format (8-bit standard) |
| rf | GEODAS grid format GRD98 (NGDC) |
| sf | Golden Software Surfer format 6 (32-bit float) |
| sd | Golden Software Surfer format 7 (64-bit float, read-only) |
| af | Atlantic Geoscience Center format AGC (32-bit float) |
| ei | ESRI Arc/Info ASCII Grid Interchange format (ASCII integer) |
| ef | ESRI Arc/Info ASCII Grid Interchange format (ASCII float) |
| gd | Import/export through GDAL |

1.46.6 Gmt Standard Netcdf Files

The standard format used for grdfiles is based on netCDF and conforms to the COARDS conventions. Files written in this format can be read by numerous third-party programs and are platform-independent. Some disk-space can be saved by storing the data as bytes or shorts in stead of integers. Use the *scale* and *offset* parameters to make this work without loss of data range or significance. For more details, see *App-file-formats*.

Multi-variable grid files

By default, GMT programs will read the first 2-dimensional grid contained in a COARDS-compliant netCDF file. Alternatively, use *ingridfile?varname* (ahead of any optional suffix *=id*) to specify the requested variable *varname*. Since *?* has special meaning as a wildcard, escape this meaning by placing the full filename and suffix between quotes.

Multi-dimensional grids

To extract one *layer* or *level* from a 3-dimensional grid stored in a COARDS-compliant netCDF file, append both the name of the variable and the index associated with the layer (starting at zero) in the form: *ingridfile?varname[layer]*. Alternatively, specify the value associated with that layer using parentheses in stead of brackets: *ingridfile?varname(layer)*.

In a similar way layers can be extracted from 4- or even 5-dimensional grids. For example, if a grid has the dimensions (parameter, time, depth, latitude, longitude), a map can be selected by using: *ingridfile?varname(parameter,time,depth)*.

Since question marks, brackets and parentheses have special meanings on the command line, escape these meanings by placing the full filename and suffix between quotes.

1.46.7 Native Binary Files

For binary native GMT files the size of the GMT *grdheader* block is *hsize* = 892 bytes, and the total size of the file is $hsize + nx * ny * item_size$, where *item_size* is the size in bytes of each element (1, 2, 4). Bit grids are stored using 4-byte integers, each holding 32 bits, so for these files the size equation is modified by using $ceil(nx / 32) * 4$ instead of *nx*. Note that these files are platform-dependent. Files written on Little Endian machines (e.g., PCs) can not be read on Big Endian machines (e.g., most workstations). Also note that it is not possible for GMT to determine uniquely if a 4-byte grid is float or int; in such cases it is best to use the *=ID* mechanism to specify the file format. In all cases a native grid is considered to be signed (i.e., there are no provision for unsigned short ints or unsigned bytes). For header and grid details, see *App-file-formats*.

1.46.8 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.46.9 Examples

To extract the second layer from a 3-dimensional grid named *temp* from a COARDS-compliant netCDF file *climate.nc*:

```
gmt grdreformat climate.nc?temp[1] temp.nc -V
```

To create a 4-byte native floating point grid from the COARDS-compliant netCDF file *data.nc*:

```
gmt grdreformat data.nc ras_data.b4=bf -V
```

To make a 2-byte short integer file, scale it by 10, subtract 32000, setting NaNs to -9999, do

```
gmt grdreformat values.nc shorts.i2=bs/10/-32000/-9999 -v
```

To create a Sun standard 8-bit rasterfile for a subset of the data file `image.nc`, assuming the range in `image.nc` is 0-1 and we need 0-255, run

```
gmt grdreformat image.nc -R-60/-40/-40/-30 image.ras8=rb/255/0 -v
```

To convert `etopo2.nc` to `etopo2.i2` that can be used by **grdraSTER**, try

```
gmt grdreformat etopo2.nc etopo2.i2=bs -N -v
```

To create a dumb file saved as a 32 bits float GeoTiff using GDAL, run

```
gmt grdmath -Rd -I10 X Y MUL = lixo.tiff=gd:GTiff
```

1.46.10 See Also

gmt.conf, *gmt*, *grdmath*

1.47 grdsample

grdsample - Resample a grid onto a new lattice

1.47.1 Synopsis

```
grdsample in_grdfile -Gout_grdfile [ -Ixinc[unit][=|+][/yinc[unit][=|+]] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -f[ilo]colinfo ] [ -n[b|c|l|n][+a][+bBC][+tthreshold] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.47.2 Description

grdsample reads a grid file and interpolates it to create a new grid file with either: a different registration (**-r** or **-T**); or, a new grid-spacing or number of nodes (**-I**), and perhaps also a new sub-region (**-R**). A bicubic [Default], bilinear, B-spline or nearest-neighbor interpolation is used; see **-n** for settings. Note that using **-R** only is equivalent to **grdcut** or **grdedit -S**. **grdsample** safely creates a fine mesh from a coarse one; the converse may suffer aliasing unless the data are filtered using **grdfft** or **grdfilter**.

When **-R** is omitted, the output grid will cover the same region as the input grid. When **-I** is omitted, the grid spacing of the output grid will be the same as the input grid. Either **-r** or **-T** can be used to change the grid registration. When omitted, the output grid will have the same registration as the input grid.

1.47.3 Required Arguments

in_grdfile The name of the input 2-D binary grid file. (See GRID FILE FORMAT below.)

-Gout_grdfile The name of the output grid file. (See GRID FILE FORMAT below.)

1.47.4 Optional Arguments

-I*inc*[*unit*][=+][*yinc*[*unit*][=+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

-T Translate between grid and pixel registration; if the input is grid-registered, the output will be pixel-registered and vice-versa.

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-n[*blclln*][+*a*][+*bBC*][+*c*][+*threshold*] (*more ...*) Select interpolation mode for grids.

-r (*more ...*) Set pixel node registration [gridline].

-^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.47.5 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.47.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify

the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.47.7 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see `-n`) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values values so they do not exceed the input min/max values by adding `+c` to your `-n` option.

1.47.8 Hints

If an interpolation point is not on a node of the input grid, then a NaN at any node in the neighborhood surrounding the point will yield an interpolated NaN. Bicubic interpolation [default] yields continuous first derivatives but requires a neighborhood of 4 nodes by 4 nodes. Bilinear interpolation [`-n`] uses only a 2 by 2 neighborhood, but yields only zero-order continuity. Use bicubic when smoothness is important. Use bilinear to minimize the propagation of NaNs.

1.47.9 Examples

To resample the 5 x 5 minute grid in `hawaii_5by5_topo.nc` onto a 1 minute grid:

```
gmt grdsample hawaii_5by5_topo.nc -I1m -Ghawaii_1by1_topo.nc
```

To translate the gridline-registered file `surface.nc` to pixel registration while keeping the same region and grid interval:

```
gmt grdsample surface.nc -T -Gpixel.nc
```

1.47.10 See Also

gmt, *grdedit*, *grdfft*, *grdfilter*

1.48 `grdtrack`

`grdtrack` - Sample grids at specified (x,y) locations

1.48.1 Synopsis

```
grdtrack [ xyfile ] -Ggrd1 -Ggrd2 ... [ -Af|p|m|r|R[+I] ] [ -Clength[u]/ds[spacing][+a] ]
[ -Ddfile ] [ -Eline[,line,...][+aaz][+d][+iinc[u]][+length[u]][+nnp][+oaz][+radius[u]] ] [ -
N ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Smethod/modifiers ] [ -T[radius[u][+elp]] ] [
-V[level] ] [ -Z ] [ -b[ncol][t][+L][+B] ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D[col]z[+|
]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -
n[bc|lln][+a][+bBC][+tthreshold] ] [ -ocols[,...] ] [ -s[cols][alr] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.48.2 Description

`grdtrack` reads one or more grid files (or a Sandwell/Smith IMG files) and a table (from file or standard input; but see **-E** for exception) with (x,y) [or (lon,lat)] positions in the first two columns (more columns may be present). It interpolates the grid(s) at the positions in the table and writes out the table with the interpolated values added as (one or more) new columns. Alternatively (**-C**), the input is considered to be line-segments and we create orthogonal cross-profiles at each data point or with an equidistant separation and sample the grid(s) along these profiles. A bicubic [Default], bilinear, B-spline or nearest-neighbor (see **-n**) interpolation is used, requiring boundary conditions at the limits of the region (see **-n**; Default uses “natural” conditions (second partial derivative normal to edge is zero) unless the grid is automatically recognized as periodic.)

1.48.3 Required Arguments

-G*gridfile* *gridfile* is a 2-D binary grid file with the function $f(x,y)$. If the specified grid is in Sandwell/Smith Mercator format you must append a comma-separated list of arguments that includes a scale to multiply the data (usually 1 or 0.1), the mode which stand for the following: (0) Img files with no constraint code, returns data at all points, (1) Img file with constraints coded, return data at all points, (2) Img file with constraints coded, return data only at constrained points and NaN elsewhere, and (3) Img file with constraints coded, return 1 at constraints and 0 elsewhere, and optionally the max latitude in the IMG file [80.738]. You may repeat **-G** as many times as you have grids you wish to sample. The grids are sampled and results are output in the order given. (See GRID FILE FORMAT below.)

1.48.4 Optional Arguments

xyfile This is an ASCII (or binary, see **-bi**) file where the first 2 columns hold the (x,y) positions where the user wants to sample the 2-D data set.

-A*f|p|m|r|R* For track resampling (if **-C** is set) we can select how this is to be performed. Append **f** to keep original points, but add intermediate points if needed [Default], **m** as **f**, but first follow meridian (along y) then parallel (along x), **p** as **f**, but first follow parallel (along y) then meridian (along x), **r** to resample at equidistant locations; input points are not necessarily included in the

output, and **R** as **r**, but adjust given spacing to fit the track length exactly. Finally, append **+l** if distances should be measured along rhumb lines (loxodromes). Ignored unless **-C** is used.

-Clength[*u*]/ds[*spacing*][+*a*] Use input line segments to create an equidistant and (optionally) equally-spaced set of crossing profiles along which we sample the grid(s) [Default simply samples the grid(s) at the input locations]. Specify two length scales that control how the sampling is done: *length* sets the full length of each cross-profile, while *ds* is the sampling spacing along each cross-profile. Optionally, append *spacing* for an equidistant spacing between cross-profiles [Default erects cross-profiles at the input coordinates]. By default, all cross-profiles have the same direction. Append **+a** to alternate the direction of cross-profiles. Append suitable units to *length*; it sets the unit used for *ds* [and *spacing*] (See UNITS below). The default unit for geographic grids is meter while Cartesian grids implies the user unit. The output columns will be *lon*, *lat*, *dist*, *azimuth*, *z1*, *z2*, ..., *zn* (The *zi* are the sampled values for each of the *n* grids)

-Dfile In concert with **-C** we can save the (possibly resampled) original lines to the file *dfile* [Default only saves the cross-profiles]. The columns will be *lon*, *lat*, *dist*, *azimuth*, *z1*, *z2*, ... (sampled value for each grid)

-Eline[*line*,...][+*aaz*][+*d*][+*iinc*[*u*]][+*length*[*u*]][+*nnp*][+*oaz*][+*radius*[*u*] Instead of reading input track coordinates, specify profiles via coordinates and modifiers. The format of each *line* is *start/stop*, where *start* or *stop* are either *lon/lat* (*x/y* for Cartesian data) or a 2-character XY key that uses the “p_{st}ext”-style justification format to specify a point on the map as [LCR][BMT]. In addition, you can use **Z-**, **Z+** to mean the global minimum and maximum locations in the grid (only available if only one grid is given). Instead of two coordinates you can specify an origin and one of **+a**, **+o**, or **+r**. You may append **+iinc[*u*]** to set the sampling interval; if not given then we default to half the minimum grid interval. The **+a** sets the azimuth of a profile of given length starting at the given origin, while **+o** centers the profile on the origin; both require **+l**. For circular sampling specify **+r** to define a circle of given radius centered on the origin; this option requires either **+n** or **+i**. The **+nnp** sets the desired number of points, while **+length** gives the total length of the profile. Append **+d** to output the along-track distances after the coordinates. Note: No track file will be read. Also note that only one distance unit can be chosen. Giving different units will result in an error. If no units are specified we default to great circle distances in km (if geographic). If working with geographic data you can prepend **-** (Flat Earth) or **+** (Geodesic) to *inc*, *length*, or *radius* to change the mode of distance calculation [Great Circle].

-N Do *not* skip points that fall outside the domain of the grid(s) [Default only output points within grid domain].

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest.

-Smethod/modifiers In conjunction with **-C**, compute a single stacked profile from all profiles across each segment. Append how stacking should be computed: **a** = mean (average), **m** = median, **p** = mode (maximum likelihood), **l** = lower, **L** = lower but only consider positive values, **u** = upper, **U** = upper but only consider negative values [**a**]. The *modifiers* control the output; choose one or more among these choices: **+a** : Append stacked values to all cross-profiles. **+d** : Append stack deviations to all cross-profiles. **+d** : Append data residuals (data - stack) to all cross-profiles. **+s[*file*]** : Save stacked profile to *file* [grdtrack_stacked_profile.txt]. **+cfact** : Compute envelope on stacked profile as +/- *fact***deviation* [2]. Notes: (1) Deviations depend on *method* and are st.dev (**a**), L1 scale (**m** and **p**), or half-range (upper-lower)/2. (2) The stacked profile file contains 1 plus groups of 4-6 columns, one group for each sampled grid. The first column holds cross distance, while the first 4 in a group hold stacked value, deviation, min value, and max value. If *method* is one of **almip** then we also write the lower and upper confidence bounds (see **+c**). When one or more of **+a**, **+d**, and **+r** are used then we append the results to the end of each row for all cross-profiles. The order is always stacked value (**+a**), followed by deviations (**+d**) and residuals (**+r**).

When more than one grid is sampled this sequence of 1-3 columns are repeated for each grid.

- T**[*radius*][*u*][+*elp*] To be used with normal grid sampling, and limited to a single, non-IMG grid. If the nearest node to the input point is NaN, search outwards until we find the nearest non-NaN node and report that value instead. Optionally specify a search radius which limits the consideration to points within this distance from the input point. To report the location of the nearest node and its distance from the input point, append +*e*. To instead replace the input point with the coordinates of the nearest node, append +*p*.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- Z** Only write out the sampled z-values [Default writes all columns].
- : Toggles between (longitude,latitude) and (latitude,longitude) input/output. [Default is (longitude,latitude)].
- bi**[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- bo**[*ncols*][*type*] (*more ...*) Select binary output. [Default is one more than input].
- f**[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- g**[*a*]*xlyldlXlYlDl*[*col*]*z*[+|-]**gap**[*u*] (*more ...*) Determine data gaps and line breaks.
- h**[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- icols**[*i*][*sscale*][*offset*][,...] (*more ...*) Select input columns (0 is first column).
- n**[*blclln*][+*a*][+*bBC*][+*c*][+*threshold*] (*more ...*) Select interpolation mode for grids.
- ocols**[,...] (*more ...*) Select output columns (0 is first column).
- s**[*cols*][*alr*] (*more ...*) Set handling of NaN records.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.48.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend - to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend + to perform exact geodesic calculations (slower but more accurate).

1.48.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted

according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.48.7 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix *=id[/scale/offset[/nan]]*, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append *?varname* to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of *?* in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The *?varname* suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.48.8 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see **-n**) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values values so they do not exceed the input min/max values by adding **+c** to your **-n** option.

1.48.9 Hints

If an interpolation point is not on a node of the input grid, then a NaN at any node in the neighborhood surrounding the point will yield an interpolated NaN. Bicubic interpolation [default] yields continuous first derivatives but requires a neighborhood of 4 nodes by 4 nodes. Bilinear interpolation [**-n**] uses only a 2 by 2 neighborhood, but yields only zeroth-order continuity. Use bicubic when smoothness is important. Use bilinear to minimize the propagation of NaNs, or lower *threshold*.

1.48.10 Examples

To sample the file *hawaii_topo.nc* along the SEASAT track *track_4.xyg* (An ASCII table containing longitude, latitude, and SEASAT-derived gravity, preceded by one header record):

```
grdtrack track_4.xyg -Ghawaii_topo.nc -h > track_4.xygt
```

To sample the Sandwell/Smith IMG format file `topo.8.2.img` (2 minute predicted bathymetry on a Mercator grid) and the Muller et al age grid `age.3.2.nc` along the lon,lat coordinates given in the file `cruise_track.xy`, try

```
grdtrack cruise_track.xy -Gtopo.8.2.img,1,1 -Gage.3.2.nc > depths-age.d
```

To sample the Sandwell/Smith IMG format file `grav.18.1.img` (1 minute free-air anomalies on a Mercator grid) along 100-km-long cross-profiles that are orthogonal to the line segment given in the file `track.xy`, erecting cross-profiles every 25 km and sampling the grid every 3 km, try

```
grdtrack track.xy -Ggrav.18.1.img,0.1,1 -C100k/3/25 -Ar > xprofiles.txt
```

To sample the grid `data.nc` along a line from the lower left to the upper right corner, using a grid spacing of 1 km, and output distances as well, try

```
grdtrack -ELB/RT+ilk+d -Gdata.nc > profiles.txt
```

1.48.11 See Also

gmt, gmtconvert, sample1d, surface

1.49 grdtrend

grdtrend - Fit trend surface to grids and compute residuals

1.49.1 Synopsis

```
grdtrend gridfile -Nn_model[r] [ -Ddiff.nc ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Ttrend.nc ] [ -Wweight.nc ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.49.2 Description

grdtrend reads a 2-D grid file and fits a low-order polynomial trend to these data by [optionally weighted] least-squares. The trend surface is defined by:

$$m_1 + m_2*x + m_3*y + m_4*x*y + m_5*x*x + m_6*y*y + m_7*x*x*x + m_8*x*x*y + m_9*x*y*y + m_{10}*y*y*y.$$

The user must specify **-N***n_model*, the number of model parameters to use; thus, **-N3** fits a bilinear trend, **-N6** a quadratic surface, and so on. Optionally, append **r** to the **-N** option to perform a robust fit. In this case, the program will iteratively reweight the data based on a robust scale estimate, in order to converge to a solution insensitive to outliers. This may be handy when separating a “regional” field from a “residual” which should have non-zero mean, such as a local mountain on a regional surface.

If data file has values set to NaN, these will be ignored during fitting; if output files are written, these will also have NaN in the same locations.

1.49.3 Required Arguments

grdfile The name of a 2-D binary grid file.

-Nn_model[r] *n_model* sets the number of model parameters to fit. Append **r** for robust fit.

1.49.4 Optional Arguments

-Ddiff.nc Write the difference (input data - trend) to the file *diff.nc*.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest. Using the **-R** option will select a subsection of the input grid. If this subsection exceeds the boundaries of the grid, only the common region will be extracted.

-Ttrend.nc Write the fitted trend to the file *trend.nc*.

-V[level] (*more ...*) Select verbosity level [c].

-Wweight.nc If *weight.nc* exists, it will be read and used to solve a weighted least-squares problem. [Default: Ordinary least-squares fit.] If the robust option has been selected, the weights used in the robust fit will be written to *weight.nc*.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.49.5 Remarks

The domain of *x* and *y* will be shifted and scaled to [-1, 1] and the basis functions are built from Legendre polynomials. These have a numerical advantage in the form of the matrix which must be inverted and allow more accurate solutions. NOTE: The model parameters listed with **-V** are Legendre polynomial coefficients; they are not numerically equivalent to the *m*#s in the equation described above. The description above is to allow the user to match **-N** with the order of the polynomial surface. See **grdmath** if you need to evaluate the trend using the reported coefficients.

1.49.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix *=id[/scale/offset[/nan]]*, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file

names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append *?varname* to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of *?* in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The *?varname* suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.49.7 Examples

To remove a planar trend from *hawaii_topo.nc* and write result in *hawaii_residual.nc*:

```
gmt grdtrend hawaii_topo.nc -N3 -Dhawaii_residual.nc
```

To do a robust fit of a bicubic surface to *hawaii_topo.nc*, writing the result in *hawaii_trend.nc* and the weights used in *hawaii_weight.nc*, and reporting the progress:

```
gmt grdtrend hawaii_topo.nc -N10r -Thawaii_trend.nc -Whawaii_weight.nc -V
```

1.49.8 See Also

gmt, *grdfft*, *grdfilter*, *grdmath*

1.50 *grdvector*

grdvector - Plot vector field from two component grids

1.50.1 Synopsis

```
grdvector comp1.nc comp2.nc -Jparameters [ -A ] [ -B[pls]parameters ] [ -Gfill ] [ -Ixinc[unit][=|+][/yinc[unit][=|+]] ] [ -K ] [ -N ] [ -O ] [ -P ] [ -Qparameters ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S[ill]scale ] [ -T ] [ -U[just/dx/dy][clabel] ] [ -Wpen ] [ -X[alcflr][x-shift[u]] ] [ -Y[alcflr][y-shift[u]] ] [ -Z ] [ -ccopies ] [ -f[ilo]colinfo ] [ -p[x|y|z]azim/elev[zlevel][+wlon0/lat0[z0]][+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.50.2 Description

grdvector reads two 2-D grid files which represents the x- and y-components of a vector field and produces a vector field plot by drawing vectors with orientation and length according to the information in the files. Alternatively, polar coordinate components may be used (r, theta). **grdvector** is basically a short-hand for using 2 calls to *grd2xyz* and pasting the output through **psxy -SV**.

1.50.3 Required Arguments

comp_x.nc Contains the x-component of the vector field.

comp_y.nc Contains the y-component of the vector field. (See GRID FILE FORMATS below.)

-Jparameters (*more ...*) Select map projection.

1.50.4 Optional Arguments

-A Means grid files have polar (r, theta) components instead of Cartesian (x, y).

-B[pls]parameters (*more ...*) Set map boundary intervals.

-C[*cptfile*] Use *cptfile* to assign colors based on vector length. Alternatively, supply the name of a GMT color master CPT [rainbow] and let **grdvector** automatically determine a 16-level continuous CPT from the grid's z-range.

-Gfill Sets color or shade for vector interiors [Default is no fill].

-I Only plot vectors at nodes every *x_inc*, *y_inc* apart (must be multiples of original grid spacing). Append **m** for arc minutes or **s** for arc seconds. [Default plots every node].

-K (*more ...*) Do not finalize the PostScript plot.

-N Do NOT clip vectors at map boundaries [Default will clip].

-O (*more ...*) Append to existing PostScript plot.

-P (*more ...*) Select "Portrait" plot orientation.

-Qparameters Modify vector parameters. For vector heads, append vector head *size* [Default is 0, i.e., stick-plot]. See VECTOR ATTRIBUTES for specifying additional attributes.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest. Specify a subset of the grid.

-S[*ill*]*scale* Sets scale for Cartesian vector length in data units per distance measurement unit [1]. Append **c**, **i**, or **p** to indicate the measurement unit (cm, inch, or point). Prepend **l** to indicate a fixed length for all vectors. For Geographic data, give scale in data units per km. Use **-Si** if it is simpler to give the reciprocal scale in measurement unit per data unit or km per data unit.

-T Means azimuth of Cartesian data sets should be adjusted for different scales in the x- and y-directions [Leave alone].

-U[*just*/*dx*/*dy*]/[*clabel*] (*more ...*) Draw GMT time stamp logo on plot.

-V[*level*] (*more ...*) Select verbosity level [c].

-Wpen Set pen attributes used for vector outlines [Default: width = default, color = black, style = solid].

-X[*alcflr*][*x-shift*[*u*]]

-Y[*alcflr*][*y-shift*[*u*]] (*more ...*) Shift plot origin.

-Z Means the angles provided are azimuths rather than direction (requires **-A**).

-ccopies (*more ...*) Specify number of plot copies [Default is 1].

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-p[*x*/*y*/*z*]*azim*/*elev*[*zlevel*][*+wlon0*/*lat0*/*z0*][*+vx0*/*y0*] (*more ...*) Select perspective view.

-t[*transp*] (*more ...*) Set PDF transparency level in percent.

- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.50.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.50.6 Vector Attributes

Several modifiers may be appended to the vector-producing options to specify the placement of vector heads, their shapes, and the justification of the vector. Below, left and right refers to the side of the vector line when viewed from the start point to the end point of the segment:

- +aangle** sets the angle of the vector head apex [30].
- +b** places a vector head at the beginning of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +e** places a vector head at the end of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +g-|fill** turns off vector head fill (if -) or sets the vector head fill [Default fill is used, which may be no fill].
- +l** draws half-arrows, using only the left side of specified heads [both sides].
- +norm** scales down vector attributes (pen thickness, head size) with decreasing length, where vectors shorter than *norm* will have their attributes scaled by `length/norm` [arrow

attributes remains invariant to length].

+oplon/plat specifies the oblique pole for the great or small circles. Only needed for great circles if **+q** is given.

+p[-][pen] sets the vector pen attributes. If *pen* has a leading - then the head outline is not drawn. [Default pen is used, and head outline is drawn]

+q means the input *angle, length* data instead represent the *start* and *stop* opening angles of the arc segment relative to the given point.

+r draws half-arrows, using only the right side of specified heads [both sides].

In addition, all but circular vectors may take these modifiers:

+jjust determines how the input *x,y* point relates to the vector. Choose from **beginning** [default], **end**, or **center**.

+s means the input *angle, length* is instead the *x, y* coordinates of the vector end point.

1.50.7 Examples

To draw the vector field given by the files *r.nc* and *theta.nc* on a linear plot with scale 5 cm per data unit, using vector rather than stick plot, scale vector magnitudes so that 10 units equal 1 inch, and center vectors on the node locations, run

```
gmt grdvector r.nc theta.nc -Jx5c -A -Q0.1i+e+jc -S10i > gradient.ps
```

To plot a geographic data sets given the files *com_x.nc* and *comp_y.nc*, using a scale of 200 km per data unit, try

```
gmt grdvector comp_x.nc comp_y.nc -JH0/20c -Q0.1i+e+jc -S200 > globe.ps
```

1.50.8 See Also

gmt, gmtcolors, grdcontour, psxy

1.51 grdview

grdview - Create 3-D perspective image or surface mesh from a grid

1.51.1 Synopsis

```
grdview relief_file -Jparameters [ -B[pls]parameters ] [ -C[cptfile] ] [ -Gdrapefile | -Ggrd_r,grd_g,grd_b ] [ -Iintensfile/intensity ] [ -Jz/Zparameters ] [ -K ] [ -Nlevel[+gfill] ] [ -O ] [ -P ] [ -Qargs[+m] ] [ -Rwest/east/south/north[zmin/zmax][r] ] [ -Ssmooth ] [ -T[s][o][pen] ] [ -U[just/dx/dy]/[clabel] ] [ -Wtypepen ] [ -X[alc/flr][x-shift][u] ] [ -Y[alc/flr][y-shift][u] ] [ -ccopies ] [ -n[blc/lln][+a][+bBC][+tthreshold] ] [ -p[x|y|z]azim/elev[zlevel][+wlon0/lat0][z0][+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.51.2 Description

grdview reads a 2-D grid file and produces a 3-D perspective plot by drawing a mesh, painting a colored/grayshaded surface made up of polygons, or by scanline conversion of these polygons to a raster image. Options include draping a data set on top of a surface, plotting of contours on top of the surface, and apply artificial illumination based on intensities provided in a separate grid file.

1.51.3 Required Arguments

relief_file 2-D gridded data set to be imaged (the relief of the surface). (See GRID FILE FORMAT below.)

-Jparameters (more ...) Select map projection.

-Jz/Zparameters (more ...) Set z-axis scaling; same syntax as **-Jx**.

1.51.4 Optional Arguments

-B[pls]parameters (more ...) Set map boundary intervals.

-C[cptfile] name of the color palette file. Must be present if you want (1) mesh plot with contours (**-Qm**), or (2) shaded/colored perspective image (**-Qs** or **-Qi**). For **-Qs**: You can specify that you want to skip a z-slice by setting red = -; to use a pattern give red = **P|pdpi/pattern[:Fcolor[Bcolor]]**. Alternatively, supply the name of a GMT color master CPT [rainbow] and let **grdview** automatically determine a 16-level continuous CPT from the grid's z-range.

-Gdrapefile | -Ggrd_r,grd_g,grd_b Drape the image in *drapefile* on top of the relief provided by *relief_file*. [Default is *relief_file*]. Note that **-Jz** and **-N** always refers to the *relief_file*. The *drapefile* only provides the information pertaining to colors, which is looked-up via the cpt file (see **-C**). Alternatively, give three grid files separated by commas. These files must contain the red, green, and blue colors directly (in 0-255 range) and no cpt file is needed. The *drapefile* may be of higher resolution than the *relief_file*.

-Iintensfile/intensity Gives the name of a grid file with intensities in the (-1,+1) range, or a constant intensity to apply everywhere. [Default is no illumination].

-K (more ...) Do not finalize the PostScript plot.

-Nlevel[+gfill] Draws a plane at this z-level. If the optional *color* is provided via the **+g** modifier, the frontal facade between the plane and the data perimeter is colored. See **-Wf** for setting the pen used for the outline.

-O (more ...) Append to existing PostScript plot.

-P (more ...) Select "Portrait" plot orientation.

-Qargs[+m] Select one of four settings: 1. Specify **m** for mesh plot [Default], and optionally append *color* for a different mesh paint [white]. 2. Specify **s** for surface plot, and optionally append **m** to have mesh lines drawn on top of surface. 3. Specify **i** for image plot, and optionally append the effective dpi resolution for the rasterization [100]. 4. Specify **c**. Same as **-Qi** but will make nodes with z = NaN transparent, using the colormasking feature in PostScript Level 3 (the PS device must support PS Level 3). For any of these choices, you may force a monochrome image by appending the modifier **+m**. Colors are then converted to shades of gray using the (monochrome television) YIQ transformation.

-R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax*. (*more ...*) This option may be used to indicate the range used for the 3-D axes [Default is region given by the *relief_file*]. You may ask for a larger *w/e/s/n* region to have more room between the image and the axes. A smaller region than specified in the *relief_file* will result in a subset of the grid.

-Ssmooth Smooth the contours before plotting (see **grdcontour**) [Default is no smoothing].

-T[s][o[*pen*]] Plot image without any interpolation. This involves converting each node-centered bin into a polygon which is then painted separately. Append **s** to skip nodes with *z* = NaN. This option is useful for categorical data where interpolating between values is meaningless. Optionally, append **o** to draw the tile outlines, and specify a custom pen if the default pen is not to your liking. As this option produces a flat surface it cannot be combined with **-JZ** or **-Jz**.

-U[*just/dx/dy*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-Wtypepen

-Wc Draw contour lines on top of surface or mesh (not image). Append pen attributes used for the contours. [Default: width = 0.75p, color = black, style = solid].

-Wm Sets the pen attributes used for the mesh. [Default: width = 0.25p, color = black, style = solid]. You must also select **-Qm** or **-Qsm** for meshlines to be drawn.

-Wf Sets the pen attributes used for the facade. [Default: width = 0.25p, color = black, style = solid]. You must also select **-N** for the facade outline to be drawn.

-X[*alclfr*][*x-shift*[*u*]]

-Y[*alclfr*][*y-shift*[*u*]] (*more ...*) Shift plot origin.

-ccopies (*more ...*) Specify number of plot copies [Default is 1].

-n[*blclln*][*+a*][*+bBC*][*+c*][*+threshold*] (*more ...*) Select interpolation mode for grids.

-p[*x|y|z*]*azim/lev*[*/zlevel*][*+wlon0/lat0/z0*][*+vx0/y0*] (*more ...*) Select perspective view.

-t[*transp*] (*more ...*) Set PDF transparency level in percent.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.51.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify

the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.51.6 Consequences of grid resampling

Except for Cartesian cases, we need to resample your geographic grid onto an equidistant projected grid. In doing so various algorithms come into play that projects data from one lattice to another while avoiding anti-aliasing, leading to possible distortions. One expected effect of resampling with splines is the tendency for the new resampled grid to slightly exceed the global min/max limits of the original grid. If this is coupled with tight CPT file limits you may find that some map areas may show up with fore- or background color due to the resampling. In that case you have two options: (1) Modify your CPT file to fit the resampled extrema (reported with `-V`) or (2) Impose clipping of resampled values so they do not exceed the input min/max values (add `+c` to your `-n` option).

1.51.7 Examples

To make a mesh plot from the file `hawaii_grav.nc` and drawing the contours given in the color palette file `hawaii.cpt` on a Lambert map at 1.5 cm/degree along the standard parallels 18 and 24, with vertical scale 20 mgal/cm, and looking at the surface from SW at 30 degree elevation, run

```
gmt grdview hawaii_grav.nc -Jl18/24/1.5c -Chawaii.cpt \  
-Jz0.05c -Qm -N-100 -p225/30 -Wc > hawaii_grav_image.ps
```

To create a illuminated color perspective plot of the gridded data set `image.nc`, using the color palette file `color.rgb`, with linear scaling at 10 cm/x-unit and tickmarks every 5 units, with intensities provided by the file `intens.nc`, and looking from the SE, use

```
gmt grdview image.nc -Jx10.0c -Ccolor.rgb -Qs -p135/30 -Iintens.nc > image3D.ps
```

To make the same plot using the rastering option with `dpi = 50`, use

```
gmt grdview image.nc -Jx10.0c -Ccolor.rgb -Qi50 -p135/30 -Iintens.nc > image3D.ps
```

To create a color PostScript perspective plot of the gridded data set `magnetics.nc`, using the color palette file `mag_intens.cpt`, draped over the relief given by the file `topography.nc`, with Mercator map width of 6 inch and tickmarks every 1 degree, with intensities provided by the file `topo_intens.nc`, and looking from the SE, run

```
gmt grdview topography.nc -JM6i -Gmagnetics.nc -Cmag_intens.cpt \  
-Qs -p140/30 -Itopo_intens.nc > draped3D.ps
```

Given `topo.nc` and the Landsat image `veggies.ras`, first run **grd2rgb** to get the red, green, and blue grids, and then drape this image over the topography and shade the result for good measure. The commands are

```
gmt grd2rgb veggies.ras -Glayer_%c.nc
gmt grdview topo.nc -JM6i -Qi -p140/30 -Itopo_intens.nc \
    -Glayer_r.nc,layer_g.nc,layer_b.nc > image.ps
```

1.51.8 Remarks

For the **-Qs** option: PostScript provides no way of smoothly varying colors within a polygon, so colors can only vary from polygon to polygon. To obtain smooth images this way you may resample the grid file(s) using **grdsample** or use a finer grid size when running gridding programs like **surface** or **nearneighbor**. Unfortunately, this produces huge PostScript files. The alternative is to use the **-Qi** option, which computes bilinear or bicubic continuous color variations within polygons by using scanline conversion to image the polygons.

1.51.9 See Also

gmt, grd2rgb, gmtcolors, grdcontour, grdimage, nearneighbor, psbasemap, pscontour, pstext, surface

1.52 grdvolume

grdvolume - Calculate grid volume and area constrained by a contour

1.52.1 Synopsis

```
grdvolume grdfile [ -Cival or -Clow/high/delta or -Crlow/high ] [ -Lbase ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S[unit] ] [ -T[clh] ] [ -V[level] ] [ -Zfact[shift] ] [ -f[ilo]colinfo ] [ -ocols[,...] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.52.2 Description

grdvolume reads a 2-D binary grid file and calculates the volume contained between the surface and the plane specified by the given contour (or zero if not given) and reports the area, volume, and maximum mean height (volume/area). Alternatively, specify a range of contours to be tried and **grdvolume** will determine the volume and area inside the contour for all contour values. Using **-T**, the contour that produced the maximum mean height (or maximum curvature of heights vs contour value) is reported as well. This feature may be used with **grdfilter** in designing an Optimal Robust Separator [Wessel, 1998].

1.52.3 Required Arguments

grdfile The name of the input 2-D binary grid file. (See GRID FILE FORMAT below.)

1.52.4 Optional Arguments

- Ccval** or **-Clow/high/delta** or **-Cr_{low/high}** find area, volume and mean height (volume/area) inside the *cval* contour. Alternatively, search using all contours from *low* to *high* in steps of *delta*. [Default returns area, volume and mean height of the entire grid]. The area is measured in the plane of the contour. The **Cr** form on the other hand computes volume between the grid surface and the plans defined by *low* and *high*. Note that this is an *outside* volume whilst the other forms compute an *inside* (bellow the surface) area volume. Use this form to compute for example the volume of water between two contours.
- Lbase** Also add in the volume from the level of the contour down to *base* [Default base is contour].
- S[unit]** Convert degrees to Flat Earth distances, append a unit from **elfklMlnlu** [Default is Cartesian].
- T[ch]** Determine the single contour that maximized the average height (= volume/area). Select **-Tc** to use the maximum curvature of heights versus contour value rather than the contour with the maximum height to pick the best contour value (requires **-C**).
- R[unit]xmin/xmax/ymin/ymax[r]** (*more ...*) Specify the region of interest.
- V[level]** (*more ...*) Select verbosity level [c].
- Zfact[/shift]** Optionally subtract *shift* before scaling data by *fact*. [Default is no scaling]. (Numbers in **-C**, **-L** refer to values after this scaling has occurred).
- f[ilo]colinfo** (*more ...*) Specify data types of input and/or output columns.
- ocols[,...]** (*more ...*) Select output columns (0 is first column).
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.52.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front

of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: "z". See *grdformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

1.52.6 Examples

To determine the volume in km³ under the surface `hawaii_topo.nc` (height in km), use

```
gmt grdvolume hawaii_topo.nc -Sk
```

To find the volume between the surface `peaks.nc` and the contour `z = 250`, use

```
gmt grdvolume peaks.nc -Sk -C250
```

To search for the contour, between 100 and 300 in steps of 10, that maximizes the ratio of volume to surface area for the file `peaks.nc`, use

```
gmt grdvolume peaks.nc -Sk -C100/300/10 -Th > results.d
```

To see the areas and volumes for all the contours in the previous example, use

```
gmt grdvolume peaks.nc -Sk -C100/300/10 > results.d
```

To find the volume of water in a lake with its free surface at 0 and max depth of 300 meters, use

```
gmt grdvolume lake.nc -Cr-300/0
```

1.52.7 Notes

grdvolume distinguishes between gridline and pixel-registered grids. In both cases the area and volume are computed up to the grid boundaries. That means that in the first case the grid cells on the boundary only contribute half their area (and volume), whereas in the second case all grid cells are fully used. The exception is when the **-C** flag is used: since contours do not extend beyond the outermost grid point, both grid types are treated the same. That means the outer rim in pixel oriented grids is ignored when using the **-C** flag.

1.52.8 See Also

gmt, *grdfilter*, *grdmask*, *grdmath*

1.52.9 References

Wessel, P., 1998, An empirical method for optimal robust regional-residual separation of geophysical data, *Math. Geol.*, **30**(4), 391-408.

1.53 greenspline

greenspline - Interpolate using Green's functions for splines in 1-3 dimensions

1.53.1 Synopsis

```
greenspline [ table ] [ -A[1|2|3|4|5],[gradfile] ] [ -C[n|v]cut[/file] ] [ -Dmode ] [ -Ggrdfile ] [ -Ixinc[/yinc[/zinc]] ] [ -L ] [ -Nnodefile ] [ -Qaz|x/y/z ] [ -Rwest/east/south/north[/zmin/zmax][r] ] [ -Sctllr|rl|plq[/pars] ] [ -Tmaskgrid ] [ -V[level] ] [ -W ] [ -b[i|o][ncol][type][w][+L+B] ] [ -f[i|o]colinfo ] [ -h[i|o][n][+c][+d][+rremark][+rtitle] ] [ -ocol[s[,...]] ] [ -:[i|o] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.53.2 Description

greenspline uses the Green's function $G(\mathbf{x}; \mathbf{x}')$ for the chosen spline and geometry to interpolate data at regular [or arbitrary] output locations. Mathematically, the solution is composed as $w(\mathbf{x}) = \sum \{c(i) G(\mathbf{x}'; \mathbf{x}(i))\}$, for $i = 1, n$, the number of data points $\{\mathbf{x}(i), w(i)\}$. Once the n coefficients $c(i)$ have been found the sum can be evaluated at any output point \mathbf{x} . Choose between minimum curvature, regularized, or continuous curvature splines in tension for either 1-D, 2-D, or 3-D Cartesian coordinates or spherical surface coordinates. After first removing a linear or planar trend (Cartesian geometries) or mean value (spherical surface) and normalizing these residuals, the least-squares matrix solution for the spline coefficients $c(i)$ is found by solving the n by n linear system $w(j) = \sum\text{-over-}i \{c(i) G(\mathbf{x}(j); \mathbf{x}(i))\}$, for $j = 1, n$; this solution yields an exact interpolation of the supplied data points. Alternatively, you may choose to perform a singular value decomposition (SVD) and eliminate the contribution from the smallest eigenvalues; this approach yields an approximate solution. Trends and scales are restored when evaluating the output.

1.53.3 Required Arguments

None.

1.53.4 Optional Arguments

table The name of one or more ASCII [or binary, see **-bi**] files holding the \mathbf{x} , w data points. If no file is given then we read standard input instead.

-A[1|2|3|4|5],[gradfile] The solution will partly be constrained by surface gradients $\mathbf{v} = v \cdot \mathbf{n}$, where v is the gradient magnitude and \mathbf{n} its unit vector direction. The gradient direction may be specified either by Cartesian components (either unit vector \mathbf{n} and magnitude v separately or gradient components \mathbf{v} directly) or angles w.r.t. the coordinate axes. Specify one of five input formats: **0**: For 1-D data there is no direction, just gradient magnitude (slope) so the input format is x , *gradient*. Options 1-2 are for 2-D data sets: **1**: records contain x , y , *azimuth*, *gradient* (*azimuth* in degrees is measured clockwise from the vertical (north) [Default]). **2**: records contain x , y , *gradient*, *azimuth* (*azimuth* in degrees is measured clockwise from the vertical (north)). Options 3-5 are for either 2-D or 3-D data: **3**: records contain \mathbf{x} , *direction(s)*, v (*direction(s)* in degrees are measured counter-clockwise from the horizontal (and for 3-D the vertical axis)). **4**: records contain \mathbf{x} , \mathbf{v} . **5**: records contain \mathbf{x} , \mathbf{n} , v . Append name of ASCII file with the surface gradients (following a comma if a format is specified).

-C[n|v]cut[/file] Find an approximate surface fit: Solve the linear system for the spline coefficients by SVD and eliminate the contribution from all eigenvalues whose ratio to the largest eigenvalue is less than *cut* [Default uses Gauss-Jordan elimination to solve the linear system and fit the data exactly]. Optionally, append */file* to save the eigenvalue ratios to the specified file for further analysis. Finally, if a negative *cut* is given then */file* is required and execution will stop after saving the

eigenvalues, i.e., no surface output is produced. Specify **-Cv** to use the largest eigenvalues needed to explain *cut %* of the data variance. Alternatively, use **-Cn** to select the *cut* largest eigenvalues. If a *file* is given with **-Cv** then we save the eigenvalues instead of the ratios.

- Dmode** Sets the distance flag that determines how we calculate distances between data points. Select *mode* 0 for Cartesian 1-D spline interpolation: **-D0** means (*x*) in user units, Cartesian distances, Select *mode* 1-3 for Cartesian 2-D surface spline interpolation: **-D1** means (*x,y*) in user units, Cartesian distances, **-D2** for (*x,y*) in degrees, Flat Earth distances, and **-D3** for (*x,y*) in degrees, Spherical distances in km. Then, if *PROJ_ELLIPSOID* is spherical, we compute great circle arcs, otherwise geodesics. Option *mode* = 4 applies to spherical surface spline interpolation only: **-D4** for (*x,y*) in degrees, use cosine of great circle (or geodesic) arcs. Select *mode* 5 for Cartesian 3-D surface spline interpolation: **-D5** means (*x,y,z*) in user units, Cartesian distances.
- Ggrdfile** Name of resulting output file. (1) If options **-R**, **-I**, and possibly **-r** are set we produce an equidistant output table. This will be written to stdout unless **-G** is specified. Note: for 2-D grids the **-G** option is required. (2) If option **-T** is selected then **-G** is required and the output file is a 2-D binary grid file. Applies to 2-D interpolation only. (3) If **-N** is selected then the output is an ASCII (or binary; see **-bo**) table; if **-G** is not given then this table is written to standard output. Ignored if **-C** or **-C0** is given.
- Ixinc[/yinc[/zinc]]** Specify equidistant sampling intervals, on for each dimension, separated by slashes.
- L** Do *not* remove a linear (1-D) or planer (2-D) trend when **-D** selects mode 0-3 [For those Cartesian cases a least-squares line or plane is modeled and removed, then restored after fitting a spline to the residuals]. However, in mixed cases with both data values and gradients, or for spherical surface data, only the mean data value is removed (and later and restored).
- Nnodefile** ASCII file with coordinates of desired output locations *x* in the first column(s). The resulting *w* values are appended to each record and written to the file given in **-G** [or stdout if not specified]; see **-bo** for binary output instead. This option eliminates the need to specify options **-R**, **-I**, and **-r**.
- Qazlx/y/z** Rather than evaluate the surface, take the directional derivative in the *az* azimuth and return the magnitude of this derivative instead. For 3-D interpolation, specify the three components of the desired vector direction (the vector will be normalized before use).
- Rxmin/xmax[/ymin/ymax[/zminzmax]]** Specify the domain for an equidistant lattice where output predictions are required. Requires **-I** and optionally **-r**.
 - 1-D*: Give *xmin/xmax*, the minimum and maximum *x* coordinates.
 - 2-D*: Give *xmin/xmax/ymin/ymax*, the minimum and maximum *x* and *y* coordinates. These may be Cartesian or geographical. If geographical, then *west*, *east*, *south*, and *north* specify the Region of interest, and you may specify them in decimal degrees or in [+]*dd*:*mm*[:*ss.xxx*][*W|E|S|N*] format. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude).
 - 3-D*: Give *xmin/xmax/ymin/ymax/zmin/zmax*, the minimum and maximum *x*, *y* and *z* coordinates. See the 2-D section if your horizontal coordinates are geographical; note the shorthands **-Rg** and **-Rd** cannot be used if a 3-D domain is specified.
- Scitlllr|plq[*pars*]** Select one of six different splines. The first two are used for 1-D, 2-D, or 3-D Cartesian splines (see **-D** for discussion). Note that all tension values are expected to be normalized tension in the range $0 < t < 1$: (**c**) Minimum curvature spline [Sandwell, 1987], (**t**) Continuous curvature spline in tension [Wessel and Bercovici, 1998]; append *tension*[/*scale*] with *tension* in the 0-1 range and optionally supply a length scale [Default is the average grid spacing]. The next is a 1-D or 2-D spline: (**l**) Linear (1-D) or Bilinear (2-D) spline; these produce output that do not exceed the range of the given data. The next is a 2-D or 3-D spline: (**r**) Regularized spline in

- tension [*Mitasova and Mitas, 1993*]; again, append *tension* and optional *scale*. The last two are spherical surface splines and both imply **-D4**: **(p)** Minimum curvature spline [*Parker, 1994*], **(q)** Continuous curvature spline in tension [*Wessel and Becker, 2008*]; append *tension*. The $G(\mathbf{x}'; \mathbf{x}')$ for the last method is slower to compute (a series solution) so we pre-calculate values and use cubic spline interpolation lookup instead. Optionally append **+nN** (an odd integer) to change how many points to use in the spline setup [10001]. The finite Legendre sum has a truncation error [1e-6]; you can lower that by appending **+elimit** at the expense of longer run-time.
- Tmaskgrid** For 2-D interpolation only. Only evaluate the solution at the nodes in the *maskgrid* that are not equal to NaN. This option eliminates the need to specify options **-R**, **-I**, and **-r**.
 - V[level]** (*more ...*) Select verbosity level [c].
 - W** Expect data weights in the final input column, typically given as $\text{weight} = 1 / \text{sigma}$, the data uncertainty. This results in a weighted least squares fit. Note that this only has an effect if **-CC** is used.
 - bi[ncols][type]** (*more ...*) Select binary input. [Default is 2-4 input columns (\mathbf{x}, w); the number depends on the chosen dimension].
 - bo[ncols][type]** (*more ...*) Select binary output.
 - f[i]o[colinfo]** (*more ...*) Specify data types of input and/or output columns.
 - h[i]o[n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
 - icols[l][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
 - ocols[,...]** (*more ...*) Select output columns (0 is first column).
 - r** (*more ...*) Set pixel node registration [gridline].
 - ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
 - ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
 - ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
 - version** Print GMT version and exit.
 - show-datadir** Print full path to GMT share directory and exit.

1.53.5 1-d Examples

To resample the x, y Gaussian random data created by *gmtmath* and stored in 1D.txt, requesting output every 0.1 step from 0 to 10, and using a minimum cubic spline, try

```
gmt gmtmath -T0/10/1 0 1 NRAND = 1D.txt
gmt psxy -R0/10/-5/5 -JX6i/3i -B2f1/1 -Sc0.1 -Gblack 1D.txt -K > 1D.ps
gmt greenspline 1D.txt -R0/10 -I0.1 -Sc -V | psxy -R -J -O -Wthin >> 1D.ps
```

To apply a spline in tension instead, using a tension of 0.7, try

```
gmt psxy -R0/10/-5/5 -JX6i/3i -B2f1/1 -Sc0.1 -Gblack 1D.txt -K > 1Dt.ps
gmt greenspline 1D.txt -R0/10 -I0.1 -St0.7 -V | psxy -R -J -O -Wthin >> 1Dt.ps
```

1.53.6 2-d Examples

To make a uniform grid using the minimum curvature spline for the same Cartesian data set from Davis (1986) that is used in the GMT Technical Reference and Cookbook example 16, try

```
gmt greenspline table_5.11 -R0/6.5/-0.2/6.5 -I0.1 -Sc -V -D1 -GS1987.nc
gmt psxy -R0/6.5/-0.2/6.5 -JX6i -B2f1 -Sc0.1 -Gblack table_5.11 -K > 2D.ps
gmt grdcontour -JX6i -B2f1 -O -C25 -A50 S1987.nc >> 2D.ps
```

To use Cartesian splines in tension but only evaluate the solution where the input mask grid is not NaN, try

```
gmt greenspline table_5.11 -Tmask.nc -St0.5 -V -D1 -GWB1998.nc
```

To use Cartesian generalized splines in tension and return the magnitude of the surface slope in the NW direction, try

```
gmt greenspline table_5.11 -R0/6.5/-0.2/6.5 -I0.1 -Sr0.95 -V -D1 -Q-45 -Gslopes.nc
```

Finally, to use Cartesian minimum curvature splines in recovering a surface where the input data is a single surface value (pt.d) and the remaining constraints specify only the surface slope and direction (slopes.d), use

```
gmt greenspline pt.d -R-3.2/3.2/-3.2/3.2 -I0.1 -Sc -V -D1 -A1,slopes.d -Gslopes.nc
```

1.53.7 3-d Examples

To create a uniform 3-D Cartesian grid table based on the data in table_5.23 in Davis (1986) that contains x,y,z locations and a measure of uranium oxide concentrations (in percent), try

```
gmt greenspline table_5.23 -R5/40/-5/10/5/16 -I0.25 -Sr0.85 -V -D5 -G3D_UO2.txt
```

1.53.8 2-d Spherical Surface Examples

To recreate Parker's [1994] example on a global 1x1 degree grid, assuming the data are in file mag_obs_1990.d, try

```
greenspline -V -Rg -Sp -D3 -I1 -GP1994.nc mag_obs_1990.d
```

To do the same problem but applying tension of 0.85, use

```
greenspline -V -Rg -Sq0.85 -D3 -I1 -GWB2008.nc mag_obs_1990.d
```

1.53.9 Considerations

(1) For the Cartesian cases we use the free-space Green functions, hence no boundary conditions are applied at the edges of the specified domain. For most applications this is fine as the region typically is arbitrarily set to reflect the extent of your data. However, if your application requires particular boundary conditions then you may consider using *surface* instead.

(2) In all cases, the solution is obtained by inverting a $n \times n$ double precision matrix for the Green function coefficients, where n is the number of data constraints. Hence, your computer's memory may place restrictions on how large data sets you can process with *greenspline* <[greenspline.html](#)>. Pre-processing

your data with *blockmean* <*blockmean.html*>, *blockmedian* <*blockmedian.html*>, or *blockmode* <*blockmode.html*> is recommended to avoid aliasing and may also control the size of n . For information, if $n = 1024$ then only 8 Mb memory is needed, but for $n = 10240$ we need 800 Mb. Note that *greenspline* <*greenspline.html*> is fully 64-bit compliant if compiled as such. For spherical data you may consider decimating using *gmtspatial* <*gmtspatial.html*> nearest neighbor reduction.

(3) The inversion for coefficients can become numerically unstable when data neighbors are very close compared to the overall span of the data. You can remedy this by pre-processing the data, e.g., by averaging closely spaced neighbors. Alternatively, you can improve stability by using the SVD solution and discard information associated with the smallest eigenvalues (see **-C**).

(4) The series solution implemented for **-Sq** was developed by Robert L. Parker, Scripps Institution of Oceanography, which we gratefully acknowledge.

1.53.10 Tension

Tension is generally used to suppress spurious oscillations caused by the minimum curvature requirement, in particular when rapid gradient changes are present in the data. The proper amount of tension can only be determined by experimentation. Generally, very smooth data (such as potential fields) do not require much, if any tension, while rougher data (such as topography) will typically interpolate better with moderate tension. Make sure you try a range of values before choosing your final result. Note: the regularized spline in tension is only stable for a finite range of *scale* values; you must experiment to find the valid range and a useful setting. For more information on tension see the references below.

1.53.11 References

- Davis, J. C., 1986, *Statistics and Data Analysis in Geology*, 2nd Edition, 646 pp., Wiley, New York,
- Mitasova, H., and L. Mitas, 1993, Interpolation by regularized spline with tension: I. Theory and implementation, *Math. Geol.*, **25**, 641-655.
- Parker, R. L., 1994, *Geophysical Inverse Theory*, 386 pp., Princeton Univ. Press, Princeton, N.J.
- Sandwell, D. T., 1987, Biharmonic spline interpolation of Geos-3 and Seasat altimeter data, *Geophys. Res. Lett.*, **14**, 139-142.
- Wessel, P., and D. Bercovici, 1998, Interpolation with splines in tension: a Green's function approach, *Math. Geol.*, **30**, 77-93.
- Wessel, P., and J. M. Becker, 2008, Interpolation using a generalized Green's function for a spherical surface spline in tension, *Geophys. J. Int.*, **174**, 21-28.
- Wessel, P., 2009, A general-purpose Green's function interpolator, *Computers & Geosciences*, **35**, 1247-1254, doi:10.1016/j.cageo.2008.08.012.

1.53.12 See Also

gmt, *gmtmath*, *nearneighbor*, *psxy*, *sphtriangulate*, *surface*, *triangulate*, *xyz2grd*

1.54 isogmt

isogmt - Run GMT command or script in isolation mode

1.54.1 Synopsis

isogmt *command*

1.54.2 Description

isogmt runs a single GMT command or shell script in **isolation mode**. This means that the files *gmt.history* and *gmt.conf* will be read from the usual locations (current directory, *~/gmt*, or home directory), but changes will only be written in a temporary directory, which will be removed after execution. The name of the temporary directory will be available to the command or script as the environment variable `GMT_TMPDIR`.

1.54.3 Examples

Run the shell script *script.gmt* in isolation mode

```
isogmt sh script.gmt
```

1.54.4 See Also

gmt, *gmt.conf*

1.55 kml2gmt

kml2gmt - Extract GMT table data from Google Earth KML files

1.55.1 Synopsis

```
kml2gmt [ kmlfiles ] [ -V[level] ] [ -Z ] [ -bo[ncols][type][w][+L+B] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.55.2 Description

kml2gmt reads a Google Earth KML file and outputs a GMT table file. Only KML files that contain points, lines, or polygons can be processed. This is a bare-bones operation that aims to extract coordinates and possibly the name and description tags of each feature. The main use intended is to capture coordinates modified in Google Earth and then reinsert the modified data into the original GMT data file. For a more complete reformatting, consider using **ogr2ogr -f "GMT"** *somefile.gmt somefile.kml*.

1.55.3 Required Arguments

None.

1.55.4 Optional Arguments

kmlfiles Name of one or more KML files to work on. If not are given, then standard input is read.

-Z Output the altitude coordinates as GMT z coordinates [Default will output just longitude and latitude].

-V[level] (*more ...*) Select verbosity level [c].

-bo[ncols][type] (*more ...*) Select binary output.

-:[ilo] (*more ...*) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.55.5 Examples

To extract the lon,lat values from the KML file google.kml, try

```
gmt kml2gmt google.kml -V > google.txt
```

1.55.6 See Also

gmt, *gmt.conf*, *img2google*, *ps2raster*, *gmt2kml*

1.56 makecpt

makecpt - Make GMT color palette tables

1.56.1 Synopsis

```
makecpt [ -A[+]transparency ] [ -Ctable ] [ -D[ilo] ] [ -F[R|r|h|c] ] [ -Gzlo/zhi ] [ -I ] [ -M ] [ -N ] [ -Q[ilo] ] [ -Tz_min/z_max[z_inc[+]] ] [ -Tztable ] [ -V[level] ] [ -W ] [ -Z ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.56.2 Description

makecpt is a utility that will help you make color palette tables (cpt files). You define an equidistant set of contour intervals or pass your own z-table, and create a new cpt file based on an existing master cpt file. The resulting cpt file can be reversed relative to the master cpt, and can be

made continuous or discrete. For color tables beyond the standard GMT offerings, visit [cpt-city: http://soliton.vm.bytemark.co.uk/pub/cpt-city/](http://soliton.vm.bytemark.co.uk/pub/cpt-city/).

The color palette includes three additional colors beyond the range of z -values. These are the background color (B) assigned to values lower than the lowest z -value, the foreground color (F) assigned to values higher than the highest z -value, and the NaN color (N) painted wherever values are undefined.

If the master cpt file includes B, F, and N entries, these will be copied into the new master file. If not, the parameters `COLOR_BACKGROUND`, `COLOR_FOREGROUND`, and `COLOR_NAN` from the `gmt.conf` file or the command line will be used. This default behavior can be overruled using the options `-D`, `-M` or `-N`.

The color model (RGB, HSV or CMYK) of the palette created by `makecpt` will be the same as specified in the header of the master cpt file. When there is no `COLOR_MODEL` entry in the master cpt file, the `COLOR_MODEL` specified in the `gmt.conf` file or on the command line will be used.

1.56.3 Required Arguments

None.

1.56.4 Optional Arguments

- `-A[+]transparency` Sets a constant level of transparency (0-100) for all color slices. Prepend `+` to also affect the fore-, back-, and nan-colors [Default is no transparency, i.e., 0 (opaque)].
- `-Ctable` Selects the master color table *table* to use in the interpolation. Choose among the built-in tables (type `makecpt` to see the list) or give the name of an existing cpt file [Default gives a rainbow cpt file].
- `-D[ilo]` Select the back- and foreground colors to match the colors for lowest and highest z -values in the output cpt file [Default uses the colors specified in the master file, or those defined by the parameters `COLOR_BACKGROUND`, `COLOR_FOREGROUND`, and `COLOR_NAN`]. Append `i` to match the colors for the lowest and highest values in the input (instead of the output) cpt file.
- `-F[Rlr|lhc]` Force output cpt file to written with r/g/b codes, gray-scale values or color name (**R**, default) or r/g/b codes only (**r**), or h-s-v codes (**h**), or c/m/y/k codes (**c**).
- `-Gzlo/zhi` Truncate the incoming CPT so that the lowest and highest z -levels are to *zlo* and *zhi*. If one of these equal NaN then we leave that end of the CPT alone. The truncation takes place before any resampling.
- `-I` Reverses the sense of color progression in the master cpt file. Also exchanges the foreground and background colors, including those specified by the parameters `COLOR_BACKGROUND` and `COLOR_FOREGROUND`.
- `-M` Override background, foreground, and NaN colors specified in the master cpt file with the values of the parameters `COLOR_BACKGROUND`, `COLOR_FOREGROUND`, and `COLOR_NAN` specified in the `gmt.conf` file or on the command line. When combined with `-D`, only `COLOR_NAN` is considered.
- `-N` Do not write out the background, foreground, and NaN-color fields [Default will write them].
- `-Q[ilo]` Selects a logarithmic interpolation scheme [Default is linear]. `-Qi` expects input z -values to be $\log_{10}(z)$, assigns colors, and writes out z [Default]. `-Qo` takes $\log_{10}(z)$ first, assigns colors, and writes out z .

- Tz_min/z_max[/z_inc[+]] | -Tztable** Defines the range of the new cpt file by giving the lowest and highest z-value and interval. When used with **-C** and *z_inc* is not specified, the number of intervals remains the same as in the input palette. If *z_inc* is specified with a trailing **+** we interpret *z_inc* as the number of desired intervals instead. Alternatively, give the name of a ASCII file that has one z-value per record. If **-T** is not given, the existing range in the master cpt file will be used intact.
- V[level] (*more ...*)** Select verbosity level [c].
- W** Do not interpolate the input color table but pick the output colors starting at the beginning of the color table, until colors for all intervals are assigned. This is particularly useful in combination with a categorical color table, like “categorical”. Cannot be used in combination with **-Z**.
- Z** Creates a continuous cpt file [Default is discontinuous, i.e., constant colors for each interval].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.56.5 Examples

To make a cpt file with z-values from -200 to 200, with discrete color changes every 25, and using a polar blue-white-red colortable:

```
gmt makecpt -Cpolar -T-200/200/25 > colors.cpt
```

To make an equidistant cpt file from z = -2 to 6, in steps of 1, using continuous default rainbow colors:

```
gmt makecpt -T-2/6/1 -Z > rainbow.cpt
```

To make a GEBCO look-alike cpt file for bathymetry, run

```
gmt makecpt -Cgebco > my_gebco.cpt
```

1.56.6 Bugs

Since **makecpt** will also interpolate from any existing .cpt file you may have in your directory, you cannot use one of the listed cpt names as an output filename; hence the `my_gebco.cpt` in the example.

1.56.7 See Also

gmt, *grd2cpt*

1.57 mapproject

mapproject - Do forward and inverse map transformations, datum conversions and geodesy

1.57.1 Synopsis

```
mapproject [ table ] -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] [ -Ab|B|f|F|l|O[lon0/lat0] ] [ -C[dx/dy] ] [ -Dcl|lp ] [ -E[datum] ] [ -F[unit] ] [ -G[x0/y0][[+|-]unit][+|-] ] [ -I ] [ -Lline.xy[/[+|-]unit][+] ] [ -N[alc|gm] ] [ -Q[dle] ] [ -S ] [ -T[h]from/to] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D[col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][...] ] [ -ocols[...] ] [ -s[cols][alr] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.57.2 Description

mapproject reads (longitude, latitude) positions from *infile*s [or standard input] and computes (x,y) coordinates using the specified map projection and scales. Optionally, it can read (x,y) positions and compute (longitude, latitude) values doing the inverse transformation. This can be used to transform linear (x,y) points obtained by digitizing a map of known projection to geographical coordinates. May also calculate distances along track, to a fixed point, or closest approach to a line. Finally, can be used to perform various datum conversions. Additional data fields are permitted after the first 2 columns which must have (longitude,latitude) or (x,y). See option **-:** on how to read (latitude,longitude) files.

1.57.3 Required Arguments

-J*parameters* (*more ...*) Select map projection.

-R[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest. Special case for the UTM projection: If **-C** is used and **-R** is not given then the region is set to coincide with the given UTM zone so as to preserve the full ellipsoidal solution (See RESTRICTIONS for more information).

1.57.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Ab|B|f|F|l|O[*lon0/lat0*] **-Af** calculates the (forward) azimuth from fixed point *lon/lat* to each data point. Use **-Ab** to get back-azimuth from data points to fixed point. Use **-Ao** to get orientations (-90/90) rather than azimuths (0/360). Upper case **F**, **B** or **O** will convert from geodetic to geocentric latitudes and estimate azimuth of geodesics (assuming the current ellipsoid is not a sphere). If no fixed point is given then we compute the azimuth (or back-azimuth) from the previous point.

-C[*dx/dy*] Set center of projected coordinates to be at map projection center [Default is lower left corner]. Optionally, add offsets in the projected units to be added (or subtracted when **-I** is set) to (from) the projected coordinates, such as false eastings and northings for particular projection zones [0/0]. The unit used for the offsets is the plot distance unit in effect (see *PROJ_LENGTH_UNIT*) unless **-F** is used, in which case the offsets are in meters.

-Dcl|lp Temporarily override *PROJ_LENGTH_UNIT* and use **c** (cm), **i** (inch), or **p** (points) instead. Cannot be used with **-F**.

-E[*datum*] Convert from geodetic (lon, lat, height) to Earth Centered Earth Fixed (ECEF) (x,y,z) coordinates (add **-I** for the inverse conversion). Append datum ID (see **-Qd**) or give *ellipsoid:dx,dy,dz*

where *ellipsoid* may be an ellipsoid ID (see **-Qe**) or given as $a[,*inv_f*]$, where a is the semi-major axis and inv_f is the inverse flattening (0 if omitted). If *datum* is - or not given we assume WGS-84.

- F[*unit*]** Force 1:1 scaling, i.e., output (or input, see **-I**) data are in actual projected meters. To specify other units, append the desired unit (see UNITS). Without **-F**, the output (or input, see **-I**) are in the units specified by *PROJ_LENGTH_UNIT* (but see **-D**).
- G[x0/y0][[+|-]*unit*][+|-]** Calculate distances along track *or* to the optional point set with **-Gx0/y0**. Append the distance unit (see UNITS), including **c** (Cartesian distance using input coordinates) or **C** (Cartesian distance using projected coordinates). The **C** unit requires **-R** and **-J** to be set. With no fixed point is given we calculate cumulate distances along track. Append **-** to obtain incremental distance between successive points. Append **+** to specify the 2nd point via two extra columns in the input file.
- I** Do the Inverse transformation, i.e., get (longitude,latitude) from (x,y) data.
- Lline.xy/[+|-]*unit*[+]** Determine the shortest distance from the input data points to the line(s) given in the ASCII multisegment file *line.xy*. The distance and the coordinates of the nearest point will be appended to the output as three new columns. Append the distance unit (see UNITS), including **c** (Cartesian distance using input coordinates) or **C** (Cartesian distance using projected coordinates). The **C** unit requires **-R** and **-J** to be set. Finally, append **+** to report the line segment id and the fractional point number instead of lon/lat of the nearest point.
- N[alc|gm]** Convert from geodetic latitudes (using the current ellipsoid; see *PROJ_ELLIPSOID*) to one of four different auxiliary latitudes (longitudes are unaffected). Choose from **authalic**, **conformal**, **geocentric**, and **meridional** latitudes [geocentric]. Use **-I** to convert from auxiliary latitudes to geodetic latitudes.
- Q[dle]** List all projection parameters. To only list datums, use **-Qd**. To only list ellipsoids, use **-Qe**.
- S** Suppress points that fall outside the region.
- T[h]from[*to*]** Coordinate conversions between datums *from* and *to* using the standard Molodensky transformation. Use **-Th** if 3rd input column has height above ellipsoid [Default assumes height = 0, i.e., on the ellipsoid]. Specify datums using the datum ID (see **-Qd**) or give *ellipsoid:dx,dy,dz* where *ellipsoid* may be an ellipsoid ID (see **-Qe**) or given as $a[,*inv_f*]$, where a is the semi-major axis and inv_f is the inverse flattening (0 if omitted). If *datum* is - or not given we assume WGS-84. **-T** may be used in conjunction with **-R -J** to change the datum before coordinate projection (add **-I** to apply the datum conversion after the inverse projection). Make sure that the *PROJ_ELLIPSOID* setting is correct for your case.
- V[*level*]** (*more ...*) Select verbosity level [*c*].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. [Default is 2 input columns].
- bo[*ncols*][*type*]** (*more ...*) Select binary output. [Default is same as input].
- f[ilo]*colinfo*** (*more ...*) Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|ID[*col*]z[+|-]*gap*[*u*]** (*more ...*) Determine data gaps and line breaks.
- h[ilo][*n*][+*c*][+*d*][+*rremark*][+*rtitle*]** (*more ...*) Skip or produce header record(s).
- i[cols][1][*sscale*][*offset*][,*...*]** (*more ...*) Select input columns (0 is first column).
- o[cols][,*...*]** (*more ...*) Select output columns (0 is first column).
- s[cols][*alr*]** (*more ...*) Set handling of NaN records.

- :-[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.57.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.57.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.57.7 Examples

To transform a file with (longitude,latitude) into (x,y) positions in cm on a Mercator grid for a given scale of 0.5 cm per degree, run

```
gmt mapproject lonlatfile -R20/50/12/25 -Jm0.5c > xyfile
```

To transform several 2-column, binary, double precision files with (latitude,longitude) into (x,y) positions in inch on a Transverse Mercator grid (central longitude 75W) for scale = 1:500000 and suppress those points that would fall outside the map area, run

```
gmt mapproject tracks.* -R-80/-70/20/40 -Jt-75/1:500000 -: -S -Di -bo -bi2 > tmfile.b
```

To convert the geodetic coordinates (lon, lat, height) in the file old.dat from the NAD27 CONUS datum (Datum ID 131 which uses the Clarke-1866 ellipsoid) to WGS 84, run

```
gmt mapproject old.dat -Th131 > new.dat
```

To compute the closest distance (in km) between each point in the input file quakes.dat and the line segments given in the multisegment ASCII file coastline.xy, run

```
gmt mapproject quakes.dat -lcoastline.xy/k > quake_dist.dat
```

1.57.8 Restrictions

The rectangular input region set with **-R** will in general be mapped into a non-rectangular grid. Unless **-C** is set, the leftmost point on this grid has $xvalue = 0.0$, and the lowermost point will have $yvalue = 0.0$. Thus, before you digitize a map, run the extreme map coordinates through **mapproject** using the appropriate scale and see what (x,y) values they are mapped onto. Use these values when setting up for digitizing in order to have the inverse transformation work correctly, or alternatively, use **awk** to scale and shift the (x,y) values before transforming.

For some projection, a spherical solution may be used despite the user having selected an ellipsoid. This occurs when the users **-R** setting implies a region that exceeds the domain in which the ellipsoidal series expansions are valid. These are the conditions: (1) Lambert Conformal Conic (**-JL**) and Albers Equal-Area (**-JB**) will use the spherical solution when the map scale exceeds $1.0E7$. (2) Transverse Mercator (**-JT**) and UTM (**-JU**) will use the spherical solution when either the west or east boundary given in **-R** is more than 10 degrees from the central meridian, and (3) same for Cassini (**-JC**) but with a limit of only 4 degrees.

1.57.9 Ellipsoids And Spheroids

GMT will use ellipsoidal formulae if they are implemented and the user have selected an ellipsoid as the reference shape (see *PROJ_ELLIPSOID*). The user needs to be aware of a few potential pitfalls: (1) For some projections, such as Transverse Mercator, Albers, and Lambert's conformal conic we use the ellipsoidal expressions when the areas mapped are small, and switch to the spherical expressions (and substituting the appropriate auxiliary latitudes) for larger maps. The ellipsoidal formulae are used as follows: (a) Transverse Mercator: When all points are within 10 degrees of central meridian, (b) Conic projections when longitudinal range is less than 90 degrees, (c) Cassini projection when all points are within 4 degrees of central meridian. (2) When you are trying to match some historical data (e.g., coordinates obtained with a certain projection and a certain reference ellipsoid) you may find that GMT gives results that are slightly different. One likely source of this mismatch is that older calculations often used less significant digits. For instance, Snyder's examples often use the Clarke 1866 ellipsoid (defined by him as having a flattening $f = 1/294.98$). From f we get the eccentricity squared to be 0.00676862818 (this is what GMT uses), while Snyder rounds off and uses 0.00676866. This difference can give discrepancies of several tens of cm. If you need to reproduce coordinates projected with this slightly different eccentricity, you should specify your own ellipsoid with the same parameters as Clarke 1866, but with $f = 1/294.97861076$. Also, be aware that older data may be referenced to different datums, and unless you know which datum was used and convert all data to a common datum you may experience mismatches of tens to hundreds of meters. (3) Finally, be aware that *PROJ_SCALE_FACTOR* have certain default values for some projections so you may have to override the setting in order to match results produced with other settings.

1.57.10 See Also

gmt, *gmt.conf*, *gmtvector*, *project*

1.57.11 References

Bomford, G., 1952, Geodesy, Oxford U. Press.

Snyder, J. P., 1987, Map Projections - A Working Manual, U.S. Geological Survey Prof. Paper 1395.

Vanicek, P. and Krakiwsky, E., 1982, Geodesy - The Concepts, North-Holland Publ., ISBN: 0 444 86149 1.

1.58 nearneighbor

nearneighbor - Grid table data using a “Nearest neighbor” algorithm

1.58.1 Synopsis

```
nearneighbor [ table ] -Gout_grdfile -Ixinc[unit][=|+][yinc[unit][=|+]] -Nsectors[/min_sectors] -R[unit]xmin/xmax/ymin/ymax[r] [ -Ssearch_radius[unit] [ -Eempty ] [ -V[level] ] [ -W ] [ -bi[ncols][type][w][+L|+B] ] [ -f[i]o]colinfo ] [ -h[i]o][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -n[b|c|l|n][+a][+bBC][+threshold] ] [ -r ] [ -: [i]o ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.58.2 Description

nearneighbor reads arbitrarily located (x,y,z[,w]) triples [quadruplets] from standard input [or *table*] and uses a nearest neighbor algorithm to assign an average value to each node that have one or more points within a radius centered on the node. The average value is computed as a weighted mean of the nearest point from each sector inside the search radius. The weighting function used is $w(r) = 1 / (1 + d^2)$, where $d = 3 * r / \text{search_radius}$ and r is distance from the node. This weight is modulated by the weights of the observation points [if supplied].

1.58.3 Required Arguments

-Gout_grdfile Give the name of the output grid file.

-Ixinc[unit][=|+][yinc[unit][=|+]] x_inc [and optionally y_inc] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If y_inc is given but set to 0 it will be reset equal to x_inc ; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max x (*east*) or y (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-Nsectors[/min_sectors] The circular area centered on each node is divided into *sectors* sectors. Average values will only be computed if there is at least one value inside each of at least *min_sectors* of the sectors for a given node. Nodes that fail this test are assigned the value NaN (but see **-E**). If *Imin_sectors* is omitted it is set to be at least 50% of *sectors* (i.e., rounded up to next integer).

[Default is a quadrant search with 100% coverage, i.e., *sectors* = *min_sectors* = 4]. Note that only the nearest value per sector enters into the averaging; the more distant points are ignored.

-R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

-S*search_radius*[*unit*] Sets the *search_radius* that determines which data points are considered close to a node. Append the distance unit (see UNITS).

1.58.4 Optional Arguments

table 3 [or 4, see **-W**] column ASCII file(s) [or binary, see **-bi**] holding (x,y,z[,w]) data values. If no file is specified, **nearneighbor** will read from standard input.

-E*empty* Set the value assigned to empty nodes [NaN].

-V[*level*] (*more ...*) Select verbosity level [*c*].

-W Input data have a 4th column containing observation point weights. These are multiplied with the geometrical weight factor to determine the actual weights used in the calculations.

-bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 3 (or 4 if **-W** is set) columns].

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-h[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).

-i[*cols*][*l*][*sscale*][*offset*][,...] (*more ...*) Select input columns (0 is first column).

-n[*blc|ln*][+*a*][+*bBC*][+*tthreshold*] Append **+bBC** to set any boundary conditions to be used, adding **g** for geographic, **p** for periodic, or **n** for natural boundary conditions. For the latter two you may append **x** or **y** to specify just one direction, otherwise both are assumed. [Default is geographic if grid is geographic].

-r (*more ...*) Set pixel node registration [gridline].

-:[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.58.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.58.6 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.58.7 Examples

To create a gridded data set from the file `seaMARCII_bathy.lon_lat_z` using a 0.5 min grid, a 5 km search radius, using an octant search with 100% sector coverage, and set empty nodes to -9999:

```
gmt nearneighbor seaMARCII_bathy.lon_lat_z -R242/244/-22/-20 -I0.5m \
-E-9999 -Gbathymetry.nc -S5k -N8/8
```

To make a global grid file from the data in `geoid.xyz` using a 1 degree grid, a 200 km search radius, spherical distances, using an quadrant search, and set nodes to NaN only when fewer than two quadrants contain at least one value:

```
gmt nearneighbor geoid.xyz -R0/360/-90/90 -I1 -Lg -Ggeoid.nc -S200k -N4
```

1.58.8 See Also

blockmean, blockmedian, blockmode, gmt, greenspline, sphtriangulate, surface, triangulate

1.59 project

`project` - Project table data onto lines or great circles, generate tracks, or translate coordinates

1.59.1 Synopsis

```
project [ table ] -Ccx/cy [ -Aazimuth ] [ -Ebx/by ] [ -Fflags ] [ -Gdist[/colat][+] ] [ -L[w][l_min/l_max] ] [ -N ] [ -Q ] [ -S ] [ -Tpx/py ] [ -V[level] ] [ -Ww_min/w_max ] [ -b[i]o][ncol][type][w][+L+B] ] [ -f[i]o][colinfo] ] [ -g[a]x|y|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[i]o][n][+c][+d][+rremark][+rtitle] ] [ -i][cols][I][sscale][ooffset][,...] ] [ -s][cols][alr] ] [ -:][i][o] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.59.2 Description

`project` reads arbitrary (x, y, z) data from standard input [or *infile*] and writes to standard output any combination of (x, y, z, p, q, r, s), where (p, q) are the coordinates in the projection, (r, s) is the position in the (x, y) coordinate system of the point on the profile ($q = 0$ path) closest to (x, y), and z is all remaining columns in the input (beyond the required x and y columns).

Alternatively, `project` may be used to generate (r, s, p) triples at equal increments *dist* along a profile. In this case (**-G** option), no input is read.

Projections are defined in any (but only) one of three ways:

(Definition 1) By a Center **-C** and an Azimuth **-A** in degrees clockwise from North.

(Definition 2) By a Center **-C** and end point E of the projection path **-E**.

(Definition 3) By a Center **-C** and a roTation pole position **-T**.

To spherically project data along a great circle path, an oblique coordinate system is created which has its equator along that path, and the zero meridian through the Center. Then the oblique longitude (p) corresponds to the distance from the Center along the great circle, and the oblique latitude (q) corresponds to the distance perpendicular to the great circle path. When moving in the increasing (p) direction, (toward B or in the *azimuth* direction), the positive (q) direction is to your left. If a Pole has been specified, then the positive (q) direction is toward the pole.

To specify an oblique projection, use the **-T** option to set the Pole. Then the equator of the projection is already determined and the **-C** option is used to locate the $p = 0$ meridian. The Center cx/cy will be taken as a point through which the $p = 0$ meridian passes. If you do not care to choose a particular point, use the South pole ($ox = 0, oy = -90$).

Data can be selectively windowed by using the **-L** and **-W** options. If **-W** is used, the projection Width is set to use only points with $w_min < q < w_max$. If **-L** is set, then the Length is set to use only those points with $l_min < p < l_max$. If the **-E** option has been used to define the projection, then **-Lw** may be selected to window the length of the projection to exactly the span from **O** to **B**.

Flat Earth (Cartesian) coordinate transformations can also be made. Set **-N** and remember that *azimuth* is clockwise from North (the y axis), NOT the usual cartesian theta, which is counterclockwise from the x axis. $azimuth = 90 - \theta$.

No assumptions are made regarding the units for $x, y, r, s, p, q, dist, l_min, l_max, w_min, w_max$. If **-Q** is selected, map units are assumed and x, y, r, s must be in degrees and $p, q, dist, l_min, l_max, w_min, w_max$ will be in km.

Calculations of specific great-circle and geodesic distances or for back-azimuths or azimuths are better done using **mapproject**.

project is CASE SENSITIVE. Use UPPER CASE for all one-letter designators which begin optional arguments. Use lower case for the $xyzpqrs$ letters in **-flags**.

1.59.3 Required Arguments

-Ccx/cy cx/cy sets the origin of the projection, in Definition 1 or 2. If Definition 3 is used (**-T**), then cx/cy are the coordinates of a point through which the oblique zero meridian ($p = 0$) should pass. The cx/cy is not required to be 90 degrees from the pole.

1.59.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[$ncols$][$type$]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Aazimuth *azimuth* defines the azimuth of the projection (Definition 1).

-Ebx/by bx/by defines the end point of the projection path (Definition 2).

-Fflags Specify your desired output using any combination of $xyzpqrs$, in any order. Do not space between the letters. Use lower case. The output will be ASCII (or binary, see **-bo**) columns of values

corresponding to *xyzpqrs* [Default]. If both input and output are using ASCII format then the *z* data are treated as textstring(s). If the **-G** option is selected, the output will be *rsp*.

- Gdist[/colat]** Generate mode. No input is read. Create (*r*, *s*, *p*) output points every *dist* units of *p*. See **-Q** option. Alternatively, append */colat* for a small circle instead [Default is a colatitude of 90, i.e., a great circle]. Use **-C** and **-E** to generate a circle that goes through the center and end point. Note, in this case the center and end point cannot be farther apart than $2 * |colat|$. Finally, if you append **+** then we will report the position of the pole as part of the segment header [no header].
- L[w][l_min/l_max]** Length controls. Project only those points whose *p* coordinate is within $l_min < p < l_max$. If **-E** has been set, then you may use **-Lw** to stay within the distance from **C** to **E**.
- N** Flat Earth. Make a Cartesian coordinate transformation in the plane. [Default uses spherical trigonometry.]
- Q** Map type units, i.e., project assumes *x*, *y*, *r*, *s* are in degrees while *p*, *q*, *dist*, *l_min*, *l_max*, *w_min*, *w_max* are in km. If **-Q** is not set, then all these are assumed to be in the same units.
- S** Sort the output into increasing *p* order. Useful when projecting random data into a sequential profile.
- Tpx/py** *px/py* sets the position of the rotation pole of the projection. (Definition 3).
- V[level] (more ...)** Select verbosity level [*c*].
- Ww_min/w_max** Width controls. Project only those points whose *q* coordinate is within $w_min < q < w_max$.
- bi[ncols][type] (more ...)** Select binary input. [Default is 2 input columns].
- bo[ncols][type] (more ...)** Select binary output. [Default is given by **-F** or **-G**].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- g[a]xly|dl|X|Y|D|[col]z[+|-]gap[u] (more ...)** Determine data gaps and line breaks.
- h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- s[cols][alr] (more ...)** Set handling of NaN records.
- :[ilo] (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.59.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in

the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the `FORMAT_FLOAT_OUT` setting.

1.59.6 Examples

To generate points every 10km along a great circle from 10N,50W to 30N,10W:

```
gmt project -C-50/10 -E-10/30 -G10 -Q > great_circle_points.xyp
```

(Note that `great_circle_points.xyp` could now be used as input for **grdtrack**, etc.).

To generate points every 10km along a small circle of colatitude 60 from 10N,50W to 30N,10W:

```
gmt project -C-50/10 -E-10/30 -G10/60 -Q > small_circle_points.xyp
```

To create a partial small circle of colatitude 80 about a pole at 40E,85N, with extent of 45 degrees to either side of the meridian defined by the great circle from the pole to a point 15E,15N, try

```
gmt project -C15/15 -T40/85 -G1/80 -L-45/45 > some_circle.xyp
```

To project the shiptrack gravity, magnetics, and bathymetry in `c2610.xygmb` along a great circle through an origin at 30S, 30W, the great circle having an azimuth of N20W at the origin, keeping only the data from NE of the profile and within +/- 500 km of the origin, run:

```
gmt project c2610.xygmb -C-30/-30 -A-20 -W-10000/0 -L-500/500 -Fpz -Q > c2610_projected.pgmb
```

(Note in this example that **-W-10000/0** is used to admit any value with a large negative q coordinate. This will take those points which are on our right as we walk along the great circle path, or to the NE in this example.)

To make a Cartesian coordinate transformation of `mydata.xy` so that the new origin is at 5,3 and the new x axis (p) makes an angle of 20 degrees with the old x axis, use:

```
gmt project mydata.xy -C5/3 -A70 -Fpq > mydata.pq
```

To take data in the file `pacific.lonlat` and transform it into oblique coordinates using a pole from the hotspot reference frame and placing the oblique zero meridian ($p = 0$ line) through Tahiti, run:

```
gmt project pacific.lonlat -T-75/68 -C-149:26/-17:37 -Fpq > pacific.pq
```

Suppose that `pacific_topo.nc` is a grid file of bathymetry, and you want to make a file of flowlines in the hotspot reference frame. If you run:

```
gmt grd2xyz pacific_topo.nc | project -T-75/68 -C0/-90 -Fxyq | xyz2grd -Retc -Ietc -Cflow.nc
```

then `flow.nc` is a file in the same area as `pacific_topo.nc`, but `flow` contains the latitudes about the pole of the projection. You now can use `grdcontour` on `flow.nc` to draw lines of constant oblique latitude, which are flow lines in the hotspot frame.

If you have an arbitrarily rotation pole px/py and you would like to draw an oblique small circle on a map, you will first need to make a file with the oblique coordinates for the small circle (i.e., lon = 0-360, lat is constant), then create a file with two records: the north pole (0/90) and the origin (0/0), and find what their oblique coordinates are using your rotation pole. Now, use the projected North pole and origin coordinates as the rotation pole and center, respectively, and project your file as in the pacific example above. This gives coordinates for an oblique small circle.

1.59.7 See Also

fitcircle, gmt, gmtvector, mapproject, grdproject

1.60 ps2raster

ps2raster - Convert [E]PS file(s) to other formats using GhostScript

1.60.1 Synopsis

```
ps2raster  psfile(s)  [ -A[u][margins][-][+r][+s]Swidth[u]/height[u] ] [ -Cgs_option
] [ -Doutdir ] [ -Eresolution ] [ -Gghost_path ] [ -I ] [ -Llistfile ] [ -
P ] [ -Q[gl][1|2|4] ] [ -S ] [ -Tble|E|f|F|j|g|G|mlt ] [ -V[level] ] [ -
W[+g][+tdocname][+nlayername][+ofoldername][+aaltmode[alt]][+lminLOD/maxLOD][+fminfade/maxfade][+uURL]
]
```

Note: No space is allowed between the option flag and the associated arguments.

1.60.2 Description

ps2raster converts one or more PostScript files to other formats (BMP, EPS, JPEG, PDF, PNG, PPM, TIFF) using GhostScript. Input file names are read from the command line or from a file that lists them. The size of the resulting images is determined by the BoundingBox (or HiResBoundingBox, if present). As an option, a tight (HiRes)BoundingBox may be computed first. As another option, it can compute ESRI type world files used to reference, for instance, tif files and make them be recognized as geotiff. Note: If the PostScript file calls on any of the Adobe PDF transparency extensions *and* PDF is not the selected output format, then the file will first be converted to a temporary PDF file (for the transparency to take effect) before converting the PDF to the desired output format.

1.60.3 Required Arguments

psfiles Names of PostScript files to be converted. The output files will have the same name (unless **-F** is used) but with the conventional extension name associated to the raster format (e.g., .jpg for the jpeg format). Use **-D** to redirect the output to a different directory.

1.60.4 Optional Arguments

-A[u][margins][-][+r][+s]Swidth[u]/height[u] Adjust the BoundingBox and HiResBoundingBox to the minimum required by the image content. Append **u** to first remove any GMT-produced time-stamps. Optionally, append extra margins to the bounding box. Give either one (uniform), two (x and y) or four (individual sides) margins; append unit [Default is set by *PROJ_LENGTH_UNIT*]. Alternatively, use **-A-** to override any automatic setting of **-A** by **-W**.

Use the **-A+snew_width** to resize the output image to exactly *new_width* units. The default is to use what is set by *PROJ_LENGTH_UNIT* but you can append a new unit and/or impose different width and height. What happens here is that GhostScript will do the re-interpolation work and the final image will retain the DPI resolution set by **-E**. Alternatively use **-A+Sscale** to scale the image by a constant factor.

- Use the **-A+r** to round the HighRes BoundingBox instead of using the *ceil* function. This is going against Adobe Law but can be useful when creating very small images where the difference of one pixel might matter.
- Cgs_option** Specify a single, custom option that will be passed on to GhostScript as is. Repeat to add several options [none].
 - Doutdir** Sets an alternative output directory (which must exist) [Default is the same directory as the PS files]. Use **-D.** to place the output in the current directory instead.
 - Eresolution** Set raster resolution in dpi [default = 720 for PDF, 300 for others].
 - F** Force the output file name. By default output names are constructed using the input names as base, which are appended with an appropriate extension. Use this option to provide a different name, but without extension. Extension is still determined automatically.
 - Gghost_path** Full path to your GhostScript executable. NOTE: For Unix systems this is generally not necessary. Under Windows, the GhostScript path is now fetched from the registry. If this fails you can still add the GS path to system's path or give the full path here. (e.g., **-Gc:\programs\gs\gs9.02\bin\gswin64c**). WARNING: because of the poor decision of embedding the bits on the gs exe name we cannot satisfy both the 32 and 64 bits GhostScript executable names. So in case of 'get from registry' failure the default name (when no **-G** is used) is the one of the 64 bits version, or gswin64c
 - I** Enforce gray-shades by using ICC profiles. GhostScript versions ≥ 9.00 change gray-shades by using ICC profiles. GhostScript 9.05 and above provide the `'-dUseFastColor=true'` option to prevent that and that is what ps2raster does by default, unless option **-I** is set. Note that for GhostScript ≥ 9.00 and < 9.05 the gray-shade shifting is applied to all but PDF format. We have no solution to offer other than upgrade GhostScript.
 - Llistfile** The *listfile* is an ASCII file with the names of the PostScript files to be converted.
 - N** This option is obsolete. Use **-S** to print the GhostScript command, if applicable. Use **-Te** to save the intermediate EPS file.
 - P** Force Portrait mode. All Landscape mode plots will be rotated back so that they show unrotated in Portrait mode. This is practical when converting to image formats or preparing EPS or PDF plots for inclusion in documents.
 - Q[gt][1|2|4]** Set the anti-aliasing options for **graphics** or **text**. Append the size of the subsample box (1, 2, or 4) [4]. Default is no anti-aliasing (same as *bits* = 1).
 - S** Print to standard error the GhostScript command after it has been executed. This option also prevent all intermediate files from being removed.
 - Tble|l|f|F|j|g|G|m|t** Sets the output format, where **b** means BMP, **e** means EPS, **E** means EPS with PageSize command, **f** means PDF, **F** means multi-page PDF, **j** means JPEG, **g** means PNG, **G** means transparent PNG (untouched regions are transparent), **m** means PPM, and **t** means TIFF [default is JPEG]. To **bjgt** you can append **-** in order to get a grayscale image. The EPS format can be combined with any of the other formats. For example, **-Tef** creates both an EPS and a PDF file. The **-TF** creates a multi-page PDF file from the list of input PS or PDF files. It requires the **-F** option.
 - V[level] (more ...)** Select verbosity level [c].
 - W[+g][+tdocname][+nlayername][+ofoldername][+aaltmode[alt]][+lminLOD/maxLOD][+fminfade/maxfade][+uU]**
Write a ESRI type world file suitable to make (e.g) .tif files be recognized as geotiff by softwares that know how to do it. Be aware, however, that different results are obtained depending on

the image contents and if the **-B** option has been used or not. The trouble with the **-B** option is that it creates a frame and very likely its annotations. That introduces pixels outside the map data extent, and therefore the map extents estimation will be wrong. To avoid this problem use `--MAP_FRAME_TYPE=inside` option which plots all annotations and ticks inside the image and therefore does not compromise the coordinate computations. Pay attention also to the cases when the plot has any of the sides with whites only because than the algorithm will fail miserably as those whites will be eaten by the GhostScript. In that case you really must use **-B** or use a slightly off-white color.

Together with **-V** it prints on screen the `gdal_translate` (`gdal_translate` is a command line tool from the GDAL package) command that reads the raster + world file and creates a true geotiff file. Use **-W+g** to do a system call to `gdal_translate` and create a geoTIFF image right away. The output file will have a `.tiff` extension.

The world file naming follows the convention of jamming a 'w' in the file extension. So, if output is `tif` **-Tt** the world file is a `.tfw`, for jpeg we have a `.jgw` and so on. This option automatically sets **-A -P**.

Use **-W+k** to create a minimalist KML file that allows loading the image in GoogleEarth. Note that for this option the image must be in geographical coordinates. If not, a warning is issued but the KML file is created anyway. Several modifier options are available to customize the KML file in the form of `+opt` strings. Append **+ttitle** to set the document title [GMT KML Document], **+nlayername** to set the layer name, and **+a/altmode[altitude]** to select one of 5 altitude modes recognized by Google Earth that determines the altitude (in m) of the image: **G** clamped to the ground, **g** append altitude relative to ground, **a** append absolute altitude, **s** append altitude relative to seafloor, and **S** clamp it to the seafloor. Control visibility of the layer with the **+lminLOD/maxLOD** and **+fminfade/maxfade** options. Finally, if you plan to leave the image itself on a server and only distribute the KML, use **+uURL** to prepend the URL to the image reference. If you are building a multi-component KML file then you can issue a KML snippet without the KML header and trailer by using the **+ofoldername** modification; it will enclose the image and associated KML code within a KML folder of the specified name. See the KML documentation for further explanation (<http://code.google.com/apis/kml/documentation/>).

Further notes on the creation of georeferenced rasters. **ps2raster** can create a georeferenced raster image with a world file OR uses GDAL to convert the GMT PostScript file to geotiff. GDAL uses Proj.4 for its projection library. To provide with the information it needs to do the georeferencing, GMT 4.5 embeds a comment near the start of the PostScript file defining the projection using Proj.4 syntax. Users with pre-GMT v4.5 PostScript files, or even non-GMT ps files, can provide the information **ps2raster** requires by manually editing a line into the PostScript file, prefixed with `%%PROJ`.

For example the command

```
gmt pscoast -JM0/12c -R-10/-4/37/43 \
-Wl -Di -Bg30m --MAP_FRAME_TYPE=inside > cara.ps
```

adds this comment line

```
%%PROJ: merc -10.0 -4.0 37.0 43.0 -1113194.908 -445277.963
4413389.889 5282821.824 +proj=merc +lon_0=0 +k=-1 +x_0=0 +y_0=0
+a=6378137.0 +b=6356752.314245
```

where 'merc' is the keyword for the coordinate conversion; the 2 to 5th elements contain the map limits, 6 to 9th the map limits in projected coordinates and the rest of the line has the regular proj4 string for this projection.

- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.60.5 Notes

The conversion to raster images (BMP, JPEG, PNG, PPM or TIFF) inherently results in loss of details that are available in the original PostScript file. Choose a resolution that is large enough for the application that the image will be used for. For web pages, smaller dpi values suffice, for Word documents and PowerPoint presentations a higher dpi value is recommended. **ps2raster** uses the loss-less DEFLATE compression technique when creating PDF and PNG files and LZW compression for TIFF images.

EPS is a vector, not a raster format. Therefore, the **-E** option has no effect on the creation of EPS files. Using the option **-Te** will remove PageSize commands from the PostScript file and will adjust the BoundingBox when the **-A** option is used. Note the original and required BoundingBox is limited to integer points, hence Adobe added the optional HiResBoundingBox to add more precision in sizing. The **-A** option calculates both and writes both to the EPS file used in the rasterization (and output if **-Te** is set).

Although PDF is also a vector format, the **-E** option has an effect on the resolution of pattern fills and fonts that are stored as bitmaps in the document. **ps2raster** therefore uses a larger default resolution when creating PDF files. In order to obtain high-quality PDF files, the */prepress* options are in effect, allowing only loss-less DEFLATE compression of raster images embedded in the PostScript file.

Although **ps2raster** was developed as part of the GMT, it can be used to convert PostScript files created by nearly any graphics program. However, **-Au** is GMT-specific.

See *include-gmt-graphics* of the **GMT Technical Reference and Cookbook** for more information on how **ps2raster** is used to produce graphics that can be inserted into other documents (articles, presentations, posters, etc.).

1.60.6 Examples

To convert the file `psfile.ps` to PNG using a tight BoundingBox and rotating it back to normal orientation in case it was in Landscape mode:

```
gmt ps2raster psfile.ps -A -P -Tg
```

To create a 5 cm PNG version at 300 dpi of our `example_01.ps` file

```
gmt ps2raster example_01.ps -A+s5c -Tg
```

To create a 3 pages PDF file from 3 individual PS files

```
gmt ps2raster -TF -Fabc a.ps b.ps c.ps
```

To create a simple linear map with `pscoast` and convert it to `tif` with a `.tfw` the tight `BoundingBox` computation.

```
gmt pscoast -JX12cd -R-10/-4/37/43 -W1 -Di -Bg30m -P -G200 --MAP_FRAME_TYPE=inside > cara.ps
gmt ps2raster cara.ps -Tt -W
```

To create a Mercator version of the above example and use `GDAL` to produce a true geotiff file.

```
gmt pscoast -JM0/12c -R-10/-4/37/43 -W1 -Di -Bg30m -P -G200 --MAP_FRAME_TYPE=inside > cara.ps
gdalwarp -s_srs +proj=merc cara.tif carageo.tiff
```

To create a Polar Stereographic geotiff file of Patagonia

```
gmt pscoast -JS-55/-60/15c -R-77/-55/-57.5/-48r -Di -Gred -P -Bg2 \
--MAP_FRAME_TYPE=inside > patagonia.ps
gmt ps2raster patagonia.ps -Tt -W+g -V
```

To create a simple `KMZ` file for use in Google Earth, try

```
gmt grdimage lonlatgrid.nc -Jx1 -Ccolors.cpt -P -B0g2 --MAP_FRAME_TYPE=inside > tile.ps
gmt ps2raster tile.ps -Tg -W+k+t"my title"+1256/-1 -V
```

(These commands assume that `GhostScript` can be found in your system's path.)

1.60.7 GhostScript Options

Most of the conversions done in `ps2raster` are handled by `GhostScript`. On most Unixes this program is available as `gs`; for Windows there is a version called `gswin32c`. `GhostScript` accepts a rich selection of command-line options that modify its behavior. Many of these are set indirectly by the options available above. However, hard-core usage may require some users to add additional options to fine-tune the result. Use `-S` to examine the actual command used, and add custom options via one or more instances of the `-C` option. For instance, to turn on image interpolation for all images, improving image quality for scaled images at the expense of speed, use `-C-dDOINTERPOLATE`. See www.ghostscript.com for complete documentation.

1.60.8 See Also

gmt

1.61 psbasemap

`psbasemap` - Plot PostScript base maps

1.61.1 Synopsis

```
psbasemap -Jparameters -Rwest/east/south/north[/zmin/zmax][r] [ -B[pls]parameters ] [ -D[unit]xmin/xmax/ymin/ymax[r]|width[/height][+cclon/clat][+ppen][+gfill] [ -K ] [ -JZparameters ]
[ -L[f][x]lon0/lat0[/slon]/slat/length[elflk]Mlnlu][+llabel][+jjust][+ppen][+gfill][+u] ] [ -O ]
[ -P ] [ -U[just/dx/dy][clabel] ] [ -T[flm][x]lon0/lat0/size[/info][:w,e,s,n]:][+gint[/mint] ] [
-V[level] ] [ -X[alcflr][x-shiftu] ] [ -Y[alcflr][y-shiftu] ] [ -ccopies ] [ -f[ilo]colinfo ] [ -p[xlyz]azim/elev/zlevel][+wlon0/lat0/z0][+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.61.2 Description

psbasemap creates PostScript code that will produce a basemap. Several map projections are available, and the user may specify separate tickmark intervals for boundary annotation, ticking, and [optionally] gridlines. A simple map scale or directional rose may also be plotted. At least one of the options **-B**, **-L[f][x]lon0/lat0[/slon]/slat/length[elflk|M|n|u][+llabel][+jjust][+ppen][+gfill][+u]**, or **-T[flm][x]lon0/lat0/size[/info][:w,e,s,n:][+gint[/mint]]** must be specified.

1.61.3 Required Arguments

-Jparameters (more ...) Select map projection.

-R[unit]xmin/xmax/ymin/ymax[r] (more ...) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax*. (more ...)

1.61.4 Optional Arguments

-B[pls]parameters (more ...) Set map boundary intervals.

-D[unit]xmin/xmax/ymin/ymax[r]width[/height][+cclon/clat][+ppen][+gfill] Draw a simple map insert box on the map. Specify the box in one of three ways: (a) Give *west/east/south/north* of geographic rectangle bounded by parallels and meridians; append **r** if the coordinates instead are the lower left and upper right corners of the desired rectangle. (b) Give **u**xmin/xmax/ymin/ymax of bounding rectangle in projected coordinates (here, **u** is the coordinate unit). (c) Give **[u]width[/height]** of bounding rectangle and use **+c** to set box center. Append any combination of the following modifiers to draw the insert box: **+c**lon/lat to specify box center. **+gfill** to paint a insert [no fill]. **+ppen** to draw the insert outline [no outline].

-Jz/Zparameters (more ...) Set z-axis scaling; same syntax as **-Jx**.

-K (more ...) Do not finalize the PostScript plot.

-L[f][x]lon0/lat0[/slon]/slat/length[elflk|M|n|u][+llabel][+jjust][+ppen][+gfill][+u] Draws a simple map scale centered on *lon0/lat0*. Use **-Lx** to specify x/y position instead. Scale is calculated at latitude *slat* (optionally supply longitude *slon* for oblique projections [Default is central meridian]), *length* is in km, or append unit from **elflk|M|n|u**. Use **-Lf** to get a “fancy” scale [Default is plain]. Append **+l** to select the default label which equals the distance unit (meter, foot, km, mile, nautical mile, US survey foot) and is justified on top of the scale [t]. Change this by giving your own label (append **+llabel**). Change label justification with **+jjustification** (choose among l(ef), r(ight), t(op), and b(ottom)). Apply **+u** to append the unit to all distance annotations along the scale. If you want to place a rectangle behind the scale, specify suitable **+ppen** and/or **+gfill** parameters. Note: Use *FONT_LABEL* to change the label font and *FONT_ANNOT_PRIMARY* to change the annotation font. The height of the map scale is controlled by *MAP_SCALE_HEIGHT*.

-O (more ...) Append to existing PostScript plot.

-P (more ...) Select “Portrait” plot orientation.

-T[flm][x]lon0/lat0/size[/info][:w,e,s,n:][+gint[/mint]] Draws a simple map directional rose centered on *lon0/lat0*. Use **-Tx** to specify x/y position instead. The *size* is the diameter of the rose, and optional label information can be specified to override the default values of W, E, S, and N (Give **::** to suppress all labels). The default [plain] map rose only labels north. Use **-Tf** to get a “fancy” rose, and specify in *info* what you want drawn. The default [**1**] draws the two principal E-W, N-S

orientations, **2** adds the two intermediate NW-SE and NE-SW orientations, while **3** adds the eight minor orientations WNW-ESE, NNW-SSE, NNE-SSW, and ENE-WSW. For a magnetic compass rose, specify **-Tm**. If given, *info* must be the two parameters *dec/dlabel*, where *dec* is the magnetic declination and *dlabel* is a label for the magnetic compass needle (specify - to format a label from *dec*). Then, both directions to geographic and magnetic north are plotted [Default is geographic only]. If the north label is * then a north star is plotted instead of the north label. Annotation and two levels of tick intervals for both geographic and magnetic directions are 30/5/1 degrees; override these settings by appending *+gints[/mints]*. Color and pen attributes for the rose are taken from **COLOR_BACKGROUND** and **MAP_TICK_PEN**, respectively, while label fonts, colors and sizes follow **FONT_TITLE** for the four major directions and **FONT_LABEL** for minor directions.

- U[*just/dx/dy/*][*clabel*] (*more ...*)** Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*)** Select verbosity level [*c*].
- X[*alcflr*][*x-shift*[*u*]]**
- Y[*alcflr*][*y-shift*[*u*]] (*more ...*)** Shift plot origin.
- ccopies (*more ...*)** Specify number of plot copies [Default is 1].
- f[*ilo*]*colinfo* (*more ...*)** Specify data types of input and/or output columns. This applies only to the coordinates specified in the **-R** option.
- p[*x|y|z*]*azim/elev*[*/zlevel*][*+wlon0/lat0*[*/z0*]][*+vx0/y0*] (*more ...*)** Select perspective view.
- t[*transp*] (*more ...*)** Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +- or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.61.5 Examples

The following section illustrates the use of the options by giving some examples for the available map projections. Note how scales may be given in several different ways depending on the projection. Also note the use of upper case letters to specify map width instead of map scale.

1.61.6 Non-geographical Projections

Linear x-y plot

To make a linear x/y frame with all axes, but with only left and bottom axes annotated, using *xscale = yscale = 1.0*, ticking every 1 unit and annotating every 2, and using *xlabel = "Distance"* and *ylabel = "No of samples"*, use

```
gmt psbasemap -R0/9/0/5 -Jx1 -Bf1a2:Distance:/:"No of samples":WeSn > linear.ps
```

Log-log plot

To make a log-log frame with only the left and bottom axes, where the x-axis is 25 cm and annotated every 1-2-5 and the y-axis is 15 cm and annotated every power of 10 but has tickmarks every 0.1, run

```
gmt psbasemap -R1/10000/1e20/1e25 -JX25c1/15c1 -B2:Wavelength:/alp3:Power:WS > loglog.ps
```

Power axes

To design an axis system to be used for a depth-sqrt(age) plot with depth positive down, ticked and annotated every 500m, and ages annotated at 1 my, 4 my, 9 my etc, use

```
gmt psbasemap -R0/100/0/5000 -Jx1p0.5/-0.001 -B1p:"Crustal age"/:500:Depth: > power.ps
```

Polar (theta,r) plot

For a base map for use with polar coordinates, where the radius from 0 to 1000 should correspond to 3 inch and with gridlines and ticks every 30 degrees and 100 units, use

```
gmt psbasemap -R0/360/0/1000 -JP6i -B30p/100 > polar.ps
```

1.61.7 Cylindrical Map Projections

Cassini

A 10-cm-wide basemap using the Cassini projection may be obtained by

```
gmt psbasemap -R20/50/20/35 -JC35/28/10c -P -B5g5:.Cassini: > cassini.ps
```

Mercator [conformal]

A Mercator map with scale 0.025 inch/degree along equator, and showing the length of 5000 km along the equator (centered on 1/1 inch), may be plotted as

```
gmt psbasemap -R90/180/-50/50 -Jm0.025i -B30g30:.Mercator: -Lx1i/1i/0/5000 > mercator.ps
```

Miller

A global Miller cylindrical map with scale 1:200,000,000 may be plotted as

```
gmt psbasemap -Rg -Jj180/1:200000000 -B30g30:.Miller: > miller.ps
```

Oblique Mercator [conformal]

To create a page-size global oblique Mercator basemap for a pole at (90,30) with gridlines every 30 degrees, run

```
gmt psbasemap -R0/360/-70/70 -Joc0/0/90/30/0.064cd -B30g30:."Oblique Mercator": > oblmerc.ps
```

Transverse Mercator [conformal]

A regular Transverse Mercator basemap for some region may look like

```
gmt psbasemap -R69:30/71:45/-17/-15:15 -Jt70/1:1000000 -B15m:."Survey area": -P > transmerc.ps
```

Equidistant Cylindrical Projection

This projection only needs the central meridian and scale. A 25 cm wide global basemap centered on the 130E meridian is made by

```
gmt psbasemap -R-50/310/-90/90 -JQ130/25c -B30g30:."Equidistant Cylindrical": > cyl\_eqdist.ps
```

Universal Transverse Mercator [conformal]

To use this projection you must know the UTM zone number, which defines the central meridian. A UTM basemap for Indo-China can be plotted as

```
gmt psbasemap -R95/5/108/20r -Ju46/1:10000000 -B3g3:."UTM": > utm.ps
```

Cylindrical Equal-Area

First select which of the cylindrical equal-area projections you want by deciding on the standard parallel. Here we will use 45 degrees which gives the Gall-Peters projection. A 9 inch wide global basemap centered on the Pacific is made by

```
gmt psbasemap -Rg -JY180/45/9i -B30g30:."Gall-Peters": > gall-peters.ps
```

1.61.8 Conic Map Projections

Albers [equal-area]

A basemap for middle Europe may be created by

```
gmt psbasemap -R0/90/25/55 -Jb45/20/32/45/0.25c -B10g10:."Albers Equal-area": > albers.ps
```

Lambert [conformal]

Another basemap for middle Europe may be created by

```
gmt psbasemap -R0/90/25/55 -Jl45/20/32/45/0.1i -B10g10:."Lambert Conformal Conic": > lambertc.ps
```

Equidistant

Yet another basemap of width 6 inch for middle Europe may be created by

```
gmt psbasemap -R0/90/25/55 -JD45/20/32/45/6i -B10g10:."Equidistant conic": > econic.ps
```

Polyconic

A basemap for north America may be created by

```
gmt psbasemap -R-180/-20/0/90 -JPoly/4i -B30g10/10g10:."Polyconic": > polyconic.ps
```

1.61.9 Azimuthal Map Projections

Lambert [equal-area]

A 15-cm-wide global view of the world from the vantage point -80/-30 will give the following basemap:

```
gmt psbasemap -Rg -JA-80/-30/15c -B30g30/15g15:."Lambert Azimuthal": > lamberta.ps
```

Follow the instructions for stereographic projection if you want to impose rectangular boundaries on the azimuthal equal-area map but substitute **-Ja** for **-Js**.

Equidistant

A 15-cm-wide global map in which distances from the center (here 125/10) to any point is true can be obtained by:

```
gmt psbasemap -Rg -JE125/10/15c -B30g30/15g15:."Equidistant": > equi.ps
```

Gnomonic

A view of the world from the vantage point -100/40 out to a horizon of 60 degrees from the center can be made using the Gnomonic projection:

```
gmt psbasemap -Rg -JF-100/40/60/6i -B30g30/15g15:."Gnomonic": > gnomonic.ps
```

Orthographic

A global perspective (from infinite distance) view of the world from the vantage point 125/10 will give the following 6-inch-wide basemap:

```
gmt psbasemap -Rg -JG125/10/6i -B30g30/15g15:."Orthographic": > ortho.ps
```

General Perspective

The **-JG** option can be used in a more generalized form, specifying altitude above the surface, width and height of the view point, and twist and tilt. A view from 160 km above -74/41.5 with a tilt of 55 and azimuth of 210 degrees, and limiting the viewpoint to 30 degrees width and height will product a 6-inch-wide basemap:

```
gmt psbasemap -Rg -JG-74/41.5/160/210/55/30/30/6i -B5g1/5g1:."General Perspective": > genper.ps
```

Stereographic [conformal]

To make a polar stereographic projection basemap with radius = 12 cm to -60 degree latitude, with plot title "Salinity measurements", using 5 degrees annotation/tick interval and 1 degree gridlines, run

```
gmt psbasemap -R-45/45/-90/-60 -Js0/-90/12c/-60 -B5g5:."Salinity measurements": > stereo1.ps
```

To make a 12-cm-wide stereographic basemap for Australia from an arbitrary view point (not the poles), and use a rectangular boundary, we must give the pole for the new projection and use the **-R** option to indicate the lower left and upper right corners (in lon/lat) that will define our rectangle. We choose a pole at 130/-30 and use 100/-45 and 160/-5 as our corners. The command becomes

```
gmt psbasemap -R100/-45/160/-5r -JS130/-30/12c -B30g30/15g15:."General Stereographic View": > stereo2.ps
```

1.61.10 Miscellaneous Map Projections

Hammer [equal-area]

The Hammer projection is mostly used for global maps and thus the spherical form is used. To get a world map centered on Greenwich at a scale of 1:200000000, use

```
gmt psbasemap -Rd -Jh0/1:200000000 -B30g30/15g15:."Hammer": > hammer.ps
```

Sinusoidal [equal-area]

To make a sinusoidal world map centered on Greenwich, with a scale along the equator of 0.02 inch/degree, use

```
gmt psbasemap -Rd -Ji0/0.02i -B30g30/15g15:."Sinusoidal": > sinus1.ps
```

To make an interrupted sinusoidal world map with breaks at 160W, 20W, and 60E, with a scale along the equator of 0.02 inch/degree, run the following sequence of commands:

```
gmt psbasemap -R-160/-20/-90/90 -Ji-90/0.02i -B30g30/15g15Wesn -K > sinus_i.ps
gmt psbasemap -R-20/60/-90/90 -Ji20/0.02i -B30g30/15g15Wesn -O -K -X2.8i >> sinus_i.ps
gmt psbasemap -R60/200/-90/90 -Ji130/0.02i -B30g30/15g15wEsn -O -X1.6i >> sinus_i.ps
```

Eckert IV [equal-area]

Pseudo-cylindrical projection typically used for global maps only. Set the central longitude and scale, e.g.,

```
gmt psbasemap -Rg -Jkf180/0.064c -B30g30/15g15:."Eckert IV": > eckert4.ps
```

Eckert VI [equal-area]

Another pseudo-cylindrical projection typically used for global maps only. Set the central longitude and scale, e.g.,

```
gmt psbasemap -Rg -Jks180/0.064c -B30g30/15g15:."Eckert VI": > eckert6.ps
```

Robinson

Projection designed to make global maps “look right”. Set the central longitude and width, e.g.,

```
gmt psbasemap -Rd -JN0/8i -B30g30/15g15:.Robinson: > robinson.ps
```

Winkel Tripel

Yet another projection typically used for global maps only. You can set the central longitude, e.g.,

```
gmt psbasemap -R90/450/-90/90 -JR270/25c -B30g30/15g15:."Winkel Tripel": > winkel.ps
```

Mollweide [equal-area]

The Mollweide projection is also mostly used for global maps and thus the spherical form is used. To get a 25-cm-wide world map centered on the Dateline:

```
psbasemap -Rg -JW180/25c -B30g30/15g15:.Mollweide: > mollweide.ps
```

Van der Grinten

The Van der Grinten projection is also mostly used for global maps and thus the spherical form is used. To get a 7-inch-wide world map centered on the Dateline:

```
gmt psbasemap -Rg -JV180/7i -B30g30/15g15:."Van der Grinten": > grinten.ps
```

1.61.11 CUSTOM LABELS OR INTERVALS

The **-B** option sets up a regular annotation interval and the annotations derive from the corresponding *x*, *y*, or *z* coordinates. However, some applications requires special control on which annotations to plot and even replace the annotation with other labels. This is achieved by using *intfile* in the **-B** option, where *intfile* contains all the information about annotations, ticks, and even gridlines. Each record is of the form *coord type [label]*, where *coord* is the coordinate for this annotation (or tick or gridline), *type* is one or more letters from **a** (annotation), **i** interval annotation, **f** tickamrk, and **g** gridline. Note that **a** and **i** are mutually exclusive and cannot both appear in the same *intfile*. Both **a** and **i** requires you to supply a *label* which is used as the plot annotation. If not given then a regular formatted annotation based on the coordinate will occur.

1.61.12 Restrictions

For some projections, a spherical earth is implicitly assumed. A warning will notify the user if **-V** is set.

1.61.13 Bugs

The **-B** option is somewhat complicated to explain and comprehend. However, it is fairly simple for most applications (see examples).

1.61.14 See Also

gmt, gmt.conf, gmtcolors

1.62 psclip

psclip - Initialize or terminate polygonal clip paths

1.62.1 Synopsis

```
psclip [ table ] -Jparameters -Rwest/east/south/north[/zmin/zmax][r] [ -A[m|p] ] [ -B[pls]parameters ]
-Jz/Zparameters ] [ -K ] [ -N ] [ -O ] [ -P ] [ -T ] [ -U[just/dx/dy/]clabel] ] [ -V[level] ]
[ -X[alc|fr][x-shift]u] ] [ -Y[alc|fr][y-shift]u] ] [ -bi[ncols][type][w][+L|+B] ] [ -ccopies ] [
-f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+|-]gapu] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -
icols[I][sscale][offset][,...] ] [ -p[x|y|z]azim|elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] ] [ -t[transp] ] [
-:[ilo] ]
```

```
psclip -C[n] [ -K ] [ -O ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.62.2 Description

psclip reads (x,y) file(s) [or standard input] and draws polygons that are activated as clipping paths. Several files may be read to create complex paths consisting of several non-connecting segments. Only marks that are subsequently drawn inside the clipping path will be shown. To determine what is inside or outside the clipping path, **psclip** uses the even-odd rule. When a ray drawn from any point, regardless of direction, crosses the clipping path segments an odd number of times, the point is inside the clipping path. If the number is even, the point is outside. The **-N** option, reverses the sense of what is the inside and outside of the paths by plotting a clipping path along the map boundary. After subsequent plotting, which will be clipped against these paths, the clipping may be deactivated by running **psclip** a second time with the **-C** option only.

1.62.3 Required Arguments

-C[*n*] Mark end of existing clip path(s). No input file will be processed. No projection information is needed unless **-B** has been selected as well. With no arguments we terminate all active clipping paths. Experts may restrict the termination to just *n* of the active clipping path by passing that as the argument. Remember to supply **-X** and **-Y** settings if you have moved since the clip started.

-Jparameters (*more ...*) Select map projection.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax***r**] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax*. (*more ...*)

1.62.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

- A**[*mlp*] By default line segments are connected as great circle arcs. To connect them as straight lines, use the **-A** flag. Alternatively, add **m** to connect the line by first following a meridian, then a parallel. Or append **p** to start following a parallel, then a meridian. (This can be practical to connect lines along parallels, for example).
- B**[*pls*]*parameters (more ...)* Set map boundary intervals.
- Jz**/**Z***parameters (more ...)* Set z-axis scaling; same syntax as **-Jx**.
- K** (*more ...*) Do not finalize the PostScript plot.
- N** Invert the sense of what is inside and outside. For example, when using a single path, this means that only points outside that path will be shown. Cannot be used together with **-B**.
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- T** Rather than read any input files, simply turn on clipping for the current map region. Basically, **-T** is a convenient way to run **psclip** with the arguments **-N /dev/null** (or, under Windows, **-N NUL**). Cannot be used together with **-B**.
- U**[*just/dx/dy/*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- X**[*alclflr*][*x-shift*][*u*]
- Y**[*alclflr*][*y-shift*][*u*] (*more ...*) Shift plot origin.
- bi**[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- ccopies** (*more ...*) Specify number of plot copies [Default is 1].
- f**[*ilo*]*colinfo (more ...)* Specify data types of input and/or output columns.
- g**[*a*]*x|y|d|X|Y|D*][*col*]*z*[+|-]*gap*[*u*] (*more ...*) Determine data gaps and line breaks.
- h**[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- icols**[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).
- p**[*x|y|z*]*azim/elev*[/*zlevel*][+*wlon0/lat0*[/*z0*]][+*vx0/y0*] (*more ...*) Select perspective view.
- t**[*transp*] (*more ...*) Set PDF transparency level in percent.
- :**[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.62.5 Examples

To make an overlay PostScript file that will set up a complex clip area to which subsequent plotting will be confined, run:

```
gmt psclip my_region.xy -R0/40/0/40 -Jm0.3i -O -K > clip_mask_on.ps
```

To deactivate the clipping in an existing plotfile, run:

```
gmt psclip -C -O >> complex_plot.ps
```

1.62.6 See Also

gmt, *grdmask*, *psbasemap*, *psmask*

1.63 pscoast

pscoast - Plot continents, shorelines, rivers, and borders on maps

1.63.1 Synopsis

```
pscoast -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] [ -Amin_area[/min_level/max_level][+as][+rll][ppercent]
] [ -B[pls]parameters ] [ -C[llr/fill ] [ -Dresolution[+] ] [ -
Fcode1,code2,...[+llL][+gfill][+ppen][+r]R[incs]] [ -Gfillc ] [ -Iriver[pen] ] [ -Jz/Zparameters ]
[ -K ] [ -L[f][x]lon0/lat0/slon]/slat/length[elfk]M/lnu][+label][+jjust][+ppen][+gfill][+u] ] [ -M ] [
-Nborder[pen] ] [ -O ] [ -P ] [ -Q ] [ -Sfillc ] [ -T[flm][x]lon0/lat0/size[/info]:w,e,s,n][+gint[mint]] ]
[ -U[just/dx/dy][clabel] ] [ -V[level] ] [ -W[level]pen] ] [ -X[alcflr][x-shift[u]] ] [ -Y[alcflr][y-shift[u]]
] [ -bo[ncols][type][w][+L+B] ] [ -ccopies ] [ -p[x/y/z]azim/elev[/zlevel][+wlon0/lat0]/z0][+vx0/y0] ] [
-t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.63.2 Description

pscoast plots grayshaded, colored, or textured land-masses [or water-masses] on maps and [optionally] draws coastlines, rivers, and political boundaries. Alternatively, it can (1) issue clip paths that will contain all land or all water areas, or (2) dump the data to an ASCII table. The data files come in 5 different resolutions: (**f**)ull, (**h**)igh, (**i**)ntermediate, (**l**)ow, and (**c**)rude. The full resolution files amount to more than 55 Mb of data and provide great detail; for maps of larger geographical extent it is more economical to use one of the other resolutions. If the user selects to paint the land-areas and does not specify fill of water-areas then the latter will be transparent (i.e., earlier graphics drawn in those areas will not be overwritten). Likewise, if the water-areas are painted and no land fill is set then the land-areas will be transparent. A map projection must be supplied. The PostScript code is written to standard output.

1.63.3 Required Arguments

-**J***parameters* (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

For perspective view **p**, optionally append *zmin/zmax*. (*more ...*)

1.63.4 Optional Arguments

-Amin_area[*min_level*/*max_level*][**+as**][**+r**][**+ppercent**] Features with an area smaller than *min_area* in km² or of hierarchical level that is lower than *min_level* or higher than *max_level* will not be plotted [Default is 0/0/4 (all features)]. Level 2 (lakes) contains regular lakes and wide river bodies which we normally include as lakes; append **+r** to just get river-lakes or **+l** to just get regular lakes. For expert users who wish to print their own Antarctica coastline and islands via *psxy* you can use **+as** to skip all GSHHG features below 60S. Finally, append **+ppercent** to exclude polygons whose percentage area of the corresponding full-resolution feature is less than *percent*. See GSHHG INFORMATION below for more details.

-B[**pls**]*parameters* (*more ...*) Set map boundary intervals.

-C[**lr**]/*fill* Set the shade, color, or pattern for lakes and river-lakes [Default is the fill chosen for “wet” areas (**-S**)]. Optionally, specify separate fills by prepending **l** for lakes and **r** for river-lakes, repeating the **-C** option as needed.

-Dresolution[**+**] Selects the resolution of the data set to use ((**f**ull, (**h**igh, (**i**ntermediate, (**l**ow, and (**c**ruce). The resolution drops off by 80% between data sets [Default is **I**]. Append **+** to automatically select a lower resolution should the one requested not be available [abort if not found].

-Fcode1,code2,...[**+l**][**+gfill**][**+ppen**][**+r**][**R**][*incs*] Select painting or dumping country polygons from the Digital Chart of the World. This is another dataset independent of GSHHG and hence the **-A** and **-D** options do not apply. Append one or more comma-separated countries using the 2-character ISO 3166-1 alpha-2 convention. To select a state of a country (if available), append .state, e.g. US.TX for Texas. To specify a whole continent, prepend = to any of the continent codes AF (Africa), AN (Antarctica), AS (Asia), EU (Europe), OC (Oceania), NA (North America), or SA (South America). Append **+l** to just list the countries and their codes [no data extraction or plotting takes place]. Use **+L** to see states/territories for Argentina, Australia, Brazil, Canada, and the US. Use **+r** to obtain the bounding box coordinates from the polygon(s). Append *inc*, *xinc*/*yinc*, or *wincl*/*incls*/*ninc* to adjust the region to be a multiple of these steps [no adjustment]. Use **+R** to extend the region outward by adding these increments instead [no extension]. Append **+ppen** to draw polygon outlines [no outline] and **+gfill** to fill them [no fill]. One of **+plg** must be specified unless **+r**, **+R**, or **-M** is in effect, and only one **-F** option can be given. To plot multiple countries in different colors, call *pscoast* multiple times. If modifiers **+r** or **+R** are used and neither **-J** nor **-M** is set then we just print the **-Rwesn** string.

-Gfillc Select filling or clipping of “dry” areas. Append the shade, color, or pattern; or use **-Gc** for clipping [Default is no fill].

-Iriver[/pen] Draw rivers. Specify the type of rivers and [optionally] append pen attributes [Default pen: width = default, color = black, style = solid].

Choose from the list of river types below; repeat option **-I** as often as necessary.

0 = Double-lined rivers (river-lakes)

1 = Permanent major rivers

2 = Additional major rivers

3 = Additional rivers

4 = Minor rivers

5 = Intermittent rivers - major

6 = Intermittent rivers - additional

7 = Intermittent rivers - minor

8 = Major canals

9 = Minor canals

10 = Irrigation canals

You can also choose from several preconfigured river groups:

a = All rivers and canals (0-10)

A = All rivers and canals except river-lakes (1-10)

r = All permanent rivers (0-4)

R = All permanent rivers except river-lakes (1-4)

i = All intermittent rivers (5-7)

c = All canals (8-10)

-Jz/Zparameters (more ...) Set z-axis scaling; same syntax as **-Jx**.

-K (more ...) Do not finalize the PostScript plot.

-L[f][x]lon0/lat0[/slon]/slat/length[eflklMlnlu][+label][+jjust][+ppen][+gfill][+u] Draws a simple map scale centered on *lon0/lat0*. Use **-Lx** to specify x/y position instead. Scale is calculated at latitude *slat* (optionally supply longitude *slon* for oblique projections [Default is central meridian]), *length* is in km, or append unit from **eflklMlnlu**. Use **-Lf** to get a “fancy” scale [Default is plain]. Append **+l** to select the default label which equals the distance unit (meter, foot, km, mile, nautical mile, US survey foot) and is justified on top of the scale [t]. Change this by giving your own label (append **+label**). Change label justification with **+jjustification** (choose among l(ef), r(ight), t(op), and b(ottom)). Apply **+u** to append the unit to all distance annotations along the scale. If you want to place a rectangle behind the scale, specify suitable **+ppen** and/or **+gfill** parameters. Note: Use *FONT_LABEL* to change the label font and *FONT_ANNOT_PRIMARY* to change the annotation font. The height of the map scale is controlled by *MAP_SCALE_HEIGHT*.

-M Dumps a single multisegment ASCII (or binary, see **-bo**) file to standard output. No plotting occurs. Specify one of **-F**, **-I**, **-N** or **-W**. Note: if **-M** is used with **-F** then **-R** or the **+r** modifier to **-F** are not required as we automatically determine the region given the selected geographic entities.

-Nborder[/pen] Draw political boundaries. Specify the type of boundary and [optionally] append pen attributes [Default pen: width = default, color = black, style = solid].

Choose from the list of boundaries below. Repeat option **-N** as often as necessary.

1 = National boundaries

2 = State boundaries within the Americas

3 = Marine boundaries

a = All boundaries (1-3)

-O (*more ...*) Append to existing PostScript plot.

-P (*more ...*) Select “Portrait” plot orientation.

-Q Mark end of existing clip path. No projection information is needed. Also supply **-X** and **-Y** settings if you have moved since the clip started.

-Sfillc Select filling or clipping of “wet” areas. Append the shade, color, or pattern; or use **-Sc** for clipping [Default is no fill].

-T[flm][x]lon0/lat0/size[/info][:w,e,s,n:][+gint[/mint]] Draws a simple map directional rose centered on *lon0/lat0*. Use **-Tx** to specify x/y position instead. The *size* is the diameter of the rose, and optional label information can be specified to override the default values of W, E, S, and N (Give **::** to suppress all labels). The default [plain] map rose only labels north. Use **-Tf** to get a “fancy” rose, and specify in *info* what you want drawn. The default **[1]** draws the two principal E-W, N-S orientations, **2** adds the two intermediate NW-SE and NE-SW orientations, while **3** adds the eight minor orientations WNW-ESE, NNW-SSE, NNE-SSW, and ENE-WSW. For a magnetic compass rose, specify **-Tm**. If given, *info* must be the two parameters *dec/dlabel*, where *dec* is the magnetic declination and *dlabel* is a label for the magnetic compass needle (specify **-** to format a label from *dec*). Then, both directions to geographic and magnetic north are plotted [Default is geographic only]. If the north label is ***** then a north star is plotted instead of the north label. Annotation and two levels of tick intervals for both geographic and magnetic directions are 30/5/1 degrees; override these settings by appending **+gints[/mints]**. Color and pen attributes for the rose are taken from **COLOR_BACKGROUND** and **MAP_TICK_PEN**, respectively, while label fonts, colors and sizes follow **FONT_TITLE** for the four major directions and **FONT_LABEL** for minor directions.

-U[just/dx/dy][clabel] (*more ...*) Draw GMT time stamp logo on plot.

-V[level] (*more ...*) Select verbosity level [c].

-W[level/pen] (*more ...*) Draw shorelines [Default is no shorelines]. Append pen attributes [Defaults: width = default, color = black, style = solid] which apply to all four levels. To set the pen for each level differently, prepend *level*, where *level* is 1-4 and represent coastline, lakeshore, island-in-lake shore, and lake-in-island-in-lake shore. Repeat **-W** as needed. When specific level pens are set, those not listed will not be drawn [Default draws all levels; but see **-A**].

-X[alclfr][x-shift[u]]

-Y[alclfr][y-shift[u]] (*more ...*) Shift plot origin.

-bo[ncols][type] (*more ...*) Select binary output.

-ccopies (*more ...*) Specify number of plot copies [Default is 1].

-p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (*more ...*) Select perspective view.

-t[transp] (*more ...*) Set PDF transparency level in percent.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).

- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.63.5 Examples

To plot a green Africa with white outline on blue background, with permanent major rivers in thick blue pen, additional major rivers in thin blue pen, and national borders as dashed lines on a Mercator map at scale 0.1 inch/degree, use

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -I1/1p,blue -N1/0.25p,- \
-I2/0.25p,blue -W0.25p,white -Ggreen -Sblue -P > africa.ps
```

To plot Iceland using the lava pattern (# 28) at 100 dots per inch, on a Mercator map at scale 1 cm/degree, run

```
gmt pscoast -R-30/-10/60/65 -Jm1c -B5 -Gp100/28 > iceland.ps
```

To initiate a clip path for Africa so that the subsequent colorimage of gridded topography is only seen over land, using a Mercator map at scale 0.1 inch/degree, use

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -Gc -P -K > africa.ps
gmt grdimage -Jm0.1i etopo5.nc -Ccolors.cpt -O -K >> africa.ps
gmt pscoast -Q -O >> africa.ps
```

pscoast will first look for coastline files in directory **\$GMT_SHAREDIR/coast** If the desired file is not found, it will look for the file **\$GMT_SHAREDIR/coastline.conf**. This file may contain any number of records that each holds the full pathname of an alternative directory. Comment lines (#) and blank lines are allowed. The desired file is then sought for in the alternate directories.

1.63.6 Gshhs Information

The coastline database is GSHHG (formerly GSHHS) which is compiled from two sources: World Vector Shorelines (WVS) and CIA World Data Bank II (WDBII). In particular, all level-1 polygons (ocean-land boundary) are derived from the more accurate WVS while all higher level polygons (level 2-4, representing land/lake, lake/island-in-lake, and island-in-lake/lake-in-island-in-lake boundaries) are taken from WDBII. Much processing has taken place to convert WVS and WDBII data into usable form for GMT: assembling closed polygons from line segments, checking for duplicates, and correcting for crossings between polygons. The area of each polygon has been determined so that the user may choose not to draw features smaller than a minimum area (see **-A**); one may also limit the highest hierarchical level of polygons to be included (4 is the maximum). The 4 lower-resolution databases were derived from the full resolution database using the Douglas-Peucker line-simplification algorithm. The classification of rivers and borders follow that of the WDBII. See the GMT Cookbook and Technical Reference Appendix K for further details.

1.63.7 Bugs

The options to fill (**-C -G -S**) may not always work if the Azimuthal equidistant projection is chosen (**-JelE**). If the antipole of the projection is in the oceans it will most likely work. If not, try to avoid using projection center coordinates that are even multiples of the coastline bin size (1, 2, 5, 10, and 20 degrees for **f, h, i, l, c**, respectively). This projection is not supported for clipping.

The political borders are for the most part 1970s-style but have been updated to reflect more recent border rearrangements in Europe and elsewhere. Let us know if you find something out of date.

The full-resolution coastlines are also from a digitizing effort in the 1970-80s and it is difficult to assess the accuracy. Users who zoom in close enough may find that the GSHHG coastline is not matching other data, e.g., satellite images, more recent coastline data, etc. We are aware of such mismatches but cannot undertake band-aid solutions each time this occurs.

Some users of **pscoast** will not be satisfied with what they find for the Antarctic shoreline. In Antarctica, the boundary between ice and ocean varies seasonally and inter-annually. There are some areas of permanent sea ice. In addition to these time-varying ice-ocean boundaries, there are also ice grounding lines where ice goes from floating on the sea to sitting on land, and lines delimiting areas of rock outcrop. For consistency's sake, we have used the World Vector Shoreline throughout the world in **pscoast**, as described in the GMT Cookbook Appendix K. Users who need specific boundaries in Antarctica should get the Antarctic Digital Database, prepared by the British Antarctic Survey, Scott Polar Research Institute, World Conservation Monitoring Centre, under the auspices of the Scientific Committee on Antarctic Research. This data base contains various kinds of limiting lines for Antarctica and is available on CD-ROM. It is published by the Scientific Committee on Antarctic Research, Scott Polar Research Institute, Lensfield Road, Cambridge CB2 1ER, United Kingdom.

1.63.8 See Also

gmt, gmt.conf, gmtcolors, grdlandmask, psbasemap

1.64 pscontour

pscontour - Contour table data by direct triangulation [method]

1.64.1 Synopsis

```
pscontour [ table ] -C[+]cptfile -Jparameters -Rwest/east/south/north[zmin/zmax][r] [ -A[-
[+]annot_int][labelinfo] ] [ -B[pls]parameters ] [ -D[template] ] [ -G[difn||Llx|X]params ]
[ -I ] [ -Jz/Zparameters ] [ -K ] [ -Lpen ] [ -N ] [ -O ] [ -P ] [ -Qindexfile
] [ -S[plt] ] [ -T[+|-][gap/length][:[labels]] ] [ -U[just/dx/dy][clabel] ] [ -V[level] ] [ -
W[+]pen ] [ -X[alcflr][x-shift][u] ] [ -Y[alcflr][y-shift][u] ] [ -b[ilo][ncol][type][w][+L+B]
] [ -copies ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,...] ] [ -
p[xlyz]azim/elev[zlevel][+wlon0/lat0][z0][+vx0/y0] ] [ -t[transp] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.64.2 Description

pscontour reads an ASCII [or binary] xyz-file and produces a raw contour plot by triangulation. By default, the optimal Delaunay triangulation is performed (using either Shewchuk's [1996] or Watson's [1982] method as selected during GMT installation; type **pscontour -** to see which method is selected), but the user may optionally provide a second file with network information, such as a triangular mesh used for finite element modeling. In addition to contours, the area between contours may be painted according to the color palette file. Alternatively, the x/y/z positions of the contour lines may be saved to one or more output files (or stdout) and no plot is produced.

1.64.3 Required Arguments

-C[+]*cont_int* The contours to be drawn may be specified in one of three possible ways:

1. If *cont_int* has the suffix ".cpt" and can be opened as a file, it is assumed to be a color palette table. The color boundaries are then used as contour levels. If the cpt-file has annotation flags in the last column then those contours will be annotated. By default all contours are labeled; use **-A-** to disable all annotations.
2. If *cont_int* is a file but not a cpt-file, it is expected to contain contour levels in column 1 and a C(ontour) OR A(nnotate) in col 2. The levels marked C (or c) are contoured, the levels marked A (or a) are contoured and annotated. Optionally, a third column may be present and contain the fixed annotation angle for this contour level.
3. If no file is found, then *cont_int* is interpreted as a constant contour interval. However, if prepended with the + sign the *cont_int* is taken as meaning draw that single contour. The **-A** option offers the same possibility so they may be used together to plot only one annotated and one non-annotated contour. If **-A** is set and **-C** is not, then the contour interval is set equal to the specified annotation interval.

If a file is given and **-T** is set, then only contours marked with upper case C or A will have tickmarks. In all cases the contour values have the same units as the file.

-Jparameters (*more ...*) Select map projection.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append *lzmin/zmax*. (*more ...*)

1.64.4 Optional Arguments

table One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-A[-|+]*annot_int*[[*labelinfo*] *annot_int* is annotation interval in data units; it is ignored if contour levels are given in a file. [Default is no annotations]. Append **-** to disable all annotations implied by **-C**. Alternatively prepend **+** to the annotation interval to plot that as a single contour. The optional *labelinfo* controls the specifics of the label formatting and consists of a concatenated string made up of any of the following control arguments:

+aangle For annotations at a fixed angle, **+an** for contour-normal, or **+ap** for contour-parallel [Default]. For **+ap**, you may optionally append **u** for up-hill and **d** for down-hill cartographic annotations.

- +cdx[/dy]** Sets the clearance between label and optional text box. Append **clip** to specify the unit or % to indicate a percentage of the label font size [15%].
- +d** Turns on debug which will draw helper points and lines to illustrate the workings of the contour line setup.
- +e** Delay the plotting of the text. This is used to build a clip path based on the text, then lay down other overlays while that clip path is in effect, then turning of clipping with **psclip -Cs** which finally plots the original text.
- +ffont** Sets the desired font [Default **FONT_ANNOT_PRIMARY** with its size changed to 9p].
- +g[*color*]** Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PS_PAGE_COLOR**].
- +jjust** Sets label justification [Default is MC].
- +ndx[/dy]** Nudges the placement of labels by the specified amount (append **clip** to specify the units). Increments are considered in the coordinate system defined by the orientation of the contour; use **+N** to force increments in the plot x/y coordinates system [no nudging]. Not allowed with **+v**.
- +o** Selects rounded rectangular text box [Default is rectangular]. Not applicable for curved text (**+v**) and only makes sense for opaque text boxes.
- +p[*pen*]** Draws the outline of text boxes [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, style = solid].
- +rmin_rad** Will not place labels where the contours's radius of curvature is less than *min_rad* [Default is 0].
- +t[*file*]** Saves contour label x, y, and text to *file* [Contour_labels.txt]. Use **+T** to save x, y, angle, text instead.
- +uunit** Appends *unit* to all contour labels. [Default is no unit]. If **z** is appended we use the z-unit from the grdf file.
- +v** Specifies curved labels following the contour [Default is straight labels].
- +w** Specifies how many (x,y) points will be used to estimate label angles [Default is 10].
- +=*prefix*** Prepends *prefix* to all contour labels. [Default is no prefix].

-B[*pls*]*parameters (more ...)* Set map boundary intervals.

-D[*template*]

Dump the (x,y,z) coordinates of each contour to one or more output files (or *stdout* if *template* is not given). No plotting will take place. If *template* contains one or more of the C-format specifiers %d, %f, %c then line segments will be written to different files; otherwise all lines are written to the specified file (*template*). The use of the C-format specifiers controls how many files are created and how the contours are organized. If the float format %f is present (standard modifications to width and precision are allowed, e.g., %f7.3f), then the filenames will contain the contour value and lines are thus separated into files based on a common contour value. If the integer format %d is present (including modifications like %05d), then all contours are written to individual segment files; if any of the other specifiers are present they just affect the file names. Finally, if the character format %c is present it is replaced with the letters C (for closed) or O (for open), reflecting the nature of each contour. Any combination of one, two, or all three modifiers are valid, resulting in different filenames and number of files. For instance, if %c appears by itself, then only two files are created, separating the open from the closed contours

(assuming both kinds are present). If just %f is used, then all segments for the same contour level will be written to the same file, resulting in N multi-segment files. If both %f and %c were combined then each contour level would be further subdivided into closed and open contours. Any combination involving %d will result in one individual file for each segment; %c, %f only modifies the file names. The files are ASCII unless **-bo** is used.

-G

The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

ddist[clip] or Ddist[dle|fk|ml|Minls] For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **f** (foot), **k** (km), **M** (mile), **n** (nautical mile) or **u** (US survey foot), and **d** (arc degree), **m** (arc minute), or **s** (arc second). [Default is 10**c** or 4**i**]. As an option, you can append */fraction* which is used to place the very first label for each contour when the cumulative along-contour distance equals *fraction * dist* [0.25].

ffile.d Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact matches and points outside the region are skipped.

llLine1[,line2,...] Give *start* and *stop* coordinates for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the quoted lines. The format of each *line* specification is *start/stop*, where *start* and *stop* are either a specified point *lon/lat* or a 2-character **XY** key that uses the justification format employed in **pstext** to indicate a point on the map, given as [LCR][BMT]. In addition, you can use **Z-**, **Z+** to mean the global minimum and maximum locations in the grid. **L** will interpret the point pairs as defining great circles [Default is straight line].

nn_label Specifies the number of equidistant labels for quoted lines line [1]. Upper case **N** starts labeling exactly at the start of the line [Default centers them along the line]. **N-1** places one justified label at start, while **N+1** places one justified label at the end of quoted lines. Optionally, append */min_dist[clip]* to enforce that a minimum distance separation between successive labels is enforced.

x|Xxfile.d Reads the multisegment file *xfile.d* and places labels at the intersections between the quoted lines and the lines in *xfile.d*. **X** will resample the lines first along great-circle arcs.

In addition, you may optionally append **+rradius[clip]** to set a minimum label separation in the x-y plane [no limitation].

-I Color the triangles using the color palette table.

-Jz/Zparameters (more ...) Set z-axis scaling; same syntax as **-Jx**.

-K (more ...) Do not finalize the PostScript plot.

-Lpen (more ...) Draw the underlying triangular mesh using the specified pen attributes [Default is no mesh].

-N Do NOT clip contours or image at the boundaries [Default will clip to fit inside region **-R**].

-O (more ...) Append to existing PostScript plot.

-P (more ...) Select "Portrait" plot orientation.

- Qindexfile** Give name of file with network information. Each record must contain triplets of node numbers for a triangle [Default computes these using Delaunay triangulation (see **triangulate**)].
- S[plt]** Skip all input *xyz* points that fall outside the region [Default uses all the data in the triangulation]. Alternatively, use **-St** to skip triangles whose three vertices are all outside the region. **-S** with no modifier is interpreted as **-Sp**.
- T[+]-[gap/length][:[labels]]** Will draw tickmarks pointing in the downward direction every *gap* along the innermost closed contours. Append *gap* and tickmark length (append units as **c**, **i**, or **p**) or use defaults [15**p**/3**p**]. User may choose to tick only local highs or local lows by specifying **-T+** or **-T-**, respectively. Append *labels* to annotate the centers of closed innermost contours (i.e, the local lows and highs). If no *labels* is appended we use - and + as the labels. Appending two characters, **:LH**, will plot the two characters (here, L and H) as labels. For more elaborate labels, separate the two label strings by a comma (e.g., **:lo,hi**). If a file is given by **-C** and **-T** is set, then only contours marked with upper case C or A will have tickmarks [and annotation].
- U[just/dx/dy/][clabel] (*more ...*)** Draw GMT time stamp logo on plot.
- V[level] (*more ...*)** Select verbosity level [*c*].
- W[+]**pen** (*more ...*)** Select contouring and set contour pen attributes. If the **+** flag is prepended then the color of the contour lines are taken from the *cpt* file (see **-C**). If the **-** flag is prepended then the color from the *cpt* file is applied both to the contours and the contour annotations.
- X[alclflr][x-shift[*u*]]**
- Y[alclflr][y-shift[*u*]] (*more ...*)** Shift plot origin.
- bi[ncols][type] (*more ...*)** Select binary input. [Default is 3 input columns]. Use 4-byte integer triplets for node ids (**-Q**).
- bo[ncols][type] (*more ...*)** Select binary output. [Default is 3 output columns].
- ccopies (*more ...*)** Specify number of plot copies [Default is 1].
- h[ilo][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*)** Skip or produce header record(s).
- icols[1][sscale][offset][,...] (*more ...*)** Select input columns (0 is first column).
- :**[ilo] (*more ...*)** Swap 1st and 2nd column on input and/or output.
- p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (*more ...*)** Select perspective view.
- t[transp] (*more ...*)** Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.64.5 Examples

To make a raw contour plot from the file `topo.xyz` and drawing the contours (pen = 2) given in the color palette file `topo.cpt` on a Lambert map at 0.5 inch/degree along the standard parallels 18 and 24, use

```
gmt pscontour topo.xyz -R320/330/20/30 -Jl18/24/0.5i -Ctopo.cpt -W0.5p > topo.ps
```

To create a color PostScript plot of the numerical temperature solution obtained on a triangular mesh whose node coordinates and temperatures are stored in `temp.xyz` and mesh arrangement is given by the file `mesh.ijk`, using the colors in `temp.cpt`, run

```
gmt pscontour temp.xyz -R0/150/0/100 -Jx0.1i -Ctemp.cpt -G -W0.25p > temp.ps
```

To save the triangulated 100-m contour lines in `topo.txt` and separate them into multisegment files (one for each contour level), try

```
gmt pscontour topo.txt -C100 -Dcontours_%.0f.txt
```

1.64.6 See Also

gmt, *gmt.conf*, *gmtcolors*, *grdcontour*, *grdimage*, *nearneighbor*, *psbasemap*, *psscale*, *surface*, *triangulate*

1.64.7 References

Watson, D. F., 1982, Acord: Automatic contouring of raw data, *Comp. & Geosci.*, **8**, 97-101.

Shewchuk, J. R., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, First Workshop on Applied Computational Geometry (Philadelphia, PA), 124-133, ACM, May 1996.

<http://www.cs.cmu.edu/~quake/triangle.html>

1.65 pshistogram

pshistogram - Calculate and plot histograms

1.65.1 Synopsis

```
pshistogram [ table ] -Jx|Xparameters -Wbin_width [ -A ] [ -B[pls]parameters ] [ -Ccptfile ] [ -D[+b][+font][+off][+r] ] [ -F ] [ -Gfill ] [ -Jz|Zparameters ] [ -I[olO] ] [ -K ] [ -Lpen ] [ -N[mode][+ppen] ] [ -O ] [ -P ] [ -Q ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ -U[just/dx/dy][cllabel] ] [ -V[level] ] [ -X[alclflr][x-shift[u]] ] [ -Y[alclflr][y-shift[u]] ] [ -Ztype ] [ -bi[ncols][type][w][+L+B] ] [ -ccopies ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][ooffset][,...] ] [ -p[x|y|z]azim/lev/zlevel [ +wlon0/lat0[z0] ] [ +vx0/y0 ] ] [ -ttransp ] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.65.2 Description

pshistogram reads *file* [or standard input] and examines the first data column (or use **-i**) to calculate histogram parameters based on the bin-width provided. Using these parameters, scaling, and optional range parameters it will generate PostScript code that plots a histogram. A cumulative histogram may also be specified.

1.65.3 Required Arguments

- Jx** *xscale* [*/yscale*] (Linear scale(s) in distance unit/data unit).
- W***bin_width* Sets the bin width used for histogram calculations.

1.65.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

- A** Plot the histogram horizontally from $x = 0$ [Default is vertically from $y = 0$].
- B**[*pls*]*parameters* (*more ...*) Set map boundary intervals.
- C***cptfile* Give a color palette file. The mid x -value for each bar is used to look-up the bar color.
- D**[**+b**][**+ffont**][**+ooff**][**+r**] Annotate each bar with the count it represents. Append any of the following modifiers: Use **+b** to place the labels beneath the bars instead of above; use **+ffont** to change to another font than the default annotation font; use **+o** to change the offset between bar and label [6p]; use **+r** to rotate the labels from horizontal to vertical.
- F** Center bin on each value. [Default is left edge].
- G***fill* Select filling of bars [Default is no fill].
- I**[**o**]**O** Inquire about min/max x and y after binning. The *xmin xmax ymin ymax* is output; no plotting is done. Append **o** to output an ASCII table of the resulting x,y data instead. Upper case **O** will output all x,y bin data even when $y == 0$.
- Jz**/**Z***parameters* (*more ...*) Set z -axis scaling; same syntax as **-Jx**.
- K** (*more ...*) Do not finalize the PostScript plot.
- L***pen* Draw bar outline using the specified pen thickness. [Default is no outline].
- N**[*mode*][**+ppen**] Draw the equivalent normal distribution; append desired pen [0.5p,black]. The *mode* selects which central location and scale to use:
 - 0 = mean and standard deviation [Default];
 - 1 = median and L1 scale;
 - 2 = LMS mode and scale.The **-N** option may be repeated to draw several of these curves.
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- Q** Draw a cumulative histogram.

-R[*unit*]*xmin*/*xmax*/*ymin*/*ymax*[*r*] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax*. (*more ...*) If not given, **pshistogram** will automatically find reasonable values for the region.

-S Draws a stairs-step diagram which does not include the internal bars of the default histogram.

-U[*just/dx/dy*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-X[*alcflr*][*x-shift*[*u*]]

-Y[*alcflr*][*y-shift*[*u*]] (*more ...*) Shift plot origin.

-Ztype Choose between 6 types of histograms:

- 0 = counts [Default]
- 1 = frequency_percent
- 2 = log (1.0 + count)
- 3 = log (1.0 + frequency_percent)
- 4 = log10 (1.0 + count)
- 5 = log10 (1.0 + frequency_percent).

-bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].

-ccopies (*more ...*) Specify number of plot copies [Default is 1].

-f[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.

-h[*ilo*][*n*][*+c*][*+d*][*+rremark*][*+rtitle*] (*more ...*) Skip or produce header record(s).

-i[*cols*][*l*][*sscale*][*offset*][*,...*] (*more ...*) Select input columns (0 is first column).

-p[*x|y|z*]*azim*/*elev*[*/zlevel*][*+wlon0*/*lat0*[*/z0*]][*+vx0*/*y0*] (*more ...*) Select perspective view.

-t[*transp*] (*more ...*) Set PDF transparency level in percent.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.65.5 Examples

To draw a histogram of the data v3206.t containing seafloor depths, using a 250 meter bin width, center bars, and draw bar outline, use:

```
gmt pshistogram v3206.t -JXh -W250 -F -LP0.5p -V > plot.ps
```

If you know the distribution of your data, you may explicitly specify range and scales. E.g., to plot a histogram of the y-values (2nd column) in the file `errors.xy` using a 1 meter bin width, plot from -10 to +10 meters @ 0.75 cm/m, annotate every 2 m and 100 counts, and use black bars, run:

```
gmt pshistogram errors.xy -W1 -R-10/10/0/0 -Jxc/0.01c \  
-B2:Error:/100:Counts: -Gblack -il -V > plot.ps
```

Since no y-range was specified, `pshistogram` will calculate `ymin` in even increments of 100.

1.65.6 Bugs

The **-W** option does not yet work properly with time series data (e.g., **-f0T**). Thus, such variable intervals as months and years are not calculated. Instead, specify your interval in the same units as the current setting of `TIME_UNIT`.

1.65.7 See Also

gmt, gmtcolors, psbasemap, psrose, psxy

1.66 psimage

`psimage` - Place images or EPS files on maps

1.66.1 Synopsis

```
psimage imagefile [ -W[-]width[/height] | -Edpi ] [ -Cxpos/ypos[/justify] ] [ -Fpen ] [ -G[b|f|t]color  
] [ -I ] [ -Jparameters ] [ -Jz|Zparameters ] [ -K ] [ -M ] [ -Nnx[/ny] ] [ -O ] [ -P ] [ -R  
west/east/south/north[/zmin/zmax][r] ] [ -U[/just/dx/dy][c|label] ] [ -V[/level] ] [ -X[a|c|f|r][x-shift[u]]  
] [ -Y[a|c|f|r][y-shift[u]] ] [ -ccopies ] [ -p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] ] [ -t  
t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.66.2 Description

psimage reads an Encapsulated PostScript file or a raster image file and plots it on a map. The image can be scaled arbitrarily, and 1-bit raster images can be (1) inverted, i.e., black pixels (on) becomes white (off) and vice versa, or (2) colorized, by assigning different foreground and background colors, and (3) made transparent where one of back- or foreground is painted only. As an option, the user may choose to convert colored raster images to grayscale using TV's YIQ-transformation. For raster files, the user can select which color to be made transparent. The user may also choose to replicate the image which, when preceded by appropriate clip paths, may allow larger custom-designed fill patterns to be implemented (the **-Gp** mechanism offered in most GMT programs is limited to rasters smaller than 146 by 146).

1.66.3 Required Arguments

imagefile This must be an Encapsulated PostScript (EPS) file or a raster image. An EPS file must contain an appropriate BoundingBox. A raster file can have a depth of 1, 8, 24, or 32 bits and is read

via GDAL. If GDAL was not configured during GMT installation then only Sun raster files are supported natively.

- Edpi** Sets the dpi of the image in dots per inch, or use **-W**.
- W[-]width[/height]** Sets the width (and height) of the image in plot coordinates (inches, cm, etc.). If *height* is not given, the original aspect ratio of the image is maintained. If *width* is negative we use the absolute value and interpolate image to the device resolution using the PostScript image operator. Alternatively, use **-E**.

1.66.4 Optional Arguments

- Cxpos/ypos[/justify]** Sets position of the image in plot coordinates (inches, cm, etc.) from the current origin of the plot. By default, this defines the position of the lower left corner of the image, but this can be changed by specifying justification [0/0/BL].
- Epen** Draws a rectangular frame around the image with the given pen [no frame].
- Jparameters (more ...)** Select map projection.
- Jz/Zparameters (more ...)** Set z-axis scaling; same syntax as **-Jx**.
- K (more ...)** Do not finalize the PostScript plot.
- M** Convert color image to monochrome grayshades using the (television) YIQ-transformation.
- Nnx[/ny]** Replicate the image *nx* times horizontally and *ny* times vertically. If *ny* is omitted, it will be identical to *nx* [Default is 1/1].
- O (more ...)** Append to existing PostScript plot.
- P (more ...)** Select “Portrait” plot orientation.
- R[unit]xmin/xmax/ymin/ymax[r] (more ...)** Specify the region of interest. (Used only with **-p**)
For perspective view **p**, optionally append */zmin/zmax. (more ...)*
- U[just/dx/dy]/[clabel] (more ...)** Draw GMT time stamp logo on plot.
- V[level] (more ...)** Select verbosity level [c].
- X[alclflr][x-shift[u]]**
- Y[alclflr][y-shift[u]] (more ...)** Shift plot origin.
- ccopies (more ...)** Specify number of plot copies [Default is 1].

These options are for 1-bit images only. They have no effect when plotting other images or PostScript files.

-G[bflft]color

- Gb** Sets background color (replace white pixel) of 1-bit images. Use **-** for transparency (and set **-Gf** to the desired color).
- Gf** Sets foreground color (replace black pixel) of 1-bit images. Use **-** for transparency (and set **-Gb** to the desired color).
- I** Invert 1-bit image before plotting. This is what is done when you use **-GP** in other GMT programs.

These options are for 8-, 24-, and 32-bit raster images only. They have no effect when plotting 1-bit images or PostScript files.

-Gt Assigns the color that is to be made transparent. Sun Raster files do not support transparency, so indicate here which color to be made transparent.

-p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (*more ...*) Select perspective view. (Requires **-R** and **-J** for proper functioning).

-t[transp] (*more ...*) Set PDF transparency level in percent.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.66.5 Examples

To plot the image contained in the 8-bit raster file `scanned_face.ras`, scaling it to 8 by 10 cm (thereby possibly changing the aspect ratio), and making the white color transparent, use

```
gmt psimage scanned_face.ras -W8c/10c -Gtwhite > image.ps
```

To plot the image `logo.jpg`, scaling it to be 1 inch wide (height is scaled accordingly), and outline with a thin, blue pen, use

```
gmt psimage logo.jpg -W1i -Fthin,blue > image.ps
```

To include an Encapsulated PostScript file `tiger.eps` with its upper right corner 2 inch to the right and 1 inch up from the current location, and have its width scaled to 3 inches, while keeping the aspect ratio, use

```
gmt psimage tiger.eps -C2i/1i/TR -W3i > image.ps
```

To replicate the 1-bit raster image template `1_bit.ras`, colorize it (brown background and red foreground), and setting each of 5 by 5 tiles to be 1 cm wide, use

```
gmt psimage 1_bit.ras -Gbbrown -Gfred -N5 -W1c > image.ps
```

1.66.6 See Also

gmt, gmtcolors, psxy, convert (1)

1.67 pslegend

pslegend - Plot legends on maps

1.67.1 Synopsis

```
pslegend [ textfile ] -D[x]lon/lat/width[/height]/just[/dx/dy] [ -B[pls]parameters ] [ -Cdx/dy ] [ -F[+gfill][+i][gap]*pen*][+p[pen]][+r[radius]][+s[[dx/dy]/shade]] ] [ -Jparameters ] [ -K ] [ -Lspacing ] [ -O ] [ -P ] [ -Rwest/east/south/north[r] ] [ -U[just/dx/dy]/[clabel] ] [ -V[level] ] [ -X[alcflr][x-shift[u]] ] [ -Y[alcflr][y-shift[u]] ] [ -ccopies ] [ -p[x|y|z]azim/elev[/zlevel]][+wlon0/lat0[/z0]][+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.67.2 Description

pslegend will make legends that can be overlaid on maps. It reads specific legend-related information from an input file [or stdin]. Unless otherwise noted, annotations will be made using the primary annotation font and size in effect (i.e., FONT_ANNOT_PRIMARY)

1.67.3 Required Arguments

-D[*x*]*lon/lat/width*[/*height*]/*just*[/*dx/dy*] Positions the legend and specifies its size. The *just* is a 2-char justification string (see *psfont*) that relates the given position to a point on the rectangular legend box. If you want to specify the position in map plot units (i.e., inches or cm), use **-Dx**; in that case the **-R** and **-J** are optional (provided it is an overlay, i.e., we are using **-O**). Use to optional *dx/dy* to shift the legend by that amount in the direction implied by the justification. If *height* is zero or not given then we estimate *height* based the expected vertical extent of the items to be placed.

1.67.4 Optional Arguments

-B[*pls*]*parameters* (*more ...*) Set map boundary intervals.

-C*dx/dy* Sets the clearance between the legend frame and the internal items [4p/4p].

-F[*+gfill*][*+i*][*gap*]**pen**][*+p*[*pen*]][*+r*[*radius*]][*+s*[[*dx/dy*]/*shade*]] Without further options, draws a rectangular border around the legend using MAP_FRAME_PEN; specify a different pen with *+ppen*. Add *+gfill* to fill the legend box [no fill] Append *+i* to draw a secondary, inner border as well. We use a *gap* between borders of 2p and the MAP_DEFAULTS_PEN unless other values are specified. Append *+r* to draw rounded rectangular borders instead, with a 6p corner radius. You can override this radius by appending another value. Finally, append *+s* to draw an offset background shaded region. Here, *dx/dy* indicates the shift relative to the foreground frame [4p/-4p] and *shade* sets the fill style to use for shading.

-J*parameters* (*more ...*) Select map projection.

-K (*more ...*) Do not finalize the PostScript plot.

-L*spacing* Sets the linespacing factor in units of the current annotation font size [1.1].

-O (*more ...*) Append to existing PostScript plot.

-P (*more ...*) Select "Portrait" plot orientation.

-R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

-U[*just/dx/dy*]/[*clabel*] (*more ...*) Draw GMT time stamp logo on plot.

-V[*level*] (*more ...*) Select verbosity level [c].

- X[*alcflr*][*x-shift*[*u*]]
- Y[*alcflr*][*y-shift*[*u*]] (*more ...*) Shift plot origin.
- ccopies (*more ...*) Specify number of plot copies [Default is 1].
- p[*xlylz*]*azim/lev*[*zlevel*][+*wlon0/lat0*[*z0*]][+*vx0/y0*] (*more ...*) Select perspective view.
- t[*transp*] (*more ...*) Set PDF transparency level in percent.
- ^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

1.67.5 Pslegend Codes

textfile This file contains instruction for the layout of items in the legend. Each legend item is described by a unique record. All records begin with a unique character that is common to all records of the same kind. The order of the legend items is implied by the order of the records. Ten different record types are recognized, and the syntax for each of these records are presented below:

comment Records starting with # and blank lines are skipped.

B *cptname offset height* [*optional arguments*] The **B** record will plot a horizontal color bar, **psscale**-style in the middle, starting at *offset* from the left edge, and of the given *height*. You may add any additional **psscale** options from the list: **-A -B -I -L -M -N -S -Z** and **-p**. See the **psscale** man page for details.

C *textcolor* The **C** record specifies the color with which the remaining text is to be printed. *textcolor* can be in the form *r/g/b*, *c/m/y/k*, or a named color. Use - to reset to default color.

D *offset pen* The **D** record results in a horizontal line with specified *pen* across the legend with one quarter of the line spacing left blank above and below the line. Two gaps of *offset* units are left blank between the horizontal line and the left and right frame sides. If no pen is given we use **MAP_GRID_PEN_PRIMARY**

G *gap* The **G** record specifies a vertical gap of the given length. In addition to the standard units (**i**, **c**, **p**) you may use **I** for lines. A negative *gap* will move the current line upwards (thus closing a gap).

H *fontsize*-*font*-*header*) The **H** record plots a centered text string using the specified font parameters. Use - to default to size and type of **FONT_TITLE**.

I *imagefile width justification* Place an EPS or Sun raster image in the legend justified relative to the current point. The image *width* determines the size of the image on the page.

L *fontsize*-*font*-*justification label*) The **L** record plots a (L)eft, (C)entered, or (R)ight-justified text string within a column using the specified font parameters. Use - to default to the size and type of **FONT_LABEL**.

M slon|- slat length flp [-Rw/e/s/n -Jparam] Place a map scale in the legend. Specify *slon slat*, the point on the map where the scale applies (*slon* is only meaningful for certain oblique projections. If not needed, you must specify - instead), *length*, the length of the scale in km (for other units append **e** (meter), **f** (foot), **M** (mile), **n** (nautical mile), or **u** (survey foot)), and **f** or **p** for fancy or plain scale. If the **-R -J** supplied to **pslegend** is different than the projection needed for the scale (or not given at all, i.e., with **-Dx**), supply the optional **-R -J** settings as well. Append **+l** to the *length* to select the default label which equals the distance unit (meter, feet, km, miles, nautical miles, survey feet) and is justified on top of the scale [t]. Change this by giving your own label (append **+llabel**). Change label justification with **+jjustification** (choose among l(ef), r(igh), t(op), and b(ottom)). Apply **+u** to append the unit to all distance annotations along the scale. If you want to place a rectangle behind the scale, specify suitable **+ppen** and/or **+ffill** parameters. All these **+)modifiers** are appended to *length* to make a single string.

N ncolumns Change the number of columns in the legend [1]. This only affects the printing of symbols (**S**) and labels (**L**). The number of columns stay in effect until **N** is used again.

S dx1 symbol size fill pen [dx2 text] Plots the selected symbol with specified diameter, fill, and outline (see *psxy*). The symbol is centered at *dx1* from the left margin of the column, with the optional explanatory *text* starting *dx2* from the margin, printed with **FONT_ANNOT_PRIMARY**. Use - if no *fill* or outline (*pen*) is required. When plotting just a symbol, without text, *dx2* and *text* can be omitted. Three **psxy** symbols may take special modifiers: front (**f**), quoted line (**q**) and vector (**v**). You can append modifiers to the symbol and affect how the fronts, quoted lines and vectors are presented (see *psxy* man page for modifiers). **pslegend** will determine default settings for all modifiers and secondary arguments if not provided. A few other symbols (the rectangles, ellipse, wedge, mathangle) may take more than a single argument size. Note that for a line segment you should use the horizontal dash symbol (-). If just a single size is given then **pslegend** will provide reasonable arguments to plot the symbol (See **Defaults**). Alternatively, combine the required arguments into a single, comma-separated string and use that as the symbol size (again, see *psxy* for details on the arguments needed).

T paragraph-text One or more of these **T** records with *paragraph-text* printed with **FONT_ANNOT_PRIMARY**. To specify special positioning and typesetting arrangements, or to enter a paragraph break, use the optional **P** record.

V offset pen The **V** record draws a vertical line between columns (if more than one) using the selected *pen offset* is analogous to the offset for the **D** records but in the vertical direction.

P paragraph-mode-header-for-pstext Start a new text paragraph by specifying all the parameters needed (see **pstext -M** record description). Note that **pslegend** knows what all those values should be, so normally you can leave the entire record (after **P**) blank or leave it out all together. If you need to set at least one of the parameters directly, you must specify all and set the ones you want to leave at their default value to -.

1.67.6 Defaults

When attributes are not provided, or extended symbol information (for symbols taking more than just an overall size) are not given as comma-separated quantities, **pslegend** will provide the following defaults:

Front: Front symbol is left-side (here, that means upper side) box, with dimensions 30% of the given symbol size.

Vector: Head size is 30% of given symbol size.

Ellipse: Minor axis is 65% of major axis (the symbol size), with an azimuth of 0 degrees.

Rectangle: Height is 65% of width (the symbol size).

Rotated rectangle: Same, with a rotation of 30 degrees.

Rounded rectangle: Same as rectangle, but with corner radius of 10% of width.

Mathangle: Angles are -10 and 45 degrees, with arrow head size 30% of symbol size.

Wedge: Angles are -30 and 30 degrees.

1.67.7 Examples

To add an example of a legend to a Mercator plot (map.ps) with the given specifications, use

```
gmt pslegend -R-10/10/-10/10 -JM6i -F+gazure1 -Dx0.5i/0.5i/5i/3.3i/BL \
-C0.1i/0.1i -L1.2 -B5f1 << EOF >> map.ps
# Legend test for pslegend
# G is vertical gap, V is vertical line, N sets # of columns, D draws horizontal line.
# H is header, L is label, S is symbol, T is paragraph text, M is map scale.
#
G -0.1i
H 24 Times-Roman My Map Legend
D 0.2i 1p
N 2
V 0 1p
S 0.1i c 0.15i p300/12 0.25p 0.3i This circle is hachured
S 0.1i e 0.15i yellow 0.25p 0.3i This ellipse is yellow
S 0.1i w 0.15i green 0.25p 0.3i This wedge is green
S 0.1i f0.1i+1+t 0.25i blue 0.25p 0.3i This is a fault
S 0.1i - 0.15i - 0.25p,- 0.3i A dashed contour
S 0.1i v0.1i+a40+e 0.25i magenta 0.25p 0.3i This is a vector
S 0.1i i 0.15i cyan 0.25p 0.3i This triangle is boring
V 0 1p
D 0.2i 1p
N 1
M 5 5 600+u f
G 0.05i
I SOEST_logo.ras 3i CT
G 0.05i
B colors.cpt 0.2i 0.2i
G 0.05i L 9 4 R Smith et al., @%5%J. Geophys. Res., 99@%, 2000
G 0.1i
P
T Let us just try some simple text that can go on a few lines.
T There is no easy way to predetermine how many lines will be required,
T so we may have to adjust the box height to get the right size box.
EOF
```

1.67.8 Note On Legend Height

As **-D** suggests, leaving the *height* off forces a calculation of the expected height. This is an exact calculation except in the case of legends that place paragraph text. Here we simply do a first-order estimate of how many typeset lines might appear. Without access to font metrics this estimate will occasionally be off by 1 line. If so, note the reported height (with **-V**) and specify a slightly larger or smaller height in **-D**.

1.67.9 Windows Remarks

Note that under Windows, the percent sign (%) is a variable indicator (like \$ under Unix). To indicate a plain percentage sign in a batch script you need to repeat it (%%); hence the font switching mechanism (@%*font*% and @%%) may require twice the number of percent signs. This only applies to text inside

a script or that otherwise is processed by DOS. Data files that are opened and read by **pslegend** do not need such duplication.

1.67.10 See Also

gmt, gmt.conf, gmtcolors, psbasemap, pstext, psxy

1.68 pslib

PSL 5.1 - A PostScript based plotting library

1.68.1 Description

PSL was created to make the generation of PostScript page description code easier. PS is a page description language developed by the Adobe for specifying how a printer should render a page of text or graphics. It uses a reverse Polish notation that puts and gets items from a stack to draws lines, text, and images and even performs calculations. PSL is a self-contained library that presents a series of functions that can be used to create plots. The resulting PostScript code is ASCII text (with some exceptions for images if so desired) and can thus be edited using any text editor. Thus, it is possible to modify a plot file even after it has been created, e.g., to change text strings, set new gray shades or colors, experiment with various pen widths, etc. Furthermore, various tools exist that can parse PostScript and let you make such edits via a graphical user interface (e.g., Adobe Illustrator). PSL is written in C but includes FORTRAN bindings and can therefore be called from both C and FORTRAN programs. To use this library, you must link your plotting program with PSL. PSL is used by the GMT graphics programs to generate PS. PSL output is freeform PostScript that conforms to the Adobe PostScript File Specification Version 3.0.

Before any PSL calls can be issued, the plotting system must be initialized. This is done by calling **PSL_beginsession**, which initializes a new PSL session; then call **PSL_setdefaults** which sets internal variables and default settings, accepts settings for measurement units and character encoding, and returns a pointer to a struct **PSL_CTRL** which must be passed as first argument to all other PSL functions. The measure unit for sizes and positions can be set to be centimeter (c), inch (i), meter (m), or points (p). A PSL session is terminated by calling **PSL_endsession**. You may create one or more plots within the same session. A new plot is started by calling **PSL_beginplot**, which defines macros, sets up the plot-coordinate system, scales, and [optionally] opens a file where all the PS code will be written. Normally, the plot code is written to *stdout*. When all plotting to this file is done, you finalize the plot by calling **PSL_endplot**.

A wide variety of output devices that support PostScript exist, including many printers and large-format plotters. Many tools exist to display PostScript on a computer screen. Open source tools such as ghostscript can be used to convert PostScript into PDF or raster images (e.g., TIFF, JPEG) at a user-defined resolution (DPI). In particular, the GMT tool *ps2raster* is a front-end to ghostscript and pre-selects the optimal options for ghostscript that will render quality PDF and images.

The PSL is fully 64-bit compliant. Integer parameters are here specified by the type **long** to distinguish them from the 32-bit **int**. Note that under standard 32-bit compilation they are equivalent. Users of this library under 64-bit mode must make sure they pass proper **long** variables (under Unix flavors) or **__int64** under Windows 64.

1.68.2 Units

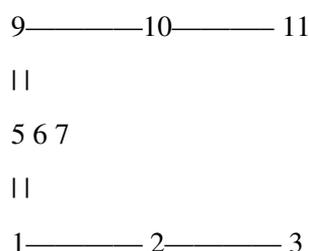
PSL can be instructed to use centimeters, inches, meters or points as input units for the coordinates and sizes of elements to be plotted. Any dimension that takes this setting as a unit is specified as *user units* or *plot units* in this manual. Excluded from this are line widths and font sizes which are always measured in *points*. The user units can be further refined by calling **PSL_beginaxes**, giving the user the opportunity to specify any linear coordinate frame. Changing the coordinate frame only affects the coordinates of plotted material indicated as measured in *plot units*, not the sizes of symbols (which remain in *user units*), nor line widths or font sizes (which remain in *points*).

1.68.3 Color

PSL uses the direct color model where red, green, and blue are given separately, each must be in the range from 0-1. If red = -1 then no fill operation takes place. If red = -3, then pattern fill will be used, and the green value will indicate the pattern to be used. Most plot-items can be plotted with or without outlines. If outline is desired (i.e., set to 1), it will be drawn using the current line width and pattern. PSL uses highly optimized macro substitutions and scales the coordinates depending on the resolution of the hardcopy device so that the output file is kept as compact as possible.

1.68.4 Justification

Text strings, text boxes and images can be “justified” by specifying the corner to which the *x* and *y* coordinates of the subroutine call apply. Nine different values are possible, as shown schematically in this diagram:



The box represents the text or image. E.g., to plot a text string with its center at (*x*, *y*), you must use *justify* == 6. *justify* == 0 means “no justification”, which generally means (*x*, *y*) is at the bottom left. Convenience values PSL_NONE, PSL_BL, PSL_BC, PSL_BL, PSL_ML, PSL_MC, PSL_MR, PSL_TL, PSL_TC and PSL_TR are available.

1.68.5 Initialization

These functions initialize or terminate the PSL system. We use the term PSL session to indicate one instance of the PSL system (a complicated program could run many PSL sessions concurrently as each would operate via its own control structure). During a single session, one or more plots may be created. Here are the functions involved in initialization:

```
struct PS_CTRL *New_PSL_Ctrl (char *session)
```

This is the first function that must be called as it creates a new PSL session. Specifically, it will allocate a new PSL control structure and initialize the session default parameters. The pointer that is returned must be passed to all subsequent PSL functions.

```
long *PSL_beginsession (struct PS_CTRL *PSL, long search, char *sharedir, char *userdir)
```

This is the second function that must be called as it initializes the new PSL session. Here, *search* is an integer that is passed as 0 in GMT but should be 1 for other users. If so we will search for the environmental parameters PSL_SHAREDIR and PSL_USERDIR should the corresponding arguments *sharedir* and *userdir* be NULL.

long PSL_endsession (struct PS_CTRL *PSL)

This function terminates the active PSL session; it is the last function you must call in your program. Specifically, this function will deallocate memory used and free up resources.

struct PS_CTRL *PSL_beginlayer (struct PS_CTRL *PSL, long layer)

Adds a DSC comment by naming this layer; give a unique integer value. Terminate layer with PSL_endlayer

struct PS_CTRL *PSL_endlayer (struct PS_CTRL *PSL)

Terminate current layer with a DSC comment.

long PSL_fopen (char *file, char *mode)

This function simply opens a file, just like fopen. The reason it is replicated here is that under Windows, file pointers must be assigned within the same DLL as they are being used. Yes, this is retarded but if we do not do so then PSL will not work well under Windows. Under non-Windows this functions is just a macro that becomes fopen.

void PSL_free (void *ptr)

This function frees up the memory allocated inside PSL. Programmers using C/C++ should now this is a macro and there is no need to cast the pointer to *void ** as this will be done by the macro. Fortran programmers should instead call **PSL_freefunction**.

void PSL_beginaxes (struct PS_CTRL *PSL, double llx, double lly, double width, double height, double x0, double y0, double x1, double y1)

This function sets up the mapping that takes the users data coordinates and converts them to the positions on the plot in PostScript units. This should be used when plotting data coordinates and is terminated with **PSL_endaxes**, which returns PSL to the default measurement units and scaling. Here, *llx* and *lly* sets the lower left position of the mapping region, while *width* and *height* sets the dimension of the plot area in user units. Finally, *x0*, *x1* and *y0*, *y1* indicate the range of the users x- and y-coordinates, respectively. Specify a reverse axis direction (e.g., to let the y-axis be positive down) by setting *y0* larger than *y1*, and similarly for an x-axis that increases to the left.

void PSL_endaxes (struct PS_CTRL *PSL)

Terminates the map scalings initialized by **PSL_beginaxes** and returns PSL to standard scaling in measurement units.

long PSL_beginplot (struct PS_CTRL *P, FILE *fp, long orientation, long overlay, long color_mode, char origin[], double offset[], double page_size[], char *title, long font_no[])

Controls the initiation (or continuation) of a particular plot within the current session. Pass file pointer *fp* where the PostScript code will be written; if NULL then the output is written to *stdout*. The Fortran interface always sends to *stdout*. The *orientation* may be landscape (PSL_LANDSCAPE or 0) or portrait (PSL_PORTRAIT or 1). Set *overlay* to PSL_OVERLAY (0) if the following PostScript code should be appended to an existing plot; otherwise pass PSL_INIT (1) to start a new plot. Let *colormode* be one of PSL_RGB (0), PSL_CMYK (1), PSL_HSV (2) or PSL_GRAY (3); this setting controls how colors are

presented in the PostScript code. The *origin* setting determines for x and y separately the origin of the specified offsets (next argument). Each of the two characters are either **r** for an offset relative to the current origin, **a** for a temporary adjustment of the origin which is undone during BD(PSL_endplot), **f** for a placement of the origin relative to the lower left corner of the page, **c** for a placement of the origin relative to the center of the page. The array *offset* specifies the offset of the new origin relative to the position indicated by **origin**. *page_size* means the physical width and height of the plotting media in points (typically 612 by 792 for Letter or 595 by 842 for A4 format). The character string *title* can be used to specify the `%%Title:` header in the PostScript file (or use NULL for the default). The array *font_no* specifies all fonts used in the plot (by number), or use NULL to leave out the `%%DocumentNeededResources:` comment in the PostScript file.

long PSL_endplot (struct PSL_CTRL *P, long last_page)

Terminates the plotting sequence and closes plot file (if other than *stdout*). If *last_page* == PSL_FINALIZE (1), then a PostScript *showpage* command is issued, which initiates the printing process on hardcopy devices. Otherwise, pass PSL_OVERLAY (0).

long PSL_setorigin (struct PSL_CTRL *P, double xorigin, double yorigin, double angle, long mode)

Changes the coordinate system by translating by (*xorigin,yorigin*) followed by a *angle*-degree rotation (*mode*=PSL_FWD or 0) or alternatively the rotation followed by translation (*mode*=PSL_INV or 1).

1.68.6 Changing Settings

The following functions are used to change various PSL settings and affect the current state of parameters such as line and fill attributes.

long PSL_define_pen (struct PSL_CTRL *P, char *name, long width, char *style, double offset, double rgb[])

Stores the specified pen characteristics in a PostScript variable called *name*. This can be used to place certain pen attributes in the PostScript file and then retrieve them later with **PSL_load_pen**. This makes the stored pen the current pen.

long PSL_define_rgb (struct PSL_CTRL *P, char *name, double rgb[])

Stores the specified color in a PostScript variable called *name*. This can be used to place certain color values in the PostScript file and then retrieve them later with **PSL_load_rgb**. This makes the stored color the current color.

long PSL_setcolor (struct PSL_CTRL *P, double rgb[], long mode)

Sets the current color for all stroked (*mode* = PSL_IS_STROKE (0)) or filled (*mode* = PSL_IS_FILL (1)) material to follow (lines, symbol outlines, text). *rgb* is a triplet of red, green and blue values in the range 0.0 through 1.0. Set the red color to -3.0 and the green color to the pattern number returned by **PSL_setpattern** to select a pattern as current paint color. For PDF transparency, set *rgb*[3] to a value between 0 (opaque) and 1 (fully transparent).

long PSL_setpattern (struct PSL_CTRL *P, long image_no, char *imagefile, long dpi, double f_rgb[], double b_rgb[])

Sets up the specified image pattern as the fill to use for polygons and symbols. Here, *image_no* is the number of the standard PSL fill patterns (1-90; use a negative number when you specify an image *filename* instead. The scaling (i.e., resolution in dots per inch) of the

pattern is controlled by the image *dpi*; if set to 0 it will be plotted at the device resolution. The two remaining settings apply to 1-bit images only and are otherwise ignored: You may replace the foreground color (the set bits) with the *f_rgb* color and the background color (the unset bits) with *b_rgb*. Alternatively, pass either color with the red component set to -1.0 and we will instead issue an image mask that is see-through for the specified fore- or back-ground component. To subsequently use the pattern as a pen or fill color, use **PSL_setcolor** or **DB(PSL_setfill)** with the a color *rgb* code made up of *r* = -3, and *b* = the pattern number returned by **PSL_setpattern**.

long PSL_setdash (struct PSL_CTRL *P, char *pattern, double offset)

Changes the current pen style attributes. The character string *pattern* contains the desired pattern using a series of lengths in points specifying the alternating lengths of dashes and gaps in points. E.g., "4 2" and *offset* = 1 will plot like

x — — —

where x is starting point of a line (The x is not plotted). That is, the line is made up of a repeating pattern of a 4 points long solid line and a 2 points long gap, starting 1 point after the x. To reset to solid line, specify *pattern* = NULL ("") and *offset* = 0.

long PSL_setfill (struct PSL_CTRL *P, double rgb[], long outline)

Sets the current fill color and whether or not outline is needed for symbols. Special cases are handled by passing the red color as -1.0 (no fill), -2.0 (do not change the outline setting) or -3.0 (select the image pattern indicated by the second (green) element of *rgb*). For PDF transparency, set *rgb*[3] to a value between 0 (opaque) and 1 (fully transparent). Set outline to **PSL_OUTLINE** (1) to draw the outlines of polygons and symbols using the current pen.

long PSL_setfont (struct PSL_CTRL *P, long fontnr)

Changes the current font number to *fontnr*. The fonts available are: 0 = Helvetica, 1 = H. Bold, 2 = H. Oblique, 3 = H. Bold-Oblique, 4 = Times, 5 = T. Bold, 6 = T. Italic, 7 = T. Bold Italic, 8 = Courier, 9 = C. Bold, 10 = C Oblique, 11 = C Bold Oblique, 12 = Symbol, 13 = AvantGarde-Book, 14 = A.-BookOblique, 15 = A.-Demi, 16 = A.-DemiOblique, 17 = Bookman-Demi, 18 = B.-DemiItalic, 19 = B.-Light, 20 = B.-LightItalic, 21 = Helvetica-Narrow, 22 = H-N-Bold, 23 = H-N-Oblique, 24 = H-N-BoldOblique, 25 = NewCenturySchlbk-Roman, 26 = N.-Italic, 27 = N.-Bold, 28 = N.-BoldItalic, 29 = Palatino-Roman, 30 = P.-Italic, 31 = P.-Bold, 32 = P.-BoldItalic, 33 = ZapfChancery-MediumItalic, 34 = ZapfDingbats, 35 = Ryumin-Light-EUC-H, 36 = Ryumin-Light-EUC-V, 37 = GothicBBB-Medium-EUC-H, and 38 = GothicBBB-Medium-EUC-V. If *fontnr* is outside this range, it is reset to 0.

long PSL_setfontdims (struct PSL_CTRL *P, double supsub, double scaps, double sup, double sdown)

Changes the settings for a variety of relative font sizes and shifts pertaining to sub-scripts, super-scripts, and small caps. Default settings are given in brackets. Here, *supsub* sets the relative size of sub- and super-scripts [0.58], *scaps* sets the relative size of small caps [0.8], *sup* indicates the upward baseline shift for placement of super-scripts [0.33], while *sdown* sets the downward baseline shift for sub-scripts [0.33].

long PSL_setformat (struct PSL_CTRL *P, long n_decimals)

Sets the number of decimals to be used when writing color or gray values. The default setting of 3 gives 1000 choices per red, green, and blue value, which is more than the 255 choices offered by most 24-bit platforms. Choosing a lower value will make the output file

smaller at the expense of less color resolution. Still, a value of 2 gives 100 x 100 x 100 = 1 million colors, more than most eyes can distinguish. For a setting of 1, you will have 10 nuances per primary color and a total of 1000 unique combinations.

long PSL_setlinewidth (struct PSL_CTRL *P, double *linewidth*)

Changes the current line width in points. Specifying 0 gives the thinnest line possible, but this is implementation-dependent (seems to work fine on most PostScript printers).

long PSL_setlinecap (struct PSL_CTRL *P, long *cap*)

Changes the current line cap, i.e., what happens at the beginning and end of a line segment. PSL_BUTT_CAP (0) gives butt line caps [Default], PSL_ROUND_CAP (1) selects round caps, while PSL_SQUARE_CAP (2) results in square caps. Thus, the two last options will visually lengthen a straight line-segment by half the line width at either end.

long PSL_setlinejoin (struct PSL_CTRL *P, long *join*)

Changes the current linejoin setting, which handles how lines of finite thickness are joined together when they meet at different angles. PSL_MITER_JOIN (0) gives a mitered joint [Default], PSL_ROUND_JOIN (1) makes them round, while PSL_BEVEL_JOIN (2) produces bevel joins.

long PSL_setmiterlimit (struct PSL_CTRL *P, long *limit*)

Changes the current miter limit used for mitered joins. PSL_MITER_DEFAULT (35) gives the default PS miter; other values are interpreted as the cutoff acute angle (in degrees) when mitering becomes active.

long PSL_settransparencymode (struct PSL_CTRL *P, char **mode*)

Changes the current PDF transparency rendering mode [Default is Normal]. Choose among Color, ColorBurn, ColorDodge, Darken, Difference, Exclusion, HardLight, Hue, Lighten, Luminosity, Multiply, Normal, Overlay, Saturation, SoftLight, and Screen.

long PSL_setdefaults (struct PSL_CTRL *P, double *xyscales*[], double *pagergb*[], char **encoding*)

Allows changes to the PSL session settings and should be called immediately after **PSL_beginsession**. The *xyscales* array affect an overall magnification of your plot [1,1]. This can be useful if you design a page-sized plot but would then like to magnify (or shrink) it by a given factor. Change the default paper media color [white; 1/1/1] by specifying an alternate page color. Passing zero (or NULL for *pagergb*) will leave the setting unchanged. Finally, pass the name of the character set encoding (if NULL we select Standard).

long PSL_defunits (struct PSL_CTRL *P, char **name*, double *value*)

Creates a PostScript variable called *name* and initializes it to the equivalent of *value* user units.

long PSL_defpoints (struct PSL_CTRL *P, char **name*, double *fontsize*)

Creates a PostScript variable called *name* and initializes it to the value that corresponds to the font size (in points) given by *fontsize*.

1.68.7 Plotting Lines And Polygons

Here are functions used to plot lines and closed polygons, which may optionally be filled. The attributes used for drawing and filling are set prior to calling these functions; see CHANGING SETTINGS above.

long PSL_plotarc (struct PSL_CTRL *P, double x, double y, double radius, double angle1, double angle2, long type)

Draws a circular arc with its center at plot coordinates (x, y), starting from angle *angle1* and end at *angle2*. Angles must be given in decimal degrees. If *angle1* > *angle2*, a negative arc is drawn. The *radius* is in user units. The *type* determines how the arc is interpreted: PSL_MOVE (1) means set new anchor point, PSL_STROKE (2) means stroke the arc, PSL_MOVE + PSL_STROKE (3) means both, whereas PSL_DRAW (0) just adds to arc path to the current path.

long PSL_plotline (struct PSL_CTRL *P, double x, double y, long n, long type)

Assemble a continuous line through *n* points whose the plot coordinates are in the *x*, *y* arrays. To continue an existing line, use *type* = PSL_DRAW (0), or if this is the first segment in a multisegment path, set *type* = PSL_MOVE (1). To end the segments and draw the lines, add PSL_STROKE (2). Thus, for a single segment, *type* must be PSL_MOVE + PSL_STROKE (3). The line is drawn using the current pen attributes. Add PSL_CLOSE (8) to *type* to close the first and last point by the PostScript operators; this is done automatically if the first and last point are equal.

long PSL_plotpoint (struct PSL_CTRL *P, double x, double y, long type)

Moves the pen from the current to the specified plot coordinates (x, y) and optionally draws and strokes the line, depending on *type*. Specify *type* as either a move (PSL_MOVE, 1), or draw (PSL_DRAW, 2), or draw and stroke (PSL_DRAW + PSL_STROKE, 3) using current pen attributes. If the coordinates are relative to the current point add PSL_REL (4) to *type*.

long PSL_plotbox (struct PSL_CTRL *P, double x0, double y0, double x1, double y1)

Creates a closed box with opposite corners at plot coordinates (x0,y0) and (x1,y1). The box may be filled and its outline stroked depending on the current settings for fill and pen attributes.

long PSL_plotpolygon (struct PSL_CTRL *P, double x, double y, long n)

Creates a closed polygon through *n* points whose plot coordinates are in the *x*, *y* arrays. The polygon may be filled and its outline stroked depending on the current settings for fill and pen attributes.

long PSL_plotsegment (struct PSL_CTRL *P, double x0, double y0, double x1, double y1)

Draws a line segment between the two points (plot coordinates) using the current pen attributes.

1.68.8 Plotting Symbols

Here are functions used to plot various geometric symbols or constructs.

long PSL_plotaxis (struct PSL_CTRL *P, double tickval, char *label, double fontsize, long side)

Plots a basic axis with tick marks, annotations, and label. Assumes that **PSL_beginaxes** has been called to set up positioning and user data ranges. Annotations will be set using the *fontsize* in points. *side* can be 0, 1, 2, or 3, which selects lower x-axis, right y-axis, upper x-axis, or left y-axis, respectively. The *label* font size is set to 1.5 times the *fontsize*.

long PSL_plotsymbol (struct PSL_CTRL *P, double x, double y, double size[], long symbol)

Plots a simple geometric symbol centered on plot coordinates (x, y) . The argument *symbol* selects the geometric symbol to use. Most symbols are scaled to fit inside a circle of diameter given as *size*[0], but some symbols take additional parameters. Choose from these 1-parameter symbols using the predefined self-explanatory integer values PSL_CIRCLE, PSL_DIAMOND, PSL_HEXAGON, PSL_INVTRIANGLE, PSL_OCTAGON, PSL_PENTAGON, PSL_SQUARE, PSL_STAR, and PSL_TRIANGLE; these may all be filled and stroked if **PSL_setfill** has been called first. In addition, you can choose several line-only symbols that cannot be filled. They are PSL_CROSS, PSL_DOT, PSL_PLUS, PSL_XDASH, and PSL_YDASH. Finally, more complicated symbols require more than one parameter to be passed via *size*. These are PSL_ELLIPSE (*size* is expected to contain the three parameter *angle*, *major*, and *minor* axes, which defines an ellipse with its major axis rotated by *angle* degrees), PSL_MANGLE (*size* is expected to contain the 8 parameters *radius*, *angle1*, and *angle2* for the math angle specification, followed by *tail-width*, *headlength*, *headwidth*, *shape*, and *status* (see PSL_VECTOR below for explanation), PSL_WEDGE (*size* is expected to contain the three parameter *radius*, *angle1*, and *angle2* for the sector specification), PSL_RECT (*size* is expected to contain the two dimensions *width* and *height*), PSL_RNDRECT (*size* is expected to contain the two dimensions *width* and *height* and the *radius* of the corners), PSL_ROTRECT (*size* is expected to contain the three parameter *angle*, *width*, and *height*, with rotation relative to the horizontal), and PSL_VECTOR (*size* is expected to contain the 7 parameters *x_tip*, *y_tip*, *tail-width*, *headlength*, *headwidth*, *shape*, and *status*. Here (x_tip, y_tip) are the coordinates to the head of the vector, while (x, y) are those of the tail. *shape* can take on values from 0-1 and specifies how far the intersection point between the base of a straight vector head and the vector line is moved toward the tip. 0.0 gives a triangular head, 1.0 gives an arrow shaped head. The *status* value is a bit-flag being the sum of several possible contributions: PSL_VEC_RIGHT (2) = only draw right half of vector head, PSL_VEC_BEGIN (4) = place vector head at beginning of vector, PSL_VEC_END (8) = place vector head at end of vector, PSL_VEC_JUST_B (0) = align vector beginning at (x,y) , PSL_VEC_JUST_C (16) = align vector center at (x,y) , PSL_VEC_JUST_E (32) = align vector end at (x,y) , PSL_VEC_JUST_S (64) = align vector center at (x,y) , PSL_VEC_OUTLINE (128) = draw vector head outline using default pen, PSL_VEC_FILL (512) = fill vector head using default fill, PSL_VEC_MARC90 (2048) = if angles subtend 90, draw straight angle symbol (PSL_MANGLE only). The symbol may be filled and its outline stroked depending on the current settings for fill and pen attributes.

1.68.9 Plotting Images

Here are functions used to read and plot various images.

long PSL_plotbitimage (**struct PSL_CTRL *P**, **double x**, **double y**, **double xsize**, **double ysize**, **int justify**, **unsigned char buffer**, **long nx**, **long ny**, **double f_rgb[]**, **double b_rgb[]**)

Plots a 1-bit image image at plot coordinates (x, y) justified as per the argument *justify* (see **JUSTIFICATION** for details). The target size of the image is given by *xsize* and *ysize* in user units. If one of these is specified as zero, the corresponding size is adjusted to the other such that the aspect ratio of the original image is retained. *buffer* is an unsigned character array in scanline orientation with 8 pixels per byte. *nx*, *ny* refers to the number of pixels in the image. The rowlength of *buffer* must be an integral number of 8; pad with zeros. *buffer*[0] is upper left corner. You may replace the foreground color (the set bits) with the *f_rgb* color and the background color (the unset bits) with *b_rgb*. Alternatively, pass either color with the red component set to -1.0 and we will instead issue an image mask that

is see-through for the specified fore- or background component. See the Adobe Systems PostScript Reference Manual for more details.

long PSL_plotcolorimage (struct PSL_CTRL *P, double x, double y, double xsize, double ysize, int justify, unsigned char *buffer, long nx, long ny, long depth)

Plots a 1-, 2-, 4-, 8-, or 24-bit deep image at plot coordinates (x, y) justified as per the argument *justify* (see **JUSTIFICATION** for details). The target size of the image is given by *xsize* and *ysize* in user units. If one of these is specified as zero, the corresponding size is adjusted to the other such that the aspect ratio of the original image is retained. This function sets up a call to the PostScript colorimage or image operators. The pixel values are stored in *buffer*, an unsigned character array in scanline orientation with gray shade or r/g/b values (0-255). *buffer*[0] is the upper left corner. *depth* is number of bits per pixel (24, 8, 4, 2, or 1). *nx*, *ny* refers to the number of pixels in image. The rowlength of *buffer* must be an integral number of $8/depth$. E.g. if *depth* = 4, then *buffer*[j]/16 gives shade for pixel[2j-1] and *buffer*[j%16 (mod 16)] gives shade for pixel[2j]. When *-depth* is passed instead then "hardware" interpolation of the image is requested (this is implementation dependent). If *-nx* is passed with 8- (or 24-) bit images then the first one (or three) bytes of *buffer* holds the gray (or r/g/b) color for pixels that are to be masked out using the PS Level 3 Color Mask method. See the Adobe Systems PostScript Reference Manual for more details.

long PSL_plotepsimage (struct PSL_CTRL *P, double x, double y, double xsize, double ysize, int justify, unsigned char *buffer, long size, long nx, long ny, long ox, long oy)

Plots an Encapsulated PostScript (EPS) image at plot coordinates (x, y) justified as per the argument *justify* (see **JUSTIFICATION** for details). The target size of the image is given by *xsize* and *ysize* in user units. If one of these is specified as zero, the corresponding size is adjusted to the other such that the aspect ratio of the original image is retained. The EPS file is stored in *buffer* and has *size* bytes. This function simply includes the image in the PostScript output stream within an appropriate wrapper. Specify position of lower left corner and size of image. *nx*, *ny*, *ox*, *oy* refers to the width, height and origin (lower left corner) of the BoundingBox in points.

long PSL_loadimage (struct PSL_CTRL *P, FILE *fp, struct imageinfo *header, unsigned char **image)

Reads the image contents of the EPS file or a raster image pointed to by the open file pointer *fp*. The routine can handle Encapsulated PostScript files or 1-, 8-, 24-, or 32-bit raster images in old, standard, run-length encoded, or RGB-style Sun format. Non-Sun rasters are automatically reformatted to Sun rasters via a system call to GraphicsMagick's or ImageMagick's **convert**, if installed. The image is returned via the *image* pointer.

1.68.10 Plotting Text

Here are functions used to read and plot text strings and paragraphs. This can be somewhat complicated since we rely on the PostScript interpreter to determine the exact dimensions of text items given the font chosen. For perfect alignment you may have to resort to calculate offsets explicitly using **long PSL_deftextdim**, **PSL_set_height** and others and issue calculations with **PSL_setcommand**.

long PSL_plottext (struct PSL_CTRL *P, double x, double y, double fontsize, char *text, double angle, long justify, long mode)

The *text* is plotted starting at plot coordinates (x, y) and will make an *angle* with the horizontal. The point (x, y) maps onto different points of the text-string by giving various values for *justify* (see **JUSTIFICATION** for details). If *justify* is negative, then all leading and

trailing blanks are stripped before plotting. Certain character sequences (flags) have special meaning to **PSL_plottext**. @~ toggles between current font and the Mathematical Symbols font. @%no% selects font *no* while @%% resets to the previous font. @- turns subscript on/off, @+ turns superscript on/off, @# turns small caps on/off, and @\ will make a composite character of the following two character. @;r/g/b; changes the font color while @;; resets it [optionally append =*transparency* to change the transparency (0–100) of the text (the Default is opaque or 0)], @:size: changes the font size (@:: resets it), and @_ toggles underline on/off. If *text* is NULL then we assume **PSL_plottextbox** was called first. Give *fontsize* in points. Normally, the text is typed using solid characters in the current color (set by **PSL_setcolor**). To draw outline characters, set *mode* == 1; the outline will get the current color and the text is filled with the current fill color (set by **PSL_setfill**). Use *mode* == 2 if the current fill is a pattern. If *fontsize* is negative it means that the current point has already been set before **PSL_plottext** was called and that (*x*, *y*) should be ignored.

long PSL_plottextbox (struct PSL_CTRL *P, double *x*, double *y*, double *fontsize*, char **text*, double *angle*, long *justify*, double *offset*[], long *mode*)

This function is used in conjugation with **PSL_plottext** when a box surrounding the text string is desired. Taking most of the arguments of **PSL_plottext**, the user must also specify *mode* to indicate whether the box needs rounded (PSL_YES = 1) or straight (PSL_NO = 0) corners. The box will be colored with the current fill style set by **PSL_setfill**. That means, if an outline is desired, and the color of the inside of the box should be set with that routine. The outline will be drawn with the current pen color (and width). The *offset* array holds the horizontal and vertical distance gaps between text and the surrounding text box in distance units. The smaller of the two determined the radius of the rounded corners (if requested).

long PSL_deftextdim (struct PSL_CTRL *P, char **prefix*, double *fontsize*, char **text*)

Computes the dimensions (width and height) required by the selected *text* given the current font and its *fontsize* (in points). The values are stored as PostScript variables called *prefix_w* and *prefix_h*, respectively. This function can be used to compute dimensions and, via **BF(PSL_setcommand)**, calculate chances to position a particular item should be plotted. For instance, if you compute a position this way and wish to plot the text there, pass the coordinates to **PSL_plottext** as NaNs. If *prefix* is BF(-w), BF(-h), BF(-d) or BF(-b), no PostScript variables will be assigned, but the values of width, height, depth, or both width and height will be left on the PostScript stack.

long PSL_setparagraph (struct PSL_CTRL *P, double *line_space*, double *par_width*, long *par_just*)

Initialize common settings to be used when typesetting paragraphs of text with **PSL_plotparagraph**. Specify the line spacing (1 equals the font size) and paragraph width (in distance units). Text can be aligned left (PSL_BL), centered (PSL_BC), right (PSL_BR), or justified (PSL_JUST) and is controlled by *par_just*.

long PSL_plotparagraphbox (struct PSL_CTRL *P, double *x*, double *y*, double *fontsize*, char **text*, double *angle*, long *justify*, double *offset*[], long *mode*)

Computes and plots the text rectangle for a paragraph using the specified *fontsize* (in points). Here, *text* is an array of the text to be typeset, using the settings initialized by **PSL_setparagraph**. The escape sequences described for **PSL_plottext** can be used to modify the text. Separate text into several paragraphs by appending \r to the last item in a paragraph. The whole text block is positioned at plot coordinates *x*, *y*, which is mapped to a point on the block specified by *justify* (see **JUSTIFICATION** for details). The whole block is then shifted by the amounts *shift*[], The box will be plotted using the current fill and outline settings. The *offset* array holds the horizontal and vertical distance gaps between

text and the surrounding text box in distance units. Use *mode* to indicate whether the box should be straight (PSL_RECT_STRAIGHT = 0), rounded (PSL_RECT_ROUNDED = 1), convex (PSL_RECT_CONVEX = 2) or concave (PSL_RECT_CONCAVE = 3).

long PSL_plotparagraph (struct PSL_CTRL *P, double x, double y, double fontsize, char *text, double angle, long justify, long mode)

Typesets paragraphs of text using the specified *fontsize* (in points). Here, *text* is an array of the text to be typeset, using the settings initialized by **PSL_setparagraph**. The escape sequences described for **PSL_plottext** can be used to modify the text. Separate text into several paragraphs by appending `\r` to the last item in a paragraph. The whole text block is positioned at plot coordinates *x*, *y*, which is mapped to a point on the block specified by *justify* (see **JUSTIFICATION** for details). See **PSL_plotparagraphbox** for laying down the surrounding text rectangle first.

long PSL_plottextline (struct PSL_CTRL *P, double *xpath, double *ypath, long *np, long nseg, void *arg1, void *arg2, char *text[], double angle[], long n_per_seg[], double fontsize, ***long* justify, double offset[], long mode)

Please text along one or more path segments. The function does different things depending on the bit flags in *mode*. A key distinction occurs if the bit flag contains the bit PSL_TXT_CURVED (64) which means we wish to typeset the text along a variable and curved baseline given by the segments in *xpath*, *ypath*; otherwise we set straight text (possibly at an angle) and the *xpath*, *ypath* are not considered for text placement [If no line drawing is desired then these two arrays may be NULL]. We will describe the action taken for each bit value. Multiple values may be passed at the same time and we process from low to high bit. PSL_TXT_INIT: When mode contains this bit (1) we will initialize all the required variables and store them in the PostScript file. PSL_TXT_SHOW: We wish to see the text strings (otherwise they may only serve as guides to set up clip paths). PSL_TXT_CLIP_ON: Use the text and the paths to set up clip paths. PSL_TXT_DRAW: Draw the lines defined by the *xpath*, *ypath* arrays. PSL_TXT_CLIP_OFF: Turn the text path clipping off. We pass the text strings via *text*. The locations of text plotting depends on whether PSL_TXT_CURVED is selected. If it is then you must pass as *arg1* the *node* array indicating at which node in the *xpath*, *ypath* array the text will be plotted; let *arg2* be NULL. For straight baselines you must instead pass another set of x,y coordinates with the locations of the text label placements via *arg1*, *arg2*. Each label has its own entry in the *angle* array. The *text* is an array of text pointers to the individual text items. The *offset* array holds the x and y distance gaps between text and the surrounding text box in user units (the clip path is the combination of all these text boxes). Use *justify* to specify how the text string relates to the coordinates (see **JUSTIFICATION**) for details). PSL_TXT_FILLBOX (128) will fill the text box (this requires you to first define the text box rgb color with **PSL_define_rgb** by setting a local PostScript variable that must be called PSL_setboxrgb). PSL_TXT_DRAWBOX (256) will draw the text box outlines (this requires you to first define the text box pen with **PSL_define_pen** by setting a local PostScript variable that must be called PSL_setboxpen). Before calling this function you must also initialize a PSL array for line pens and text fonts.

1.68.11 Clipping

Here are functions used to activate and deactivate clipping regions.

long PSL_beginclipping (struct PSL_CTRL *P, double x, double y, long n, double rgb[], long flag)

Sets up a user-definable clip path as a series on *n* points with plot coordinates (*x*, *y*). Plotting outside this polygon will be clipped until **PSL_endclipping** is called. If *rgb*[0] = -1 the

inside of the path is left empty, otherwise it is filled with the specified color. *flag* is used to create complex clip paths consisting of several disconnected regions, and takes on values 0-3. *flag* = PSL_PEN_MOVE_ABS (1) means this is the first path in a multisegment clip path. *flag* = PSL_PEN_DRAW_ABS (2) means this is the last segment. Thus, for a single path, *flag* = PSL_PEN_DRAW_AND_STROKE_ABS (3).

long PSL_endclipping (struct PSL_CTRL *P, long mode)

Depending on the *mode* it restores the clip path. The *mode* values can be: - *n* will restore *n* levels of text-based clipping, *n* will restore *n* levels of polygon clipping, PSL_ALL_CLIP_TXT will undo all levels of text-based clipping, and PSL_ALL_CLIP_POL will undo all levels of polygon-based clipping.

1.68.12 Miscellaneous Functions

Here are functions used to issue comments or to pass custom PostScript commands directly to the output PostScript file. In C these functions are declared as macros and they can accept a variable number of arguments. However, from FORTRAN only a single text argument may be passed.

long PSL_setcommand (struct PSL_CTRL *P, char *text) Writes a raw PostScript command to the PostScript output file, e.g., “1 setlinejoin0.

long PSL_comment (struct PSL_CTRL *P, char *text) Writes a comment (*text*) to the PostScript output file, e.g., “Start of graph 20. The comment are prefixed with with %% .

1.68.13 Authors

Paul Wessel, School of Ocean and Earth Science and Technology, <http://www.soest.hawaii.edu>.

Remko Scharroo, EUMETSAT, Darmstadt, Germany, <http://www.eumetsat.int>.

1.68.14 Bugs

Caveat Emptor: The authors are **not** responsible for any disasters, suicide attempts, or ulcers caused by correct **or** incorrect use of PSL. If you find bugs, please report them to the authors by electronic mail. Be sure to provide enough detail so that we can recreate the problem.

1.68.15 References

Adobe Systems Inc., 1990, PostScript language reference manual, 2nd edition, Addison-Wesley, (ISBN 0-201-18127-4).

1.69 psmask

psmask - Use data tables to clip or mask map areas with no coverage

1.69.1 Synopsis

```
psmask [ table ] -I $xinc[unit][=+][/yinc[unit][=+]]$  -J $parameters$  -R $[unit]xmin/xmax/ymin/ymax[r]$  [ -B $[pls]parameters$  ] [ -D $dumpfile$  ] [ -G $fill$  ] [ -Jz $Zparameters$  ] [ -K ] [ -N ] [ -O ] [ -P ] [ -Q $cut$  ] [ -S $search\_radius[unit]$  ] [ -T ] [ -U $[just/dx/dy][c]label$  ] [ -V $[level]$  ] [ -X $[alc]fir[x-shift[u]]$  ] [ -Y $[alc]fir[y-shift[u]]$  ] [ -bi $[ncols][type][w][+L+B]$  ] [ -ccopies ] [ -h $[ilo][n][+c][+d][+rremark][+rtitle]$  ] [ -i $[cols][l][sscale][ooffset][,...]$  ] [ -p $[x|y|z]azim/elev[zlevel][+wlon0/lat0[z0]][+vx0/y0]$  ] [ -r ] [ -t $[transp]$  ] [ -: $[ilo]$  ]
```

```
psmask -C [ -K ] [ -O ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.69.2 Description

psmask reads a (x,y,z) file [or standard input] and uses this information to find out which grid cells are reliable. Only grid cells which have one or more data points are considered reliable. As an option, you may specify a radius of influence. Then, all grid cells that are within *radius* of a data point are considered reliable. Furthermore, an option is provided to reverse the sense of the test. Having found the reliable/not reliable points, **psmask** will either paint tiles to mask these nodes (with the **-T** switch), or use contouring to create polygons that will clip out regions of no interest. When clipping is initiated, it will stay in effect until turned off by a second call to **psmask** using the **-C** option.

1.69.3 Required

-I $xinc[unit][=+][/yinc[unit][=+]]$ x_inc [and optionally y_inc] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If y_inc is given but set to 0 it will be reset equal to x_inc ; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max x (*east*) or y (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R $grdfile$** is used then the grid spacing has already been initialized; use **-I** to override the values.

-J $parameters$ (more ...) Select map projection.

-R $[unit]xmin/xmax/ymin/ymax[r]$ (more ...) Specify the region of interest.

For perspective view **p**, optionally append $lzmin/zmax$. (*more ...*)

1.69.4 Optional Arguments

table One or more ASCII (or binary, see **-bi $[ncols][type]$**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-B $[pls]parameters$ (more ...) Set map boundary intervals.

- C Mark end of existing clip path. No input file is needed. Implicitly sets -O. Also supply -X and -Y settings if you have moved since the clip started.
- D*dumpfile* Dump the (x,y) coordinates of each clipping polygon to one or more output files (or *stdout* if *template* is not given). No plotting will take place. If *template* contains the C-format specifier %d (including modifications like %05d) then polygons will be written to different files; otherwise all polygons are written to the specified file (*template*). The files are ASCII unless -bo is used. See -Q to exclude small polygons from consideration.
- p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (*more ...*) Select perspective view.
- Gfill Paint the clip polygons (or tiles) with a selected fill [Default is no fill].
- Jz/Zparameters (*more ...*) Set z-axis scaling; same syntax as -Jx.
- K (*more ...*) Do not finalize the PostScript plot.
- N Invert the sense of the test, i.e., clip regions where there is data coverage.
- O (*more ...*) Append to existing PostScript plot.
- P (*more ...*) Select “Portrait” plot orientation.
- Q Do not dump polygons with less than *cut* number of points [Dumps all polygons]. Only applicable if -D has been specified.
- S*search_radius*[*unit*] Sets radius of influence. Grid nodes within *radius* of a data point are considered reliable. [Default is 0, which means that only grid cells with data in them are reliable]. Append the distance unit (see UNITS).
- T Plot tiles instead of clip polygons. Use -G to set tile color or pattern. Cannot be used with -D.
- U[*just/dx/dy/*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*) Select verbosity level [c].
- X[*alclflr*][*x-shift*[*u*]]
- Y[*alclflr*][*y-shift*[*u*]] (*more ...*) Shift plot origin.
- bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- ccopies (*more ...*) Specify number of plot copies [Default is 1].
- h[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s). Not used with binary data.
- i*cols*[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).
- p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (*more ...*) Select perspective view.
- r (*more ...*) Set pixel node registration [gridline].
- t[*transp*] (*more ...*) Set PDF transparency level in percent.
- :[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.69.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.69.6 Examples

To make an overlay PostScript file that will mask out the regions of a contour map where there is no control data using clip polygons, use:

```
gmt psmask africa_grav.xyg -R20/40/20/40 -I5m -JM10i -O -K > mask.ps
```

We do it again, but this time we wish to save the clipping polygons to file all_pols.txt:

```
gmt psmask africa_grav.xyg -R20/40/20/40 -I5m -Dall_pols.txt
```

A repeat of the first example but this time we use white tiling:

```
gmt psmask africa_grav.xyg -R20/40/20/40 -I5m -JM10i -T -O -K -Gwhite > mask.ps
```

1.69.7 See Also

gmt, *gmtcolors*, *grdmask*, *surface*, *psbasemap*, *psclip*

1.70 psrose

psrose - Plot a polar histogram (rose, sector, windrose diagrams)

1.70.1 Synopsis

```
psrose [ table ] [ -A[r]*sector_width* ] [ -B[p]parameters ] [ -C[mode_file] ] [ -D ] [ -I ]
[ -Gfill ] [ -J ] [ -K ] [ -L[wlabel/elabel/slabel/nlabel] ] [ -Mparameters ] [ -O ] [ -P ] [ -
Rr0/r1/az_0/az_1 ] [ -S[n]*radial_scale* ] [ -T ] [ -U[just/dx/dy]/[clabel] ] [ -V[level] ] [ -
W[v]pen ] [ -X[alc/flr]/[x-shift][u] ] [ -Y[alc/flr]/[y-shift][u] ] [ -Zscale ] [ -bi[ncols]/[type]/[w]/[+L+B]
] [ -ccopies ] [ -h[ilo]/[n]/[+c]/[+d]/[+rremark]/[+rtitle] ] [ -icols[l]/[sscale]/[offset]/[,...] ] [ -
p[xly/z]/[azim/elev]/[zlevel]/[+wlon0/lat0/z0]/[+vx0/y0] ] [ -t[transp] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.70.2 Description

psrose reads (length,azimuth) pairs from *file* [or standard input] and generates PostScript code that will plot a windrose diagram. Optionally (with **-A**), polar histograms may be drawn (sector diagram or rose diagram). Options include full circle and half circle plots. The PostScript code is written to standard output. The outline of the windrose is drawn with the same color as **MAP_DEFAULT_PEN**.

1.70.3 Required Arguments

None.

1.70.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input. If a file with only azimuths are given, use **-i** to indicate the single column with azimuths; then all lengths are set to unity (see **-Zu** to set actual lengths to unity as well).

-A[*r*]***sector_width*** Gives the sector width in degrees for sector and rose diagram. [Default 0 means windrose diagram]. Use **-Ar** to draw rose diagram instead of sector diagram.

-B[*pls*]*parameters (more ...)* Set map boundary intervals.

Remember that “x” here is radial distance and “y” is azimuth. The *ylabel* may be used to plot a figure caption.

-C[*mode_file*] Plot vectors showing the principal directions given in the *modes* file. If no file is given, compute and plot mean direction. See **-M** to control vector attributes.

-D Shift sectors so that they are centered on the bin interval (e.g., first sector is centered on 0 degrees).

-F Do not draw the scale length bar [Default plots scale in lower right corner]

-G*fill* Selects shade, color or pattern for filling the sectors [Default is no fill].

-I Inquire. Computes statistics needed to specify useful **-R**. No plot is generated.

-K (more ...) Do not finalize the PostScript plot.

-L[*wlabel/elabel/slabel/nlabel*] Specify labels for the 0, 90, 180, and 270 degree marks. For full-circle plot the default is WEST/EAST/SOUTH/NORTH and for half-circle the default is 90W/90E/-/0. A - in any entry disables that label. Use **-L** with no argument to disable all four labels

-M*parameters* Used with **-C** to modify vector parameters. For vector heads, append vector head *size* [Default is 0, i.e., a line]. See VECTOR ATTRIBUTES for specifying additional attributes.

-O (more ...) Append to existing PostScript plot.

-P (more ...) Select “Portrait” plot orientation.

-R*r0/r1/az_0/az_1* Specifies the ‘region’ of interest in (r,azimuth) space. *r0* is 0, *r1* is max length in units. For azimuth, specify either -90/90 or 0/180 for half circle plot or 0/360 for full circle.

-S[*n*]***radial_scale*** Specifies radius of circle. Use **-Sn** to normalize input radii to go from 0 to 1.

- T** Specifies that the input data is orientation data (has a 180 degree ambiguity) instead of true 0-360 degree directions [Default].
- U***[just/dx/dy/][clabel] (more ...)* Draw GMT time stamp logo on plot.
- V***[level] (more ...)* Select verbosity level [c].
- Wpen** Set pen attributes for sector outline or rose plot. [Default is no outline]. Use **-Wvpen** to change pen used to draw vector (requires **-C**) [Default is same as sector outline].
- X***[alclfr][x-shift[u]]*
- Y***[alclfr][y-shift[u]] (more ...)* Shift plot origin.
- Zscale** Multiply the data radii by *scale*. E.g., use **-Z0.001** to convert your data from m to km. To exclude the radii from consideration, set them all to unity with **-Zu** [Default is no scaling].
- : Input file has (azimuth,radius) pairs rather than the expected (radius,azimuth).
- bi***[ncols][type] (more ...)* Select binary input. [Default is 2 input columns].
- ccopies** *(more ...)* Specify number of plot copies [Default is 1].
- h***[ilo][n][+c][+d][+rremark][+rtitle] (more ...)* Skip or produce header record(s).
- icols***[1][sscale][offset][,...] (more ...)* Select input columns (0 is first column).
- p***[x|y|z]azim/lev[zlevel][+wlon0/lat0/z0][+vx0/y0] (more ...)* Select perspective view.
- t***[transp] (more ...)* Set PDF transparency level in percent.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.70.5 Vector Attributes

Several modifiers may be appended to the vector-producing options to specify the placement of vector heads, their shapes, and the justification of the vector. Below, left and right refers to the side of the vector line when viewed from the start point to the end point of the segment:

- +**a***angle* sets the angle of the vector head apex [30].
- +**b** places a vector head at the beginning of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +**e** places a vector head at the end of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +**g***-fill* turns off vector head fill (if -) or sets the vector head fill [Default fill is used, which may be no fill].
- +**l** draws half-arrows, using only the left side of specified heads [both sides].

+n*norm* scales down vector attributes (pen thickness, head size) with decreasing length, where vectors shorter than *norm* will have their attributes scaled by $\text{length}/\text{norm}$ [arrow attributes remains invariant to length].

+o*plon/plat* specifies the oblique pole for the great or small circles. Only needed for great circles if **+q** is given.

+p*[-][pen]* sets the vector pen attributes. If *pen* has a leading - then the head outline is not drawn. [Default pen is used, and head outline is drawn]

+q means the input *angle, length* data instead represent the *start* and *stop* opening angles of the arc segment relative to the given point.

+r draws half-arrows, using only the right side of specified heads [both sides].

In addition, all but circular vectors may take these modifiers:

+j*just* determines how the input *x,y* point relates to the vector. Choose from **beginning** [default], **end**, or **center**.

+s means the input *angle, length* is instead the *x, y* coordinates of the vector end point.

1.70.6 Examples

To plot a half circle rose diagram of the data in the file `fault_segments.az_r` (containing pairs of (azimuth, length in meters), using a 10 degree bin sector width, on a circle of radius = 3 inch, grid going out to radius = 150 km in steps of 25 km with a 30 degree sector interval, radial direction annotated every 50 km, using a light blue shading outlined by a solid red pen (width = 0.75 points), draw the mean azimuth, and shown in Portrait orientation, use:

```
gmt psrose fault_segments.az_r -R0/150/-90/90 -B50g25:"Fault
length":/g30:."Rose diagram":-S3i -Ar10 -Glightblue
-W0.75p,red -Z0.001 -C -P -T -: > half_rose.ps
```

To plot a full circle wind rose diagram of the data in the file `lines.r_az`, on a circle of radius = 5 cm, grid going out to radius = 500 units in steps of 100 with a 45 degree sector interval, using a solid pen (width = 0.5 point, and shown in landscape [Default] orientation with UNIX timestamp and command line plotted, use:

```
gmt psrose lines.az_r -R0/500/0/360 -S5c -Bg100/g45:."Windrose diagram": -W0.5p -Uc | lpr
```

1.70.7 Bugs

No default radial scale and grid settings for polar histograms. User must run **psrose -I** to find max length in binned data set.

1.70.8 See Also

gmt, gmt.conf, gmtcolors, pshistogram

1.71 psscale

`psscale` - Plot a gray or color scale-bar on maps

1.71.1 Synopsis

```
psscale -Dxpos/ypos/length/width[h] [ -A[alllc] ] [ -B[pls]parameters ] [ -Ccpt_file ] [ -E[blf][length][+n[text]] ] [ -Gzlo/zhi ] [ -I[max_intens/low_i/high_i] ] [ -Jparameters ] [ -JzZparameters ] [ -K ] [ -L[i][gap] ] [ -M ] [ -N[pldpi] ] [ -O ] [ -P ] [ -Q ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ -T[+ppen][+gfill][+lrlb|toff] ] [ -U[just/dx/dy][c|label] ] [ -V[level] ] [ -X[alc|fir][x-shift[u]] ] [ -Y[alc|fir][y-shift[u]] ] [ -Zzfile ] [ -ccopies ] [ -p[x|y|z]azim/levl[zlevel][+wlon0/lat0[z0]][+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.71.2 Description

psscale plots gray scales or color scales on maps. Both horizontal and vertical scales are supported. For *cpt_files* with gradational colors (i.e., the lower and upper boundary of an interval have different colors) **psscale** will interpolate to give a continuous scale. Variations in intensity due to shading/illumination may be displayed by setting the option **-I**. Colors may be spaced according to a linear scale, all be equal size, or by providing a file with individual tile widths. The font used for the annotations along the scale and optional units is specified by *FONT_ANNOT_PRIMARY*. If a label is requested, it is plotted with *FONT_LABEL*.

1.71.3 Required Arguments

-D*xpos/ypos/length/width***[h]** Defines the position of the center/top (for horizontal scale) or center/left (for vertical scale) and the dimensions of the scale. Give a negative length to reverse the scalebar. Append *h* to get a horizontal scale [Default is vertical].

1.71.4 Optional Arguments

-A**[alllc]** Place annotations and labels above (instead of below) horizontal scalebars and to the left (instead of the right) of vertical scalebars. Append **a** or **l** to move only the annotations or the label to the other side. Append **c** if you want to print a vertical label as a column of characters (does not work with special characters).

-B**[pls]***parameters* Set annotation, tick, and gridline interval for the colorbar. The x-axis label will plot beneath a horizontal bar (or vertically to the right of a vertical bar), except when using **-A**. As an option, use the y-axis label to plot the data unit to the right of a horizontal bar (and above a vertical bar). When using **-Ba** or **-Baf** annotation and/or minor tick intervals are chosen automatically. If **-B** is omitted, or no annotation intervals are provided, the default is to annotate every color level based on the numerical entries in the *cpt* file (which may be overridden by ULB flags in the *cpt* file). To specify custom text annotations for intervals, you must append *;annotation* to each *z*-slice in the *cpt* file.

-C*cpt_file* *cpt_file* is the color palette file to be used. By default all color changes are annotated. To use a subset, add an extra column to the *cpt*-file with a L, U, or B to annotate Lower, Upper, or Both color segment boundaries (but see **-B**). If not given, **psscale** will read *stdin*. Like **grdview**, **psscale** can understand pattern specifications in the *cpt* file. For CPT files where the *z* range is in meters, it may be useful to change to another unit when plotting. To do so, append **+Unit** to the file name. Likewise, if the CPT file uses another unit than meter and you wish to plot the CPT versus meters, append **+unit**.

- E[**blf**][*length*][+**n**][*text*] Add sidebar triangles for **back**- and/or **foreground** colors. Add **f** or **b** for only one sidebar triangle [Default gives both]. Optionally, append triangle height [Default is half the barwidth]. Finally, you can plot a rectangle with the NaN color at the start of the bar, labeled with *text* [NaN].
- G*zlo/zhi* Truncate the incoming CPT so that the lowest and highest z-levels are to *zlo* and *zhi*. If one of these equal NaN then we leave that end of the CPT alone. The truncation takes place before the plotting.
- I[*max_intens*][*low_i/high_i*] Add illumination effects. Optionally, set the range of intensities from - to + *max_intens*. If not specified, 1 is used. Alternatively, append *low/high* intensities to specify an asymmetric range [Default is no illumination].
- J*parameters (more ...)* Select map projection.
- J*Zparameters (more ...)* Set z-axis scaling; same syntax as -J*x*.
- K (*more ...*) Do not finalize the PostScript plot.
- L[**i**][*gap*] Gives equal-sized color rectangles. Default scales rectangles according to the z-range in the cpt-file (Also see -Z). If set, any equal interval annotation set with -B will be ignored. If *gap* is appended and the cpt table is discrete we will center each annotation on each rectangle, using the lower boundary z-value for the annotation. If **i** is prepended we annotate the interval range instead. If -I is used then each rectangle will have its constant color modified by the specified intensity.
- M Force a monochrome graybar using the (television) YIQ transformation.
- N[**pldpi**] Controls how the color scale is represented by the PostScript language. To preferentially draw color rectangles (e.g., for discrete colors), append **p**. Otherwise we will preferentially draw images (e.g., for continuous colors). Optionally append effective dots-per-inch for rasterization of color scales [600].
- O (*more ...*) Append to existing PostScript plot.
- P (*more ...*) Select "Portrait" plot orientation.
- Q Select logarithmic scale and power of ten annotations. All z-values in the cpt file will be converted to $p = \log_{10}(z)$ and only integer p values will be annotated using the 10^p format [Default is linear scale].
- R[*unit*][*west/east/south/north*][*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands -R**g** and -R**d** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the -R settings (and grid spacing, if applicable) are copied from the grid. Using -R*unit* expects projected (Cartesian) coordinates compatible with chosen -J and we inversely project to determine actual rectangular geographic region. For perspective view (-**p**), optionally append *zmin/zmax*. In case of perspective view (-**p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the -J**z** option, not when using only the -**p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

For perspective view **p**, optionally append *zmin/zmax*. (*more ...*)

- S Do not separate different color intervals with black grid lines.
- T[+**ppen**][+**gfill**][+**llr|bltoff**] Place a rectangle as background to the color scale. You must specify either a pen outline (with modifier +**ppen**) or a fill (with modifier +**gfill**), or both. The size of the rectangle is computed from -D and the current fontsize and offset parameters. You can add (or

remove) additional space on any side by appending *+soff*, where *s* is one of **left**, **right**, **bottom**, or **top**.

- U[*just/dx/dy/*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*) Select verbosity level [*c*].
- X[*alclflr*][*x-shift*][*u*]
- Y[*alclflr*][*y-shift*][*u*] (*more ...*) Shift plot origin.
- Z*zfile* File with colorbar-width per color entry. By default, width of entry is scaled to color range, i.e., *z* = 0-100 gives twice the width as *z* = 100-150 (Also see **-L**).
- ccopies* (*more ...*) Specify number of plot copies [Default is 1].
- p[*xlylz*]*azim/lev*[*zlevel*][*+wlon0/lat0*][*z0*][*+vx0/y0*] (*more ...*) Select perspective view. (Required **-R** and **-J** for proper functioning).
- t[*transp*] (*more ...*) Set PDF transparency level in percent.
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- + or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.71.5 Examples

To append a vertical color scale (7.5 cm long; 1.25 cm wide) to the right of a plot that is 6 inch wide and 4 inch high, using illumination, and show back- and foreground colors, and annotating every 5 units, use

```
gmt psscale -D6.5i/2i/7.5c/1.25c -O -Ccolors.cpt -I -E -B5:BATHYMETRY:/:m: > map.ps
```

1.71.6 Notes

When the *cpt* file is discrete and no illumination is specified, the color bar will be painted using polygons. For all other cases we must paint with an image. Some color printers may give slightly different colors for the two methods given identical RGB values.

1.71.7 See Also

gmt, *makecpt*, *grd2cpt*

1.72 pstext

pstext - Plot or typeset text on maps

1.72.1 Synopsis

```
pstext [ textfiles ] -Jparameters -Rwest/east/south/north[lzmin/lzmax][r] [ -A ] -B[pls]parameters
[ -D[j|J]dx/dy][v]pen] ] [ -F[+a]angle][+c]justify][+f]font][+h][+j]justify][+l] ] [ -Gcolor ] [ -
Jz/Zparameters ] [ -K ] [ -L ] [ -M ] [ -N ] [ -O ] [ -P ] [ -Qlu ] [ -TlOlC ] [ [ -Wpen ] [
-X[alcflr][x-shift][u] ] [ -Y[alcflr][y-shift][u] ] [ -U[just/dx/dy][cllabel] ] [ -Z ] [ -acol=name[...] ] [
-ccopies ] [ -filo]colinfo ] [ -hilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][ooffset][...] ] [
-p[x|y|z]azim/lev[zlevel][+wlon0/lat0][z0][+vx0/y0] ] [ -t[transp] ] [ -ocols[...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.72.2 Description

pstext plots text strings of variable size, font type, and orientation. Various map projections are provided, with the option to draw and annotate the map boundaries. PostScript code is written to standard output. Greek characters, subscript, superscript, and small caps are supported as follows: The sequence @~ toggles between the selected font and Greek (Symbol). @%no% sets the font to *no*; @%% resets the font to the starting font, @- toggles subscripts on/off, @+ toggles superscript on/off, @# toggles small caps on/off, @;color; changes the font color (@; resets it), @:size: changes the font size (@:: resets it), and @_ toggles underline on/off. @@ prints the @ sign. @e, @o, @a, @E, @O, @A give the accented Scandinavian characters. Composite characters (overstrike) may be indicated with the @!<char1><char2> sequence, which will print the two characters on top of each other. To learn the octal codes for symbols not available on the keyboard and some accented European characters, see Section 4.16 and Appendix F in the GMT Technical Reference and Cookbook. Note that *PS_CHAR_ENCODING* must be set to an extended character set in your *gmt.conf* file in order to use the accented characters. Using the **-G** or **-W** options, a rectangle underlying the text may be plotted (does not work for strings with sub/super scripts, symbols, or composite characters, except in paragraph mode (**-M**)).

1.72.3 Required Arguments

-J*parameters* (*more ...*) Select map projection.

-R[*unit*]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append *lzmin/lzmax*. (*more ...*)

1.72.4 Optional Arguments

textfiles This is one or more files containing 1 or more records with (*x*, *y*[, *font*, *angle*, *justify*], *text*). The attributes in brackets can alternatively be set directly via **-F**. If no files are given, **pstext** will read standard input. *font* is a font specification with format [*size*,][*font*,][*color*] where *size* is text size in points, *font* is the font to use, and *color* sets the font color. To draw outline fonts you append =*pen* to the font specification. The *angle* is measured in degrees counter-clockwise from horizontal, and *justify* sets the alignment. If *font* is not an integer, then it is taken to be a text string with the desired fontname (see **-L** for available fonts). The alignment refers to the part of the text string that will be mapped onto the (*x*,*y*) point. Choose a 2 character combination of L, C, R (for left, center, or right) and T, M, B for top, middle, or bottom. e.g., BL for lower left.

-A Angles are given as azimuths; convert them to directions using the current projection.

-B[*pls*]*parameters* (*more ...*) Set map boundary intervals.

- Cdx/dy** Sets the clearance between the text and the surrounding box [15%]. Only used if **-W** or **-G** are specified. Append the unit you want (**cm**, **inch**, or **point**; if not given we consult *PROJ_LENGTH_UNIT*) or **%** for a percentage of the font size.
- D[jlJ]dx/dy[v[pen]]** Offsets the text from the projected (*x,y*) point by *dx,dy* [0/0]. If *dy* is not specified then it is set equal to *dx*. Use **-Dj** to offset the text away from the point instead (i.e., the text justification will determine the direction of the shift). Using **-DJ** will shorten diagonal offsets at corners by $\sqrt{2}$. Optionally, append **v** which will draw a line from the original point to the shifted point; append a *pen* to change the attributes for this line.
- F[+a[angle]][+c[justify]][+f[font]][+h][+j[justify]][+l]** By default, text will be placed horizontally, using the primary annotation font attributes (*FONT_ANNOT_PRIMARY*), and centered on the data point. Use this option to override these defaults by specifying up to three text attributes (font, angle, and justification) directly on the command line. Use **+f** to set the font (size,fontname,color); if no font info is given then the input file must have this information in one of its columns. Use **+a** to set the angle; if no angle is given then the input file must have this as a column. Use **+j** to set the justification; if no justification is given then the input file must have this as a column. Items read from the data file should be in the same order as specified with the **-F** option. Example: **-F+f12p,Helvetica-Bold,red+j+a** selects a 12p red Helvetica-Bold font and expects to read the justification and angle from the file, in that order, after *x, y* and before *text*. In addition, the **+c** justification lets us use *x,y* coordinates extracted from the **-R** string instead of providing them in the input file. For example **-F+cTL** gets the *x_min, y_max* from the **-R** string and plots the text at the Upper Left corner of the map. Normally, the text to be plotted comes from the data record. Instead, use **+h** or **+l** to select the text as the most recent segment header or segment label, respectively in a multisegment input file.
- Gcolor** Sets the shade or color used for filling the text box [Default is no fill]. Alternatively, use **-Gc** to plot the text and then use the text dimensions (and **-C**) to build clip paths and turn clipping on. This clipping can then be turned off later with **psclip -C**. To **not** plot the text but activate clipping, use **-GC** instead.
- Jz/Zparameters (more ...)** Set z-axis scaling; same syntax as **-Jx**.
- K (more ...)** Do not finalize the PostScript plot.
- L** Lists the font-numbers and font-names available, then exits.
- M** Paragraph mode. Files must be multiple segment files. Segments are separated by a special record whose first character must be *flag* [Default is **>**]. Starting in the 3rd column, we expect to find information pertaining to the typesetting of a text paragraph (the remaining lines until next segment header). The information expected is (*x y [font angle justify] linespace parwidth parjust*), where *x y font angle justify* are defined above (*font, angle, and justify* can be set via **-F**), while *linespace* and *parwidth* are the linespacing and paragraph width, respectively. The justification of the text paragraph is governed by *parjust* which may be **l**(eft), **c**(enter), **r**(ight), or **j**(ustified). The segment header is followed by one or more lines with paragraph text. Text may contain the escape sequences discussed above. Separate paragraphs with a blank line. Note that here, the justification set via **-F+j** applies to the box alignment since the text justification is set by *parjust*.
- N** Do NOT clip text at map boundaries [Default will clip].
- O (more ...)** Append to existing PostScript plot.
- P (more ...)** Select "Portrait" plot orientation.
- Q** Change all text to either **lower** or **upper** case [Default leaves all text as is].
- T** Specify the shape of the textbox when using **-G** and/or **-W**. Choose lower case **o** to get a straight

rectangle [Default]. Choose upper case **O** to get a rounded rectangle. In paragraph mode (**-M**) you can also choose lower case **c** to get a concave rectangle or upper case **C** to get a convex rectangle.

- U***[just/dx/dy/][clabel] (more ...)* Draw GMT time stamp logo on plot.
- V***[level] (more ...)* Select verbosity level [c].
- Wpen** Sets the pen used to draw a rectangle around the text string (see **-T**) [Default is width = default, color = black, style = solid].
- X***[alclflr][x-shift[u]]*
- Y***[alclflr][y-shift[u]] (more ...)* Shift plot origin.
- Z** For 3-D projections: expect each item to have its own level given in the 3rd column, and **-N** is implicitly set. (Not implemented for paragraph mode).
- acol=name[...]** *(more ...)* Set aspatial column associations *col=name*.
- ccopies** *(more ...)* Specify number of plot copies [Default is 1].
- f***[ilo]colinfo (more ...)* Specify data types of input and/or output columns.
- h***[ilo][n][+c][+d][+rremark][+rtitle] (more ...)* Skip or produce header record(s).
- :***[ilo] (more ...)* Swap 1st and 2nd column on input and/or output.
- p***[x|y|z]azim/lev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] (more ...)* Select perspective view. (Not implemented for paragraph mode).
- t***[transp] (more ...)* Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.72.5 Examples

To plot just the red outlines of the (lon at text strings) stored in the file text.d on a Mercator plot with the given specifications, use

```
gmt pstext text.d -R-30/30/-10/20 -Jm0.1i -P -F+f18p,Helvetica,==0.5p,red -B5 > plot.ps
```

To plot a text at the upper left corner of a 10 cm map

```
echo TopLeft | gmt pstext -R1/10/1/10 -JX10 -F+cTL -P > plot.ps
```

To add a typeset figure caption for a 3-inch wide illustration, use

```
gmt pstext -R0/3/0/5 -JX3i -O -H -M -N << EOF >> figure.ps
```

This is an optional header record

```
> 0 -0.5 12 0 4 LT 13p 3i j
@%5%Figure 1.%% This illustration shows nothing useful, but it still needs
a figure caption. Highlighted in @;255/0/0;red@;; you can see the locations
of cities where it is @\_impossible@\_ to get any good Thai food; these are to be avoided.
EOF
```

1.72.6 Windows Remarks

Note that under Windows, the percent sign (%) is a variable indicator (like \$ under Unix). To indicate a plain percentage sign in a batch script you need to repeat it (%%); hence the font switching mechanism (@%*font*% and @%%) may require twice the number of percent signs. This only applies to text inside a script or that otherwise is processed by DOS. Data files that are opened and read by **ps_{text}** do not need such duplication.

1.72.7 Limitations

In paragraph mode, the presence of composite characters and other escape sequences may lead to unfortunate word splitting. Also, if a font is requested with an outline pen it will not be used in paragraph mode. Note if any single word is wider than your chosen paragraph width then the paragraph width is automatically enlarged to fit the widest word.

The **-N** option does not adjust the BoundingBox information so you may have to post-process the PostScript output with **ps2raster** to obtain a correct BoundingBox.

1.72.8 See Also

gmt, gmt.conf, gmtcolors, ps2raster, psbasemap, pslegend, psxy

1.73 pswiggle

pswiggle - Plot $z = f(x,y)$ anomalies along tracks

1.73.1 Synopsis

```
pswiggle [ table ] -Jparameters -Rwest/east/south/north[/zmin/zmax][r] [ -Zscale [ -Aazimuth ] [ -B[pls]parameters ] [ -Ccenter ] [ -G[+|-|=fill] ] [ -Jz/Zparameters ] [ -Ifix_az ] [ -K ] [ -O ] [ -P ] [ -S[x]/lon0/lat0/length[/units] ] [ -Tpen ] [ -U[just/dx/dy]/[c]label ] [ -V[level] ] [ -Wpen ] [ [ -X[alcflr]/[x-shift][u] ] [ -Y[alcflr]/[y-shift][u] ] [ -bi[ncols]/[type][w][+L]+B ] [ -ccopies ] [ -f[ilo]/colinfo ] [ -g[a]/x|y|d][X|Y|D]/[col][z][+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I]/[sscale][ooffset][,...] ] [ -p[x|y|z]/azim/elev[/zlevel][+w]/lon0/lat0[/z0]][+vx0/y0] ] [ -t[transp] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.73.2 Description

pswiggle reads (x,y,z) triplets from files [or standard input] and plots z as a function of distance along track. This means that two consecutive (x,y) points define the local distance axis, and the local z axis

is then perpendicular to the distance axis. The user may set a preferred positive anomaly plot direction, and if the positive normal is outside the plus/minus 90 degree window around the preferred direction, then 180 degrees are added to the direction. Either the positive or the negative wiggle may be shaded. The resulting PostScript code is written to standard output.

1.73.3 Required Arguments

- Jparameters** (*more ...*) Select map projection.
 - R[unit]xmin/xmax/ymin/ymax[r]** (*more ...*) Specify the region of interest.
- For perspective view **p**, optionally append */zmin/zmax*. (*more ...*)
- Zscale** Gives anomaly scale in data-units/distance-unit.

1.73.4 Optional Arguments

- table** One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.
- Aazimuth** Sets the preferred positive azimuth. Positive wiggles will “gravitate” towards that direction.
- B[pls]parameters** (*more ...*) Set map boundary intervals.
- Ccenter** Subtract *center* from the data set before plotting [0].
- G[+|-|=]fill** Set fill shade, color or pattern for positive and/or negative wiggles [Default is no fill].
Optionally, prepend **+** to fill positive areas (this is the default behavior). Prepend **-** to fill negative areas. Prepend **=** to fill both positive and negative areas with the same fill.
- Ifix_az** Set a fixed azimuth projection for wiggles [Default uses track azimuth, but see **-A**].
- Jz/Zparameters** (*more ...*) Set z-axis scaling; same syntax as **-Jx**.
- K** (*more ...*) Do not finalize the PostScript plot.
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- S[x]lon0/lat0/length[units]** Draws a simple vertical scale centered on *lon0/lat0*. Use **-Sx** to specify cartesian coordinates instead. *length* is in z units, append unit name for labeling. **FONT_ANNOT_PRIMARY** is used as font.
- Tpen** Draw track [Default is no track]. Append pen attributes to use [Defaults: width = 0.25p, color = black, style = solid].
- U[just/dx/dy]/[clabel]** (*more ...*) Draw GMT time stamp logo on plot.
- V[level]** (*more ...*) Select verbosity level [c].
- Wpen**
- bi[ncols][type]** (*more ...*) Select binary input. [Default is 3 input columns].
- ccopies** (*more ...*) Specify number of plot copies [Default is 1].
- f[ilo]colinfo** (*more ...*) Specify data types of input and/or output columns.
- g[a]x|y|d|X|Y|D|[col]z[+|-]gap[u]** (*more ...*) Determine data gaps and line breaks.

- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
- icols[1][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
- p[x|y|z]azim/lev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0]** (*more ...*) Select perspective view.
- t[transp]** (*more ...*) Set PDF transparency level in percent.
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.73.5 Examples

To plot the magnetic anomaly stored in the file `track.xym` along track @ 1000 nTesla/cm (after removing a mean value of 32000 nTesla), using a 15-cm-wide Polar Stereographic map ticked every 5 degrees in Portrait mode, with positive anomalies in red on a blue track of width 0.25 points, use

```
gmt pswiggle track.xym -R-20/10/-80/-60 -JS0/90/15c -Z1000 -B5 \
-C32000 -P -Gred -T0.25p,blue -S1000 -V > track_xym.ps
```

1.73.6 Bugs

Sometimes the (x,y) coordinates are not printed with enough significant digits, so the local perpendicular to the track swings around a lot. To see if this is the problem, you should do this:

```
awk '{ if (NR > 1) print atan2(y-$1, x-$2); y=$1; x=$2; }' yourdata.xyz | more
```

(note that output is in radians; on some machines you need “nawk” to do this). Then if these numbers jump around a lot, you may do this:

```
awk '{ print NR, $0 }' yourdata.xyz | filter1d -Fb5 -N4/0 \
--FORMAT_FLOAT_OUT=%.12g > smoothed.xyz
```

and plot this data set instead.

1.73.7 See Also

gmt, *gmtcolors*, *filter1d*, *psbasemap*, *splitxyz*

1.74 psxy

psxy - Plot lines, polygons, and symbols on maps

1.74.1 Synopsis

```
psxy [ table ] -Jparameters -Rwest/east/south/north[/zmin/zmax][r] [ -A[m|p] ] [ -B[pls]parameters ] [
-Ccptfile ] [ -Ddx/dy ] [ -E[x|y|X|Y][n][cap][/-|+][pen] ] [ -Gfill ] [ -Iintens ] [ -Jz[Zparameters] ] [ -K
] [ -L ] [ -N ] [ -O ] [ -P ] [ -S[symbol][size[u] ] ] [ -U[just/dx/dy][cllabel] ] [ -V[level] ] [ -W[-|+][pen]
] [ -X[alc|fr][x-shift[u] ] ] [ -Y[alc|fr][y-shift[u] ] ] [ -acol=name[...] ] [ -bi[ncols][type][w][+L|+B] ] [
-ccopies ] [ -f[i|o]colinfo ] [ -g[a|x|y|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[i|o][n][+c][+d][+rremark][+rtitle]
] [ -i[cols][l][sscale][offset][,...] ] [ -p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0] ] [ -t[transp] ]
[ -:i|o ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.74.2 Description

psxy reads (x,y) pairs from *files* [or standard input] and generates PostScript code that will plot lines, polygons, or symbols at those locations on a map. If a symbol is selected and no symbol size given, then **psxy** will interpret the third column of the input data as symbol size. Symbols whose *size* is ≤ 0 are skipped. If no symbols are specified then the symbol code (see **-S** below) must be present as last column in the input. If **-S** is not used, a line connecting the data points will be drawn instead. To explicitly close polygons, use **-L**. Select a fill with **-G**. If **-G** is set, **-W** will control whether the polygon outline is drawn or not. If a symbol is selected, **-G** and **-W** determines the fill and outline/no outline, respectively. The PostScript code is written to standard output.

1.74.3 Required Arguments

-Jparameters (*more ...*) Select map projection.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append */zmin/zmax*. (*more ...*)

1.74.4 Optional Arguments

table One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input. Use **-T** to ignore all input files, including standard input (see below).

-A[m|p] By default line segments are drawn as great circle arcs. To draw them as straight lines, use the **-A** flag. Alternatively, add **m** to draw the line by first following a meridian, then a parallel. Or append **p** to start following a parallel, then a meridian. (This can be practical to draw a line along parallels, for example).

-B[pls]parameters (*more ...*) Set map boundary intervals.

-Ccptfile Give a color palette file. If **-S** is set, let symbol fill color be determined by the z-value in the third column. Additional fields are shifted over by one column (optional size would be 4th rather than 3rd field, etc.). If **-S** is not set, then **psxy** expects the user to supply a multisegment file where each segment header contains a **-Zval** string. The *val* will control the color of the line or polygon (if **-L** is set) via the cpt file.

-Ddx/dy Offset the plot symbol or line locations by the given amounts *dx/dy* [Default is no offset]. If *dy* is not given it is set equal to *dx*.

- E**[x|y|X|Y][n][cap][[-|+]*pen*] Draw error bars. Append *x* and/or *y* to indicate which bars you want to draw (Default is both *x* and *y*). The *x* and/or *y* errors must be stored in the columns after the (*x*,*y*) pair [or (*x*,*y*,*size*) triplet]. The *cap* parameter indicates the length of the end-cap on the error bars [7p]. Pen attributes for error bars may also be set [Defaults: width = default, color = black, style = solid]. A leading + will use the lookup color (via **-C**) for both symbol fill and error pen color, while a leading - will set error pen color and turn off symbol fill. If upper case **X** and/or **Y** is used we will instead draw “box-and-whisker” (or “stem-and-leaf”) symbols. The *x* (or *y*) coordinate is then taken as the median value, and 4 more columns are expected to contain the minimum (0% quantile), the 25% quantile, the 75% quantile, and the maximum (100% quantile) values. The 25-75% box may be filled by using **-G**. If *n* is appended to **X** (or **Y**) we draw a notched “box-and-whisker” symbol where the notch width reflects the uncertainty in the median. Then a 5th extra data column is expected to contain the number of points in the distribution.
- Gfill** Select color or pattern for filling of symbols or polygons [Default is no fill]. Note that **psxy** will search for **-G** and **-W** strings in all the segment headers and let any values thus found over-ride the command line settings.
- Iintens** Use the supplied *intens* value (nominally in the -1 to + 1 range) to modulate the fill color by simulating illumination [none].
- Jz/Zparameters (more ...)** Set z-axis scaling; same syntax as **-Jx**.
- K (more ...)** Do not finalize the PostScript plot.
- L** Force closed polygons: connect the endpoints of the line-segment(s) and draw polygons. Also, in concert with **-C** and any **-Z** settings in the headers will use the implied color for polygon fill [Default is polygon pen color].
- N** Do NOT skip symbols that fall outside map border [Default plots points inside border only]. The option does not apply to lines and polygons which are always clipped to the map region.
- O (more ...)** Append to existing PostScript plot.
- P (more ...)** Select “Portrait” plot orientation.
- S[symbol][size[u]]** Plot symbols. If present, *size* is symbol size in the unit set in *gmt.conf* (unless **c**, **i**, or **p** is appended). If the symbol code (see below) is not given it will be read from the last column in the input data; this cannot be used in conjunction with binary input. Optionally, append **c**, **i**, or **p** to indicate that the size information in the input data is in units of cm, inch, or point, respectively [Default is *PROJ_LENGTH_UNIT*]. Note: if you provide *both* size and symbol via the input file you must use *PROJ_LENGTH_UNIT* to indicate the unit used for the symbol size or append the units to the sizes in the file. If symbol sizes are expected via the third data column then you may convert those values to suitable symbol sizes via the **-i** mechanism.

The uppercase symbols **A**, **C**, **D**, **G**, **H**, **I**, **N**, **S**, **T** are normalized to have the same area as a circle with diameter *size*, while the size of the corresponding lowercase symbols refers to the diameter of a circumscribed circle.

You can change symbols by adding the required **-S** option to any of your multisegment headers.

Choose between these symbol codes:

- S-** x-dash (-). *size* is the length of a short horizontal (x-dir) line segment.
- S+** plus (+). *size* is diameter of circumscribing circle.
- Sa** star. *size* is diameter of circumscribing circle.

- Sb***[size[clilplu]][b[base]]* Vertical bar extending from *base* to *y*. *size* is bar width. Append **u** if *size* is in x-units [Default is plot-distance units]. By default, *base* = *ymin*. Append **b[base]** to change this value. If *base* is not appended then we read it from the last input data column.
- SB***[size[clilplu]][b[base]]* Horizontal bar extending from *base* to *x*. *size* is bar width. Append **u** if *size* is in y-units [Default is plot-distance units]. By default, *base* = *xmin*. Append **b[base]** to change this value. If *base* is not appended then we read it from the last input data column.
- Sc** circle. *size* is diameter of circle.
- Sd** diamond. *size* is diameter of circumscribing circle.
- Se** ellipse. Direction (in degrees counter-clockwise from horizontal), *major_axis*, and *minor_axis* must be found in columns 3, 4, and 5.
- SE** Same as **-Se**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Se** leaves the directions unchanged.) Furthermore, the axes lengths must be given in km instead of plot-distance units. An exception occurs for a linear projection in which we assume the ellipse axes are given in the same units as **-R**. For degenerate ellipses (circles) with diameter given in km, use **-SE-**.
- Sfgap***[/size][+ll+r][+b+c+f+s+t][+offset]*. Draw a front. Supply distance gap between symbols and symbol size. If *gap* is negative, it is interpreted to mean the number of symbols along the front instead. If *size* is missing it is set to 30% of the *gap*, except when *gap* is negative and *size* is thus required. Append **+l** or **+r** to plot symbols on the left or right side of the front [Default is centered]. Append **+type** to specify which symbol to plot: **box**, **circle**, **fault**, **slip**, or **triangle**. [Default is **fault**]. Slip means left-lateral or right-lateral strike-slip arrows (centered is not an option). Append **+offset** to offset the first symbol from the beginning of the front by that amount [0]. Note: By placing **-Sf** options in the segment header you can change the front types on a segment-by-segment basis.
- Sg** octagon. *size* is diameter of circumscribing circle.
- Sh** hexagon. *size* is diameter of circumscribing circle.
- Si** inverted triangle. *size* is diameter of circumscribing circle.
- Sj** Rotated rectangle. Direction (in degrees counter-clockwise from horizontal), *x-dimension*, and *y-dimension* must be found in columns 3, 4, and 5.
- SJ** Same as **-Sj**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Sj** leaves the directions unchanged.) Furthermore, the dimensions must be given in km instead of plot-distance units. An exception occurs for a linear projection in which we assume the dimensions are given in the same units as **-R**.
- Sk** kustom symbol. Append *name/size*, and we will look for a definition file called *name.def* in (1) the current directory or (2) in *~/gmt* or (3) in **\$GMT_SHAREDIR/custom**. The symbol as defined in that file is of size 1.0 by default; the appended *size* will scale symbol accordingly. Users may add their own custom *.def files; see CUSTOM SYMBOLS below.
- Sl** letter or text string (less than 256 characters). Give *size*, and append **+tstring** after the size. Note that the size is only approximate; no individual scaling is done for different characters. Remember to escape special characters like *****. Optionally, you may append **+ffont** to select a particular font [Default is *FONT_ANNOT_PRIMARY*] and **+jjustify** to change justification [CM].

- Sm** math angle arc, optionally with one or two arrow heads [Default is no arrow heads]. The *size* is the length of the vector head. Arc width is set by **-W**. The radius of the arc and its start and stop directions (in degrees counter-clockwise from horizontal) must be given in columns 3-5. See VECTOR ATTRIBUTES for specifying other attributes.
- SM** Same as **-Sm** but switches to straight angle symbol if angles subtend 90 degrees exactly.
- Sn** pentagon. *size* is diameter of circumscribing circle.
- Sp** point. No size needs to be specified (1 pixel is used).
- Sq** quoted line, i.e., lines with annotations such as contours. Append **[d|D|f|l|L|n|x|X]info[:labelinfo]**. The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

ddist[clilp] or **Ddist[dleflklmlMlnls]** For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **f** (foot), **k** (km), **M** (mile), **n** (nautical mile) or **u** (US survey foot), and **d** (arc degree), **m** (arc minute), or **s** (arc second). [Default is 10**c** or 4**i**]. As an option, you can append */fraction* which is used to place the very first label for each contour when the cumulative along-contour distance equals *fraction * dist* [0.25].

ffile.d Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact matches and points outside the region are skipped.

llLine1[,line2,...] Give the coordinates of the end points for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the quoted lines. The format of each *line* specification is *start_lon/start_lat/stop_lon/stop_lat*. Both *start_lon/start_lat* and *stop_lon/stop_lat* can be replaced by a 2-character key that uses the justification format employed in **pstext** to indicate a point on the frame or center of the map, given as [LCR][BMT]. **L** will interpret the point pairs as defining great circles [Default is straight line].

n|Nn_label Specifies the number of equidistant labels for quoted lines line [1]. Upper case **N** starts labeling exactly at the start of the line [Default centers them along the line]. **N-1** places one justified label at start, while **N+1** places one justified label at the end of quoted lines. Optionally, append */min_dist[clilp]* to enforce that a minimum distance separation between successive labels is enforced.

x|Xxfile.d Reads the multisegment file *xfile.d* and places labels at the intersections between the quoted lines and the lines in *xfile.d*. **X** will resample the lines first along great-circle arcs. In addition, you may optionally append **+rradius[clilp]** to set a minimum label separation in the x-y plane [no limitation].

The optional *labelinfo* controls the specifics of the label formatting and consists of a concatenated string made up of any of the following control arguments:

- +aangle** For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default].
- +cdx[/dy]** Sets the clearance between label and optional text box. Append **clilp** to specify the unit or *%* to indicate a percentage of the label font size [15%].

- +d** Turns on debug which will draw helper points and lines to illustrate the workings of the quoted line setup.
- +e** Delay the plotting of the text. This is used to build a clip path based on the text, then lay down other overlays while that clip path is in effect, then turning of clipping with `psclip -Cs` which finally plots the original text.
- +ffont** Sets the desired font [Default `FONT_ANNOT_PRIMARY` with its size changed to 9p].
- +g[*color*]** Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is `PS_PAGE_COLOR`].
- +jjust** Sets label justification [Default is MC]. Ignored when `-SqNln+|-1` is used.
- +label** Sets the constant label text.
- +Lflag** Sets the label text according to the specified flag:
 - +Lh** Take the label from the current segment header (first scan for an embedded `-Llabel` option, if not use the first word following the segment flag). For multiple-word labels, enclose entire label in double quotes.
 - +Ld** Take the Cartesian plot distances along the line as the label; append `clip` as the unit [Default is `PROJ_LENGTH_UNIT`].
 - +LD** Calculate actual map distances; append `d|e|f|k|l|M|n|s` as the unit [Default is `d`(egrees), unless label placement was based on map distances along the lines in which case we use the same unit specified for that algorithm]. Requires a map projection to be used.
 - +Lf** Use text after the 2nd column in the fixed label location file as the label. Requires the fixed label location setting.
 - +Lx** As **+Lh** but use the headers in the `xfile.d` instead. Requires the crossing file option.
- +ndx[/*dy*]** Nudges the placement of labels by the specified amount (append `clip` to specify the units). Increments are considered in the coordinate system defined by the orientation of the line; use **+N** to force increments in the plot x/y coordinates system [no nudging]. Not allowed with **+v**.
- +o** Selects rounded rectangular text box [Default is rectangular]. Not applicable for curved text (**+v**) and only makes sense for opaque text boxes.
- +p[*pen*]** Draws the outline of text boxes [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, style = solid].
- +rmin_rad** Will not place labels where the line's radius of curvature is less than `min_rad` [Default is 0].
- +t[*file*]** Saves line label x, y, and text to `file` [Line_labels.txt]. Use **+T** to save x, y, angle, text instead.
- +uunit** Appends `unit` to all line labels. If `unit` starts with a leading hyphen (-) then there will be no space between label value and the unit. [Default is no unit].
- +v** Specifies curved labels following the path [Default is straight labels].
- +w** Specifies how many (x,y) points will be used to estimate label angles [Default is 10].
- +=prefix** Prepends `prefix` to all line labels. If `prefix` starts with a leading hyphen (-) then there will be no space between label value and the prefix. [Default is no prefix].

Note: By placing **-Sq** options in the segment header you can change the quoted text attributes on a segment-by-segment basis.

- Sr** rectangle. No size needs to be specified, but the x- and y-dimensions must be found in columns 3 and 4.
- SR** Rounded rectangle. No size needs to be specified, but the x- and y-dimensions and corner radius must be found in columns 3, 4, and 5.
- Ss** square. *size* is diameter of circumscribing circle.
- St** triangle. *size* is diameter of circumscribing circle.
- Sv** vector. Direction (in degrees counter-clockwise from horizontal) and length must be found in columns 3 and 4. The *size* is the length of the vector head. Vector width is set by **-W**. See VECTOR ATTRIBUTES for specifying other attributes.
- SV** Same as **-Sv**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Sv** leaves the directions unchanged.) See VECTOR ATTRIBUTES for specifying other attributes.
- Sw** pie wedge. Start and stop directions (in degrees counter-clockwise from horizontal) for pie slice must be found in columns 3 and 4.
- SW** Same as **-Sw**, except azimuths (in degrees east of north) should be given instead of the two directions. The azimuths will be mapped into angles based on the chosen map projection (**-Sw** leaves the directions unchanged.)
- Sx** cross (x). *size* is diameter of circumscribing circle.
- Sy** y-dash (l). *size* is the length of a short vertical (y-dir) line segment.
- S=** geovector. Azimuth (in degrees east from north) and length (in km) must be found in columns 3 and 4. The *size* is the length of the vector head. Vector width is set by **-W**. See VECTOR ATTRIBUTES for specifying attributes.
- T** Ignore all input files, including standard input. This is the same as specifying /dev/null (or NUL for Windows users) as input file. Use this to activate only the options that are not related to plotting of lines or symbols, such as **psxy -R -J -O -T** to terminate a sequence of GMT plotting commands without producing any plotting output.
- U[*just/dx/dy*][*cllabel*] (*more ...*)** Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*)** Select verbosity level [*c*].
- W[*-|+*][*pen*] (*more ...*)** Set pen attributes for lines or the outline of symbols [Defaults: width = default, color = black, style = solid]. A leading **+** will use the lookup color (via **-C**) for both symbol fill and outline pen color, while a leading **-** will set outline pen color and turn off symbol fill.
- X[*alclflr*][*x-shift*[*u*]]**
- Y[*alclflr*][*y-shift*[*u*]] (*more ...*)** Shift plot origin.
- bi[*ncols*][*type*] (*more ...*)** Select binary input. [Default is the required number of columns given the chosen settings].
- acol=name[...]** (*more ...*) Set aspatial column associations *col=name*.
- ccopies** (*more ...*) Specify number of plot copies [Default is 1].
- f[*ilo*]*colinfo*** (*more ...*) Specify data types of input and/or output columns.

- g[a]x|y|d|X|Y|D|[col]z[+|-]gap[u]** (*more ...*) Determine data gaps and line breaks. The **-g** option is ignored if **-S** is set.
- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
- icols[l][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- p[x|y|z]azim/lev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0]** (*more ...*) Select perspective view.
- t[transp]** (*more ...*) Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.74.5 Vector Attributes

Several modifiers may be appended to the vector-producing options to specify the placement of vector heads, their shapes, and the justification of the vector. Below, left and right refers to the side of the vector line when viewed from the start point to the end point of the segment:

- +aangle** sets the angle of the vector head apex [30].
- +b** places a vector head at the beginning of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +e** places a vector head at the end of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +g-|fill** turns off vector head fill (if **-**) or sets the vector head fill [Default fill is used, which may be no fill].
- +l** draws half-arrows, using only the left side of specified heads [both sides].
- +norm** scales down vector attributes (pen thickness, head size) with decreasing length, where vectors shorter than *norm* will have their attributes scaled by $\text{length}/\text{norm}$ [arrow attributes remains invariant to length].
- +oplon/plat** specifies the oblique pole for the great or small circles. Only needed for great circles if **+q** is given.
- +p[-][pen]** sets the vector pen attributes. If *pen* has a leading **-** then the head outline is not drawn. [Default pen is used, and head outline is drawn]
- +q** means the input *angle*, *length* data instead represent the *start* and *stop* opening angles of the arc segment relative to the given point.
- +r** draws half-arrows, using only the right side of specified heads [both sides].

In addition, all but circular vectors may take these modifiers:

+jjust determines how the input *x,y* point relates to the vector. Choose from **beginning** [default], **end**, or **center**.

+s means the input *angle, length* is instead the *x, y* coordinates of the vector end point.

1.74.6 Examples

To plot solid red circles (diameter = 0.25 cm) at the positions listed in the file `DSDP.xy` on a Mercator map at 5 cm/degree of the area 150E to 154E, 18N to 23N, with tickmarks every 1 degree and gridlines every 15 minutes, use

```
gmt psxy DSDP.xy R150/154/18/23 -Jm5c -Sc0.25c -Gred -Blg15m > map.ps
```

To plot the *xyz* values in the file `quakes.xyzm` as circles with size given by the magnitude in the 4th column and color based on the depth in the third using the color palette `cpt` on a linear map, use

```
gmt psxy quakes.xyzm -R0/1000/0/1000 -JX6i -Sc -Ccpt -B200 > map.ps
```

To plot the file `trench.xy` on a Mercator map, with white triangles with sides 0.25 inch on the left side of the line, spaced every 0.8 inch, use

```
gmt psxy trench.xy -R150/200/20/50 -Jm0.15i -Sf0.8i/0.1i+l+t -Gwhite -W -B10 > map.ps
```

To plot the data in the file `misc.d` as symbols determined by the code in the last column, and with size given by the magnitude in the 4th column, and color based on the third column via the color palette `cpt` on a linear map, use

```
gmt psxy misc.d -R0/100/-50/100 -JX6i -S -Ccpt -B20 > map.ps
```

1.74.7 Segment Header Parsing

Segment header records may contain one of more of the following options:

- Gfill** Use the new *fill* and turn filling on
- G-** Turn filling off
- G** Revert to default fill (none if not set on command line)
- Wpen** Use the new *pen* and turn outline on
- W** Revert to default pen `MAP_DEFAULT_PEN` (if not set on command line)
- W-** Turn outline off
- Zzval** Obtain fill via `cpt` lookup using *z-value* *zval*
- ZNaN** Get the NaN color from the `cpt` file

1.74.8 Custom Symbols

`psxy` allows users to define and plot their own custom symbols. This is done by encoding the symbol using our custom symbol macro code described in Appendix N. Put all the macro codes for your new symbol in a file whose extension must be `.def`; you may then address the symbol without giving the extension (e.g., the symbol file `tsunami.def` is used by specifying **-Sktsunami/size**). The definition file can contain any number of plot code records, as well as blank lines and comment lines (starting with

#). **psxy** will look for the definition files in (1) the current directory, (2) the `~/gmt` directory, and (3) the `$GMT_SHAREDIR/custom` directory, in that order. Freeform polygons (made up of straight line segments and arcs of circles) can be designed - these polygons can be painted and filled with a pattern. Other standard geometric symbols can also be used. See Appendix *App-custom_symbols* for macro definitions.

1.74.9 See Also

gmt, *gmt.conf*, *gmtcolors*, *psbasemap*, *psxyz*

1.75 psxyz

psxyz - Plot lines, polygons, and symbols in 3-D

1.75.1 Synopsis

```
psxyz [ table ] -Jparameters -Jz/Zparameters -Rwest/east/south/north[zmin/zmax][r] [ -B[pls]parameters ] [ -Ddx/dy/dz ] [ -Gfill ] [ -Iintens ] [ -K ] [ -L ] [ -N ] [ -O ] [ -P ] [ -Q ] [ -S[symbol][size[unit]][/size_y] ] [ -U[just/dx/dy]/[cllabel] ] [ -V[level] ] [ -W[-][+][pen] ] [ -X[alc][f][x-shift][u] ] [ -Y[alc][f][y-shift][u] ] [ -acol=name[...] ] [ -bi[ncols][type][w][+L][B] ] [ -ccopies ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+][-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,,...] ] [ -p[x|y|z]azim/elev[/zlevel][+wlon0/lat0][/z0]][+vx0/y0] ] [ -t[transp] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.75.2 Description

psxyz reads (x,y,z) triplets from *files* [or standard input] and generates PostScript code that will plot lines, polygons, or symbols at those locations in 3-D. If a symbol is selected and no symbol size given, then **psxyz** will interpret the fourth column of the input data as symbol size. Symbols whose *size* is ≤ 0 are skipped. If no symbols are specified then the symbol code (see **-S** below) must be present as last column in the input. If **-S** is not used, a line connecting the data points will be drawn instead. To explicitly close polygons, use **-L**. Select a fill with **-G**. If **-G** is set, **-W** will control whether the polygon outline is drawn or not. If a symbol is selected, **-G** and **-W** determines the fill and outline/no outline, respectively. The PostScript code is written to standard output.

1.75.3 Required Arguments

-J*parameters* (*more ...*) Select map projection.

-Jz/Z*parameters* (*more ...*) Set z-axis scaling; same syntax as **-Jx**.

-R[**unit**]*xmin/xmax/ymin/ymax*[**r**] (*more ...*) Specify the region of interest.

For perspective view **p**, optionally append *zmin/zmax*. (*more ...*)

1.75.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-B[*pls*]*parameters* (*more ...*) Set map boundary intervals.

-C*cptfile* Give a color palette file. If **-S** is set, let symbol fill color be determined by the t-value in the fourth column. Additional fields are shifted over by one column (optional size would be in 5th rather than 4th field, etc.). If **-S** is not set, then **psxyz** expects the user to supply a multisegment file (where each segment header contains a **-Z***val* string. The *val* will control the color of the line or polygon (if **-L** is set) via the *cpt* file.

-D*dx/dy/dz* Offset the plot symbol or line locations by the given amounts *dx/dy/dz* [Default is no offset].

-G*fill* Select color or pattern for filling of symbols or polygons [Default is no fill]. Note that **psxyz** will search for **-G** and **-W** strings in all the segment headers and let any values thus found over-ride the command line settings.

-I*intens* Use the supplied *intens* value (nominally in the -1 to + 1 range) to modulate the fill color by simulating illumination [none].

-K (*more ...*) Do not finalize the PostScript plot.

-L Force closed polygons: connect the endpoints of the line-segment(s) and draw polygons. Also, in concert with **-C** and any **-Z** settings in the headers will use the implied color for polygon fill [Default is polygon pen color]. **-N** Do NOT skip symbols that fall outside map border [Default plots points inside border only].

-O (*more ...*) Append to existing PostScript plot.

-P (*more ...*) Select "Portrait" plot orientation.

-Q Turn off the automatic sorting of items based on their distance from the viewer. The default is to sort the items so that items in the foreground are plotted after items in the background.

-S[*symbol*][*size*[*u*]][/*size_y*] Plot symbols. If present, *size* is symbol size in the unit set in *gmt.conf* (unless **c**, **i**, or **p** is appended). If the symbol code (see below) is not given it will be read from the last column in the input data; this cannot be used in conjunction with binary input. Optionally, append **c**, **i**, or **p** to indicate that the size information in the input data is in units of cm, inch, or point, respectively [Default is **PROJ_LENGTH_UNIT**]. Note: if you give *both* size and symbol via the input file you must use **PROJ_LENGTH_UNIT** to indicate the units used for the symbol size or append the units to the size in the file. Some 2-dimensional symbols optionally take a second size via *size_y*. If symbol sizes are expected via the fourth data column then you may convert those values to suitable symbol sizes via the **-i** mechanism.

The uppercase symbols **A**, **C**, **D**, **G**, **H**, **I**, **N**, **S**, **T** are normalized to have the same area as a circle with diameter *size*, while the size of the corresponding lowercase symbols refers to the diameter of a circumscribed circle.

You can change symbols by adding the required **-S** option to any of your multisegment headers.

Choose between these symbol codes:

-S- x-dash (-). *size* is the length of a short horizontal (x-dir) line segment.

-S+ plus (+). *size* is diameter of circumscribing circle.

-Sa star. *size* is diameter of circumscribing circle.

- Sb** Vertical bar extending from *base* to *y*. *size* is bar width. Append **u** if *size* is in x-units [Default is plot-distance units]. By default, *base* = *ymin*. Append **b[base]** to change this value. If *base* is not appended then we read it from the last input data column.
- SB** Horizontal bar extending from *base* to *x*. *size* is bar width. Append **u** if *size* is in y-units [Default is plot-distance units]. By default, *base* = *xmin*. Append **b[base]** to change this value. If *base* is not appended then we read it from the last input data column.
- Sc** circle. *size* is diameter of circle.
- Sd** diamond. *size* is diameter of circumscribing circle.
- Se** ellipse. Direction (in degrees counter-clockwise from horizontal), *major_axis*, and *minor_axis* must be found in columns 4, 5, and 6.
- SE** Same as **-Se**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Se** leaves the directions unchanged.) Furthermore, the axes lengths must be given in km instead of plot-distance units. An exception occurs for a linear projection in which we assume the ellipse axes are given in the same units as **-R**. For degenerate ellipses (circles) with diameter given in km, use **-SE-**.
- Sf** front. **-Sfgap[/size][+l+r][+b+c+f+s+t][+offset]**. Supply distance gap between symbols and symbol size. If *gap* is negative, it is interpreted to mean the number of symbols along the front instead. If *size* is missing it is set to 30% of the *gap*, except when *gap* is negative and *size* is thus required. Append **+l** or **+r** to plot symbols on the left or right side of the front [Default is centered]. Append **+type** to specify which symbol to plot: **box**, **circle**, **fault**, **slip**, or **triangle**. [Default is **fault**]. Slip means left-lateral or right-lateral strike-slip arrows (centered is not an option). Append **+offset** to offset the first symbol from the beginning of the front by that amount [0]. Note: By placing **-Sf** options in the segment header you can change the front types on a segment-by-segment basis.
- Sg** octagon. *size* is diameter of circumscribing circle.
- Sh** hexagon. *size* is diameter of circumscribing circle.
- Si** inverted triangle. *size* is diameter of circumscribing circle.
- Sj** Rotated rectangle. Direction (in degrees counter-clockwise from horizontal), *x-dimension*, and *y-dimension* must be found in columns 4, 5, and 6.
- SJ** Same as **-Sj**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Sj** leaves the directions unchanged.) Furthermore, the dimensions must be given in km instead of plot-distance units. An exception occurs for a linear projection in which we assume the dimensions are given in the same units as **-R**.
- Sk** kustom symbol. Append *<name>/size*, and we will look for a definition file called *<name>.def* in (1) the current directory or (2) in *~/gmt* or (3) in **\$GMT_SHAREDIR/custom**. The symbol as defined in that file is of size 1.0 by default; the appended *size* will scale symbol accordingly. The symbols are plotted in the *x-y* plane. Users may add their own custom *.def files; see CUSTOM SYMBOLS below.
- Sl** letter or text string (less than 64 characters). Give *size*, and append */string* after the size. Note that the size is only approximate; no individual scaling is done for different characters. Remember to escape special characters like *. Optionally, you may append *%font* to select a particular font [Default is **FONT_ANNOT_PRIMARY**].

- Sm** math angle arc, optionally with one or two arrow heads [Default is no arrow heads]. The *size* is the length of the vector head. Arc width is set by **-W**. The radius of the arc and its start and stop directions (in degrees counter-clockwise from horizontal) must be given in columns 4-6. See VECTOR ATTRIBUTES for specifying attributes.
- SM** Same as **-Sm** but switches to straight angle symbol if angles subtend 90 degrees exactly.
- Sn** pentagon. *size* is diameter of circumscribing circle.
- So** column (3-D) extending from *base* to *z*. The *size* sets base width (Use *xsize/ysize* if not the same). Append **u** if *size* is in *x*-units [Default is plot-distance units]. If no *size* is given we expect both *xsize* and *ysize* as two extra data columns. By default, *base* = 0. Append **bbase** to change this value. The facet colors will be modified to simulate shading. Use **-SO** to disable such 3-D illumination. If *base* is not appended then we read it from the last input data column.
- Sp** point. No size needs to be specified (1 pixel is used).
- Sq** quoted line, i.e., lines with annotations such as contours. It is assumed that each individual line has a constant *z* level (i.e., each line must lie in the *x-y* plane). Append [**d|D|f|l|L|n|n|x|X**]*info[:labelinfo]*. The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

ddist[clilp] or **Ddist[dleflklm|M|nls]** For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **f** (foot), **k** (km), **M** (mile), **n** (nautical mile) or **u** (US survey foot), and **d** (arc degree), **m** (arc minute), or **s** (arc second). [Default is 10**c** or 4**i**]. As an option, you can append */fraction* which is used to place the very first label for each contour when the cumulative along-contour distance equals *fraction * dist* [0.25]. **ffile.d** Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact matches and points outside the region are skipped. **llline1[,*line2*,...]** Give *start* and *stop* coordinates for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the quoted lines. The format of each *line* specification is *start/stop*, where *start* and *stop* are either a specified point *lon/lat* or a 2-character **XY** key that uses the justification format employed in **pstext** to indicate a point on the map, given as [LCR][BMT]. **L** will interpret the point pairs as defining great circles [Default is straight line]. **nn_label** Specifies the number of equidistant labels for quoted lines line [1]. Upper case **N** starts labeling exactly at the start of the line [Default centers them along the line]. **N-1** places one justified label at start, while **N+1** places one justified label at the end of quoted lines. Optionally, append */min_dist[clilp]* to enforce that a minimum distance separation between successive labels is enforced. **x|Xxfile.d** Reads the multisegment file *xfile.d* and places labels at the intersections between the quoted lines and the lines in *xfile.d*. **X** will resample the lines first along great-circle arcs. In addition, you may optionally append **+rradius[clilp]** to set a minimum label separation in the *x-y* plane [no limitation].

The optional *labelinfo* controls the specifics of the label formatting and consists of a concatenated string made up of any of the following control arguments:

- +aangle** For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default].
- +cdx[dy]** Sets the clearance between label and optional text box. Append **clilp** to specify the unit or *%* to indicate a percentage of the label font size [15%].

- +d** Turns on debug which will draw helper points and lines to illustrate the workings of the quoted line setup.
- +e** Delay the plotting of the text. This is used to build a clip path based on the text, then lay down other overlays while that clip path is in effect, then turning of clipping with `psclip -Cs` which finally plots the original text.
- +ffont** Sets the desired font [Default **FONT_ANNOT_PRIMARY** with its size changed to 9p].
- +g[*color*]** Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PS_PAGE_COLOR**].
- +jjust** Sets label justification [Default is MC]. Ignored when `-SqNln+|-1` is used.
- +l*label*** Sets the constant label text.
- +L*flag*** Sets the label text according to the specified flag:
 - +Lh** Take the label from the current segment header (first scan for an embedded `-Llabel` option, if not use the first word following the segment flag). For multiple-word labels, enclose entire label in double quotes.
 - +Ld** Take the Cartesian plot distances along the line as the label; append `clip` as the unit [Default is **PROJ_LENGTH_UNIT**].
 - +LD** Calculate actual map distances; append `d|e|f|k|l|n|M|n|s` as the unit [Default is **d**(egrees), unless label placement was based on map distances along the lines in which case we use the same unit specified for that algorithm]. Requires a map projection to be used.
 - +Lf** Use text after the 2nd column in the fixed label location file as the label. Requires the fixed label location setting.
 - +Lx** As **+Lh** but use the headers in the `xfile.d` instead. Requires the crossing file option.
- +ndx[/*dy*]** Nudges the placement of labels by the specified amount (append `clip` to specify the units). Increments are considered in the coordinate system defined by the orientation of the line; use **+N** to force increments in the plot x/y coordinates system [no nudging]. Not allowed with **+v**.
- +o** Selects rounded rectangular text box [Default is rectangular]. Not applicable for curved text (**+v**) and only makes sense for opaque text boxes.
- +p[*pen*]** Draws the outline of text boxes [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, style = solid].
- +r*min_rad*** Will not place labels where the line's radius of curvature is less than *min_rad* [Default is 0].
- +t[*file*]** Saves line label x, y, and text to *file* [Line_labels.txt]. Use **+T** to save x, y, angle, text instead.
- +u*unit*** Appends *unit* to all line labels. If *unit* starts with a leading hyphen (-) then there will be no space between label value and the unit. [Default is no unit].
- +v** Specifies curved labels following the path [Default is straight labels].
- +w** Specifies how many (x,y) points will be used to estimate label angles [Default is 10].
- +=*prefix*** Prepends *prefix* to all line labels. If *prefix* starts with a leading hyphen (-) then there will be no space between label value and the prefix. [Default is no prefix].

Note: By placing **-Sq** options in the segment header you can change the quoted text attributes on a segment-by-segment basis.

- Sr** rectangle. No size needs to be specified, but the x- and y-dimensions must be found in columns 4 and 5.
- SR** Rounded rectangle. No size needs to be specified, but the x- and y-dimensions and corner radius must be found in columns 4, 5, and 6.
- Ss** square. *size* is diameter of circumscribing circle.
- St** triangle. *size* is diameter of circumscribing circle.
- Su** cube (3-D). The *size* sets length of all sides. Append **u** if *size* is in x-units [Default is plot-distance units]. The facet colors will be modified to simulate shading. Use **-SU** to disable such 3-D illumination.
- Sv** vector. Direction (in degrees counter-clockwise from horizontal) and length must be found in columns 4 and 5. The *size* is the length of the vector head. Vector width is set by **-W**. See VECTOR ATTRIBUTES for specifying attributes.
- SV** Same as **-Sv**, except azimuth (in degrees east of north) should be given instead of direction. The azimuth will be mapped into an angle based on the chosen map projection (**-Sv** leaves the directions unchanged.) See VECTOR ATTRIBUTES for specifying attributes.
- Sw** pie wedge. Start and stop directions (in degrees counter-clockwise from horizontal) for pie slice must be found in columns 4 and 5.
- SW** Same as **-Sw**, except azimuths (in degrees east of north) should be given instead of the two directions. The azimuths will be mapped into angles based on the chosen map projection (**-Sw** leaves the directions unchanged.)
- Sx** cross (x). *size* is diameter of circumscribing circle.
- Sy** y-dash (l). *size* is the length of a short horizontal (y-dir) line segment.
- S=** geovector. Azimuth (in degrees east from north) and length (in km) must be found in columns 4 and 5. The *size* is the length of the vector head. Vector width is set by **-W**. See VECTOR ATTRIBUTES for specifying attributes.
- U[just/dx/dy/][clabel] (*more ...*)** Draw GMT time stamp logo on plot.
- V[level] (*more ...*)** Select verbosity level [c].
- W[-l+][pen] (*more ...*)** Set pen attributes for lines or the outline of symbols [Defaults: width = default, color = black, style = solid]. A leading + will use the lookup color (via **-C**) for both symbol fill and outline pen color, while a leading - will set outline pen color and turn off symbol fill.
- X[alcflr][x-shift[u]]**
- Y[alcflr][y-shift[u]] (*more ...*)** Shift plot origin.
- acol=name[...] (*more ...*)** Set aspatial column associations *col=name*.
- bi[ncols][type] (*more ...*)** Select binary input. [Default is the required number of columns given the chosen settings].
- ccopies (*more ...*)** Specify number of plot copies [Default is 1].
- f[ilo]colinfo (*more ...*)** Specify data types of input and/or output columns.

- g[*a*]*xlyldlXlYlDl*[*col*]*z*[+|-]*gap*[*u*] (*more ...*) Determine data gaps and line breaks. The **-g** option is ignored if **-S** is set.
- h[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- i*cols*[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).
- p[*xlylz*]*azim/lev*[*zlevel*][+*wlon0/lat0*][*z0*][+*vx0/y0*] (*) Select perspective view.
- t[*transp*] (*more ...*) Set PDF transparency level in percent.
- :[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- + or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.75.5 Vector Attributes

Several modifiers may be appended to the vector-producing options to specify the placement of vector heads, their shapes, and the justification of the vector. Below, left and right refers to the side of the vector line when viewed from the start point to the end point of the segment:

- +*aangle* sets the angle of the vector head apex [30].
- +*b* places a vector head at the beginning of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +*e* places a vector head at the end of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +*g*-/*fill* turns off vector head fill (if -) or sets the vector head fill [Default fill is used, which may be no fill].
- +*l* draws half-arrows, using only the left side of specified heads [both sides].
- +*norm* scales down vector attributes (pen thickness, head size) with decreasing length, where vectors shorter than *norm* will have their attributes scaled by *length/norm* [arrow attributes remains invariant to length].
- +*oplon/plat* specifies the oblique pole for the great or small circles. Only needed for great circles if +*q* is given.
- +*p*[-][*pen*] sets the vector pen attributes. If *pen* has a leading - then the head outline is not drawn. [Default pen is used, and head outline is drawn]
- +*q* means the input *angle*, *length* data instead represent the *start* and *stop* opening angles of the arc segment relative to the given point.
- +*r* draws half-arrows, using only the right side of specified heads [both sides].

In addition, all but circular vectors may take these modifiers:

+jjust determines how the input *x,y* point relates to the vector. Choose from **beginning** [default], **end**, or **center**.

+s means the input *angle, length* is instead the *x, y* coordinates of the vector end point.

1.75.6 Examples

To plot blue columns (width = 1.25 cm) at the positions listed in the file *heights.xyz* on a 3-D projection of the space (0-10), (0-10), (0-100), with tickmarks every 2, 2, and 10, viewing it from the southeast at 30 degree elevation, use:

```
gmt psxyz heights.xyz -R0/10/0/10/0/100 -Jx1.25c -Jz0.125c -So1.25c \
  -Gblue -B2:XLABEL:/2:YLABEL:/10:ZLABEL:."3-D PLOT":15 -p135/30 \
  -Uc -W -P > heights.ps
```

1.75.7 Segment Header Parsing

Segment header records may contain one of more of the following options:

- Gfill** Use the new *fill* and turn filling on
- G-** Turn filling off
- G** Revert to default fill (none if not set on command line)
- Wpen** Use the new *pen* and turn outline on
- W** Revert to default pen *MAP_DEFAULT_PEN* (if not set on command line)
- W-** Turn outline off
- Zzval** Obtain fill via cpt lookup using z-value *zval*
- ZNaN** Get the NaN color from the cpt file

1.75.8 Custom Symbols

psxyz allows users to define and plot their own custom symbols. This is done by encoding the symbol using our custom symbol macro code described in Appendix N. Put all the macro codes for your new symbol in a file whose extension must be *.def*; you may then address the symbol without giving the extension (e.g., the symbol file *tsunami.def* is used by specifying **-S k tsunami/size**). The definition file can contain any number of plot code records, as well as blank lines and comment lines (starting with #). **psxyz** will look for the definition files in (1) the current directory, (2) the *~/gmt* directory, and (3) the *\$GMT_SHAREDIR/custom* directory, in that order. Freeform polygons (made up of straight line segments and arcs of circles) can be designed - these polygons can be painted and filled with a pattern. Other standard geometric symbols can also be used. See Appendix *App-custom_symbols* for macro definitions.

1.75.9 Bugs

No hidden line removal is employed for polygons and lines. Symbols, however, are first sorted according to their distance from the viewpoint so that nearby symbols will overprint more distant ones should they project to the same *x,y* position.

psxyz cannot handle filling of polygons that contain the south or north pole. For such a polygon, make a copy and split it into two and make each explicitly contain the polar point. The two polygons will combine to give the desired effect when filled; to draw outline use the original polygon.

The **-N** option does not adjust the BoundingBox information so you may have to post-process the PostScript output with *ps2raster* **-A** to obtain the correct BoundingBox.

1.75.10 See Also

gmt, *gmt.conf*, *gmtcolors*, *psbasemap*, *psxy*

1.76 sample1d

sample1d - Resample 1-D table data using splines

1.76.1 Synopsis

```
sample1d [ table ] [ -Afl|mlr|R[+l] ] [ -F|al|cn ] [ -Iinc[unit] ] [ -Nknotfile ] [ -Sstart[/stop] ] [ -Tcol ] [ -V[level] ] [ -b[i]o][ncol][type][w][+L|+B] ] [ -f[i]o]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[i]o][n][+c][+d][+rremark][+rtitle] ] [ -i[cols][l][sscale][offset][,...] ] [ -o[cols][,...] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.76.2 Description

sample1d reads a multi-column ASCII [or binary] data set from file [or standard input] and interpolates the timeseries/profile at locations where the user needs the values. The user must provide the column number of the independent (monotonically increasing **or** decreasing) variable. Equidistant or arbitrary sampling can be selected. All columns are resampled based on the new sampling interval. Several interpolation schemes are available. Extrapolation outside the range of the input data is not supported.

1.76.3 Required Arguments

None.

1.76.4 Optional Arguments

table This is one or more ASCII [of binary, see **-bi**] files with one column containing the independent variable (which must be monotonically in/de-creasing) and the remaining columns holding other data values. If no file is provided, **sample1d** reads from standard input.

-Afl|mlr|R For track resampling (if **-T...unit** is set) we can select how this is to be performed. Append **f** to keep original points, but add intermediate points if needed; note this selection does not necessarily yield equidistant points [Default], **m** as **f**, but first follow meridian (along y) then parallel (along x), **p** as **f**, but first follow parallel (along y) then meridian (along x), **r** to resample at equidistant locations; input points are not necessarily included in the output, and **R** as **r**, but adjust given spacing to fit the track length exactly. Finally, append **+l** if distances should be measured along rhumb lines (loxodromes).

- Fllalcn** Choose from **l** (Linear), **a** (Akima spline), **c** (natural cubic spline), and **n** (no interpolation: nearest point) [Default is **-Fa**]. You may change the default interpolant; see *GMT_INTERPOLANT* in your *gmt.conf* file.
- linc[*unit*]** *inc* defines the sampling interval [Default is the separation between the first and second abscissa point in the *infile*]. Append a distance unit (see UNITS) to indicate that the first two columns contain longitude, latitude and you wish to resample this path with a spacing of *inc* in the chosen units. For sampling of (x, y) Cartesian tracks, specify the unit as c. Use **-A** to control how path resampling is performed.
- Nknotfile** *knotfile* is an optional ASCII file with the x locations where the data set will be resampled in the first column. Note: If **-H** is selected it applies to both *infile* and *knotfile*. Also note that **-i** never applies to *knotfile* since we always consider the first column only.
- Sstart[*stop*]** For equidistant sampling, *start* indicates the location of the first output value. [Default is the smallest even multiple of *inc* inside the range of *infile*]. Optionally, append *stop* to indicate the location of the last output value [Default is the largest even multiple of *inc* inside the range of *infile*].
- Tcol** Sets the column number of the independent variable [Default is 0 (first)].
- V[*level*]** (*more ...*) Select verbosity level [c].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. [Default is 2 (or at least the number of columns implied by **-T**)].
- bo[*ncols*][*type*]** (*more ...*) Select binary output. [Default is same as input].
- f[*ilo*]*colinfo*** (*more ...*) Specify data types of input and/or output columns.
- g[*a*]*x*|*y*|*d*|*X*|*Y*|*D*][*col*]*z*[+]*l*-**gap[*u*]** (*more ...*) Determine data gaps and line breaks.**
- h[*ilo*][*n*][+]*c*][+]*d*][+]*rremark*][+]*rtitle*** (*more ...*) Skip or produce header record(s).
- i[*cols*][*l*][*sscale*][*offset*][*,...*]** (*more ...*) Select input columns (0 is first column).
- o[*cols*][*,...*]** (*more ...*) Select output columns (0 is first column).
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.76.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

1.76.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.76.7 Calendar Time Sampling

If the abscissa are calendar times then you must use the **-f** option to indicate this. Furthermore, **-I** then expects an increment in the current *TIME_UNIT* units. There is not yet support for variable intervals such as months.

1.76.8 Examples

To resample the file *profiles.tdgmb*, which contains (time,distance,gravity,magnetics,bathymetry) records, at 1km equidistant intervals using Akima's spline, use

```
gmt sample1d profiles.tdgmb -I1 -Fa -T1 > profiles_equi_d.tdgmb
```

To resample the file *depths.dt* at positions listed in the file *grav_pos.dg*, using a cubic spline for the interpolation, use

```
gmt sample1d depths.dt -Ngrav_pos.dg -Fc > new_depths.dt
```

To resample the file *track.txt* which contains lon, lat, depth every 2 nautical miles, use

```
gmt sample1d track.txt -I2n -AR > new_track.dt
```

To do approximately the same, but make sure the original points are included, use

```
gmt sample1d track.txt -I2n -Af > new_track.dt
```

To obtain a rhumb line (loxodrome) sampled every 5 km instead, use

```
gmt sample1d track.txt -I5k -AR+1 > new_track.dt
```

1.76.9 See Also

gmt, *gmt.conf*, *greenspline*, *filter1d*

1.77 spectrum1d

spectrum1d - Compute auto- [and cross-] spectra from one [or two] time-series

1.77.1 Synopsis

```
spectrum1d [ table ] -Ssegment_size [ -C[xycnpago] ] [ -Ddt ] [ -L[hlm] ] [ -N[+]name_stem ] [ -b[ilo][ncol][type][w][+L+B] ] [ -W ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+|-]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[1][sscale][offset][,...] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.77.2 Description

spectrum1d reads X [and Y] values from the first [and second] columns on standard input [or *x[y]file*]. These values are treated as timeseries X(t) [Y(t)] sampled at equal intervals spaced *dt* units apart. There may be any number of lines of input. **spectrum1d** will create file[s] containing auto- [and cross-] spectral density estimates by Welch's method of ensemble averaging of multiple overlapped windows, using standard error estimates from Bendat and Piersol.

The output files have 3 columns: f or w, p, and e. f or w is the frequency or wavelength, p is the spectral density estimate, and e is the one standard deviation error bar size. These files are named based on *name_stem*. If the **-C** option is used, up to eight files are created; otherwise only one (xpower) is written. The files (which are ASCII unless **-bo** is set) are as follows:

name_stem.xpower Power spectral density of X(t). Units of $X * X * dt$.

name_stem.ypower Power spectral density of Y(t). Units of $Y * Y * dt$.

name_stem.cpower Power spectral density of the coherent output. Units same as ypower.

name_stem.npower Power spectral density of the noise output. Units same as ypower.

name_stem.gain Gain spectrum, or modulus of the transfer function. Units of (Y / X).

name_stem.phase Phase spectrum, or phase of the transfer function. Units are radians.

name_stem.admit Admittance spectrum, or real part of the transfer function. Units of (Y / X).

name_stem.coh (Squared) coherency spectrum, or linear correlation coefficient as a function of frequency. Dimensionless number in [0, 1]. The Signal-to-Noise-Ratio (SNR) is $coh / (1 - coh)$. $SNR = 1$ when $coh = 0.5$.

1.77.3 Required Arguments

-S*segment_size* *segment_size* is a radix-2 number of samples per window for ensemble averaging. The smallest frequency estimated is $1.0/(segment_size * dt)$, while the largest is $1.0/(2 * dt)$. One standard error in power spectral density is approximately $1.0 / \sqrt{n_data / segment_size}$, so if *segment_size* = 256, you need 25,600 data to get a one standard error bar of 10%. Cross-spectral error bars are larger and more complicated, being a function also of the coherency.

1.77.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**) files holding X(t) [Y(t)] samples in the first 1 [or 2] columns. If no files are specified, **spectrum1d** will read from standard input.

- C**[*xycnpag*o] Read the first two columns of input as samples of two time-series, X(t) and Y(t). Consider Y(t) to be the output and X(t) the input in a linear system with noise. Estimate the optimum frequency response function by least squares, such that the noise output is minimized and the coherent output and the noise output are uncorrelated. Optionally specify up to 8 letters from the set { **x y c n p a g o** } in any order to create only those output files instead of the default [all]. **x** = xpower, **y** = ypower, **c** = cpower, **n** = npower, **p** = phase, **a** = admit, **g** = gain, **o** = coh.
- Ddt** *dt* Set the spacing between samples in the time-series [Default = 1].
- L** Leave trend alone. By default, a linear trend will be removed prior to the transform. Alternatively, append **m** to just remove the mean value or **h** to remove the mid-value.
- N**[+]*name_stem* *name_stem* Supply the name stem to be used for output files [Default = "spectrum"]. To place all the computed output columns in a single table, use **-N+**.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- W** Write Wavelength rather than frequency in column 1 of the output file[s] [Default = frequency, (cycles / *dt*)].
- bi**[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- bo**[*ncols*][*type*] (*more ...*) Select binary output. [Default is 2 output columns].
- f**[*ilo*]*colinfo* (*more ...*) Specify data types of input and/or output columns.
- g**[*a*]*x**y**l**d**X**Y**D*][*col*]*z*[+]*l*-**gap**[*u*] (*more ...*) Determine data gaps and line breaks.
- h**[*ilo*][*n*][+]**c**][+]**d**][+]**rremark**][+]**rtitle**] (*more ...*) Skip or produce header record(s).
- icol**s[*l*][*sscale*][+]**ooffset**][*,...*] (*more ...*) Select input columns (0 is first column).
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.77.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.77.6 Examples

Suppose data.g is gravity data in mGal, sampled every 1.5 km. To write its power spectrum, in mGal**2-km, to the file data.xpower, use

```
gmt spectrum1d data.g -S256 -D1.5 -Ndata
```

Suppose in addition to data.g you have data.t, which is topography in meters sampled at the same points as data.g. To estimate various features of the transfer function, considering data.t as input and data.g as output, use

```
paste data.t data.g | gmt spectrum1d -S256 -D1.5 -Ndata -C
```

1.77.7 Tutorial

The output of spectrum1d is in units of power spectral density, and so to get units of data-squared you must divide by delta_t, where delta_t is the sample spacing. (There may be a factor of 2 pi somewhere, also. If you want to be sure of the normalization, you can determine a scale factor from Parseval's theorem: the sum of the squares of your input data should equal the sum of the squares of the outputs from spectrum1d, if you are simply trying to get a periodogram. [See below.]

Suppose we simply take a data set, x(t), and compute the discrete Fourier transform (DFT) of the entire data set in one go. Call this X(f). Then suppose we form X(f) times the complex conjugate of X(f).

$P_{\text{raw}}(f) = X(f) * X'(f)$, where the ' indicates complex conjugation.

P_raw is called the periodogram. The sum of the samples of the periodogram equals the sum of the samples of the squares of x(t), by Parseval's theorem. (If you use a DFT subroutine on a computer, usually the sum of P_raw equals the sum of x-squared, times M, where M is the number of samples in x(t).)

Each estimate of X(f) is now formed by a weighted linear combination of all of the x(t) values. (The weights are sometimes called "twiddle factors" in the DFT literature.) So, no matter what the probability distribution for the x(t) values is, the probability distribution for the X(f) values approaches [complex] Gaussian, by the Central Limit Theorem. This means that the probability distribution for P_raw(f) approaches chi-squared with two degrees of freedom. That reduces to an exponential distribution, and the variance of the estimate of P_raw is proportional to the square of the mean, that is, the expected value of P_raw.

In practice if we form P_raw, the estimates are hopelessly noisy. Thus P_raw is not useful, and we need to do some kind of smoothing or averaging to get a useful estimate, P_useful(f).

There are several different ways to do this in the literature. One is to form P_raw and then smooth it. Another is to form the auto-covariance function of x(t), smooth, taper and shape it, and then take the Fourier transform of the smoothed, tapered and shaped auto-covariance. Another is to form a parametric model for the auto-correlation structure in x(t), then compute the spectrum of that model. This last approach is what is done in what is called the "maximum entropy" or "Berg" or "Box-Jenkins" or "ARMA" or "ARIMA" methods.

Welch's method is a tried-and-true method. In his method, you choose a segment length, -SN, so that estimates will be made from segments of length N. The frequency samples (in cycles per delta_t unit) of your P_useful will then be at $k/(N * \text{delta}_t)$, where k is an integer, and you will get N samples (since the spectrum is an even function of f, only N/2 of them are really useful). If the length of your entire data set, x(t), is M samples long, then the variance in your P_useful will decrease in proportion to N/M. Thus you need to choose $N \ll M$ to get very low noise and high confidence in P_useful. There is a trade-off here; see below.

There is an additional reduction in variance in that Welch's method uses a Von Hann spectral window on each sample of length N. This reduces side lobe leakage and has the effect of smoothing the (N segment) periodogram as if the X(f) had been convolved with [1/4, 1/2, 1/4] prior to forming P_useful. But this

slightly widens the spectral bandwidth of each estimate, because the estimate at frequency sample k is now a little correlated with the estimate at frequency sample $k+1$. (Of course this would also happen if you simply formed P_{raw} and then smoothed it.)

Finally, Welch's method also uses overlapped processing. Since the Von Hann window is large in the middle and tapers to near zero at the ends, only the middle of the segment of length N contributes much to its estimate. Therefore in taking the next segment of data, we move ahead in the $x(t)$ sequence only $N/2$ points. In this way, the next segment gets large weight where the segments on either side of it will get little weight, and vice versa. This doubles the smoothing effect and ensures that (if $N \ll M$) nearly every point in $x(t)$ contributes with nearly equal weight in the final answer.

Welch's method of spectral estimation has been widely used and widely studied. It is very reliable and its statistical properties are well understood. It is highly recommended in such textbooks as "Random Data: Analysis and Measurement Procedures" by Bendat and Piersol.

In all problems of estimating parameters from data, there is a classic trade-off between resolution and variance. If you want to try to squeeze more resolution out of your data set, then you have to be willing to accept more noise in the estimates. The same trade-off is evident here in Welch's method. If you want to have very low noise in the spectral estimates, then you have to choose $N \ll M$, and this means that you get only N samples of the spectrum, and the longest period that you can resolve is only $N * \text{delta}_t$. So you see that reducing the noise lowers the number of spectral samples and lowers the longest period. Conversely, if you choose N approaching M , then you approach the periodogram with its very bad statistical properties, but you get lots of samples and a large fundamental period.

The other spectral estimation methods also can do a good job. Welch's method was selected because the way it works, how one can code it, and its effects on statistical distributions, resolution, side-lobe leakage, bias, variance, etc. are all easily understood. Some of the other methods (e.g. Maximum Entropy) tend to hide where some of these trade-offs are happening inside a "black box".

1.77.8 See Also

gmt, grdfft

1.77.9 References

Bendat, J. S., and A. G. Piersol, 1986, Random Data, 2nd revised ed., John Wiley & Sons.

Welch, P. D., 1967, The use of Fast Fourier Transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms, IEEE Transactions on Audio and Electroacoustics, Vol AU-15, No 2.

1.78 sph2grd

sph2grd - Compute grid from spherical harmonic coefficients

1.78.1 Synopsis

```
sph2grd [ table ] -Ggrdfile -Ixinc[unit][=|+][yinc[unit][=|+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -D[gln] ] [ -E ] [ -F[k]filter ] [ -N[norm] ] [ -Q ] [ -V[level] ] [ -bi[ncols][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+remark][+rtitle] ] [ -icols ] [ -sscale ] [ -ooffset ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.78.2 Description

sph2grd reads a spherical harmonics coefficient table with records of L, M, C[L,M], S[L,M] and evaluates the spherical harmonic model on the specified grid.

1.78.3 Required Arguments

-Ggrdfile *grdfile* is the name of the binary output grid file. (See GRID FILE FORMAT below.)

-I x_{inc} [unit][=+][y_{inc} [unit][=+]] x_{inc} [and optionally y_{inc}] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If y_{inc} is given but set to 0 it will be reset equal to x_{inc} ; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max x (*east*) or y (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit] $x_{min}/x_{max}/y_{min}/y_{max}$ [r] (*more ...*) Specify the region of interest.

1.78.4 Optional Arguments

table One or more ASCII [or binary, see **-bi**] files holding the spherical harmonic coefficients. We expect the first four columns to hold the degree L, the order M, followed by the cosine and sine coefficients.

-D[$g|n$] Will evaluate a derived field from a geopotential model. Choose between **Dg** which will compute the gravitational field or **Dn** to compute the geoid [Add **-E** for anomalies on the ellipsoid].

-E Evaluate expansion on the current ellipsoid [Default is sphere].

-F[d]filter Filter coefficients according to one of two kinds of filter specifications: . Select **-Fk** if values are given in km [Default is coefficient harmonic degree L]. a) Cosine band-pass: Append four wavelengths $lc/lp/hp/hc$. Coefficients outside lc/hc are cut; those inside lp/hp are passed, while the rest are tapered. Replace wavelength by **-** to skip, e.g., **-F-/50/75** is a low-pass filter. b) Gaussian band-pass: Append two wavelengths lo/hi where filter amplitudes = 0.5. Replace wavelength by **-** to skip, e.g., **-F70/-** is a high-pass Gaussian filter.

-N[norm] Normalization used for coefficients. Choose among **m**: Mathematical normalization - inner products summed over surface equal 1 [Default]. **g** Geodesy normalization - inner products summed over surface equal 4pi. **s**: Schmidt normalization - as used in geomagnetism.

-V[level] (*more ...*) Select verbosity level [c].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 4 input columns].

- h[ilo][n][+c][+d][+rremark][+rtitle] (*more ...*)** Skip or produce header record(s). Not used with binary data.
- icols[1][sscale][offset][,...] (*more ...*)** Select input columns (0 is first column).
- r (*more ...*)** Set pixel node registration [gridline].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.78.5 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.78.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When writing a netCDF file, the grid is stored by default with the variable name “z”. To specify another variable name *varname*, append `?varname` to the file name. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes.

1.78.7 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x** **-f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.78.8 Examples

To create a 1 x 1 degree global grid file from the ASCII coefficients in EGM96_to_360.txt, use

```
gmt sph2grd EGM96_to_360.txt -GEGM96_to_360.nc -Rg -I1 -V
```

1.78.9 Reference

Holmes, S. A., and Featherstone, W. E., 2002, A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalized associated Legendre functions: *J. Geodesy*, v. 76, p. 279–299.

1.78.10 See Also

gmt, *grdfft*, *grdmath*

1.79 sphdistance

sphdistance - Make grid of distances to nearest points on a sphere

1.79.1 Synopsis

```
sphdistance [ table ] -Ggrdfile [ -C ] [ -E ] [ -Ixinc[unit][=|+][yinc[unit][=|+]] ] [ -Lunit ] [ -QVoronoi.txt ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.79.2 Description

sphdistance reads one or more ASCII [or binary] files (or standard input) containing lon, lat and performs the construction of Voronoi polygons. These polygons are then processed to calculate the nearest distance to each node of the lattice and written to the specified grid. The Voronoi algorithm used is STRIPACK. As an option, you may provide pre-calculated Voronoi polygon file in the format written by *sphtriangulate*, thus bypassing the memory- and time-consuming triangularization.

1.79.3 Required Arguments

-G*grdfile* Name of the output grid to hold the computed distances.

1.79.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

- C For large data sets you can save some memory (at the expense of more processing) by only storing one form of location coordinates (geographic or Cartesian 3-D vectors) at any given time, translating from one form to the other when necessary [Default keeps both arrays in memory]. Not applicable with -Q.
- E Instead of computing distances, return the ID numbers of the Voronoi polygons that each grid node is inside [Default computes distances].
- I $xinc$ [*unit*][=|+][*yinc*[*unit*][=|+]] x_inc [and optionally y_inc] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If y_inc is given but set to 0 it will be reset equal to x_inc ; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max x (*east*) or y (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if -R $grdfile$ is used then the grid spacing has already been initialized; use -I to override the values.
- L*unit* Specify the unit used for distance calculations. Choose among **d** (spherical degree), **e** (m), **f** (feet), **k** (km), **M** (mile), **n** (nautical mile) or **u** survey foot. A spherical approximation is used unless *PROJ_ELLIPSOID* is set to an actual ellipsoid.
- N Read the information pertaining to each Voronoi polygon (the unique node lon, lat and polygon area) from a separate file [Default acquires this information from the ASCII segment headers of the output file]. Required if binary input via -Q is used.
- Q*voronoi.txt* Append the name of a file with pre-calculated Voronoi polygons [Default performs the Voronoi construction on input data]. For binary data -**bi** you must specify the node information separately (via -N).
- R[*unit*]*west/east/south/north*[*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*dd*:*mm*:*ss*.*xxx*[**W**|**E**|**S**|**N**] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands -**Rg** and -**Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the -**R** settings (and grid spacing, if applicable) are copied from the grid. Using -**Runit** expects projected (Cartesian) coordinates compatible with chosen -**J** and we inversely project to determine actual rectangular geographic region. For perspective view (-**p**), optionally append *zmin/zmax*. In case of perspective view (-**p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the -**Jz** option, not when using only the -**p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.
- V[*level*] (*more ...*) Select verbosity level [*c*].
- bi**[*ncols*][*type*] (*more ...*) Select binary input. [Default is 2 input columns].
- bo**[*ncols*][*type*] (*more ...*) Select binary output. [Default is same as input].
- h**[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- icols**[*l*][*sscale*][*offset*][,...] (*more ...*) Select input columns (0 is first column).

- r** (*more ...*) Set pixel node registration [gridline].
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.79.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.79.6 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.79.7 Examples

To construct Voronoi polygons from the points in the file *testdata.txt* and then calculate distances from the data to a global 1x1 degree grid, use

```
gmt sphdistance testdata.txt -Rg -I1 -Gglobedist.nc
```

To generate the same grid in two steps using **sphtriangulate** separately, try

```
gmt sphtriangulate testdata.txt -Qv > voronoi.txt
gmt sphdistance -Qvoronoi.txt -Rg -I1 -Gglobedist.nc
```

1.79.8 See Also

gmt, *sphtriangulate*, *triangulate*

1.79.9 References

Renka, R. J., 1997, Algorithm 772: STRIPACK: Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere, *AMC Trans. Math. Software*, **23**(3), 416-434.

1.80 sphinterpolate

sphinterpolate - Spherical gridding in tension of data on a sphere

1.80.1 Synopsis

```
sphinterpolate [ table ] -Ggrdfile [ -Ixinc[unit][=|+][/yinc[unit][=|+]] ] [ -Qmode[/options] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -Z ] [ -b[ilo][ncol][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.80.2 Description

sphinterpolate reads one or more ASCII [or binary] files (or standard input) containing lon, lat, f and performs a Delaunay triangulation to set up a spherical interpolation in tension. The final grid is saved to the specified file. Several options may be used to affect the outcome, such as choosing local versus global gradient estimation or optimize the tension selection to satisfy one of four criteria.

1.80.3 Required Arguments

-G*grdfile* Name of the output grid to hold the interpolation.

1.80.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-I*xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.

- Qmode[*loptions*]** Specify one of four ways to calculate tension factors to preserve local shape properties or satisfy arc constraints [Default is no tension].
- Q0** Piecewise linear interpolation; no tension is applied.
- Q1** Smooth interpolation with local gradient estimates.
- Q2** Smooth interpolation with global gradient estimates. You may optionally append */N/M/U*, where *N* is the number of iterations used to converge at solutions for gradients when variable tensions are selected (e.g., **-T** only) [3], *M* is the number of Gauss-Seidel iterations used when determining the global gradients [10], and *U* is the maximum change in a gradient at the last iteration [0.01].
- Q3** Smoothing. Optionally append */E/U* [*/0/0*], where *E* is Expected squared error in a typical (scaled) data value, and *U* is Upper bound on weighted sum of squares of deviations from data.
- R[*unit*]*west/east/south/north*[*zmin/zmax*][*r*]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][*W|E|S|N*] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.
- T** Use variable tension (ignored with **-Q0** [constant])
- V[*level*]** (*more ...*) Select verbosity level [*c*].
- Z** Before interpolation, scale data by the maximum data range [no scaling].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. [Default is 3 input columns].
- bo[*ncols*][*type*]** (*more ...*) Select binary output. [Default is same as input].
- h[*ilo*][*n*][*+c*][*+d*][*+rremark*][*+rtitle*]** (*more ...*) Skip or produce header record(s).
- :[*ilo*]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- r** (*more ...*) Set pixel node registration [gridline].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.80.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted

according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.80.6 Examples

To interpolate the points in the file `testdata.txt` on a global 1x1 degree grid with no tension, use

```
sphinterpolate testdata.txt -Rg -I1 -Gsolution.nc
```

1.80.7 See Also

gmt, *greenspline*, *nearneighbor*, *sphdistance*, *sphtriangulate*, *surface*, *triangulate*

1.80.8 References

Renka, R. J., 1997, Algorithm 772: STRIPACK: Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere, *AMC Trans. Math. Software*, **23**(3), 416-434.

Renka, R. J., 1997, Algorithm 773: SSRFPACK: Interpolation of scattered data on the Surface of a Sphere with a surface under tension, *AMC Trans. Math. Software*, **23**(3), 435-442.

1.81 sphtriangulate

sphtriangulate - Delaunay or Voronoi construction of spherical lon,lat data

1.81.1 Synopsis

```
sphtriangulate [ table ] [ -A ] [ -C ] [ -D ] [ -Lunit ] [ -Nnfile ] [ -Qdlv ] [ -T ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L|+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.81.2 Description

sphtriangulate reads one or more ASCII [or binary] files (or standard input) containing lon, lat and performs a spherical Delaunay triangulation, i.e., it find how the points should be connected to give the most equilateral triangulation possible on the sphere. Optionally, you may choose **-Qv** which will do further processing to obtain the Voronoi polygons. Normally, either set of polygons will be written as fillable segment output; use **-T** to write unique arcs instead. As an option, compute the area of each triangle or polygon. The algorithm used is STRIPACK.

1.81.3 Required Arguments

None.

1.81.4 Optional Arguments

- table** One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.
- A** Compute the area of the spherical triangles (**-Qd**) or polygons (**-Qv**) and write the areas (in chosen units; see **-L**) in the output segment headers [no areas calculated].
 - C** For large data set you can save some memory (at the expense of more processing) by only storing one form of location coordinates (geographic or Cartesian 3-D vectors) at any given time, translating from one form to the other when necessary [Default keeps both arrays in memory].
 - D** Used with **-m** to skip the last (repeated) input vertex at the end of a closed segment if it equals the first point in the segment. Requires **-m** [Default uses all points].
 - Lunit** Specify the unit used for distance and area calculations. Choose among **e** (m), **f** (foot), **k** (km), **m** (mile), **n** (nautical mile), **u** (survey foot), or **d** (spherical degree). A spherical approximation is used unless *PROJ_ELLIPSOID* is set to an actual ellipsoid, in which case we convert latitudes to authalic latitudes before calculating areas. When degree is selected the areas are given in steradians.
 - Nnfile** Write the information pertaining to each polygon (for Delaunay: the three node number and the triangle area (if **-A** was set); for Voronoi the unique node lon, lat and polygon area (if **-A** was set)) to a separate file. This information is also encoded in the segment headers of ASCII output files]. Required if binary output is needed.
- dlv** Select between **delaunay** or **voronoi** mode [Delaunay].
- T** Write the unique arcs of the construction [Default writes fillable triangles or polygons]. When used with **-A** we store arc length in the segment header in chosen unit (see **-L**).
 - V[level]** (*more ...*) Select verbosity level [c].
 - bi[ncols][type]** (*more ...*) Select binary input. [Default is 2 input columns].
 - bo[ncols][type]** (*more ...*) Select binary output. [Default is same as input].
 - h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
 - :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
 - r** (*more ...*) Set pixel node registration [gridline].
 - ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
 - ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
 - ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
 - version** Print GMT version and exit.
 - show-datadir** Print full path to GMT share directory and exit.

1.81.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted

according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.81.6 Examples

To triangulate the points in the file `testdata.txt`, and make a Voronoi diagram via *psxy*, use

```
gmt sphtriangulate testdata.txt -Qv | psxy -Rg -JG30/30/6i -L -P -Wlp -B0g30 | gv -
```

To compute the optimal Delaunay triangulation network based on the multiple segment file `globalnodes.d` and save the area of each triangle in the header record, try

```
gmt sphtriangulate globalnodes.d -Qd -A > global_tri.d
```

1.81.7 See Also

gmt, *triangulate*, *sphdistance*

1.81.8 References

Renka, R. J., 1997, Algorithm 772: STRIPACK: Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere, *AMC Trans. Math. Software*, **23**(3), 416-434.

1.82 splitxyz

splitxyz - Split xyz[dh] data tables into individual segments

1.82.1 Synopsis

```
splitxyz [ table ] [ -Aazimuth/tolerance ] [ -Ccourse_change ] [ -Dminimum_distance ] [ -Fxy_filter/z_filter ] [ -Ntemplate ] [ -Qflags ] [ -S ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L][B] ] [ -f[ilo]colinfo ] [ -g[a]x|y|d|X|Y|D][col]z[+]l]gap[u] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.82.2 Description

splitxyz reads a series of (x,y[,z]) records [or optionally (x,y,z,d,h); see **-S** option] from standard input [or *xyz[dh]file*] and splits this into separate lists of (x,y[,z]) series, such that each series has a nearly constant azimuth through the x,y plane. There are options to choose only those series which have a certain orientation, to set a minimum length for series, and to high- or low-pass filter the z values and/or the x,y values. **splitxyz** is a useful filter between data extraction and *pswiggle* plotting, and can also be used to divide a large x,y[,z] dataset into segments.

1.82.3 Required Arguments

none.

1.82.4 Optional Arguments

table One or more ASCII [or binary, see **-bi**] files with 2, 3, or 5 columns holding (x,y,[z[,d,h]]) data values. To use (x,y,z,d,h) input, sorted so that d is non-decreasing, specify the **-S** option; default expects (x,y,z) only. If no files are specified, **splitxyz** will read from standard input.

-Aazimuth/tolerance Write out only those segments which are within +/- *tolerance* degrees of *azimuth* in heading, measured clockwise from North, [0 - 360]. [Default writes all acceptable segments, regardless of orientation].

-Ccourse_change Terminate a segment when a course change exceeding *course_change* degrees of heading is detected [ignore course changes].

-Dminimum_distance Do not write a segment out unless it is at least *minimum_distance* units long [0]

-Fxy_filter/z_filter Filter the z values and/or the x,y values, assuming these are functions of d coordinate. *xy_filter* and *z_filter* are filter widths in distance units. If a filter width is zero, the filtering is not performed. The absolute value of the width is the full width of a cosine-arch low-pass filter. If the width is positive, the data are low-pass filtered; if negative, the data are high-pass filtered by subtracting the low-pass value from the observed value. If *z_filter* is non-zero, the entire series of input z values is filtered before any segmentation is performed, so that the only edge effects in the filtering will happen at the beginning and end of the complete data stream. If *xy_filter* is non-zero, the data is first divided into segments and then the x,y values of each segment are filtered separately. This may introduce edge effects at the ends of each segment, but prevents a low-pass x,y filter from rounding off the corners of track segments. [Default = no filtering].

-Ntemplate Write each segment to a separate output file [Default writes a multiple segment file to stdout]. Append a format template for the individual file names; this template **must** contain a C format specifier that can format an integer argument (the running segment number across all tables); this is usually %d but could be %08d which gives leading zeros, etc. [Default is splitxyz_segment_%d.{txt|bin}, depending on **-bo**]. Alternatively, give a template with two C format specifiers and we will supply the table number and the segment number within the table to build the file name.

-Qflags Specify your desired output using any combination of *xyzdh*, in any order. Do not space between the letters. Use lower case. The output will be ASCII (or binary, see **-bo**) columns of values corresponding to *xyzdh* [Default is **-Qxyzdh** (**-Qxydh** if only 2 input columns)].

-S Both d and h are supplied. In this case, input contains x,y,z,d,h. [Default expects (x,y,z) input, and d,h are computed from delta x, delta y. Use **-fg** to indicate map data; then x,y are assumed to be in degrees of longitude, latitude, distances are considered to be in kilometers, and angles are actually azimuths. Otherwise, distances are Cartesian in same units as x,y and angles are counter-clockwise from horizontal].

-V[level] (*more ...*) Select verbosity level [c].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 2, 3, or 5 input columns as set by **-S**].

-bo[ncols][type] (*more ...*) Select binary output. [Default is 1-5 output columns as set by **-Q**].

-f[ilo]colinfo (*more ...*) Specify data types of input and/or output columns.

- g[a]xly|d|X|Y|ID|[col]z[+|-]gap[u]** (*more ...*) Determine data gaps and line breaks. Do not let a segment have a gap exceeding *gap*; instead, split it into two segments. [Default ignores gaps].
- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s).
- icols[l][sscale][offset][,...]** (*more ...*) Select input columns (0 is first column).
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.82.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.82.6 Distance Calculations

The type of input data is dictated by the **-f** option. If **-fg** is given then x,y are in degrees of longitude, latitude, distances are in kilometers, and angles are azimuths. Otherwise, distances are Cartesian in same units as x,y and angles are counter-clockwise from horizontal.

1.82.7 Examples

Suppose you want to make a wiggle plot of magnetic anomalies on segments oriented approximately east-west from a NGDC-supplied cruise called JA020015 in the region **-R300/315/12/20**. You want to use a 100 km low-pass filter to smooth the tracks and a 500km high-pass filter to detrend the magnetic anomalies. Try this:

```
gmt mgd77list JA020015 -R300/315/12/20 -Flon,lat,mag,dist,azim | gmt splitxyz -A90/15 -F100/-500 \  
-D100 -S -V -fg | gmt pswiggle -R300/315/12/20 -Jm0.6i -Baf -B+tJA020015 -T1 \  
-W0.75p -Ggray -Z200 > JA020015_wiggles.ps
```

MGD-77 users: For this application we recommend that you extract dist,azim from *mgd77list* rather than have **splitxyz** compute them separately.

Suppose you have been given a binary, double-precision file containing lat, lon, gravity values from a survey, and you want to split it into profiles named *survey_###.txt* (when gap exceeds 100 km). Try this:

```
gmt splitxyz survey.bin -Nsurvey_%03d.txt -V -gd100k -D100 -: -fg -bi3d
```

1.82.8 See Also

gmt, filter1d, mgd77list, pswiggle

1.83 surface

surface - Grid table data using adjustable tension continuous curvature splines

1.83.1 Synopsis

```
surface [ table ] -Goutputfile.nc -Iinc[unit][=|+][yinc[unit][=|+]] -R[unit]xmin/xmax/ymin/ymax[r]
[ -Aaspect_ratio ] [ -Cconvergence_limit ] [ -Llower ] [ -Uupper ] [ -Nmax_iterations ] [ -Q
] [ -Ssearch_radius[mls] ] [ -T[ilb] ] tension_factor [ -V[level] ] [ -Zover-relaxation_factor ] [ -
acol=name[...] ] [ -b[ncols][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle]
] [ -i[cols][l][sscale][offset][...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.83.2 Description

surface reads randomly-spaced (x,y,z) triples from standard input [or *table*] and produces a binary grid file of gridded values z(x,y) by solving:

$$(1 - T) * L (L (z)) + T * L (z) = 0$$

where T is a tension factor between 0 and 1, and L indicates the Laplacian operator. T = 0 gives the “minimum curvature” solution which is equivalent to SuperMISP and the ISM packages. Minimum curvature can cause undesired oscillations and false local maxima or minima (See Smith and Wessel, 1990), and you may wish to use T > 0 to suppress these effects. Experience suggests T ~ 0.25 usually looks good for potential field data and T should be larger (T ~ 0.35) for steep topography data. T = 1 gives a harmonic surface (no maxima or minima are possible except at control data points). It is recommended that the user pre-process the data with **blockmean**, **blockmedian**, or **blockmode** to avoid spatial aliasing and eliminate redundant data. You may impose lower and/or upper bounds on the solution. These may be entered in the form of a fixed value, a grid with values, or simply be the minimum/maximum input data values. Natural boundary conditions are applied at the edges, except for geographic data with 360-degree range where we apply periodic boundary conditions in the longitude direction.

1.83.3 Required Arguments

-G*outputfile.nc* Output file name. Output is a binary 2-D *.nc* file. Note that the smallest grid dimension must be at least 4.

-I*inc[unit][=|+][yinc[unit][=|+]]* *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the

conversion depends on *PROJ_ELLIPSOID*). If *ly_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

1.83.4 Optional Arguments

table One or more ASCII (or binary, see **-bi[ncols][type]**) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Aspect_ratio Aspect ratio. If desired, grid anisotropy can be added to the equations. Enter *aspect_ratio*, where $dy = dx / aspect_ratio$ relates the grid dimensions. [Default = 1 assumes isotropic grid.]

-Cconvergence_limit Convergence limit. Iteration is assumed to have converged when the maximum absolute change in any grid value is less than *convergence_limit*. (Units same as data *z* units). [Default is scaled to 0.1 percent of typical gradient in input data.]

-Llower and -Luupper Impose limits on the output solution. *lower* sets the lower bound. *lower* can be the name of a grid file with lower bound values, a fixed value, **d** to set to minimum input value, or **u** for unconstrained [Default]. *upper* sets the upper bound and can be the name of a grid file with upper bound values, a fixed value, **d** to set to maximum input value, or **u** for unconstrained [Default]. Grid files used to set the limits may contain NaNs. In the presence of NaNs, the limit of a node masked with NaN is unconstrained.

-Nmax_iterations Number of iterations. Iteration will cease when *convergence_limit* is reached or when number of iterations reaches *max_iterations*. [Default is 250.]

-Q Suggest grid dimensions which have a highly composite greatest common factor. This allows surface to use several intermediate steps in the solution, yielding faster run times and better results. The sizes suggested by **-Q** can be achieved by altering **-R** and/or **-I**. You can recover the **-R** and **-I** you want later by using **grdsample** or **grdcut** on the output of **surface**.

-Ssearch_radius[mls] Search radius. Enter *search_radius* in same units as *x,y* data; append **m** to indicate arc minutes or **s** for arc seconds. This is used to initialize the grid before the first iteration; it is not worth the time unless the grid lattice is prime and cannot have regional stages. [Default = 0.0 and no search is made.]

-T[ilb] [tension_factor] Tension factor[s]. These must be between 0 and 1. Tension may be used in the interior solution (above equation, where it suppresses spurious oscillations) and in the boundary conditions (where it tends to flatten the solution approaching the edges). Using zero for both values results in a minimum curvature surface with free edges, i.e., a natural bicubic spline. Use **-Titension_factor** to set interior tension, and **-Tbtension_factor** to set boundary tension. If you do not prepend **i** or **b**, both will be set to the same value. [Default = 0 for both gives minimum curvature solution.]

-V[level] (*more ...*) Select verbosity level [c]. **-V3** will report the convergence after each iteration; **-V** will report only after each regional grid is converged.

- Zover-relaxation_factor** Over-relaxation factor. This parameter is used to accelerate the convergence; it is a number between 1 and 2. A value of 1 iterates the equations exactly, and will always assure stable convergence. Larger values overestimate the incremental changes during convergence, and will reach a solution more rapidly but may become unstable. If you use a large value for this factor, it is a good idea to monitor each iteration with the **-VI** option. [Default = 1.4 converges quickly and is almost always stable.]
- acol=name[...]** (*more ...*) Set aspatial column associations *col=name*.
- bi[ncols][type]** (*more ...*) Select binary input. [Default is 3 input columns].
- f[ilo]colinfo** (*more ...*) Specify data types of input and/or output columns.
- h[ilo][n][+c][+d][+rremark][+rtitle]** (*more ...*) Skip or produce header record(s). Not used with binary data.
- icols[l][sscale][ooffset][,...]** (*more ...*) Select input columns (0 is first column).
- :[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.83.5 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.83.6 Examples

To grid 5 by 5 minute gravity block means from the ASCII data in `hawaii_5x5.xy`, using a *tension_factor* = 0.25, a *convergence_limit* = 0.1 milligal, writing the result to a file called `hawaii_grd.nc`, and monitoring each iteration, try:

```
gmt surface hawaii_5x5.xy -R198/208/18/25 -I5m -Ghawaii_grd.nc -T0.25 -C0.1 -VI
```

1.83.7 Bugs

surface will complain when more than one data point is found for any node and suggest that you run **blockmean**, **blockmedian**, or *blockmode* first. If you did run **blockm*** and still get this message it usually means that your grid spacing is so small that you need more decimals in the output format used

by **blockm***. You may specify more decimal places by editing the parameter **FORMAT_FLOAT_OUT** in your *gmt.conf* file prior to running **blockm***, or choose binary input and/or output using single or double precision storage.

Note that only gridline registration is possible with **surface**. If you need a pixel-registered grid you can resample a gridline registered grid using *grdsample -T*.

1.83.8 See Also

blockmean, blockmedian, blockmode, gmt, greenspline, nearneighbor, triangulate, sphtriangulate

1.83.9 References

Smith, W. H. F. and P. Wessel, 1990, Gridding with continuous curvature splines in tension, *Geophysics*, 55, 293-305.

1.84 trend1d

trend1d - Fit a [weighted] [robust] polynomial [or Fourier] model for $y = f(x)$ to *xy[w]* data

1.84.1 Synopsis

trend1d [*table*] **-F***xymrwlp* **-N**[*f*]*n_model*[**r**] [*xy[w]file*] [**-C***condition_number*] [**-I**[*confidence_level*]] [**-V**[*level*]] [**-W**] [**-b**[*ilo*][*ncol*][**t***ype*][*w*][**+L**+**B**]] [**-f**[*ilo*]*colinfo*] [**-h**[*ilo*][*n*][**+c**][**+d**][**+r***emark*][**+r***itle*]] [**-i***cols*[**I**][*sscale*][**o***ffset*][*,...*]] [**-:**[*ilo*]]

Note: No space is allowed between the option flag and the associated arguments.

1.84.2 Description

trend1d reads *x,y* [and *w*] values from the first two [three] columns on standard input [or *file*] and fits a regression model $y = f(x) + e$ by [weighted] least squares. The functional form of $f(x)$ may be chosen as polynomial or Fourier, and the fit may be made robust by iterative reweighting of the data. The user may also search for the number of terms in $f(x)$ which significantly reduce the variance in *y*.

1.84.3 Required Arguments

-F*xymrwlp* Specify up to five letters from the set {**x y m r w**} in any order to create columns of ASCII [or binary] output. **x** = *x*, **y** = *y*, **m** = model $f(x)$, **r** = residual $y - m$, **w** = weight used in fitting. Alternatively choose **-Fp** (i.e., no other of the 5 letters) to output only the model coefficients (Polynomial).

-N[*f*]*n_model*[**r**] Specify the number of terms in the model, *n_model*, whether to fit a Fourier (**-Nf**) or polynomial [Default] model, and append **r** to do a robust fit. E.g., a robust quadratic model is **-N3r**.

1.84.4 Optional Arguments

- table** One or more ASCII [or binary, see **-bi**] files containing x,y [w] values in the first 2 [3] columns. If no files are specified, **trend1d** will read from standard input.
- Ccondition_number** Set the maximum allowed condition number for the matrix solution. **trend1d** fits a damped least squares model, retaining only that part of the eigenvalue spectrum such that the ratio of the largest eigenvalue to the smallest eigenvalue is *condition_#*. [Default: *condition_#* = 1.0e06.].
- I[confidence_level]** Iteratively increase the number of model parameters, starting at one, until *n_model* is reached or the reduction in variance of the model is not significant at the *confidence_level* level. You may set **-I** only, without an attached number; in this case the fit will be iterative with a default confidence level of 0.51. Or choose your own level between 0 and 1. See remarks section.
- V[level] (more ...)** Select verbosity level [c].
- W** Weights are supplied in input column 3. Do a weighted least squares fit [or start with these weights when doing the iterative robust fit]. [Default reads only the first 2 columns.]
- bi[ncols][type] (more ...)** Select binary input. [Default is 2 (or 3 if **-W** is set) columns].
- bo[ncols][type] (more ...)** Select binary output. [Default is 1-5 columns as given by **-F**].
- f[i]lo[colinfo (more ...)** Specify data types of input and/or output columns.
- h[i]lo[n][+c][+d][+rremark][+rtile] (more ...)** Skip or produce header record(s).
- i[cols][l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- :[i]lo (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.84.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.84.6 Remarks

If a Fourier model is selected, the domain of x will be shifted and scaled to $[-\pi, \pi]$ and the basis functions used will be $1, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots$. If a polynomial model is selected, the domain of x will be shifted and scaled to $[-1, 1]$ and the basis functions will be Chebyshev polynomials. These have a numerical advantage in the form of the matrix which must be inverted and allow more accurate solutions. The Chebyshev polynomial of degree n has $n+1$ extrema in $[-1, 1]$, at all of which its value is either -1 or $+1$. Therefore the magnitude of the polynomial model coefficients can be directly compared. NOTE: The stable model coefficients are Chebyshev coefficients. The corresponding polynomial coefficients in $a + bx + cxx + \dots$ are also given in Verbose mode but users must realize that they are NOT stable beyond degree 7 or 8. See Numerical Recipes for more discussion. For evaluating Chebyshev polynomials, see **gmtmath**.

The **-Nr** (robust) and **-I** (iterative) options evaluate the significance of the improvement in model misfit Chi-Squared by an F test. The default confidence limit is set at 0.51; it can be changed with the **-I** option. The user may be surprised to find that in most cases the reduction in variance achieved by increasing the number of terms in a model is not significant at a very high degree of confidence. For example, with 120 degrees of freedom, Chi-Squared must decrease by 26% or more to be significant at the 95% confidence level. If you want to keep iterating as long as Chi-Squared is decreasing, set *confidence_level* to zero.

A low confidence limit (such as the default value of 0.51) is needed to make the robust method work. This method iteratively reweights the data to reduce the influence of outliers. The weight is based on the Median Absolute Deviation and a formula from Huber [1964], and is 95% efficient when the model residuals have an outlier-free normal distribution. This means that the influence of outliers is reduced only slightly at each iteration; consequently the reduction in Chi-Squared is not very significant. If the procedure needs a few iterations to successfully attenuate their effect, the significance level of the F test must be kept low.

1.84.7 Examples

To remove a linear trend from `data.xy` by ordinary least squares, use:

```
gmt trend1d data.xy -Fxr -N2 > detrended_data.xy
```

To make the above linear trend robust with respect to outliers, use:

```
gmt trend1d data.xy -Fxr -N2r > detrended_data.xy
```

To find out how many terms (up to 20, say in a robust Fourier interpolant are significant in fitting `data.xy`, use:

```
gmt trend1d data.xy -Nf20r -I -V
```

1.84.8 See Also

gmt, *gmtmath*, *grdtrend*, *trend2d*

1.84.9 References

Huber, P. J., 1964, Robust estimation of a location parameter, *Ann. Math. Stat.*, **35**, 73-101.

Menke, W., 1989, Geophysical Data Analysis: Discrete Inverse Theory, Revised Edition, Academic Press, San Diego.

1.85 trend2d

trend2d - Fit a [weighted] [robust] polynomial model for $z = f(x,y)$ to $xyz[w]$ data

1.85.1 Synopsis

trend2d [*table*] **-Fxyzmrw** **-Nn_model**[**r**] [*xyz[w]file*] [**-Ccondition_number**] [**-I**[*confidence_level*]] [**-V**[*level*]] [**-W**] [[**-b**[*ilo*][*ncol*][**type**][*w*][**+L**+**B**]] [**-f**[*ilo*]*colinfo*] [**-h**[*ilo*][*n*][**+c**][**+d**][**+rremark**][**+rtitle**]] [**-i***cols*[**I**][*sscale*][**ooffset**][,...]] [**-:**[*ilo*]]

Note: No space is allowed between the option flag and the associated arguments.

1.85.2 Description

trend2d reads x,y,z [and w] values from the first three [four] columns on standard input [or *xyz[w]file*] and fits a regression model $z = f(x,y) + e$ by [weighted] least squares. The fit may be made robust by iterative reweighting of the data. The user may also search for the number of terms in $f(x,y)$ which significantly reduce the variance in z . *n_model* may be in [1,10] to fit a model of the following form (similar to `grdtrend`):

$$m1 + m2*x + m3*y + m4*x*y + m5*x*x + m6*y*y + m7*x*x*x + m8*x*x*y + m9*x*y*y + m10*y*y*y.$$

The user must specify **-Nn_model**, the number of model parameters to use; thus, **-N4** fits a bilinear trend, **-N6** a quadratic surface, and so on. Optionally, append **r** to perform a robust fit. In this case, the program will iteratively reweight the data based on a robust scale estimate, in order to converge to a solution insensitive to outliers. This may be handy when separating a “regional” field from a “residual” which should have non-zero mean, such as a local mountain on a regional surface.

1.85.3 Required Arguments

- Fxyzmrw** Specify up to six letters from the set {**x y z m r w**} in any order to create columns of ASCII [or binary] output. **x** = x , **y** = y , **z** = z , **m** = model $f(x,y)$, **r** = residual $z - m$, **w** = weight used in fitting.
- Nn_model**[**r**] Specify the number of terms in the model, *n_model*, and append **r** to do a robust fit. E.g., a robust bilinear model is **-N4r**.

1.85.4 Optional Arguments

- table** One or more ASCII [or binary, see **-bi**] files containing x,y,z [w] values in the first 3 [4] columns. If no files are specified, **trend2d** will read from standard input.
- Ccondition_number** Set the maximum allowed condition number for the matrix solution. **trend2d** fits a damped least squares model, retaining only that part of the eigenvalue spectrum such that the ratio of the largest eigenvalue to the smallest eigenvalue is *condition_#*. [Default: *condition_#* = 1.0e06.].
- I**[*confidence_level*] Iteratively increase the number of model parameters, starting at one, until *n_model* is reached or the reduction in variance of the model is not significant at the *confidence_level* level.

You may set **-I** only, without an attached number; in this case the fit will be iterative with a default confidence level of 0.51. Or choose your own level between 0 and 1. See remarks section.

- V[level] (more ...)** Select verbosity level [c].
- W** Weights are supplied in input column 4. Do a weighted least squares fit [or start with these weights when doing the iterative robust fit]. [Default reads only the first 3 columns.]
- bi[ncols][type] (more ...)** Select binary input. [Default is 3 (or 4 if **-W** is set) input columns].
- bo[ncols][type] (more ...)** Select binary output. [Default is 1-6 columns as set by **-F**].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- icols[l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
- :[ilo] (more ...)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.85.5 Remarks

The domain of x and y will be shifted and scaled to $[-1, 1]$ and the basis functions are built from Chebyshev polynomials. These have a numerical advantage in the form of the matrix which must be inverted and allow more accurate solutions. In many applications of **trend2d** the user has data located approximately along a line in the x,y plane which makes an angle with the x axis (such as data collected along a road or ship track). In this case the accuracy could be improved by a rotation of the x,y axes. **trend2d** does not search for such a rotation; instead, it may find that the matrix problem has deficient rank. However, the solution is computed using the generalized inverse and should still work out OK. The user should check the results graphically if **trend2d** shows deficient rank. NOTE: The model parameters listed with **-V** are Chebyshev coefficients; they are not numerically equivalent to the m 's in the equation described above. The description above is to allow the user to match **-N** with the order of the polynomial surface. For evaluating Chebyshev polynomials, see [grdmath](#).

The **-Nn_modelr** (robust) and **-I** (iterative) options evaluate the significance of the improvement in model misfit Chi-Squared by an F test. The default confidence limit is set at 0.51; it can be changed with the **-I** option. The user may be surprised to find that in most cases the reduction in variance achieved by increasing the number of terms in a model is not significant at a very high degree of confidence. For example, with 120 degrees of freedom, Chi-Squared must decrease by 26% or more to be significant at the 95% confidence level. If you want to keep iterating as long as Chi-Squared is decreasing, set *confidence_level* to zero.

A low confidence limit (such as the default value of 0.51) is needed to make the robust method work. This method iteratively reweights the data to reduce the influence of outliers. The weight is based on the Median Absolute Deviation and a formula from Huber [1964], and is 95% efficient when the model

residuals have an outlier-free normal distribution. This means that the influence of outliers is reduced only slightly at each iteration; consequently the reduction in Chi-Squared is not very significant. If the procedure needs a few iterations to successfully attenuate their effect, the significance level of the F test must be kept low.

1.85.6 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.85.7 Examples

To remove a planar trend from data.xyz by ordinary least squares, use:

```
gmt trend2d data.xyz -Fxyr -N2 > detrended_data.xyz
```

To make the above planar trend robust with respect to outliers, use:

```
gmt trend2d data.zxy -Fxyr -N2r > detrended_data.xyz
```

To find out how many terms (up to 10 in a robust interpolant are significant in fitting data.xyz, use:

```
gmt trend2d data.xyz -N10r -I -V
```

1.85.8 See Also

gmt, *grdmath*, *grdtrend*, *trend1d*

1.85.9 References

Huber, P. J., 1964, Robust estimation of a location parameter, *Ann. Math. Stat.*, **35**, 73-101.

Menke, W., 1989, Geophysical Data Analysis: Discrete Inverse Theory, Revised Edition, Academic Press, San Diego.

1.86 triangulate

triangulate - Do optimal (Delaunay) triangulation and gridding of Cartesian table data [method]

1.86.1 Synopsis

```
triangulate [ table ] [ -Dxly ] [ -Empty ] [ -Ggrdfile ] [ -Ixinc[unit][=+][lyinc[unit][=+]] ] [ -Jparameters ] [ -M ] [ -N ] [ -Q ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ -V[level] ] [ -Z ] [ -b[i]lo][ncol][type][w][+L+B] ] [ -f[i]lo]colinfo ] [ -h[i]lo][n][+c][+d][+rremark][+rtitle] ] [ -i ] [ -I ] [ -s[scale][ooffset][,...] ] [ -r ] [ -:[i]lo ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.86.2 Description

triangulate reads one or more ASCII [or binary] files (or standard input) containing x,y[,z] and performs Delaunay triangulation, i.e., it find how the points should be connected to give the most equilateral triangulation possible. If a map projection (give **-R** and **-J**) is chosen then it is applied before the triangulation is calculated. By default, the output is triplets of point id numbers that make up each triangle and is written to standard output. The id numbers refer to the points position (line number, starting at 0 for the first line) in the input file. As an option, you may choose to create a multiple segment file that can be piped through *psxy* to draw the triangulation network. If **-G -I** are set a grid will be calculated based on the surface defined by the planar triangles. The actual algorithm used in the triangulations is either that of Watson [1982] [Default] or Shewchuk [1996] (if installed; type **triangulate -** to see which method is selected). This choice is made during the GMT installation.

1.86.3 Required Arguments

None.

1.86.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Dxly Take either the *x*- or *y*-derivatives of surface represented by the planar facets (only used when **-G** is set).

-Empty Set the value assigned to empty nodes when **-G** is set [NaN].

-Ggridfile Use triangulation to grid the data onto an even grid (specified with **-R -I**). Append the name of the output grid file. The interpolation is performed in the original coordinates, so if your triangles are close to the poles you are better off projecting all data to a local coordinate system before using **triangulate** (this is true of all gridding routines).

-I *x_inc* [and optionally *y_inc*] sets the grid size for optional grid output (see **-G**). Append **m** to indicate arc minutes or **s** to indicate arc seconds.

-Jparameters (*more ...*) Select map projection.

-M Output triangulation network as multiple line segments separated by a segment header record.

-N Used in conjunction with **-G** to also write the triplets of the ids of all the Delaunay vertices [Default only writes the grid].

-Q Output the edges of the Voronoi cells instead [Default is Delaunay triangle edges]. Requires **-R** and is only available if linked with the Shewchuk [1996] library. Note that **-Z** is ignored on output.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest.

-S Output triangles as polygon segments separated by a segment header record. Requires Delaunay triangulation.

-V[level] (*more ...*) Select verbosity level [*c*].

- Z** Controls whether we read (x,y) or (x,y,z) data and if z should be output when **-M** or **-S** are used [Read (x,y) only].
- bi[*ncols*][*type*] (*more ...*)** Select binary input. [Default is 2 input columns].
- bo[*ncols*][*type*] (*more ...*)** Select binary output. [Default is same as input]. Node ids are stored as double triplets.
- f[*ilo*]*colinfo* (*more ...*)** Specify data types of input and/or output columns.
- h[*ilo*][*n*][*+c*][*+d*][*+rremark*][*+rtitle*] (*more ...*)** Skip or produce header record(s).
- i[*cols*][*sscale*][*offset*][*,...*] (*more ...*)** Select input columns (0 is first column).
- r (*more ...*)** Set pixel node registration [gridline]. (Only valid with **-G**).
- :[*ilo*] (*more ...*)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

1.86.5 Ascii Format Precision

The ASCII output formats of numerical data are controlled by parameters in your *gmt.conf* file. Longitude and latitude are formatted according to *FORMAT_GEO_OUT*, whereas other values are formatted according to *FORMAT_FLOAT_OUT*. Be aware that the format in effect can lead to loss of precision in the output, which can lead to various problems downstream. If you find the output is not written with enough precision, consider switching to binary output (**-bo** if available) or specify more decimals using the *FORMAT_FLOAT_OUT* setting.

1.86.6 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.86.7 Examples

To triangulate the points in the file *samples.xyz*, store the triangle information in a binary file, and make a grid for the given area and spacing, use

```
gmt triangulate samples.xyz -bo -R0/30/0/30 -I2 -Gsurf.nc > samples.ijk
```

To draw the optimal Delaunay triangulation network based on the same file using a 15-cm-wide Mercator map, use

```
gmt triangulate samples.xyz -M -R-100/-90/30/34 -JM15c | gmt psxy \  
-R-100/-90/30/34 -JM15c -W0.5p -B1 > network.ps
```

To instead plot the Voronoi cell outlines, try

```
gmt triangulate samples.xyz -M -Q -R-100/-90/30/34 -JM15c | \  
gmt psxy -R-100/-90/30/34 -JM15c -W0.5p -B1 > cells.ps
```

1.86.8 See Also

gmt, *greenspline*, *nearneighbor*, *pscontour*, *sphinterpolate*, *sphtriangulate*, *surface*

1.86.9 References

Watson, D. F., 1982, Acord: Automatic contouring of raw data, *Comp. & Geosci.*, **8**, 97-101.

Shewchuk, J. R., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, First Workshop on Applied Computational Geometry (Philadelphia, PA), 124-133, ACM, May 1996.

[Shewchuk's Homepage](#)

1.87 xyz2grd

xyz2grd - Convert data table to a grid file

1.87.1 Synopsis

```
xyz2grd [ table ] -Ggrdfile -Ixinc[unit][=|+][/yinc[unit][=|+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -A[fillmlnr|rs|ulz] ] [ -Dxname/yname/zname/scale/offset/invalid/title/remark ] [ -Nnodata ] [ -S[zfile] ] [ -V[level] ] [ -Z[flags] ] [ -bi[ncols][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -r ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

1.87.2 Description

xyz2grd reads one or more z or xyz tables and creates a binary grid file. **xyz2grd** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the mean value. As an option (using **-Z**), a 1-column z-table may be read assuming all nodes are present (z-tables can be in organized in a number of formats, see **-Z** below.)

1.87.3 Required Arguments

- G***grdfile* *grdfile* is the name of the binary output grid file. (See GRID FILE FORMAT below.)
- I***inc*[*unit*][=|+][*yinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.
- R**[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

1.87.4 Optional Arguments

- table** One or more ASCII [or binary, see **-bi**] files holding *z* or (*x,y,z*) values. The *xyz* triplets do not have to be sorted. One-column *z* tables must be sorted and the **-Z** must be set.
- A**[*fillm|nlr|slulz*] By default we will calculate mean values if multiple entries fall on the same node. Use **-A** to change this behavior, except it is ignored if **-Z** is given. Append **f** or **s** to simply keep the first or last data point that was assigned to each node. Append **l** or **u** to find the lowest (minimum) or upper (maximum) value at each node, respectively. Append **m** or **r** to compute mean or RMS value at each node, respectively. Append **n** to simply count the number of data points that were assigned to each node. Append **z** to sum multiple values that belong to the same node.
- D***xname/yname/zname/scale/offset/invalid/title/remark* Give values for *xname*, *yname*, *zname* (give the names of those variables and in square bracket their units, e.g., “distance [km]”), *scale* (to multiply grid values after read [normally 1]), *offset* (to add to grid after scaling [normally 0]), *invalid* (a value to represent missing data [NaN]), *title* (anything you like), and *remark* (anything you like). To leave some of these values untouched, leave field blank. Empty fields in the end may be skipped. Alternatively, to allow “/” to be part of one of the values, use any non-alphanumeric character (and not the equal sign) as separator by both starting and ending with it. For example: **-D:xname:yname:zname:scale:offset:invalid:title:remark:** Use quotes to group texts with more than one word. Note that for geographic grids (**-fg**) *xname* and *yname* are set automatically.
- N***nodata* No data. Set nodes with no input *xyz* triplet to this value [Default is NaN]. For *z*-tables, this option is used to replace *z*-values that equal *nodata* with NaN.
- S**[*zfile*] Swap the byte-order of the input only. No grid file is produced. You must also supply the **-Z** option. The output is written to *zfile* (or stdout if not supplied).
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- Z**[*flags*] Read a 1-column ASCII [or binary] table. This assumes that all the nodes are present and sorted according to specified ordering convention contained in *flags*. If incoming data represents rows, make *flags* start with **T**(op) if first row is *y* = *y*max or **B**(ottom) if first row is *y* = *y*min. Then, append **L** or **R** to indicate that first element is at left or right end of row. Likewise for

column formats: start with **L** or **R** to position first column, and then append **T** or **B** to position first element in a row. Note: These two row/column indicators are only required for grids; for other tables they do not apply. For gridline registered grids: If data are periodic in x but the incoming data do not contain the (redundant) column at $x = x_{max}$, append **x**. For data periodic in y without redundant row at $y = y_{max}$, append **y**. Append *sn* to skip the first *n* number of bytes (probably a header). If the byte-order or the words needs to be swapped, append **w**. Select one of several data types (all binary except **a**):

A ASCII representation of one or more floating point values per record

a ASCII representation of a single item per record

c int8_t, signed 1-byte character

u uint8_t, unsigned 1-byte character

h int16_t, signed 2-byte integer

H uint16_t, unsigned 2-byte integer

i int32_t, signed 4-byte integer

I uint32_t, unsigned 4-byte integer

l int64_t, long (8-byte) integer

L uint64_t, unsigned long (8-byte) integer

f 4-byte floating point single precision

d 8-byte floating point double precision

Default format is scanline orientation of ASCII numbers: **-ZTLa**. Note that **-Z** only applies to 1-column input. The difference between **A** and **a** is that the latter can decode both *dateTclock* and *ddd:mm:ss[.xx]* formats while the former is strictly for regular floating point values.

-bi[ncols][type] (more ...) Select binary input. [Default is 3 input columns]. This option only applies to xyz input files; see **-Z** for z tables.

-f[ilo]colinfo (more ...) Specify data types of input and/or output columns.

-h[ilo][n][+c][+d][+rremark][+rtitle] (more ...) Skip or produce header record(s). Not used with binary data.

-icol[s][l][sscale][offset][,...] (more ...) Select input columns (0 is first column).

-r (more ...) Set pixel node registration [gridline].

-:[ilo] (more ...) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

1.87.5 Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

1.87.6 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. See *gridreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When writing a netCDF file, the grid is stored by default with the variable name “z”. To specify another variable name *varname*, append `?varname` to the file name. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes.

1.87.7 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x** **-f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

1.87.8 Swapping Limitations

All data types can be read, even 64-bit integers, but internally grids are stored using floats. Hence, integer values exceeding the float type’s 23-bit mantissa may not be represented exactly. When **-S** is used no grids are implied and we read data into an intermediate double container. This means all but 64-bit integers can be represented using the double type’s 53-bit mantissa.

1.87.9 Examples

To create a grid file from the ASCII data in *hawaii_grv.xyz*, use

```
gmt xyz2grd hawaii_grv.xyz -Ddegree/degree/mGal/1/0/"Hawaiian Gravity"/"GRS-80 Ellipsoid used"
-Ghawaii_grv_new.nc -R198/208/18/25 -I5m -V
```

To create a grid file from the raw binary (3-column, single-precision scanline-oriented data *raw.b*), use

```
gmt xyz2grd raw.b -Dm/m/m/1/0 -Graw.nc -R0/100/0/100 -I1 -V -Z -bi3f
```

To make a grid file from the raw binary USGS DEM (short integer scanline-oriented data topo30. on the NGDC global relief Data CD-ROM, with values of -9999 indicate missing data, one must on some machine reverse the byte-order. On such machines (like Sun, use

```
gmt xyz2grd topo30. -Dm/m/m/1/0 -Gustopo.nc -R234/294/24/50 -I30s -N-9999 -B -  
ZTLhw
```

Say you have received a binary file with 4-byte floating points that were written on a machine of different byte-order than yours. You can swap the byte-order with

```
gmt xyz2grd floats.bin -Snew_floats.bin -V -Zf
```

1.87.10 See Also

gmt, grd2xyz, grdedit, grdreformat

Supplementary modules

2.1 gshhg

`gshhg` - Extract data tables from binary GSHHG or WDBII data files

2.1.1 Synopsis

```
gshhg binaryfile.b [ -Amin ] [ -G ] [ -Iid ] [ -L ] [ -Nlevel ] [ -Qeli ] [ -bo[ncols][type][w][+L+B] ] [ -ocols[,...] ] > asciifile.txt
```

Note: No space is allowed between the option flag and the associated arguments.

2.1.2 Description

`gshhg` reads the binary coastline (GSHHG) or political boundary or river (WDBII) files and extracts an ASCII listing. It automatically handles byte-swabbing between different architectures. Optionally, only segment header info can be displayed. The header info has the format *ID npoints hierarchical-level source area f_area west east south north container ancestor*, where hierarchical levels for coastline polygons go from 1 (shoreline) to 4 (lake inside island inside lake inside land). Source is either W (World Vector Shoreline) or C (CIA World Data Bank II); lower case is used if a lake is a river-lake. The *west east south north* is the enclosing rectangle, *area* is the polygon area in km² while *f_area* is the actual area of the ancestor polygon, *container* is the ID of the polygon that contains this polygon (-1 if none), and *ancestor* is the ID of the polygon in the full resolution set that was reduced to yield this polygon (-1 if full resolution since there is no ancestor). For line data the header is simply *ID npoints hierarchical-level source west east south north*. For more information about the file formats, see TECHNICAL INFORMATION below.

2.1.3 Required Arguments

binaryfile.b GSHHG or WDBII binary data file as distributed with the GSHHG data supplement. Any of the 5 standard resolutions (full, high, intermediate, low, crude) can be used.

2.1.4 Optional Arguments

-Amin Only output information for the polygon if its area equals or exceeds *min* [Default outputs all polygons].

- G** Write output that can be imported into GNU Octave or Matlab by ending segments with a NaN-record.
- Id** Only output information for the polygon that matches *id*. Use **-Ic** to get all the continents only [Default outputs all polygons].
- L** Only output a listing of polygon or line segment headers [Default outputs headers and data records].
- N** Only output features whose level matches the given *level* [Default will output all levels].
- Qeli** Control what to do with river-lakes (river sections large enough to be stored as closed polygons). Use **-Qe** to exclude them and **-Qi** to exclude everything else instead [Default outputs all polygons].
- bo[ncols][type]** (*more ...*) Select binary output.
- ocol[s],[...]** (*more ...*) Select output columns (0 is first column).

2.1.5 Examples

To convert the entire intermediate GSHHG binary data to ASCII files for Octave/Mathlab, run

```
gmt gshhg gshhs_i.b --IO_SEGMENT_MARKER=N > gshhs_i.txt
```

To only get a listing of the headers for the river data set at full resolution, try

```
gmt gshhg wdb_rivers_f.b -L > riverlisting.txt
```

To only extract lakes, excluding river-lakes, from the high resolution file, try

```
gmt gshhg gshhs_h.b -Ee -N2 > all_lakes.txt
```

2.1.6 Technical Information

Users who wish to access the GSHHG or WDBII data directly from their custom programs should consult the `gshhg.c` and `gshhg.h` source code and familiarize themselves with the data format and how various information flags are packed into a single 4-byte integer. While we do not maintain any Octave/Matlab code to read these files we are aware that both Mathworks and IDL have made such tools available to their users. However, they tend not to update their code and our file structure has evolved considerably over time, breaking their code. Here, some general technical comments on the binary data files are given. **GSHHG**: These files contain completely closed polygons of continents and islands (level 1), lakes (level 2), islands-in-lakes (level 3) and ponds-in-islands-in-lakes (level 4); a particular level can be extracted using the **-N** option. Continents are identified as the first 6 polygons and can be extracted via the **-Ic** option. The IDs for the continents are Eurasia (0), Africa (1), North America (2), South America (3), Antarctica (4), and Australia (5). Files are sorted on area from large to small. There are two sub-groups for level 2: Regular lakes and the so-called “river-lakes”, the latter being sections of a river that are so wide to warrant a polygon representation. These river-lakes are flagged in the header (also see **-Q**). All five resolutions are free of self-intersections. Areas of all features have been computed using a Lambert azimuthal equal-area projection centered on the polygon centroids, using WGS-84 as the ellipsoid. GMT use the GSHHG as a starting point but then partition the polygons into pieces using a resolution-dependent binning system; parts of the world are then rebuilt into closed polygons on the fly as needed. For more information on GSHHG processing, see Wessel and Smith (1996). **WDBII**. These files contain sets of line segments not necessarily in any particular order. Thus, it is not possible to extract information pertaining to just one river or one country. Furthermore, the 4 lower resolutions derive directly from the full resolution by application of the Douglas-Peucker algorithm (see `gshhg_dp`), hence self-intersections are increasingly likely as the resolution is degraded. Note that the river-lakes included

in GSHHG are also duplicated in the WDBII river files so that each data set can be a stand-alone representation. Users who wish to access both data sets can recognize the river-lakes features by examining the header structure (see the source code for details); they are also the only closed polygons in the WDBII river file. There are many levels (classes) in the river file: River-lakes (0), Permanent major rivers (1), Additional major rivers (2), Additional rivers (3), Minor rivers (4), Intermittent rivers – major (6), Intermittent rivers – additional (7), Intermittent rivers – minor (8), Major canals (10), Canals of lesser importance (11), and Canals – irrigation type (12). For the border file there are three levels: National boundaries (1), Internal domestic boundaries (2), and international maritime boundaries (3). Individual levels or classes may be extracted via **-N**.

2.1.7 References

Douglas, D. H., and T. K. Peucker, 1973, Algorithms for the reduction of the number of points required to represent a digitized line of its caricature, *Can. Cartogr.*, 10, 112-122.

Gorny, A. J., 1977, *World Data Bank II General User Guide Rep. PB 271869*, 10pp, Central Intelligence Agency, Washington, DC.

Soluri, E. A., and V. A. Woodson, 1990, World Vector Shoreline, *Int. Hydrograph. Rev.*, LXVII(1), 27-35.

Wessel, P., and W. H. F. Smith, 1996, A global, self-consistent, hierarchical, high-resolution shoreline database, *J. Geophys. Res.*, 101(B4), 8741-8743.*

2.1.8 See Also

gmt

2.2 img2grd

img2grd - Extract subset of img file in Mercator or Geographic format

2.2.1 Synopsis

```
img2grd imgfile -Ggrdfile -R[unit]xmin/xmax/ymin/ymax[r] -Ttype [ -C ] [ -D[minlat/maxlat] ] [ -E ] [ -Iminutes ] [ -M ] [ -Nnavg ] [ -S[scale] ] [ -V[level] ] [ -Wmaxlon ] [ -n[blclln][+a][+bBC][+tthreshold] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.2.2 Description

img2grd reads an img format file, extracts a subset, and writes it to a grid file. The **-M** option dictates whether or not the Spherical Mercator projection of the img file is preserved or if a Geographic grid should be written by undoing the Mercator projection. If geographic grid is selected you can also request a resampling onto the exact **-R** given.

2.2.3 Required Arguments

imgfile A Mercator img format file such as the marine gravity or seafloor topography fields estimated from satellite altimeter data by Sandwell and Smith. If the user has set an environment variable **\$GMT_DATADIR**, then **img2grd** will try to find *imgfile* in **\$GMT_DATADIR**; else it will try to open *imgfile* directly.

-Ggrdfile *grdfile* is the name of the output grid file.

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

2.2.4 Optional Arguments

-C Set the x and y Mercator coordinates relative to projection center [Default is relative to lower left corner of grid]. Requires **-M**.

-D[*minlat/maxlat*] Use the extended latitude range -80.738/+80.738. Alternatively, append *minlat/maxlat* as the latitude extent of the input img file. [Default is -72.006/72.006]. Not usually required since we can determine the extent from inspection of the file size.

-E Can be used when **-M** is not set to force the final grid to have the exact same region as requested with **-R**. By default, the final region is a direct projection of the original Mercator region and will typically extend slightly beyond the requested latitude range, and furthermore the grid increment in latitude does not match the longitude increment. However, the extra resampling introduces small interpolation errors and should only be used if the output grid must match the requested region and have *x_inc* = *y_inc*. In this case the region set by **-R** must be given in multiples of the increment (.e.g. **-R0/45/45/72**).

-I Indicate *minutes* as the width of an input img pixel in minutes of longitude. [Default is 2.0]. Not usually required since we can determine the pixel size from inspection of the size.

-M Output a Spherical Mercator grid [Default is a geographic lon/lat grid]. The Spherical Mercator projection of the img file is preserved, so that the region **-R** set by the user is modified slightly; the modified region corresponds to the edges of pixels [or groups of *navg* pixels]. The grid file header is set so that the x and y axis lengths represent distance from the west and south edges of the image, measured in user default units, with **-Jm1** and the adjusted **-R**. By setting the default **PROJ_ELLIPSOID** = Sphere, the user can make overlays with the adjusted **-R** so that they match. See **EXAMPLES** below. The adjusted **-R** is also written in the *grdheader* remark, so it can be found later. See **-C** to set coordinates relative to projection center.

-Nnavg Average the values in the input img pixels into *navg* by *navg* squares, and create one output pixel for each such square. If used with **-T3** it will report an average constraint between 0 and 1. If used with **-T2** the output will be average data value or NaN according to whether average

constraint is > 0.5 . *navg* must evenly divide into the dimensions of the imgfile in pixels. [Default *l* does no averaging].

- S[*scale*]** Multiply the img file values by *scale* before storing in grid file. [Default is 1.0]. For recent img files: img topo files are stored in (corrected) meters [-S1]; free-air gravity files in mGal*10 [-S0.1 to get mGal]; vertical deflection files in microradians*10 [-S0.1 to get microradians], vertical gravity gradient files in Eotvos*50 [-S0.02 to get Eotvos, or -S0.002 to get mGal/km]). If no *scale* is given we try to determine the scale by examining the file name for clues.
- T*type*** *type* handles the encoding of constraint information. *type* = 0 indicates that no such information is encoded in the img file (used for pre-1995 versions of the gravity data) and gets all data. *type* > 0 indicates that constraint information is encoded (1995 and later (current) versions of the img files) so that one may produce a grid file as follows: **-T1** gets data values at all points, **-T2** gets data values at constrained points and NaN at interpolated points; **-T3** gets 1 at constrained points and 0 at interpolated points [Default is 1].
- V[*level*] (*more ...*)** Select verbosity level [*c*]. Particularly recommended here, as it is helpful to see how the coordinates are adjusted.
- W*maxlon*** Indicate *maxlon* as the maximum longitude extent of the input img file. Versions since 1995 have had *maxlon* = 360.0, while some earlier files had *maxlon* = 390.0. [Default is 360.0].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.2.5 Geographic Examples

The **-M** option should be excluded if you need the output grid to be in geographic coordinates. To extract data in the region **-R-40/40/-70/-30** from *world_grav.img.7.2* and reproject to yield geographic coordinates, you can try

```
img2grd world_grav.img.16.1 -Gmerc_grav.nc -R-40/40/-70/-30 -V
```

Because the latitude spacing in the img file is equidistant in Mercator units, the resulting grid will not match the specified **-R** exactly, and the latitude spacing will not equal the longitude spacing. If you need an exact match with your **-R** and the same spacing in longitude and latitude, use the **-E** option:

```
img2grd world_grav.img.16.1 -Gmerc_grav.nc -R-40/40/-70/-30 -E -V
```

2.2.6 Mercator Examples

Since the img files are in a Mercator projection, you should NOT extract a geographic grid if your plan is to make a Mercator map. If you did that you end of projecting and reprojecting the grid, losing short-wavelength detail. Better to use **-M** and plot the grid using a linear projection with the same scale as the desired Mercator projection (see GMT Example 29). To extract data in the region **-R-40/40/-70/-30** from *world_grav.img.7.2*, run

```
gmt img2grd -M world_grav.img.7.2 -Gmerc_grav.nc -R-40/40/-70/-30 -V
```

Note that the **-V** option tells us that the range was adjusted to **-R-40/40/-70.0004681551/-29.9945810754**. We can also use *grdinfo* to find that the grid file header shows its region to be **R0/80/0/67.9666667** This is the range of x,y we will get from a Spherical Mercator projection using **-R-40/40/-70.0004681551/-29.9945810754** and **-Jm1**. Thus, to take *ship.lonlatgrav* and use it to sample the *merc_grav.nc*, we can do this:

```
gmt set PROJ_ELLIPSOID Sphere

gmt mapproject -R-40/40/-70.0004681551/-29.9945810754 -Jm1i ship.lonlatgrav | \
  gmt grdtrack -Gmerc_grav.nc | gmt mapproject \
  -R-40/40/-70.0004681551/-29.9945810754 -Jm1i -I > ship.lonlatgravsat
```

It is recommended to use the above method of projecting and unprojecting the data in such an application, because then there is only one interpolation step (in *grdtrack*). If one first tries to convert the grid file to lon,lat and then sample it, there are two interpolation steps (in conversion and in sampling).

To make a lon,lat grid from the above grid we can use

```
gmt grdproject merc_grav.nc -R-40/40/-70.0004681551/-29.9945810754 -Jm1i -I -D2m -Ggrav.nc
```

In some cases this will not be easy as the **-R** in the two coordinate systems may not align well. When this happens, we can also use (in fact, it may be always better to use)

```
gmt grd2xyz merc_grav.nc | gmt mapproject \
  -R-40/40/-70.0004681551/-29.994581075 -Jm1i -I | \
  gmt surface -R-40/40/-70/70 -I2m -Ggrav.nc
```

To make a Mercator map of the above region, suppose our *gmt.conf* value for *PROJ_LENGTH_UNIT* is inch. Then since the above *merc_grav.nc* file is projected with **-Jm1i** it is 80 inches wide. We can make a map 8 inches wide by using **-Jx0.1i** on any map programs applied to this grid (e.g., *grdcontour*, *grdimage*, *grdview*), and then for overlays which work in lon,lat (e.g., *psxy*, *pscoast*) we can use the above adjusted **-R** and **-Jm0.1** to get the two systems to match up.

However, we can be smarter than this. Realizing that the input *img* file had pixels 2.0 minutes wide (or checking the *nx* and *ny* with *grdinfo merc_grav.nc*) we realize that *merc_grav.nc* used the full resolution of the *img* file and it has 2400 by 2039 pixels, and at 8 inches wide this is 300 pixels per inch. We decide we do not need that many and we will be satisfied with 100 pixels per inch, so we want to average the data into 3 by 3 squares. (If we want a contour plot we will probably choose to average the data much more (e.g., 6 by 6) to get smooth contours.) Since 2039 isn't divisible by 3 we will get a different adjusted **-R** this time:

```
gmt img2grd -M world_grav.img.7.2 -Gmerc_grav_2.nc -R-40/40/-70/-30 -N3 -V
```

This time we find the adjusted region is **-R-40/40/-70.023256525/-29.9368261101** and the output is 800 by 601 pixels, a better size for us. Now we can create an artificial illumination file for this using *grdgradient*:

```
gmt grdgradient merc_grav_2.nc -Gillum.nc -A0/270 -Ne0.6
```

and if we also have a *cpt* file called "grav.cpt" we can create a color shaded relief map like this:

```
gmt grdimage merc_grav_2.nc -Iillum.nc -Cgrav.cpt -Jx0.1i -K > map.ps
gmt psbasemap -R-40/40/-70.023256525/-29.9368261101 -Jm0.1i -Ba10 -O >> map.ps
```

Suppose you want to obtain only the constrained data values from an *img* file, in lat/lon coordinates. Then run **img2grd** with the **-T2** option, use *grd2xyz* to dump the values, pipe through *grep -v NaN* to eliminate NaNs, and pipe through *mapproject* with the inverse projection as above.

2.2.7 See Also

gmt

2.3 pscoupe

pscoupe - Plot cross-sections of focal mechanisms

2.3.1 Synopsis

```
pscoupe [ files ] -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] -Aparameters [ -B[pls]parameters ] [ -Efill ] [ -Fmode[args] ] [ -Gfill ] [ -K ] [ -L[pen] ] [ -M ] [ -N ] [ -O ] [ -Q ] [ -S<symbol><scale>[/d] ] [ -Tn ] [ -U[just/dx/dy]/[cllabel] ] [ -V[level] ] [ -Wpen ] [ -X[alcflr]/[x-shift][u] ] [ -Y[alcflr]/[y-shift][u] ] [ -Zcpt ] [ -ccopies ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.3.2 Description

pscoupe reads data values from *files* [or standard input] and generates PostScript code that will plot symbols, lines or polygons on a cross-section. Focal mechanisms may be specified and require additional columns of data. The PostScript code is written to standard output.

Unless **-Q** is used, new file is created with the new coordinates (**x**, **y**) and the mechanism (from lower focal half-sphere for horizontal plane, to half-sphere behind a vertical plane). When the plane is not horizontal, - north direction becomes upwards steepest descent direction of the plane (**u**) - east direction becomes strike direction of the plane (**s**) - down direction (= north^east) becomes **u**^**s** Axis angles are defined in the same way as in horizontal plane in the new system. Moment tensor (initially in **r**, **t**, **f** system that is up, south, east) is defined in (-**u**^**s**, -**u**, **s**) system.

2.3.3 Required Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-J*parameters* (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*dd*:*mm*[:*ss.xxx*][**W**|**E**|**S**|**N**] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. If frame is defined from cross-section parameters (see **-A** this option is not taken into account, but must be present.

-A selects the cross-section.

-Aa*lon1/lat1/lon2/lat2/dip/p_width/dmin/dmax*[**f**] *lon* and *lat* are the longitude and latitude of points 1 and 2 limiting the length of the cross-section. *dip* is the dip of the plane on which the cross-section is made. *p_width* is the width of the cross-section on each side of a vertical plane or above and under an oblique plane. *dmin* and *dmax* are the distances min and max from horizontal plane, along steepest descent direction. Add **f** to get the frame from the cross-section parameters.

-Ablon1/lat1/strike/p_length/dip/p_width/dmin/dmax[f**] *lon1* and *lat1* are the longitude and latitude of the beginning of the cross-section. *strike* is the azimuth of the direction of the cross-section. *p_length* is the length along which the cross-section is made. The other parameters are the same as for **-Aa** option.**

-Acx1/y1/x2/y2/dip/p_width/dmin/dmax[f**] The same as **-Aa** option with *x* and *y* cartesian coordinates.**

-Adx1/y1/strike/p_length/dip/p_width/dmin/dmax[f**] The same as **-Ab** option with *x* and *y* cartesian coordinates.**

-S selects the meaning of the columns in the data file and the figure to be plotted.

-Sascale[/*fontsize*[/*offset*[**u**]]] Focal mechanisms in Aki and Richards convention. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. The *scale* is the size for magnitude = 5 in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). Use the **-T** option to render the beach ball transparent by drawing only the nodal planes and the circumference. The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order)

3: depth of event in kilometers

4,5,6: strike, dip and rake

7: magnitude

8,9: not used; can be 0 0; allows use of the psmeca file format

10: text string to appear above the beach ball (default) or under (add **u**).

-Scscale Focal mechanisms in Harvard CMT convention. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. The *scale* is the size for magnitude = 5 (that is $M_0 = 4E+23$ dynes-cm.) in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). Use the **-T** option to render the beach ball transparent by drawing only the nodal planes and the circumference. The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order)

3: depth of event in kilometers

4,5,6: strike, dip, and slip of plane 1

7,8,9: strike, dip, and slip of plane 2

10,11: mantissa and exponent of moment in dyne-cm (if magnitude is uses instead of scalar moment, magnitude is in column 10 and 0 must be in column 11)

12,13: not used; can be 0 0; allows use of the psmeca file format

14: text string to appear above the beach ball (default) or under (add **u**).

-Spscale*[/fontsize[/offset[u]]]* Focal mechanisms given with partial data on both planes. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. The *scale* is the size for magnitude = 5 in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order)

3: depth

4,5: strike, dip of plane 1

6: strike of plane 2

7: must be -1/+1 for a normal/inverse fault

8: magnitude

9,10: not used; can be 0 0; allows use of the psmeca file format

11: text string to appear above the beach ball (default) or under (add **u**).

-Smdlzscale*[/fontsize[/offset[u]]]* Seismic moment tensor (Harvard CMT, with zero trace). *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. The *scale* is the size for magnitude = 5 (that is seismic scalar moment = 4E+23 dynes-cm) in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). (**-T0** option overlays best double couple transparently.)

-Sdscale*[/fontsize[/offset[u]]]* to plot the only double couple part of moment tensor.

-Szscales*[/fontsize[/offset[u]]]* to plot anisotropic part of moment tensor (zero trace). The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order)

3: depth of event in kilometers

4,5,6,7,8,9: mrr, mtt, mff, mrt, mrf, mtf in 10*exponent dynes-cm

10: exponent

11,12: Not used; can be 0 0; allows use of the psmeca file format

13: Text string to appear above the beach ball (default) or under (add **u**).

-Sxscales*[/fontsize[/offset[u]]]* Principal axis. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. The *scale* is the size for magnitude = 5 (that is seismic scalar moment = 4*10e+23 dynes-cm) in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). (**-T0** option overlays best double couple transparently.)

-Syscales*[/fontsize[/offset[u]]]* to plot the only double couple part of moment tensor.

-Stscale*[/fontsize[/offset[u]]]* to plot anisotropic part of moment tensor (zero trace). The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (**-:** option interchanges order)

3: depth of event in kilometers

4,5,6,7,8,9,10,11,12: value (in 10*exponent dynes-cm), azimuth, plunge of the T, N, and P axes.

13: exponent

14,15: longitude, latitude at which to place beach ball. Entries in these columns are necessary with the **-C** option. Using 0,0 in columns 9 and 10 will plot the beach ball at the longitude, latitude given in columns 1 and 2. The **-:** option will interchange the order of columns (1,2) and (9,10).

16: Text string to appear above the beach ball (optional).

2.3.4 Optional Arguments

-B[pls]parameters (*more ...*) Set map boundary intervals.

-Efill Sets color or fill pattern for extensive quadrants [Default is white].

-Fmode[args] Sets one or more attributes; repeatable. The various combinations are

-Fssymbol*[size[/fontsize[/offset[u]]]* selects a symbol instead of mechanism. Choose from the following: (**c**) circle, (**d**) diamond, (**i**) itriangle, (**s**) square, (**t**) triangle, (**x**) cross. *size* is the symbol size in **PROJ_LENGTH_UNIT** (unless **c**, **i**, **m**, or **p** is appended to indicate that the size information is in units of cm, inches, meters, or points, respectively). If *size* must be read, it must be in column 4 and the text string will start in column 5. Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (**-:** option interchanges order)

3: depth of event in kilometers

4: Text string to appear above the beach ball (default) or under (add **u**).

-Fa*[size][[/P_symbol[/T_symbol]]]* Computes and plots P and T axes with symbols. Optionally specify *size* and (separate) P and T axis symbols from the following: (**c**) circle, (**d**) diamond, (**h**) hexagon, (**i**) inverse triangle, (**p**) point, (**s**) square, (**t**) triangle, (**x**) cross. [Default: 6p/cc]

-Fefill Sets the color or fill pattern for the T axis symbol. [Default as set by **-E**]

-Fgfill Sets the color or fill pattern for the P axis symbol. [Default as set by **-G**]

-Fp[pen] Draws the P axis outline using current pen (see **-W**), or sets pen attributes.

-Fr[fill] Draw a box behind the label (if any). [Default fill is white]

-Ft[pen] Draws the T axis outline using current pen (see **-W**), or sets pen attributes.

-Gfill Sets color or fill pattern for compressional quadrants [Default is black].

-K (*more ...*) Do not finalize the PostScript plot.

-L[pen] Draws the “beach ball” outline using current pen (see **-W**) or sets pen attributes.

-M Same size for any magnitude.

- N Does **not** skip symbols that fall outside map border [Default plots points inside border only].
- O (*more ...*) Append to existing PostScript plot.
- P (*more ...*) Select “Portrait” plot orientation.
- Q Suppress the production of files with cross-section and mechanism information.
- T[*num_of_planes*] Plots the nodal planes and outlines the bubble which is transparent. If *num_of_planes* is 0: both nodal planes are plotted; 1: only the first nodal plane is plotted; 2: only the second nodal plane is plotted [Default: 0].
- U[*just/dx/dy*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*) Select verbosity level [c].
- W set pen attributes for text string or default pen attributes for fault plane edges. [Defaults: width = default, color = black, style = solid].
- X[*alcflr*][*x-shift*[u]]
- Y[*alcflr*][*y-shift*[u]] (*more ...*) Shift plot origin.
- Z*cptfile* Give a color palette file and let compressive part color be determined by the z-value in the third column.
- ccopies* (*more ...*) Specify number of plot copies [Default is 1].
- h[*ilo*][*n*][*+c*][*+d*][*+rremark*][*+rtitle*] (*more ...*) Skip or produce header record(s).
- icols*[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).
- :*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or **just** - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.3.5 See Also

gmt, *psbasemap*, *psmeca*, *psxy*

2.3.6 References

Bomford, G., Geodesy, 4th ed., Oxford University Press, 1980.

Aki, K. and P. Richards, Quantitative Seismology, Freeman, 1980.

F. A. Dahlen and Jeroen Tromp, Theoretical Seismology, Princeton, 1998, p.167. Definition of scalar moment.

Cliff Frohlich, Cliff's Nodes Concerning Plotting Nodal Lines for P, Sh and Sv

Seismological Research Letters, Volume 67, Number 1, January-February, 1996

Thorne Lay, Terry C. Wallace, Modern Global Seismology, Academic Press, 1995, p.384.

W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University press (routine jacobi)

2.3.7 Author

Genevieve Patau, *Laboratory of Seismogenesis* <<http://www.ipgp.fr/rech/sismogenese/>>, Institut de Physique du Globe de Paris, Departement de Sismologie, Paris, France

2.4 psmeca

psmeca - Plot focal mechanisms on maps

2.4.1 Synopsis

```
psmeca [ table ] -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] [ -B[pls]parameters ] [ -C[pen][Ppointsize] ] [ -Ddepmin/depmax ] [ -Efill] [ -Fmode[args] ] [ -Gfill] [ -K] [ -L[pen] ] [ -M] [ -N] [ -O] [ -P] [ -S<format><scale>[d] ] [ -Tnum_of_plane[pen] ] [ -U[just/dx/dy][cllabel] ] [ -V[level] ] [ -Wpen] [ -X[alcflr][x-shift][u] ] [ -Y[alcflr][y-shift][u] ] [ -Zcptfile] [ -ccopies] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][offset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.4.2 Description

psmeca reads data values from *files* [or standard input] and generates PostScript code that will plot focal mechanisms on a map. Most options are the same as for **psxy**. The PostScript code is written to standard output.

2.4.3 Required Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Jparameters (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-S<format><scale>[/d]

Selects the meaning of the columns in the data file. In order to use the same file to plot cross-sections, depth is in third column. Nevertheless, it is possible to use “old style” **psvelomeca** input files without depth in third column using the **-o** option.

-Sascale[/fontsize[/offset[**u**]]]

Focal mechanisms in Aki and Richards convention. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. Scale is the size for magnitude = 5 in inch (unless **c**, **i**, **m**, or **p** is appended). Use the **-T** option to render the beach ball transparent by drawing only the nodal planes and the circumference. The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Append **u** to have the text appear below the beach ball (default is above). Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order) **3**: depth of event in kilometers
4,5,6: strike, dip and rake in degrees **7**: magnitude **8,9**: longitude, latitude at which to place beach ball. Entries in these columns are necessary with the **-C** option. Using 0,0 in columns 8 and 9 will plot the beach ball at the longitude, latitude given in columns 1 and 2. The -: option will interchange the order of columns (1,2) and (8,9). **10**: Text string to appear above or below the beach ball (optional).

-Scscale[/fontsize[/offset[**u**]]]

Focal mechanisms in Harvard CMT convention. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. Scale is the size for magnitude = 5 (that is $M_0 = 4.0E23$ dynes-cm) in inch (unless **c**, **i**, **m**, or **p** is appended). Use the **-T** option to render the beach ball transparent by drawing only the nodal planes and the circumference. The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Append **u** to have the text appear below the beach ball (default is above). Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order) **3**: depth of event in kilometers
4,5,6: strike, dip, and rake of plane 1 **7,8,9**: strike, dip, and rake of plane 2 **10,11**: mantissa and exponent of moment in dyne-cm **12,13**: longitude, latitude at which to place beach ball. Entries in these columns are necessary with the **-C** option. Using (0,0) in columns 12 and 13 will plot the beach ball at the longitude, latitude given in columns 1 and 2. The -: option will interchange the order of columns (1,2) and (12,13). **14**: Text string to appear above or below the beach ball (optional).

-Smldlzscale[/fontsize[/offset[**u**]]]

Seismic moment tensor (Harvard CMT, with zero trace). *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. Scale is the size for magnitude = 5 (that is scalar seismic moment = $4.0E23$ dynes-cm) in inch (unless **c**, **i**, **m**, or **p** is appended). (**-T0** option overlays best double couple transparently.) Use **-Sm** to plot the Harvard CMT seismic moment tensor with zero trace. Use **-Sd** to plot only the double couple part of moment tensor. Use **-Sz** to plot the anisotropic part of moment tensor (zero trace). The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Append **u** to have the text appear below the beach ball (default is above). Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (-: option interchanges order) **3**: depth of event in kilometers
4,5,6,7,8,9: mrr, mtt, mff, mrt, mrf, mtf in 10^* exponent dynes-cm **10**: exponent **11,12**: longitude, latitude at which to place beach ball. Entries in these columns are necessary with

the **-C** option. Using (0,0) in columns 11 and 12 will plot the beach ball at the longitude, latitude given in columns 1 and 2. The **-:** option will interchange the order of columns (1,2) and (11,12). **13**: Text string to appear above or below the beach ball (optional).

-Spyscale[/fontsize[/offset[**u**]]]

Focal mechanisms given with partial data on both planes. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. Scale is the size for magnitude = 5 in inch (unless **c**, **i**, **m**, or **p** is appended). The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Append **u** to have the text appear below the beach ball (default is above). Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (**-:** option interchanges order) **3**: depth of event in kilometers
4,5: strike, dip of plane 1 **6**: strike of plane 2 **7**: must be -1/+1 for a normal/inverse fault **8**:
magnitude **9,10**: longitude, latitude at which to place beach ball. Entries in these columns
are necessary with the **-C** option. Using (0,0) in columns 9 and 10 will plot the beach ball at
the longitude, latitude given in columns 1 and 2. The **-:** option will interchange the order of
columns (1,2) and (9,10). **11**: Text string to appear above or below the beach ball (optional).

-Sxlyltscale[/fontsize[/offset[**u**]]]

Principal axis. *scale* adjusts the scaling of the radius of the “beach ball”, which will be proportional to the magnitude. Scale is the size for magnitude = 5 (that is seismic scalar moment = 4×10^{23} dynes-cm) in inch (unless **c**, **i**, **m**, or **p** is appended). (**-T0** option overlays best double couple transparently.) Use **-Sx** to plot standard Harvard CMT. Use **-Sy** to plot only the double couple part of moment tensor. Use **-St** to plot zero trace moment tensor. The color or shade of the compressive quadrants can be specified with the **-G** option. The color or shade of the extensive quadrants can be specified with the **-E** option. Append **u** to have the text appear below the beach ball (default is above). Parameters are expected to be in the following columns:

1,2: longitude, latitude of event (**-:** option interchanges order) **3**: depth of event in kilometers
4,5,6,7,8,9,10,11,12: value (in 10^{exponent} dynes-cm), azimuth, plunge of T, N, P axis. **13**:
exponent **14,15**: longitude, latitude at which to place beach ball. Entries in these columns
are necessary with the **-C** option. Using (0,0) in columns 14 and 15 will plot the beach
ball at the longitude, latitude given in columns 1 and 2. The **-:** option will interchange the
order of columns (1,2) and (14,15). **16**: Text string to appear above or below the beach ball
(optional).

2.4.4 Optional Arguments

-B[pls]parameters (more ...) Set map boundary intervals.

-C[pen][Ppointsize] Offsets focal mechanisms to the longitude, latitude specified in the last two columns of the input file before the (optional) text string. A small circle is plotted at the initial location and a line connects the beachball to the circle. Specify *pen* and/or *pointsize* to change the line style and/or size of the circle. [Defaults: *pen* as given by **-W**; *pointsize* 0].

-Ddepmin/depmax Plots events between *depmin* and *depmax*.

-Efill Selects filling of extensive quadrants. Usually white. Set the color [Default is white].

-Fmode[args] Sets one or more attributes; repeatable. The various combinations are

-Fa[size][/P_axis_symbol[T_axis_symbol]] Computes and plots P and T axes with symbols. Optionally specify *size* and (separate) P and T axis symbols from the following: (**c**) circle, (**d**) diamond, (**h**)

- hexagon, (i) inverse triangle, (p) point, (s) square, (t) triangle, (x) cross. [Default: 6p/cc]
- Fefill** Sets the color or fill pattern for the T axis symbol. [Default as set by -E]
 - Fgfill** Sets the color or fill pattern for the P axis symbol. [Default as set by -G]
 - Fo** Use the **psvelomeca** input format without depth in the third column.
 - Fp[pen]** Draws the P axis outline using default pen (see -W), or sets pen attributes.
 - Fr[fill]** Draw a box behind the label (if any). [Default fill is white]
 - Ft[pen]** Draws the T axis outline using default pen (see -W), or sets pen attributes.
 - Fz[pen]** Overlay zero trace moment tensor using default pen (see -W), or sets pen attributes.
 - Gfill** Selects filling of focal mechanisms. By convention, the compressional quadrants of the focal mechanism beach balls are shaded. Set the color [Default is black].
 - K (more ...)** Do not finalize the PostScript plot.
 - Lpen** Draws the “beach ball” outline with *pen* attributes instead of with the default pen set by -W.
 - M** Use the same size for any magnitude. Size is given with -S.
 - N** Does **not** skip symbols that fall outside frame boundary specified by -R [Default plots symbols inside frame only].
 - O (more ...)** Append to existing PostScript plot.
 - P (more ...)** Select “Portrait” plot orientation.
 - T[num_of_planes][/pen]** Plots the nodal planes and outlines the bubble which is transparent. If *num_of_planes* is
 - 0: both nodal planes are plotted;
 - 1: only the first nodal plane is plotted;
 - 2: only the second nodal plane is plotted.
 Append */pen* to set the pen attributes for this feature. Default pen is as set by -W.
 - U[just/dx/dy][clabel] (more ...)** Draw GMT time stamp logo on plot.
 - V[level] (more ...)** Select verbosity level [c].
 - Wpen** Set pen attributes for all lines and the outline of symbols [Defaults: width = default, color = black, style = solid]. This setting applies to -C, -L, -T, -p, -t, and -z, unless overruled by options to those arguments.
 - X[alclfr][x-shift[u]]**
 - Y[alclfr][y-shift[u]] (more ...)** Shift plot origin.
 - Zcptfile** Give a color palette file and let compressive part color be determined by the z-value in the third column.
 - ccopies (more ...)** Specify number of plot copies [Default is 1].
 - h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
 - icols[l][sscale][ooffset][,...] (more ...)** Select input columns (0 is first column).
 - :[ilo] (more ...) Swap 1st and 2nd column on input and/or output.

- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.4.5 Examples

The following file should give a normal-faulting CMT mechanism:

```
gmt psmeqa -R239/240/34/35.2 -Jm4c -Sc0.4 -h1 << END > test.ps
lon lat depth str dip slip st dip slip mant exp plon plat
239.384 34.556 12. 180 18 -88 0 72 -90 5.5 0 0 0
END
```

2.4.6 See Also

gmt, *psbasemap*, *psxy*

2.4.7 References

- Bomford, G., Geodesy, 4th ed., Oxford University Press, 1980.
- Aki, K. and P. Richards, Quantitative Seismology, Freeman, 1980.
- F. A. Dahlen and Jeroen Tromp, Theoretical Seismology, Princeton, 1998, p.167.
- Cliff Frohlich, Cliff's Nodes Concerning Plotting Nodal Lines for P, Sh and Sv
Seismological Research Letters, Volume 67, Number 1, January-February, 1996
- Thorne Lay, Terry C. Wallace, Modern Global Seismology, Academic Press, 1995, p.384.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University press (routine jacobi)

2.4.8 Authors

Genevieve Patau, *Laboratory of Seismogenesis* <<http://www.ipgp.fr/rech/sismogenese/>>, Institut de Physique du Globe de Paris, Departement de Sismologie, Paris, France

2.5 pspolar

pspolar - Plot polarities on the inferior focal half-sphere on maps

2.5.1 Synopsis

```
pspolar [ table ] -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] -Msize -S<symbol><size> [ -B[pls]parameters ] [ -Clon/lat[ldash_width/pointsize] ] [ -Fcolor ] [ -Gfill ] [ -K ] [ -L ] [ -N ] [ -O ] [ -Qmode[args] ] [ -Tangle/form/justify/fontsize ] [ -U[just/dx/dy]/[cllabel] ] [ -V[level] ] [ -Wpen ] [ -X[alcflr]/[x-shift][u] ] [ -Y[alcflr]/[y-shift][u] ] [ -ccopies ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.5.2 Description

pspolar reads data values from *files* [or standard input] and generates PostScript code that will plot stations on focal mechanisms on a map. The PostScript code is written to standard output.

Parameters are expected to be in the following columns:

1,2,3: station_code, azimuth, take-off angle

4: polarity:

- compression can be c,C,u,U,+
- rarefaction can be d,D,r,R,-
- not defined is anything else

2.5.3 Required Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Jparameters (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][r**]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-Dlongitude/latitude) Maps the bubble at given longitude and latitude point.

-Msize) Sets the size of the beach ball to plot polarities in. *Size* is in inch (unless **c**, **i**, **m**, or **p** is appended).

-S<symbol_type><size>) Selects *symbol_type* and *symbol size*. Size is in inch (unless **c**, **i**, **m**, or **p** is appended). Choose symbol type from *st*(a)r, (c)ircle, (d)iamond, (h)exagon, (i)nverted triangle, (p)oint, (s)quare, (t)riangle, (x)cross.

2.5.4 Optional Arguments

- B**[*pls*]*parameters* (*more ...*) Set map boundary intervals.
- C** Offsets focal mechanisms to the latitude and longitude specified in the last two columns of the input file.
- Efill**) Selects filling of symbols for stations in extensive quadrants. Set the color [Default is 250]. If **-Efill** is the same as **-Ffill**, use **-e** to outline.
- Ffill** Sets background color of the beach ball. Default is no fill.
- Gfill** Selects filling of symbols for stations in compressional quadrants. Set the color [Default is black].
- K** (*more ...*) Do not finalize the PostScript plot.
- N** Does **not** skip symbols that fall outside map border [Default plots points inside border only].
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- Qmode**[*args*] Sets one or more attributes; repeatable. The various combinations are
- Qe**[*pen*]) Outline symbols in extensive quadrants using *pen* or the default pen (see **-W**).
- Qf**[*pen*]) Outline the beach ball using *pen* or the default pen (see **-W**).
- Qg**[*pen*]) Outline symbols in compressional quadrants using *pen* or the default pen (see **-W**).
- Qh** Use special format derived from HYPO71 output
- Qshalf-size**/[**V**[*v_width/h_length/h_width/shape*]][**Gcolor**][**L**] Plots S polarity azimuth. S polarity is in last column. It may be a vector (**V** option) or a segment. Give half-size, *v_width*, *h_length*, *h_width* in inch (unless **c**, **i**, **m**, or **p** is appended). [**L**] option is for outline.
- Qtpen** Set pen color to write station code. Default uses the default pen (see **-W**).
- Tangle/form/justify/fontsize in points** To write station code. [Default is 0.0/0/5/12].
- U**[*just/dx/dy*]/[*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V**[*level*] (*more ...*) Select verbosity level [*c*].
- Wpen** Set current pen attributes [Defaults: width = default, color = black, style = solid].
- X**[*alcflr*][*x-shift*][*u*]
- Y**[*alcflr*][*y-shift*][*u*] (*more ...*) Shift plot origin.
- ccopies** (*more ...*) Specify number of plot copies [Default is 1].
- icols**[*l*][*sscale*][*ooffset*][*,...*] (*more ...*) Select input columns (0 is first column).
- :**[ilo]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- +** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.5.5 Examples

```
gmt psolar -R239/240/34/35.2 -JM8c -N -Sc0.4 -h1 -D39.5/34.5 -M5 << END > test.ps
#stat azim ih pol
0481 11 147 c
6185 247 120 d
0485 288 114 +
0490 223 112 -
0487 212 109 .
END
```

or

```
gmt psolar -R239/240/34/35.2 -JM8c -N -Sc0.4 -h1 -D239.5/34.5 -M5 <<END > test.ps
#Date Or. time stat azim ih
910223 1 22 0481 11 147 ipu0
910223 1 22 6185 247 120 ipd0
910223 1 22 0485 288 114 epu0
910223 1 22 0490 223 112 epd0
910223 1 22 0487 212 109 epu0
END
```

2.5.6 See Also

gmt, *psbasemap*, *psxy*

2.5.7 References

Bomford, G., *Geodesy*, 4th ed., Oxford University Press, 1980.

Aki, K. and P. Richards, *Quantitative Seismology*, Freeman, 1980.

2.5.8 Authors

Genevieve Patau, *Laboratory of Seismogenesis* <<http://www.ipgp.fr/rech/sismogenese/>>, Institut de Physique du Globe de Paris, Departement de Sismologie, Paris, France

2.6 psvelo

psvelo - Plot velocity vectors, crosses, and wedges on maps

2.6.1 Synopsis

```
psvelo [ table ] -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] [ -Aparameters ] [ -B[pls]parameters ] [ -Fcolor ] [ -Ecolor ] [ -Gfill ] [ -K ] [ -L ] [ -N ] [ -O ] [ -P ] [ -Ssymbol/scale/conf/font_size ] [ [ -U[just/dx/dy][clabel] ] [ -V[level] ] [ -Wpen ] [ -X[alclflr][x-shift[u]] ] [ -Y[alclflr][y-shift[u]] ] [ -ccopies ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.6.2 Description

psvelo reads data values from *files* [or standard input] and generates PostScript code that will plot velocity arrows on a map. Most options are the same as for *psxy*, except **-S**. The PostScript code is written to standard output. The previous version (**psvelomeca**) is now obsolete. It has been replaced by **psvelo** and *psmecca*.

2.6.3 Required Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Jparameters (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][*W|E|S|N*] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-S

Selects the meaning of the columns in the data file and the figure to be plotted.

-Sevelscale/confidence/fontsize.

Velocity ellipses in (N,E) convention. *Vscale* sets the scaling of the velocity arrows. This scaling gives inches (unless **c**, **i**, **m**, or **p** is appended). *Confidence* sets the 2-dimensional confidence limit for the ellipse, e.g., 0.95 for 95% confidence ellipse. *FontSize* sets the size of the text in points. The ellipse will be filled with the color or shade specified by the **-G** option [default transparent]. The arrow and the circumference of the ellipse will be drawn with the pen attributes specified by the **-W** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude of station (**-:** option interchanges order) **3,4**: eastward, northward velocity (**-:** option interchanges order) **5,6**: uncertainty of eastward, northward velocities (1-sigma) (**-:** option interchanges order) **7**: correlation between eastward and northward components **8**: name of station (optional).

-Snbarscale.

Anisotropy bars. *Barscale* sets the scaling of the bars This scaling gives inches (unless **c**, **i**, **m**, or **p** is appended). Parameters are expected to be in the following columns:

1,2: longitude, latitude of station (**-:** option interchanges order) **3,4**: eastward, northward components of anisotropy vector (**-:** option interchanges order)

-Srvscale/confidence/fontsize

Velocity ellipses in rotated convention. *Vscale* sets the scaling of the velocity arrows. This scaling gives inches (unless **c**, **i**, **m**, or **p** is appended). *Confidence* sets the 2-dimensional confidence limit for the ellipse, e.g., 0.95 for 95% confidence ellipse. *FontSize* sets the size of the text in points. The ellipse

will be filled with the color or shade specified by the **-G** option [default transparent]. The arrow and the circumference of the ellipse will be drawn with the pen attributes specified by the **-W** option. Parameters are expected to be in the following columns:

1,2: longitude, latitude, of station (**-:** option interchanges order) **3,4**: eastward, northward velocity (**-:** option interchanges order) **5,6**: semi-major, semi-minor axes **7**: counter-clockwise angle, in degrees, from horizontal axis to major axis of ellipse. **8**: name of station (optional)

-Sw*wedge_scale/wedge_mag*.

Rotational wedges. *Wedge_scale* sets the size of the wedges in inches (unless **c**, **i**, **m**, or **p** is appended). Values are multiplied by *Wedge_mag* before plotting. For example, setting *Wedge_mag* to 1.e7 works well for rotations of the order of 100 nanoradians/yr. Use **-G** to set the fill color or shade for the wedge, and **-E** to set the color or shade for the uncertainty. Parameters are expected to be in the following columns:

1,2: longitude, latitude, of station (**-:** option interchanges order) **3**: rotation in radians **4**: rotation uncertainty in radians

-Sx*cross_scale*

gives Strain crosses. *Cross_scale* sets the size of the cross in inches (unless **c**, **i**, **m**, or **p** is appended). Parameters are expected to be in the following columns:

1,2: longitude, latitude, of station (**-:** option interchanges order) **3**: eps1, the most extensional eigenvalue of strain tensor, with extension taken positive. **4**: eps2, the most compressional eigenvalue of strain tensor, with extension taken positive. **5**: azimuth of eps2 in degrees CW from North.

2.6.4 Optional Arguments

-A*parameters* Modify vector parameters. For vector heads, append vector head *size* [Default is 9p]. See VECTOR ATTRIBUTES for specifying additional attributes.

-B[*p*]*parameters (more ...)* Set map boundary intervals.

-D*Sigma_scale* can be used to rescale the uncertainties of velocities (**-Se** and **-Sr**) and rotations (**-Sw**). Can be combined with the *confidence* variable.

-F*fill* Sets the color or shade used for frame and annotation. [Default is black]

-E*fill* Sets the color or shade used for filling uncertainty wedges (**-Sw**) or velocity error ellipses (**-Se** or **-Sr**). [If **-E** is not specified, the uncertainty regions will be transparent.]

-G*fill* Specify color (for symbols/polygons) or pattern (for polygons) [Default is black]. Optionally, specify **-G***icon_size/pattern*, where *pattern* gives the number of the image pattern (1-90) OR the name of a icon-format file. *icon_size* sets the unit size in inches. To invert black and white pixels, use **-GP** instead of **-Gp**. See **pspatterns** for information on individual patterns.

-K (*more ...*) Do not finalize the PostScript plot.

-L Draw lines. Ellipses and fault planes will have their outlines drawn using current pen (see **-W**).

-N Do **NOT** skip symbols that fall outside the frame boundary specified by **-R**. [Default plots symbols inside frame only].

-O (*more ...*) Append to existing PostScript plot.

-P (*more ...*) Select "Portrait" plot orientation.

- U[*just/dx/dy/*][*clabel*] (*more ...*) Draw GMT time stamp logo on plot.
- V[*level*] (*more ...*) Select verbosity level [*c*].
- W Set pen attributes for velocity arrows, ellipse circumference and fault plane edges. [Defaults: width = default, color = black, style = solid].
- X[*alclfr*][*x-shift*][*u*]
- Y[*alclfr*][*y-shift*][*u*] (*more ...*) Shift plot origin.
- ccopies (*more ...*) Specify number of plot copies [Default is 1].
- h[*ilo*][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- icols[*l*][*sscale*][*offset*][*,...*] (*more ...*) Select input columns (0 is first column).
- :[*ilo*] (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

2.6.5 Vector Attributes

Several modifiers may be appended to the vector-producing options to specify the placement of vector heads, their shapes, and the justification of the vector. Below, left and right refers to the side of the vector line when viewed from the start point to the end point of the segment:

- +*angle* sets the angle of the vector head apex [30].
- +**b** places a vector head at the beginning of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +**e** places a vector head at the end of the vector path [none]. Append **llr** to only draw the left or right side of this head [both sides].
- +**g-*fill*** turns off vector head fill (if -) or sets the vector head fill [Default fill is used, which may be no fill].
- +**l** draws half-arrows, using only the left side of specified heads [both sides].
- +**norm** scales down vector attributes (pen thickness, head size) with decreasing length, where vectors shorter than *norm* will have their attributes scaled by length/*norm* [arrow attributes remains invariant to length].
- +*oplon/plat* specifies the oblique pole for the great or small circles. Only needed for great circles if +**q** is given.
- +**p**[-][*pen*] sets the vector pen attributes. If *pen* has a leading - then the head outline is not drawn. [Default pen is used, and head outline is drawn]

+q means the input *angle, length* data instead represent the *start* and *stop* opening angles of the arc segment relative to the given point.

+r draws half-arrows, using only the right side of specified heads [both sides].

In addition, all but circular vectors may take these modifiers:

+jjust determines how the input *x,y* point relates to the vector. Choose from **beginning** [default], **end**, or **center**.

+s means the input *angle, length* is instead the *x, y* coordinates of the vector end point.

2.6.6 Examples

The following should make big red arrows with green ellipses, outlined in red. Note that the 39% confidence scaling will give an ellipse which fits inside a rectangle of dimension *Esig* by *Nsig*.

```
gmt psvelo << END -h2 -R-10/10/-10/10 -W0.25p,red -Ggreen -L -Se0.2/0.39/18 \
  -Blg1 -Jx0.4/0.4 -A0.3p -P -V > test.ps
#Long. Lat. EVEL NVEL Esig Nsig CorEN SITE
#(deg) (deg) (mm/yr) (mm/yr)
0. -8. 0.0 0.0 4.0 6.0 0.500 4x6
-8. 5. 3.0 3.0 0.0 0.0 0.500 3x3
0. 0. 4.0 6.0 4.0 6.0 0.500
-5. -5. 6.0 4.0 6.0 4.0 0.500 6x4
5. 0. -6.0 4.0 6.0 4.0 -0.500 -6x4
0. -5. 6.0 -4.0 6.0 4.0 -0.500 6x-4
END
```

This example should plot some residual rates of rotation in the Western Transverse Ranges, California. The wedges will be dark gray, with light gray wedges to represent the 2-sigma uncertainties.

```
gmt psvelo << END -Sw0.4/1.e7 -W0.75p -Gdarkgray -Elightgray -h1 -D2 -Jm2.2 \
  -R240./243./32.5/34.75 -Bf10ma60m/WeSn -P > test.ps
#lon lat spin(rad/yr) spin_sigma (rad/yr)
241.4806 34.2073 5.65E-08 1.17E-08
241.6024 34.4468 -4.85E-08 1.85E-08
241.0952 34.4079 4.46E-09 3.07E-08
241.2542 34.2581 1.28E-07 1.59E-08
242.0593 34.0773 -6.62E-08 1.74E-08
241.0553 34.5369 -2.38E-07 4.27E-08
241.1993 33.1894 -2.99E-10 7.64E-09
241.1084 34.2565 2.17E-08 3.53E-08
END
```

2.6.7 See Also

gmt, psbasemap, psxy

2.6.8 References

Bomford, G., Geodesy, 4th ed., Oxford University Press, 1980.

2.6.9 Authors

Kurt L. Feigl, Department of Geology and Geophysics at University of Wisconsin-Madison, Madison, Wisconsin, USA

Genevieve Patau, *Laboratory of Seismogenesis*, Institut de Physique du Globe de Paris, Departement de Sismologie, Paris, France

2.7 mgd77convert

mgd77convert - Convert MGD77 data to other file formats

2.7.1 Synopsis

```
mgd77convert NGDC-ids -Falclm lt -T[+]alclm lt [ -C ] [ -D ] [ -L[w][e][+ ] ] [ -V[level] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.7.2 Description

mgd77convert reads versions of MGD77 files and writes the same data in (probably) another format to a new file in the current directory. Both pre- and post-Y2K MGD77 formats can be processed.

2.7.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See mgd77info **-L** for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use **-I** to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in **\$MGD77_HOME/mgd77_paths.txt** [If **\$MGD77_HOME** is not set it will default to **\$GMT_SHAREDIR/mgd77**].

-F*alclm lt* Specifies the format of the input (From) files. Choose from **a** for standard MGD77 ASCII table (with extension .mgd77), **c** for the new MGD77+ netCDF format (with extension .nc), **m** for the new MGD77 t format (extension .m77t) and **t** for a plain ASCII tab-separated table dump (with extension .dat). Use **-FC** to recover the original MGD77 setting from the MGD77+ file [Default will apply any E77 corrections encoded in the file].

-T[+]*alclm lt* Specifies the format of the output (To) files. Choose from **a** for standard MGD77 ASCII table (with extension .mgd77), **c** for the new MGD77+ netCDF format (with extension .nc), **m** for the new MGD77 t format (extension .m77t) and **t** for a plain ASCII tab-separated table dump (with extension .dat). We will refuse to create the file(s) if they already exist in the current directory. Prepend **+** to override this policy.

2.7.4 Optional Arguments

- C Convert from NGDC two-file data sets *.h77, *.a77 to single file *.mgd77. No other options (except -V) are allowed. Give one or more names of *.h77 files, *.a77 files, or just the file prefixes.
- D By default, the storage types used in a MGD77+ netCDF file greatly exceed the precision imposed by the ASCII MGD77 format. However, for the five items **faa**, **eot**, **mag**, **diur** and **msd** we use 2-byte integers with implied precisions of 0.1 mGal, 0.1 nTesla, and 1 m as in the MGD77 format. It is possible that at some point these items will need to be stored as 4-byte ints which would allow precisions of 10 fTesla, 1 nGal, and 0.01 mm, respectively. This option activates such storage [Default uses 2-byte integers].
- L[w][e][+] Set the level of verification reporting [none] and where to send such reports [stderr]. Append a combination of **w** for warnings, **e** for errors, and **+** to send such log information to stdout.
- V[level] (*more ...*) Select verbosity level [c].
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

2.7.5 Examples

To convert a large set of a77,h77 pairs to proper mgd77 files, try

```
gmt mgd77convert -C *.h77
```

To convert 01010047.mgd77 and 01010008.mgd77 to new netCDF .nc files, and capture all verification messages, try

```
gmt mgd77convert 01010047 01010008 -Fa -Tc -V -Lew+ > log.lis
```

To convert 01010047.nc back to MGD77 ASCII and make sure it is identical to the original file, try (Bourne shell syntax)

```
orig=`gmt mgd77path 01010047 -Ic`
gmt mgd77convert 01010047 -Fc -Ta -V
diff $orig 01010047.mgd77
```

To convert 01010047.nc to a plain ASCII table for manual editing, overwriting any existing table, try

```
gmt mgd77convert 01010047 -Fc -T+t -V
```

To recover the original NGDC MGD77 version of 01020051.nc and ignore any E77 corrections, use

```
gmt mgd77convert 01020051 -FC -Ta -V
```

2.7.6 File Formats

mgd77convert handles three different formats. (1) The MGD77 ASCII tables are the established standard for distribution of underway geophysical data to and from the NGDC data center. Normally, only the ship-operations people and the cruise PI might be involved in *making* an MGD77 ASCII file for transmission to NGDC; users are more interested in *reading* such files. (2) The MGD77+ netCDF format was developed to facilitate the use of MGD77 data by scientists. It contains all the information of the original MGD77 file and if you convert back and forth you end up with the original. However, file sizes are typically ~30% of the original ASCII format and is much faster to operate on. (3) The plain ASCII tab-separated dump is available for users who need to manually edit the content of a MGD77 file. This is usually easier to do when the columns are tab-separated than when they are all crunched together in the MGD77 punch-card format.

2.7.7 Other Tools

The MGD77+ netCDF files are CF-1.0 and COARDS compliant and can be examined with general-purpose tools such as ncBrowse and ncView.

2.7.8 See Also

mgd77manage, *mgd77list*, *mgd77info*, *mgd77track* *x2sys_init*

2.7.9 References

ncBrowse, see <http://www.epic.noaa.gov/java/ncBrowse/> ncView, see http://meteora.ucsd.edu/~pierce/ncview_home_page/ The Marine Geophysical Data Exchange Format - "MGD77", see <http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt>

2.8 mgd77info

mgd77info - Extract information about MGD77 files

2.8.1 Synopsis

mgd77info *NGDC-ids* [**-C**[*mle*]] [**-E**[*mle*]] [**-I***gnore*] [**-Mf**[*item*]**|r|e|l|h**] [**-L**[*v*]] [**-V**[*level*]]

Note: No space is allowed between the option flag and the associated arguments.

2.8.2 Description

mgd77info reads <legid>.[mgd77]nc files and produces a single record of information about each cruise specified. The information includes beginning and end times, total track distances in km, longitude and latitude range, and the total number of geophysical observations. Optionally, choose instead to see the original MGD77 header meta-data section or its individual members.

If you need to know which tracks are crossing through a given region and what kinds of geophysical observations are available, consider using the *x2sys* tools to set up a tracks index data base (see **x2sys_init** for more information).

2.8.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See `mgd77info -L` for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use `-I` to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in `$MGD77_HOME/mgd77_paths.txt` [If `$MGD77_HOME` is not set it will default to `$GMT_SHAREDIR/mgd77`].

2.8.4 Optional Arguments

-C[*mle*] List abbreviations for all columns present in the MGD77[+] files. Append **m** or **e** to limit the display to the MGD77 standard or MGD77+ extended set only.

-E[*mle*] Give a one-line summary for each cruise listed.

-Mf[*item*]rlrlh List the meta-data (header) and (if present) the MGD77+ history for each cruise. Append **f** for a formatted display. This will list individual parameters and their values, one entry per output line, in a format that can be searched using standard UNIX text tools. Alternatively, append the name of a particular parameter (you only need to give enough characters - starting at the beginning - to uniquely identify the item). Give **-** to display the list of all parameter names. You may also specify the number of a parameter. For the raw, punchcard-formatted MGD77 original header block, append **r** instead. For the MGD77+ E77 status, append **e** instead. Finally, for the MGD77+ history, append **h** instead.

-ignore Ignore certain data file formats from consideration. Append **alclt** to ignore MGD77 ASCII, MGD77+ netCDF, or plain tab-separated ASCII table files, respectively. The option may be repeated to ignore more than one format. [Default ignores none].

-L[*v*] No cruise information is listed. Instead, we just display a list of the GEODAS institution 2-character codes and their names. Optionally, append **v** to also display the vessels and their 4-character codes for each institution. The following is the list of institutions: (01) LAMONT (LDEO), (02) WOODS HOLE O.I., (03) NOAA, (04) US ARMY, (05) NEW ZEALAND, (06) US GEOL. SURVEY, (07) OREGON ST. UNIV, (08) U.HAWAII SOEST, (09) US NAVY, (10) UNIV OF TEXAS, (11) RICE UNIV., (12) CANADA, (13) UNIV OF CONN., (14) U.MIAMI (RSMAS), (15) SCRIPPS INST.OC, (16) CHINA, (17) U RHODE ISLAND, (18) DUKE UNIVERSITY, (19) UNITED KINGDOM, (20) U.WASHINGTON, (22) WESTERN GEOPHY., (23) TEXAS A&M UNIV., (24) AUSTRALIA, (25) MONACO, (29) RUSSIA, (30) SPAIN, (35) NIMA, (58) NETHERLANDS, (60) MIN MGMT SVC, (63) ISRAEL, (67) FRANCE, (71) SOUTH AFRICA, (75) US COAST GUARD, (76) BRAZIL, (77) INT. GRAV. BUR, (83) GERMANY, (84) ORSTOM NEW CAL, (86) CUBA, (87) ARGENTINA, (88) US NSF, (89) INDIA, (90) PORTUGAL, (92) FINLAND, (93) CHILE, (J1) HYDR DEPT JAPAN, (J2) GEOL SRVY JAPAN, (J4) UNIV TOKYO, (J5) KOBE UNIV, (J7) UNIV OF RYUKYUS, (J8) J.O.D.C. JAPAN, (J9) CHIBA UNIV, (JA) INST.POLAR RES., (ZZ) INST NOT CODED.

-V[*level*] (*more ...*) Select verbosity level [*c*].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.8.5 Examples

To get one-line summary information about the cruises 01010047.mgd77 and 01010008.mgd77, try

```
gmt mgd77info 01010047 01010008 -E > listing.lis
```

To see the original raw MGD77 header meta-data for cruise 01010047.mgd77, run

```
gmt mgd77info 01010047 -Mr
```

To determine all the parameters related to Gravity during cruise 01010047.mgd77, run

```
gmt mgd77info 01010047 -Mf | grep Gravity
```

To determine the Magnetic sampling rate used during cruise 01010047.mgd77, run

```
gmt mgd77info 01010047 -MfMagnetics_Sampling_Rate
```

To see all the columns that the MGD77+ cruise 01010047.nc contains, run

```
gmt mgd77info 01010047 -C
```

To see the E77 status of all MGD77+ cruises collected by the University of Hawaii (institution 08), run

```
gmt mgd77info 08 -Ia -Me
```

2.8.6 See Also

mgd77list, *mgd77manage*, *mgd77path*, *mgd77track*, *x2sys_init*

2.8.7 References

The Marine Geophysical Data Exchange Format - MGD77, see <http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt>.

2.9 mgd77list

mgd77list - Extract data from MGD77 files

2.9.1 Synopsis

```
mgd77list NGDC-ids -Fcolumns[,logic][:bittests] [ -A[+]cldflmltcode ] [ -Cfigle ] [ -DAstartdate ] [ -DBlbstopdate ] [ -E ] [ -Gstartrec ] [ -Gstoprec ] [ -Ignore ] [ -L[corrtable] ] [ -Ndlsunit ] [ -Qalcvmin/max ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Sstartdist[unit] ] [ -Sstopdist[unit] ] [ -T[mle] ] [ -V[level] ] [ -Wweight ] [ -Z+|- ] [ -bo[ncols][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.9.2 Description

mgd77list reads <NGDC-id>.[mgd77]nc files and produces an ASCII [or binary] table. The <NGDC-id>.[mgd77]nc files contain track information such as leg-id, time and position, geophysical observables such as gravity, magnetics, and bathymetry, and control codes and corrections such as Eotvos and diurnal corrections. The MGD77+ extended netCDF files may also contain additional user columns (for a listing of available columns, use *mgd77info* **-C**, and to learn how to add your own custom columns, see *mgd77manage*). The user may extract any combination of these parameters, any of 8 computed quantities (distance, heading, course-change, velocity, Carter correction, Eotvos correction and gravity and magnetic global reference fields), calendar sub-units of time (year, month, day, hour, min, sec), the NGDC id, and finally a preset weight (see **-W**). A sub-section can be specified by passing time- or distance-intervals along track or by selecting a geographical region. Finally, each output record may be required to pass any number of logical tests involving data values or bit flags. If multiple cruises are requested then they are separated by segment headers.

2.9.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See *mgd77info* **-L** for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use **-I** to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in **\$MGD77_HOME/mgd77_paths.txt** [If **\$MGD77_HOME** is not set it will default to **\$GMT_SHAREDIR/mgd77**].

-Fcolumns[,*logic*][:*bittests*] The required *columns* string must be a comma-separated list of parameter abbreviations given in the desired output order. Any parameters given in UPPER case must not be NaN in a record for output to occur. Unless specified separately, the output format (if ASCII) is controlled by the GMT parameter *FORMAT_FLOAT_OUT*. The available column abbreviations for information stored in the files (some columns may be NaN) are:

recno The record number counter.

drt The digital record type, usually 3 or 5 (for Y2K-compliant cruises).

id The survey ID string (leg name).

- ngdcid** The 8-character NGDC cruise ID string (usually the file prefix).
- time** Choose between Absolute calendar time (**atime**, the default) in the format dictated by the GMT parameters *FORMAT_DATE_OUT* and *FORMAT_CLOCK_OUT*, Relative time (**rtime**) in the format dictated by the GMT parameters *FORMAT_FLOAT_OUT* and *TIME_SYSTEM* (or *TIME_EPOCH* and *TIME_UNIT*), or Fractional year (**ytime**) in the format dictated by *FORMAT_FLOAT_OUT*.
- lon** Longitude in the format dictated by the GMT parameter *FORMAT_GEO_OUT*.
- lat** Longitude in the format dictated by the GMT parameter *FORMAT_GEO_OUT*.
- twt** Two-Way Travel time (in s).
- depth** Corrected bathymetry (in m, positive below sealevel).
- mtf1** Magnetic Total Field intensity from sensor 1 (in nTesla).
- mtf2** Magnetic Total Field intensity from sensor 2 (in nTesla).
- mag** Residual magnetic anomaly (in nTesla).
- gobs** Observed gravity (in mGal).
- faa** Free-air gravity anomaly (in mGal).
- ptc** Position Type Code (1 = fix, 3 = interpolated, 9 = unspecified).
- bcc** Bathymetric Correction Code, indicating the procedure used to convert travel time to depth. (01-55 = Matthews' zone used to correct the depth, 59 = Matthews' corrections used but the zones is unspecified in the data record, 60 = S. Kuwahara formula for T-S, 61 = Wilson formula for T-S, 62 = Del Grosso formula for T-S, 63 = Carter's tables, 88 = Other, described in header sections, 99 = unspecified).
- btc** Bathymetric Type Code, indicating how the bathymetry value was obtained (1 = observed, 3 = interpolated, 9 = unspecified).
- msens** Magnetic sensor for used to evaluate the residual field (1 = 1st or leading sensor, 2 = 2nd or trailing sensor, 9 = unspecified).
- msd** Depth (or altitude) of the magnetic sensor (in m, positive below sealevel).
- diur** Magnetic diurnal correction (in nTesla).
- eot** Eotvos correction (in mGal).
- sln** Seismic Line Number string.
- sspn** Seismic Shot Point Number string.
- nqc** Navigation Quality Code (5 = suspected, by source institution, 6 = suspected, by NGDC, 9 = no problems identified).
- In addition, the following derived navigational quantities can be requested:
- year** The year of each record.
- month** The month of each record.
- day** The day of the month of each record.
- hour** The hour of each record.
- min** The minutes of each record.

sec The decimal seconds of each record.

dist Along-track distance from start of leg. For method of calculation, see **-C** [spherical great circle distances], and for distance units, see **-N** [km].

az Ship azimuth (heading) measured clockwise from north (in degrees).

cc Ship course change (change in heading) measured clockwise from north (in degrees).

vel Ship speed; see **-N** for units [m/s].

Finally, the following computed quantities can be requested:

weight Weight assigned to this data set (see **-W**).

carter Carter depth correction, if **tw**t is present in file (in m). Sign: Correction is to be added to uncorrected depths to yield a corrected depth.

igrf International geomagnetic reference field (total field) (in nTesla).

ngrav International Gravity reference Field (“normal gravity”) (in mGal). Field is selected based on the parameter Gravity Theoretical Formula Code in the cruise’s MGD77 header. If this is not set or is invalid we default to the IGF 1980. Alternatively, specify the field directly using **-Af** (see that option for more details).

ceot Calculated Eotvos correction from navigation, using $E = 7.5038 * V * \cos(\text{lat}) * \sin(\text{az}) + 0.004154 * V^2$ (in mGal). Sign: Correction is to be added to uncorrected faa to yield a corrected faa.

The following short-hand flags are also recognized:

mgd77 This results in all 27 MGD77 fields being written out in the official MGD77 order.

mgd77t The full set of all 26 columns in the MGD77T specification.

geo This limits the output to 10 fields (**time**, **lon**, **lat** plus the seven geophysical observations **tw**t, **depth**, **mtf1**, **mtf2**, **mag**, **gobs**, and **faa**). By appending + to either of these set we will also append **dist**, **azim**, **cc**, **vel**, and **weight** as listed above.

all This returns all data columns in the file.

allt As **mgd77t** but with time items written as a date-time string.

As an option, logical tests may be added for any of the observations by appending *logic*, which is itself composed of one or more comma-separated instructions of the form *parOPvalue*, where *par* is one of the parameters listed above, **OP** is a logical operator (<, <=, =, !=, >=, >, !), and *value* is a constant used in the comparison. Floating point parameters are compared numerically; character parameters are compared lexically (after leading and trailing blanks have been removed). The bit comparison (!) means that at least one of the bits in *value* must be turned on in *par*. At least one of the tests must be true for the record to be output, except for tests using UPPER case parameters which all must be true for output to occur. Note that specifying a test does not imply that the corresponding column will be included in the output stream; it must be present in *columns* for that to occur. Note: some of the operators are special UNIX characters and you are advised to place quotes around the entire argument to **-F**.

Finally, for MGD77+ files you may optionally append *:bittests* which is : (a colon) followed by one or more comma-separated *+col* terms. This compares specific bitflags only for each listed column. Here, + means the chosen bit must be 1 (ON) whereas - means it must be 0 (OFF). All bit tests given must be passed. By default, MGD77+ files that have the special **MGD77_flags** column present will use those flags, and observations associated with ON-bits (meaning they are flagged

as bad) will be set to NaN; append : with no trailing information to turn this behavior off (i.e., no bit flags will be consulted).

2.9.4 Optional Arguments

- A[+]cldflmltcode** By default, corrected depth (**depth**), magnetic residual anomaly (**mag**), free-air gravity anomaly (**faa**), and the derived quantity Carter depth correction (**carter**) are all output as is (if selected in **-F**); this option adjusts that behavior. For each of these columns there are 2-4 ways to adjust the data. Append **c**(arter), **d**(ePTH), **f**(aa), or **m**(ag) and select the *code* for the procedure you want applied. You may select more than one procedure for a data column by summing their numerical *codes* (1, 2, 4, and 8). E.g., **-Ac3** will first try method **-Ac1** to estimate a Carter correction but if **depth** is NaN we will next try **-Ac2** which only uses **twT**. In all cases, if any of the values required by an adjustment procedure is NaN then the result will be NaN. This is also true if the original anomaly is NaN. Specify **-A+** to recalculate anomalies even if the anomaly in the file is NaN. Additionally, you can use **-At** to create fake times for cruises that has no time; these are based on distances and cruise duration.
- Ac** Determines how the **carter** correction term is calculated. Below, **C(twT)** stands for the Carter-corrected depth (it also depends on **lon**, **lat**), **U(twT, v)** is the uncorrected depth ($= \text{twT} * v / 2$) using as *v* the “Assumed Sound Velocity” parameter in the MGD77 header (if it is a valid velocity, otherwise we default to 1500 m/s); alternatively, append your preferred velocity *v* in m/s, **TU(depth, v)** is the 2-way travel time estimated from the (presumably) uncorrected **depth**, and **TC(depth)** is the 2-way travel time obtained by inverting the (presumably) corrected **depth** using the Carter correction formula. Select from
- Ac1[,v]** returns difference between **U(twT, v)** and **depth** [Default].
 - Ac2[,v]** returns difference between **U(twT, v)** and Carter (**twT**).
 - Ac4[,v]** returns difference between (assumed uncorrected) **depth** and Carter (**TU(depth)**).
 - Ac8[,v]** returns difference between **U(TC(depth), v)** and **depth**.
- Ad** Determines how the **depth** column output is obtained:
- Ad1** returns **depth** as stored in the data set [Default].
 - Ad2[,v]** returns calculated uncorrected depth **U(twT, v)**.
 - Ad4** returns calculated corrected depth **C(twT)**.
- Af** Determines how the **faa** column output is obtained. If **ngrav** (i.e., the International Gravity reference Field (IGF), or “normal gravity”) is required it is selected based on the MGD77 header parameter “Theoretical Gravity Formula Code”; if this code is not present or is invalid we default to 4. Alternatively, append the preferred *field* (1-4) to select 1 (Heiskanen 1924), 2 (IGF 1930), 3 (IGF 1967) or 4 (IGF 1980). Select from
- Af1[,field]** returns **faa** as stored in the data set [Default]. Optionally, sets the IGF *field* to use if you also have requested **ngrav** as an output column in **-F**.
 - Af2[,field]** returns the difference between **gobs** and **ngrav** (with optional *field* directive).
 - Af4[,field]** returns the combination of **gobs + eot - ngrav** (with optional *field* directive).
 - Af8[,field]** returns the combination of **gobs + pred_eot - ngrav** (with optional *field* directive).

- Am** Determines how the **mag** column output is obtained. There may be one or two total field measurements in the file (**mtf1** and **mtf2**), and the column **msens** may state which one is the leading sensor (1 or 2; it may also be undefined). Select from
- Am1** returns **mag** as stored in the data set [Default].
 - Am2** returns the difference between **mgfx** and **igrf**, where **x** is the leading sensor (1 or 2) indicated by the **msens** data field (defaults to 1 if unspecified).
 - Am4** returns the difference between **mgfx** and **igrf**, where **x** is the sensor (2 or 1) *not* indicated by the **msens** data field (defaults to 2 if unspecified).
 - Amc<offset>[unit]** Apply a correction that tries to compensate the fact that the magnetic field was not acquired at the same position as the navigation refer (ship's position). This is accomplished by reinterpolating the total magnetic field to what it would have had if it was measured at the ship's position (remember, it probably was measured <offset> meters behind). Due to this interpolation step, bad navigation, namely too many repeated points, may cause troubles. Measures are taken to minimize this effect but they aren't 100% full prof. The interpolation method is controlled by the gmtdefaults GMT_INTERPOLANT. Append **e** for meter, **f** for feet, **k** for km, **m** for miles, **n** for nautical miles, or **u** for survey feet [Default is e (meters)].
- Cfgle** Append a one-letter code to select the procedure for along-track distance calculation (see **-N** for selecting units):
- f** Flat Earth distances.
 - g** Great circle distances [Default].
 - e** Geodesic distances on current GMT ellipsoid.
- Dstartdate** Do not list data collected before *startdate* (yyyy-mm-ddBD(T)[hh:mm:ss]) [Default is start of cruise]. Use **-DA** to exclude records whose time is undefined (i.e., NaN). [Default reports those records].
- Dbstopdate** Do not list data collected on or after *stopdate* (yyyy-mm-ddBD(T)[hh:mm:ss]). [Default is end of cruise]. Use **-DB** to exclude records whose time is undefined (i.e., NaN). [Default reports those records].
- E** Exact match: Only output records that match all the requested geophysical columns [Default outputs records that matches at least one of the observed columns].
- Gastartrec** Do not list records before *startrec* [Default is 0, the first record].
- Gbstoprec** Do not list data after *stoprec*. [Default is the last record].
- Iignore** Ignore certain data file formats from consideration. Append **alclt** to ignore MGD77 ASCII, MGD77+ netCDF, or plain tab-separated ASCII table files, respectively. The option may be repeated to ignore more than one format. [Default ignores none].
- L[corrtable]** Apply optimal corrections to columns where such corrections are available. Append the correction table to use [Default uses the correction table *mgd77_corrections.txt* in the **\$MGD77_HOME** directory]. For the format of this file, see CORRECTIONS below.
- n** Issue a segment header record with cruise ID for each cruise.
- Ndlsunit** Append **d** for distance or **s** for speed, then give the desired *unit* as **e** (meter or m/s), **f** (feet or feet/s), **k** (km or km/hr), **m** (miles or miles/hr), **n** (nautical miles or knots), or **u** (survey feet or sfeet/s). [Default is **-Ndk -Nse** (km and m/s)].

- Qamin/max** Specify an accepted range (*min/max*) of azimuths. Records whose track azimuth falls outside this range are ignored [0/360].
- Qcmin/max** Specify an accepted range (*min/max*) of course changes. Records whose track course change falls outside this range are ignored [-360/+360]. Use **-QC** to take the absolute value of the course change before the test [Default uses signed course changes].
- Qvmin/max** Specify an accepted range (*min/max*; or just *min* if there is no upper limit) of velocities. Records whose track speed falls outside this range are ignored [0/infinity].
- R[*unit*]*west/east/south/north*[*zmin/zmax*][*r*]** *west, east, south, and north* specify the region of interest, and you may specify them in decimal degrees or in [+*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin, zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.
- Sastardist[*unit*]** Do not list data that are less than *startdist* meter along track from port of departure. Append **e** for meter, **f** for feet, **k** for km, **m** for miles, **n** for nautical miles, or **u** for survey feet [Default is 0e (meters)].
- Sbstopdist[*unit*]** Do not list data that are *stopdist* or more meters along track from port of departure. Append **e** for meter, **f** for feet, **k** for km, **m** for miles, **n** for nautical miles, or **u** for survey feet [Default is end of track].
- T[*mle*]** Turns OFF the otherwise automatic adjustment of values based on correction terms that are stored in the MGD77+ file and used to counteract such things as wrong units used by the source institution when creating the original MGD77 file from which the MGD77+ file derives (the option has no effect on plain MGD77 ASCII files). Append **m** or **e** to limit the option to the MGD77 or extended columns set only [Default applies to both].
- V[*level*] (*more ...*)** Select verbosity level [*c*].
- Wweight** Set the weight for these data. Weight output option must be set in **-F**. This is useful if the data are to be processed with the weighted averaging techniques offered by **blockmean**, **blockmedian**, and **blockmode** [1].
- Z+|-** Append the sign you want for **depth**, **carter**, and **msd** values below sea level (**-Z-** gives negative bathymetry) [Default is positive down].
- bo[*ncols*][*type*] (*more ...*)** Select binary output. ignored if **-bo** is selected. Likewise, string-fields cannot be selected. Note that if time is one of the binary output columns it will be stored as Unix-time (seconds since 1970). To read this information in GMT to obtain absolute calendar time will require you to use **--TIME_SYSTEM=1**.
- h[*ilo*][*n*][*+c*][*+d*][*+rremark*][*+rtitle*] (*more ...*)** Skip or produce header record(s).
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.9.5 Examples

To get a (distance, heading, gravity, bathymetry) listing from 01010047.mgd77, starting at June 3 1971 20:45 and ending at distance = 5000 km, use the following command:

```
gmt mgd77list 01010047 -Da1971-06-03T20:45 -Sb5000 -Fdist,azim,faa,depth > myfile.d
```

To make input for **blockmean** and **surface** using free-air anomalies from all the cruises listed in the file cruises.lis, but only the data that are inside the specified area, and make the output binary:

```
gmt mgd77list `cat cruises.lis` -Flon,lat,faa -R-40/-30/25/35 -bo > allgrav.b
```

To extract the locations of depths exceeding 9000 meter that were not interpolated (**btc != 1**) from all the cruises listed in the file cruises.lis:

```
gmt mgd77list `cat cruises.lis` -F"depth,DEPTH>9000,BTC!=1" > really_deep.d
```

To extract dist, faa, and grav12_2 from records whose depths are shallower than 3 km and where none of the requested fields are NaN, from all the MGD77+ netCDF files whose cruise ids are listed in the file cruises.lis, we try

```
gmt mgd77list `cat cruises.lis` -E -Ia -F"dist,faa,grav12_2,depth<3000" > \
shallow_grav.d
```

To extract dist, faa, and grav12_2 from all the MGD77+ netCDF files whose cruise ids are listed in the file cruises.lis, but only retrieve records whose bitflag for faa indicates BAD values, we try

```
gmt mgd77list `cat cruises.lis` -E -Ia -F"dist,faa,grav12_2:+faa" > bad_grav.d
```

To output lon, lat, mag, and faa from all the cruises listed in the file cruises.lis, but recalculate the two residuals based on the latest reference fields, try:

```
gmt mgd77list `cat cruises.lis` -Flon,lat,mag,faa -Af2,4 -Am2 > data.d
```

2.9.6 Recalculated Anomalies

When recalculated anomalies are requested (either explicitly via the **-A** option or implicitly via E77 metadata in the MGD77+ file) we only do so for the records whose original anomaly was not a NaN. This restriction is implemented since many anomaly columns contains corrections, usually in the form of hand-edited changes, that cannot be duplicated from the corresponding observation.

2.9.7 Igrf

The IGRF calculations are based on a Fortran program written by Susan Macmillan, British Geological Survey, translated to C via f2c by Joaquim Luis, U Algarve, and adapted to GMT-style by Paul Wessel.

2.9.8 I_{gf}

The equations used are reproduced here using coefficients extracted directly from the source code (let us know if you find errors):

$$(1) g = 978052.0 * [1 + 0.005285 * \sin^2(\text{lat}) - 7e-6 * \sin^2(2*\text{lat}) + 27e-6 * \cos^2(\text{lat}) * \cos^2(\text{lon}-18)]$$

$$(2) g = 978049.0 * [1 + 0.0052884 * \sin^2(\text{lat}) - 0.0000059 * \sin^2(2*\text{lat})]$$

$$(3) g = 978031.846 * [1 + 0.0053024 * \sin^2(\text{lat}) - 0.0000058 * \sin^2(2*\text{lat})]$$

$$(4) g = 978032.67714 * [(1 + 0.00193185138639 * \sin^2(\text{lat})) / \sqrt{1 - 0.00669437999013 * \sin^2(\text{lat})}]$$

2.9.9 Corrections

The correction table is an ASCII file with coefficients and parameters needed to carry out corrections. Comment records beginning with # are allowed. All correction records are of the form

cruiseID observation correction

where *cruiseID* is a NGDC prefix, *observation* is one of the abbreviations for geophysical observations listed under **-F** above, and *correction* consists of one or more *terms* that will be summed up and then **subtracted** from the observation before output. Each *term* must have this exact syntax:

factor[**function*](*scale*)(*abbrev*[*-origin*]))[^*power*]

where terms in brackets are optional (the brackets themselves are not used but regular parentheses must be used as indicated). No spaces are allowed except between *terms*. The *factor* is the amplitude of the basis function, while the optional *function* can be one of sin, cos, or exp. The optional *scale* and *origin* can be used to translate the argument (before giving it to the optional function). The argument *abbrev* is one of the abbreviations for observations listed above. If *origin* is given as **T** it means that we should replace it with the value of *abbrev* for the very first record in the file (this is usually only done for *time*). If the first record entry is NaN we revert *origin* to zero. Optionally, raise the entire expression to the given *power*, before multiplying by the amplitude. The following is an example of fictitious corrections to the cruise 99999999, implying the **depth** should have the Carter correction removed, **faa** should have a linear trend removed, the magnetic anomaly (**mag**) should be corrected by a strange dependency on ship heading and latitude, and **gobs** needs to have 10 mGal added (hence given as -10):

```
99999999 depth 1.0*((carter))
```

```
99999999 faa 14.1 1e-5*((time-T))
```

```
99999999 mag 0.5*cos(0.5*(azim-19))^2 1.0*exp(-1e-3(lat))^1.5
```

```
99999999 gobs -10
```

2.9.10 See Also

mgd77convert, *mgd77info*, *mgd77manage*, *mgd77track*

2.9.11 References

The Marine Geophysical Data Exchange Format - MGD77, see <http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt>

IGRF, see <http://www.ngdc.noaa.gov/IAGA/vmod/igrf/>

2.10 mgd77magref

mgd77magref - Evaluate the IGRF or CM4 magnetic field models

2.10.1 Synopsis

```
mgd77magref [ inputfile ] [ -A[+aalt+tdate+y] ] [ -Ccm4file ] [ -Ddstfile ] [ -Ef107file ] [ -Fflags ] [ -G ] [ -Scllow/high ] [ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -f[ilo]colinfo ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.10.2 Description

mgd77magref will evaluate the IGRF or the CM4 geomagnetic models at the specified locations and times.

2.10.3 Required Arguments

None.

2.10.4 Optional Arguments

inputfile Contains the moments in space-time where we want to evaluate the magnetic reference field. The first two columns must contain longitude and latitude (however, see **-:** for latitude and longitude instead). Normally, the third and fourth columns must contain altitude (in km) and time, respectively, but if one or both of these are constant for all records they can be supplied via the **-A** option instead and are thus not expected in the input file. If no input file is given we read *stdin*. A note about the CM4 validity domain. The core field of CM4 is valid from 1960-2002.5 but the ionospheric and magnetospheric fields are computed after the *Dst* and *F10.7* coefficient files. We extended here those coefficient files up to 2006, which means that one can compute external contributions up until 2006 but the Secular Variation will be biased (non reliable). New indices files may be retrieved from ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/DST/ (the *Dst* coefficients) and http://umbra.nascom.nasa.gov/sdb/yohkoh/ys_dbase/indices_flux_raw/Penticton_Absolute/monthly/M (The *F10.7* index file is a MONTHPLT.ABS). NOTE: since the *Dst* files in the .../DST/ directory are still only up to 2006, for GMT4.5.3 and after we extended the *Dst* until August 2009 by reformatting the data in the preliminary file *Est_Ist_index_0_mean.pli*, which is at ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/EST_IST/. But since this site is now also outdated, we now get the DST indices from <http://wdc.kugi.kyoto-u.ac.jp/dstae/index.html> However, for the most recent dates those indices are “Quick Look” (the best are the “Definitive” type). Because the *F10.7* from the MONTHPLT.ABS file mentioned above are apparently no being updated, we found another place where they are, which is: ftp://ftp.ngdc.noaa.gov/STP/space-weather/solar-data/solar-features/solar-radio/noontime-flux/penticton/penticton_absolute/listings/listing_drao_noontime-flux-absolute_monthly.txt

-A[**+aalt**+**tdate**+**y**] Adjusts how the input record is interpreted. Append **+a** to set a fixed *altitude* (in km) that should apply to all data records [Default expects *altitude* to be in the 3rd column of all records]. Append **+t** to set a fixed *time* that should apply to all data records [Default expects *time*

to be in the 4th column of all records]. Finally, append **+y** to indicate that all times are specified as decimal years [Default is ISO *dateTclock* format, see *TIME_EPOCH*].

- Ccm4file** Specify an alternate CM4 coefficient file [umdl.CM4].
- DDstfile** Specify an alternate file with hourly means of the Dst index for CM4 [Dst_all.wdc]. Alternatively, simply specify a single index to apply for all records.
- Ef107file** Specify an alternate file with monthly means of absolute F10.7 solar radio flux for CM4 [F107_mon.plt]. Alternatively, simply specify a single flux to apply for all records.
- Fflags** Selects output items; *flags* is a string made up of one or more of these characters:

r means output all input columns before adding the items below

t means list total field (nT).

h means list horizontal field (nT).

x means list X component (nT, positive north).

y means list Y component (nT, positive east).

z means list Z component (nT, positive down).

d means list declination (deg, clockwise from north).

i means list inclination (deg, positive down).

Append one or more number to indicate the requested field contribution(s):

0 means IGRF field (no combinations allowed)

1 means CM4 Core field

2 means CM4 Lithospheric field

3 means CM4 Primary Magnetospheric field

4 means CM4 Induced Magnetospheric field

5 means CM4 Primary ionospheric field

6 means CM4 Induced ionospheric field

7 means CM4 Toroidal field

9 means Core field from IGRF and other contributions from CM4. DO NOT USE BOTH 0 AND 9.

Appending several numbers (1-7) will add up the different contributions. For example **-Ft/12** computes the total field due to Core and Lithospheric sources. Two special cases are allowed, which mix which Core field from IGRF and other sources from CM4. **-Ft/934** computes Core field due to IGRF plus terms 3 and 4 from CM4 (but you can add others). **-Ft/934** the same as above but output the field components. The data is written out in the order they appear in *flags* [Default is **-Frthxyzdi/1**].

- G** Specifies that coordinates are geocentric [geodetic].

- L** Computes J field vectors from certain external sources.

r means output all input columns before adding the items below (all in Ampers/m).

t means list magnitude field.

x means list X component.

y means list Y component.

z means list Z or current function Psi.

Append a number to indicate the requested J contribution:

1 means Induced Magnetospheric field.

2 means Primary ionospheric field.

3 means Induced ionospheric field.

4 means Poloidal field.

-Sclow/high Limits the wavelengths of the core field contribution to the band indicated by the low and high spherical harmonic order [1/13].

-Sslow/high Limits the wavelengths of the lithosphere field contribution to the band indicated by the low and high spherical harmonic order [14/65].

-V[level] (*more ...*) Select verbosity level [c].

-bi[ncols][type] (*more ...*) Select binary input. [Default is 4 input columns unless **-A** is used].

-bo[ncols][type] (*more ...*) Select binary output. [Default is reflected by **-F**].

-h[i/o][n][+c][+d][+rremark][+rtitle] (*more ...*) Skip or produce header record(s).

-:[i/o] (*more ...*) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.10.5 Time Settings

If binary input files are used then absolute time are stored as time relative to the selected epoch. However, since the epoch used is not stored in the data files there can be problems decoding the correct time. The `mgd77` supplement uses the Unix time system as its default; thus you should make sure that binary data files with time uses the same system (see the GMT default `TIME_SYSTEM`).

2.10.6 Examples

To get the CM4 Total field, Declination and Inclination due to all but lithospheric and toroidal field at a one point location and decimal time 2000.0, try

```
echo -28 38 0 2000.0 | gmt mgd77magref -A+y -Ftdi/13456
```

To do the same as above but at noon (Universal Time) of first May 2001, try

echo -28 38 0 2001-05-01T12:00:00 | gmt mgd77magref -Ftdi/13456

2.10.7 See Also

gmt, mgd77info, mgd77list, mgd77manage, mgd77track, gmt.conf

2.10.8 References

Comprehensive Modeling of the Geomagnetic Field, see <http://denali.gsfc.nasa.gov/cm/>

The International Geomagnetic Reference Field (IGRF), see http://www.iugg.org/IAGA/iaga_pages/pubs_prods/igrf.htm

2.11 mgd77manage

mgd77manage - Manage the content of MGD77+ files

2.11.1 Synopsis

mgd77manage *NGDC-ids* [**-A**[+]**a****c****d****i****d****l****e****e****l****i****n****t****f****i****e****i****n****f****o**] [**-C****f****i****l****e**] [**-D****a****b****b****r****e****v****1**,**a****b****b****r****e****v****2**,...)] [**-E****e****m****p****t****y**] [**-F**] [**-I****a****b****b****r****e****v**/**n****a****m****e**/**u****n****i****t**/**t**/**s****c****a****l****e**/**o****f****f****s****e****t**/**c****o****m****m****e****n****t**] [**-N****u****n****i****t**] [**-R**[**u****n****i****t**]**x****m****i****n**/**x****m****a****x**/**y****m****i****n**/**y****m****a****x**[**r**]] [**-V**[**l****e****v****e****l**]] [**-b****i**[**n****c****o****l****s**][**t****y****p****e**][**w**][**+L****+B**]] [**-n**[**b****i****c****l****l****n**][**+a**][**+b****B**C][**+t****t****h****r****e****s****h****o****l****d**]]

Note: No space is allowed between the option flag and the associated arguments.

2.11.2 Description

mgd77manage deals with maintaining extra custom columns in MGD77+ netCDF files. You can either delete one or more columns, add a new column, update an existing column with new data, or supply error correction information (*.e77 files). New data may come from a table (ASCII unless **-bi** is used), be based on existing columns and certain theoretical expressions, or they may be obtained by sampling a grid (choose between GMT grid or a Sandwell/Smith Mercator *.img grid) along track. The new data will be appended to the MGD77+ file in the form of an extra data column of specified type. The data file will be modified; no new file will be created. For the big issues, see the DISCUSSION section below.

2.11.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See `mgd77info -L` for agency and vessel codes). The “.mgd77” or “.nc” extensions will automatically be appended, if needed (use `-I` to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in `$MGD77_HOME/mgd77_paths.txt` [If `$MGD77_HOME` is not set it will default to `$GMT_SHAREDIR/mgd77`].

2.11.4 Optional Arguments

-A[+]*alcldlDleElglilnlTfileinfo* Add a new data column. If an existing column with the same abbreviation already exists in the file we will cowardly refuse to update the file. Specifying `-A+` overcomes this reluctance (However, sometimes an existing column cannot be upgraded without first deleting it; if so you will be warned). Select a column source code among **a**, **c**, **d**, **D**, **e**, **g**, **i**, **n**, **t**, or **T**; detailed descriptions for each choice follow:

a Append filename of a single column table to add. File must have the same number of rows as the MGD77+ file. If no file is given we read from stdin instead.

c Create a new column that derives from existing data or formulas for corrections and reference fields. Append **c** for the Carter corrections subtracted from uncorrected depths, **g** for the IGF gravity reference field (a.k.a “normal gravity”), **m** for the IGRF total field magnetic reference field, and **r** for recomputed magnetic anomaly (append 1 or 2 to specify which total field column to use [1]). For gravity we choose the reference field based on the parameter Gravity Theoretical Formula Code in the cruise’s MGD77 header. If this is not set or is invalid we default to the IGF 1980. You can override this behaviour by appending the desired code: 1 = Heiskanen 1924, 2 = International 1930, 3 = IGF1967, or 4 = IGF1980.

d Append filename of a two-column table with the first column holding distances along track and the second column holding data values. If no file is given we read from stdin instead. Records with matching distances in the MGD77+ file will be assigned the new values; at other distances we set them to NaN. Alternatively, give upper case **D** instead and we will interpolate the column at all record distances. See `-N` for choosing distance units and `-C` for choosing how distances are calculated.

e Expects to find an `e77` error/correction log from `mgd77sniffer` with the name `NGDC_ID.e77` in the current directory or in `$MGD77_HOME/E77`; this file will be examined and used to make modifications to the header values, specify a systematic correction for certain columns (such as scale and offset), specify that a certain anomaly should be recalculated from the observations (e.g., recalculate mag from `mtf1` and the latest IGRF), and add or update the special column **flag** which may hold bitflags (0 = GOOD, 1 = BAD) for each data field in the standard MGD77 data set. Any fixed correction terms found (such as needing to scale a field by 0.1 or 10 because the source agency used incorrect units) will be written as attributes to the netCDF MGD77+ file and applied when the data are read by `mgd77list`. Ephemeral corrections such as those determined by crossover analysis are not kept in the data files but reside in correction tables (see `mgd77list` for details). By default, the first character of each header line in the `e77` file (which is `?`, `Y` or `N`) will be consulted to see if the corresponding adjustment should be applied. If any undecided settings are found (i.e., `?`) we will abort and make no changes. Only records marked `Y` will be processed. You can override this behavior by appending one or more modifiers to the `-Ae` command: **h** will ignore all header corrections, **f** will ignore all fixed systematic trend corrections, **n**, **v**, and **s** will ignore bitflags pertaining to navigation, data values, and data slopes, respectively. Use `-A+e` to replace any existing `E77` corrections in the file with the new values. Finally, `e77` corrections will not be applied if the `E77` file has not been verified. Use `-AE` to ignore the verification status.

g Sample a GMT geographic (lon, lat) grid along the track given by the MGD77+ file using bicubic interpolation (however, see `-n`). Append name of a GMT grid file.

- i** Sample a Sandwell/Smith Mercator *.img grid along the track given by the MGD77+ file using bicubic interpolation (however, see **-n**). Append the img grid filename, followed by the comma-separated data scale (typically 1 or 0.1), the IMG file mode (0-3), and optionally the img grid max latitude [80.738]. The modes stand for the following: (0) Img files with no constraint code, returns data at all points, (1) Img file with constraints coded, return data at all points, (2) Img file with constraints coded, return data only at constrained points and NaN elsewhere, and (3) Img file with constraints coded, return 1 at constraints and 0 elsewhere.
- n** Append filename of a two-column table with the first column holding the record number (0 to nrows - 1) and the second column holding data values. If no file is given we read from stdin instead. Records with matching record numbers in the MGD77+ file will be assigned the new values; at other records we set them to NaN.
- t** Append filename of a two-column table with the first column holding absolute times along track and the second column holding data values. If no file is given we read from stdin instead. Records with matching times in the MGD77+ file will be assigned the new values; at other times we set them to NaN. Alternatively, give upper case **T** instead and we will interpolate the column at all record times.
- Cflgle** Append a one-letter code to select the procedure for along-track distance calculation when using **-AdID** (see **-N** for selecting distance units):
- f** Flat Earth distances.
 - g** Great circle distances [Default].
 - e** Geodesic distances on current GMT ellipsoid.
- Dabbrev1,abbrev2,...)** Give a comma-separated list of column abbreviations that you want to delete from the MGD77+ files. Do NOT use this option to remove columns that you are replacing with new data (use **-A+** instead). Because we cannot remove variables from netCDF files we must create a new file without the columns to be deleted. Once the file is successfully created we temporarily rename the old file, change the new filename to the old filename, and finally remove the old, renamed file.
- Empty** Give a single character that will be repeated to fill empty string values, e.g., “9” will yield a string like “99999...” [9].
- F** Force mode. When this mode is active you are empowered to delete or replace even the standard MGD77 set of columns. You better know what you are doing!
- Iabbrev/name/unit/t/scale/offset/comment** In addition to file information we must specify additional information about the extra column. Specify a short (16 char or less, using lower case letters, digits, or underscores only) abbreviation for the selected data, its more descriptive name, the data unit, the data type 1-character code (**byte**, **short**, **float**, **int**, **double**, or **text**) you want used for storage in the netCDF file, any scale and offset we should apply to the data to make them fit inside the range implied by the chosen storage type, and a general comment (< 128 characters) regarding what these data represent. Note: If text data type is selected then the terms “values” in the **-A** discussion refer to your text data. Furthermore, the discussion on interpolation does not apply and the NaN value becomes a “no string” value (see **-E** for what this is). Place quotes around terms with more than one word (e.g., “Corrected Depth”).
- Nunit** Append the distance unit (see UNITS). [Default is **-Nk** (km)]. Only relevant when **-Agi** is selected.
- R[unit]xmin/xmax/ymin/ymax[r] (more ...)** Specify the region of interest. Only relevant when **-Agi** is selected.

- V[*level*]** (*more ...*) Select verbosity level [*c*].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. This applies to the input 1- or 2-column data files specified under some of the **-A** options. The binary input option is only available for numerical data columns.
- n[*blc|lln*][*+a*][*+bBC*][*+c*][*+threshold*]** (*more ...*) Select interpolation mode for grids.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.11.5 Units

For map distance unit, append *unit* **d** for arc degree, **m** for arc minute, and **s** for arc second, or **e** for meter [Default], **f** for foot, **k** for km, **M** for statute mile, **n** for nautical mile, and **u** for US survey foot. By default we compute such distances using a spherical approximation with great circles. Prepend **-** to a distance (or the unit is no distance is given) to perform “Flat Earth” calculations (quicker but less accurate) or prepend **+** to perform exact geodesic calculations (slower but more accurate).

2.11.6 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see **-n**) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values values so they do not exceed the input min/max values by adding **+c** to your **-n** option.

2.11.7 Examples

To append Geosat/ERS-1 gravity version 11.2 as an extra data column in the cruises 01010047.nc and 01010008.nc, storing the values as mGal*10 in a 2-byte short integer, try

```
gmt mgd77manage 01010047 01010008 -A10/1/grav.11.2.img \
  -Isatgrav/"Geosat/ERS-1 gravity"/"mGal"/s/10/0/"Sandwell/Smith version 11.2" -V
```

To append a filtered version of magnetics as an extra data column of type float for the cruise 01010047.nc, and interpolate the filtered data at the times given in the MGD77+ file, try

```
gmt mgd77manage 01010047 -ATmymag.tm -Ifiltmag/"Intermediate-wavelength \
  magnetic residuals"/"nTesla"/f/1/0/"Useful for looking for isochrons" -V
```

To delete the existing extra columns satfaa, coastdist, and satvgg from all MGD77+ files, try

```
gmt mgd77manage =allmgd77.lis -Dsatfaa,coastdist,satvgg -V
```

To create a 4-byte float column with the correct IGRF reference field in all MGD77+ files, try

```
gmt mgd77manage =allmgd77.lis -Acm -Igrf/"IGRF reference \  
field"/"nTesla"/f/1/0/"IGRF version 10 for 1990-2010" -V
```

2.11.8 Discussion

1. Preamble

The `mgd77` supplement is an attempt to (1) improve on the limited functionality of the existing `mgg` supplement, (2) incorporate some of the ideas from Scripps' `gmt+` supplement by allowing extra data columns, and (3) add new capabilities for managing marine geophysical trackline data stored in an architecture-independent CF-1.0- and COARDS-compliant netCDF file format. Here are some of the underlying ideas and steps you need to take to maintain your files.

2. Introduction

Our starting point is the MGD77 ASCII data files distributed from NGDC on CD-ROMS, DVD-ROMS, and via FTP. Using Geodas to install the files locally we choose the "Carter corrected depth" option which will fill in the depth column using the two-way traveltimes and the Carter tables if `twt` is present. This step yields ~5000 individual cruise files. Place these in one or more sub-directories of your choice, list these sub-directories (one per line) in the file `mgd77_paths.txt`, and place that file in the directory pointed to by `$MGD77_HOME`; if not set this variable defaults to `$GMT_SHAREDIR/mgd77`.

3. Conversion

Convert the ASCII MGD77 files to the new netCDF MGD77+ format using `mgd77convert`. Typically, you will make a list of all the cruises to be converted (with or without extension), and you then run

```
mgd77convert =cruises.lis -Fa -Tc -V -Lwe+ > log.txt
```

The verbose settings will ensure that all problems found during conversion will be reported. The new `*.nc` files may also be placed in one or more separate sub-directories and these should also be listed in the `mgd77_paths.txt` file. We suggest you place the directories with `*.nc` files ahead of the `*.mgd77` directories. When you later want to limit a search to files of a certain extension you should use the `-I` option.

4. Adding new columns

`mgd77manage` will allow you to add additional data columns to your `*.nc` files. These can be anything, including text strings, but most likely are numerical values sampled along the track from a supplied grid or an existing column that have been filtered or manipulated for a particular purpose. The format supports up to 32 such extra columns. See this man page for how to add columns. You may later decide to remove some of these columns or update the data associated with a certain column. Data extraction tools such as `mgd77list` can be used to extract a mix of standard MGD77 columns (navigation, time, and the usual geophysical observations) and your custom columns.

5. Error sources

Before we discuss how to correct errors we will first list the different classes of errors associated with MGD77 data: (1) Header record errors occur when some of the information fields in the header do not comply with the MGD77 specification or required information is missing. `mgd77convert` will list these errors when the extended verbose setting is selected. These errors typically do not affect the data and are instead errors in the *meta-data* (2). Fixed systematic errors occur when a particular data column, despite the MGD77 specification, has been encoded incorrectly. This usually means the data will be off by a constant factor such as 10 or 0.1, or in some cases even 1.8288 which converts fathoms to meters. (3) Unknown systematic errors occur when the instrument that recorded the data or the processing that followed introduced signals that appear to be systematic functions of time along track, latitude, heading,

or some other combination of terms that have a physical or logical explanation. These terms may sometimes be resolved by data analysis techniques such as along-track and across-track investigations, and will result in correction terms that when applied to the data will remove these unwanted signals in an optimal way. Because these correction terms may change when new data are considered in their determination, such corrections are considered to be ephemeral. (4) Individual data points or sequences of data may violate rules such as being outside of possible ranges or in other ways violate sanity. Furthermore, sequences of points that may be within valid ranges may give rise to data gradients that are unreasonable. The status of every point can therefore be determined and this gives rise to bitflags GOOD or BAD. Our policy is that error sources 1, 2, and 4 will be corrected by supplying the information as meta-data in the relevant *.nc files, whereas the corrections for error source 3 (because they will constantly be improved) will be maintained in a separate list of corrections.

6. Finding errors

The **mgd77sniffer** is a tool that does a thorough along-track sanity check of the original MGD77 ASCII files and produces a corresponding *.e77 error log. All problems found are encoded in the error log, and recommended fixed correction terms are given, if needed. An analyst may verify that the suggested corrections are indeed valid (we only want to correct truly obvious unit errors), edit these error logs and modify such correction terms and activate them by changing the relevant code key (see **mgd77sniffer** for more details). **mgd77manage** can ingest these error logs and (1) correct bad header records given the suggestions in the log, (2) insert scale/offset correction terms to be used when reading certain columns, and (3) insert any bit-flags found. Rerun this step if you later find other problems as all E77 settings or flags will be recreated based on the latest E77 log.

7. Error corrections

The extraction program *mgd77list* allows for corrections to be applied on-the-fly when data are requested. First, data with BAD bitflags are suppressed. Second, data with fixed systematic correction terms are corrected accordingly. Third, data with ephemeral correction terms will have those corrections applied (if a correction table is supplied). All of these steps require the presence of the relevant meta-data and all can be overruled by the user. In addition, users may add their own bitflags as separate data columns and use **mgd77list**'s logical tests to further dictate which data are suppressed from output.

2.11.9 Credits

The IGRF calculations are based on a Fortran program written by Susan Macmillan, British Geological Survey, translated to C via f2c by Joaquim Luis, and adapted to GMT style by Paul Wessel.

2.11.10 See Also

mgd77convert, *mgd77list*, *mgd77info*, *mgd77sniffer* *mgd77track* *x2sys_init*

2.11.11 References

The Marine Geophysical Data Exchange Format - MGD77, see <http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt>

IGRF, see <http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html>

2.12 mgd77path

mgd77path - Return paths to MGD77 cruises and directories

2.12.1 Synopsis

mgd77path *NGDC-ids* [**-A**[-]] [**-D**] [**-I**ignore] [**-V**[level]]

Note: No space is allowed between the option flag and the associated arguments.

2.12.2 Description

mgd77path returns the full pathname to one or more MGD77 files. The pathname returned for a given cruise may change with time due to reshuffling of disks/subdirectories.

2.12.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See **mgd77info -L** for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use **-I** to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in **\$MGD77_HOME/mgd77_paths.txt** [If **\$MGD77_HOME** is not set it will default to **\$GMT_SHAREDIR/mgd77**].

2.12.4 Optional Arguments

- A**[-] Display the full path to each cruise [Default]. Optionally, append **-** which will list just the cruise IDs instead.
- D** Instead of cruise listings, just show the directory paths currently used in the search.
- I**ignore Ignore certain data file formats from consideration. Append **alcIt** to ignore MGD77 ASCII, MGD77+ netCDF, or plain tab-separated ASCII table files, respectively. The option may be repeated to ignore more than one format. [Default ignores none].
- V**[level] (*more ...*) Select verbosity level [c]. Reports the total number of cruises found.
- ^** or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- ++** or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ?** or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.12.5 Examples

To obtain pathnames for cruises 01010008 and 01010007, run

```
gmt mgd77path 01010008 01010007
```

To obtain pathnames for cruises 01010008 and 01010007, but only if there are MGD77+ version in netCDF, run

```
gmt mgd77path 01010008 01010007 -Ia -It
```

To see the list of active directories where MGD77 files might be stored, run

```
gmt mgd77path -D
```

2.12.6 See Also

gmt mgd77info mgd77list mgd77manage mgd77track

2.12.7 References

The Marine Geophysical Data Exchange Format - MGD77, see [*http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt*](http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt).

2.13 mgd77sniffer

mgd77sniffer - Along-track quality control of MGD77 cruises

2.13.1 Synopsis

```
mgd77sniffer NGDC-ids [ -Afieldabbrev,scale,offset ] [ -Cmaxspd ] [ -DdelE|fl|ms|v[r] ] [ -Gfieldabbrev,imggrid,scale,mode or -Gfieldabbrev,grid ] [ -H ] [ -Ifieldabbrev,rec1,recN ] [ -K ] [ -Lcustom-limits-file ] [ -N ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Sd|s|t ] [ -Tgap ] [ -V[level] ] [ -Wc|g|l|s|t|v|x ] [ -bo[ncols][type][w][+L|+B] ] [ -n[bc|lln][+a][+bBC][+threshold] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.13.2 Description

mgd77sniffer scans old (pre-Y2K) and new format ASCII MGD77 files for errors using point-by-point sanity checking, along-track detection of excessive slopes, and optional comparison of cruise data with global gravity and predicted bathymetry grids. Detected data problems are output by default as verbose descriptions of each detected error, often resulting in multiple messages per scanned record. Data problems are optionally output (**-De** option) using a computer-parseable format (see E77 ERROR FORMAT description below). Default error thresholds are derived from histograms of all MGD77 geophysical data collected between 1952 and January, 2006. Thresholds are adjustable with the **-L** option.

2.13.3 Requirements

The `mgd77sniffer` links with Generic Mapping Tools 4.0 or later along with the supplemental GMT packages `x2sys` and `mgd77`. See <http://gmt.soest.hawaii.edu> for GMT details. Grids for comparison with cruise data may be downloaded via the web.

2.13.4 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See `mgd77info -L` for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use `-I` to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in `$MGD77_HOME/mgd77_paths.txt` [If `$MGD77_HOME` is not set it will default to `$GMT_SHAREDIR/mgd77`].

2.13.5 Optional Arguments

-Afieldabbrev,scale,offset Apply scale factor and DC adjustment to specified data field. Allows adjustment of cruise data prior to along-track analysis. CAUTION: data must be thoroughly examined before applying these global data adjustments. May not be used for multiple cruises.

-Cmaxspd Set maximum ship speed in m/s, or knots with `-N` option. Ship speeds exceeding 10 m/s (~20 knots) are flagged as excessive by default.

-Ddle|E|fill|ms|v[r] Suppress default warning output and only dump cruise data row-by-row such as values, gradients, grid-cruise differences, E77 error summaries for each record, re-created MGD77 records or sniffer limits. Append `r` to include all records (default omits records where navigation errors were detected).

-Dd output differences between cruise and grid data. Requires `-G` option. Output columns include:

lat lon dist cruiseZ gridZ diff [cruiseZ2 gridZ2 diff2 ...]

Note: grid values are subtracted from cruise data so a positive difference implies cruise > grid. For multiple grid comparison, *cruiseZ gridZ diff* are repeated for each grid comparison in command line order.

-De output E77 error classification format. Error output is divided into (1) a header containing information globally applicable to the cruise and (2) individual error records summarizing all errors encountered in each cruise record. `mgd77sniffer` writes E77 directly to <ngdc_id.e77> file handle. See **E77 ERROR FORMAT** below for additional details.

-DE Same as `-De` but no regression tests will be carried out.

-Df output delta Z (change in geophysical field) column and delta S (change in distance) for each geophysical field. Distance between observations often differ for different fields depending on

instrument sampling rate, so *ds* is included for each geophysical observation. Output columns include:

d[twt] ds d[depth] ds d[mtf1] ds d[mtf2] ds d[mag] ds d[diur] ds d[msd] ds d[gobs] ds d[eot] ds d[faa] ds

-Dl display mgd77sniffer limits. Customize this output to create a custom limits file for the **-L** option. No additional arguments are required. Output columns include:

fieldabbrev min max maxSlope maxArea

-Dm output MGD77 format records in Y2K-compliant MGD77 format

-Dn output distance to coast for each record. Requires the **-Gnav** option. Output columns include:

lat lon dist distToCoast

-Ds output calculated gradients for speed and geophysical fields. Gradients correspond to the gradient type selected in the **-S** option (spatial derivatives by default). Output columns include:

speed d[twt] d[depth] d[mtf1] d[mtf2] d[mag] d[diur] d[msd] d[gobs] d[eot] d[faa]

See **MGD77 FIELD INFO** below for field and abbreviations descriptions.

-Dv display values for the twelve position and geophysical fields for each MGD77 data record (in this order):

lat lon twt depth mtf1 mtf2 mag diur msens gobs eot faa

See below for **MGD77 FIELD INFO**.

-Ginformation Compare cruise data to GMT or IMG grids. Use one of the formats below. **-Gfieldabbrev,imggrid,scale,mode** Compare cruise data to the specified grid in Sandwell/Smith Mercator format. Requires a valid MGD77 field abbreviation (see **MGD77 FIELD INFO** below) followed by a comma, the path (if not in current directory) and grid filename, a scale to multiply the data (1 or 0.1), and mode which stand for the following: (0) Img files with no constraint code, returns data at all points, (1) Img file with constraints coded, return data at all points, (2) Img file with constraints coded, return data only at constrained points and NaN elsewhere, and (3) Img file with constraints coded, return 1 at constraints and 0 elsewhere. **-Gfieldabbrev,grid** Compare cruise data to the specified grid. Requires a valid MGD77 field abbreviation (see **MGD77 FIELD INFO** below) followed by a comma, then the path (if not in current directory) and grid filename. Multiple grid comparison is supported by using separate **-G** calls for each grid. See **GRID FILE INFO** below.

Grid comparison activates several additional error checks. (1) Re-weighted Least Squares Regression of ship versus grid data determines slope and DC shift, which when differing from expected 1 and 0, respectively, may indicate incorrectly scaled ship data, including incorrect units or instrument drift as well as erroneous gravity tie-in. (2) Accumulated ship grid offsets are computed along-track and excessive offsets are flagged according to *maxArea* threshold (use **-L** option to adjust *maxArea*). Warning: predicted bathymetry grids are constrained by cruise data so grids and cruise data are not always independent. Comparison of cruise bathymetry with predicted bathymetry grids also activates a “navigation crossing over land” check.

-H (with **-Glg** only) disable (or force) decimation during RLS analysis of ship and gridded data. By default mgd77sniffer analyses both the full and decimated data sets then reports RLS statistics for the higher correlation regression.

-Hb analyze both (default), report better of two.

-Hd to disable data decimation (equivalent to **-H** with no argument).

-Hf to force data decimation.

-Ifieldabbrev,rec1,recN Append a field abbreviation and the first and last record in a range of records that should be flagged as bad (and set to NaN prior to the analysis). Repeat as many times as needed. May not be used for multiple cruises.

-K Reverse navigation quality flags (good to bad and vice versa). May be necessary when a majority of navigation fixes are erroneously flagged bad, which can happen when a cruise's first navigation fix is extremely erroneous. Caution! This will affect sniffer output and should only be attempted after careful manual navigation review.

-Lcustom-limits-file Override mgd77sniffer default error detection limits. Supply path and filename to the custom limits file. Rows not beginning with a valid MGD77 field abbreviation are ignored. Field abbreviations are listed below in exact form under MGD77 FIELD INFO. Multiple field limits may be modified using one default file, one field per line. Field min, max, max slope and max area may be changed for each field. Max slope pertains to the gradient type selected using the **-S** option. Max area is used by the **-G** option as the threshold for flagging excessive offsets from the specified grid. Dump defaults **-DI** to view syntax or to quickly create an editable custom limits file.

Example custom default file contents (see below for units):

| # abbrev | min | max | maxSlope | maxArea |
|----------|------|-------|----------|---------|
| twf | 0 | 15 | 1 | 0 |
| depth | 0 | 11000 | 500 | 5000 |
| mag | -800 | 800 | - | - |
| faa | -300 | 300 | 100 | 2500 |

Use a dash '-' to retain a default limit. Hint: to test your custom limits, try: mgd77sniffer **-DI** **-L<yourlimitsfile>**

-N Use nautical units.

-P Flag regression statistics that are outside the specified confidence level. (i.e., **-P5** flags coefficients *m*, *b*, *rms*, and *r* that fall outside 95%.)

-R[unit]west/east/south/north[/zmin/zmax][r] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+ -]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-Sdslt Specify gradient type for along-track excessive slope checking. **-Sd** Calculate change in *z* values along track (*dz*). Output is given in geophysical units, e.g., mGal. **-Ss** Calculate spatial gradients (*dz/ds*). Output is given in geophysical units per km along the survey track, e.g., mGal/km. **-St** Calculate time gradients (*dz/dt*) [default]. Output is given in geophysical units per second along the survey track, e.g., mGal/sec.

-Tgap Adjusts mgd77sniffer gap handling. By default, data gaps greater than 5 km are skipped. Set to zero to de-activate gap skipping.

- Wc|g|l|o|l|s|t|l|v|x** Print out only certain warning types for verbose error messages. Comma delimit any combination of **c|g|l|o|l|s|t|l|v|x**: where (**c**) type code warnings, (**g**)radient out of range, (**o**)ffsets from grid (requires **-G|g**), (**s**)peed out of range, (**t**)ime warnings, (**v**)alue out of range, (**x**) warning summaries. By default ALL warning messages are printed. Not compatible with any **-D** options.
- V[level]** (*more ...*) Select verbosity level [c].
- bo[ncols][type]** (*more ...*) Select binary output. Output binary data for **-Dd|f|l|s|v** option.
- n[blc|lln][+a][+bBC][+c][+tthreshold]** (*more ...*) Select interpolation mode for grids.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- +- or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.13.6 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see **-n**) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values values so they do not exceed the input min/max values by adding **+c** to your **-n** option.

2.13.7 Mgd77 Field Info

| <i>Field</i> | <i>Abbreviation</i> | <i>Units</i> |
|--------------|---------------------|--------------|
| Two-way | Travel | Time |
| Corrected | Depth | depth |
| Mag | Total | Field1 |
| Mag | Total | Field2 |
| Residual | Magnetic | mag |
| Diurnal | Correction | diur |
| Mag | Sensor | Depth/Alt |
| Observed | Gravity | gobs |
| Eotvos | Correction | eot |
| Free | Air | Anomaly |

2.13.8 Grid File Info

For **-G** the grids must either be in the format used by Sandwell & Smith, which is a spherical Mercator 2-byte grid with no header, or any grid type supported by GMT and therefore must contain a GMT header. A correctly formatted *.i2 grid file can be generated using `grdraster` as shown below.

```
gmtset GRIDFILE_SHORTHAND TRUE
```

Create/edit `.gmt_io` file to include the following rows:

```
# GMT I/O shorthand file
# suffix format_id scale offset NaN
grd 0 - - - -
i2 2 - - 32767

gmt grdraster 1 -R0/359:55/-90/90 -Getopo5_hdr.i2
```

The new grid, `etopo5_hdr.i2` in this example, contains a GMT header and can be used in the `-G` option to compare cruise depth with grid values.

2.13.9 E77 Error Format

Header Information pertaining to an entire cruise, such as NGDC and survey institution identification codes, cruise examination time, two-way travel time corrector information, data precision warnings, as well as systematic scales, DC shifts and correlation coefficients from global grid comparisons are reported as E77 header information.

Sample

```
# Cruise 08010039 ID 74010908 MGD77 FILE VERSION: 19801230 N_RECS: 3066
# Examined: Wed Oct 3 16:30:13 2007 by mtchndl
# Arguments: -De -Gdepth,/data/GRIDS/etopo5_hdr.i2
N Errata table verification status

# mgd77manage applies corrections if the errata table is verified (toggle 'N' above to 'Y' after
review)

# For instructions on E77 format and usage, see http://gmt.soest.hawaii.edu/mgd77/errata.php
# Verified by:
# Comments:
# Errata: Header
Y-E-08010039-H13-02: Invalid Magnetism Sampling Rate: (99) [ ]
Y-W-08010039-H13-10: Survey year (1975) outside magnetic reference field IGRF 1965 time
range (1965-1970)
Y-I-08010039-depth-00: RLS m: 1.00053 b: 0 rms: 127.851 r: 0.973422 significant: 1 decimation:
0
Y-W-08010039-twt-09: More recent bathymetry correction table available
Y-W-08010039-mtf1-10: Integer precision
Y-W-08010039-mag-10: Integer precision
```

Error Record Individual error records have strict format. Included is a time or distance column followed by record number, a formatted error code string, and finally a verbose description of errors detected in the record. Three error classes are encoded into the error code string with different alphabetic characters representing unique error types. See below for error code format description.

Format <time/distance> <record number> <error code string> <description>

Sample

Errata: Data

Y 08010039 1975-05-10T22:16:05.88 74 C-0-0 NAV: excessive speed

Error Code Description Each of the three error classes is separated by a dash - and described by a combination of alphabetic characters or 0 signifying no detected problems.

Error classes: NAV-VAL-GRAD

Error Class Descriptions

NAV (navigation):

- 0 - fine
- A - time out of range
- B - time decreasing
- C - excessive speed
- D - above sea level
- E - lat undefined
- F - lon undefined

VAL (value):

- 0 - fine
- K - twt invalid
- L - depth invalid
- O - mtf1 invalid

etc.

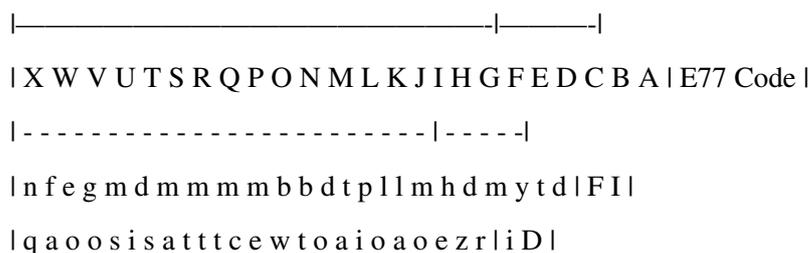
GRAD (gradient):

- 0 - fine
- K - d[twt] excessive
- L - d[depth] excessive

O - d[mtf1] excessive etc.

The NAV error class has unique cases while VAL and GRAD classes are described by alphabetic characters for each of the 24 numeric fields in MGD77 format order.

MGD77 bit-pattern w/ E77 alpha characters



2.14.2 Description

mgd77track reads NGDC MGD77 cruises and creates PostScript code that will plot one or more ship tracks on a map using the specified projection. The PostScript code is written to standard output.

2.14.3 Required Arguments

NGDC-ids Can be one or more of five kinds of specifiers:

1. 8-character NGDC IDs, e.g., 01010083, JA010010etc., etc.
2. 2-character <agency> codes which will return all cruises from each agency.
3. 4-character <agency><vessel> codes, which will return all cruises from those vessels.
4. =<list>, where <list> is a table with NGDC IDs, one per line.
5. If nothing is specified we return all cruises in the data base.

(See **mgd77info -L** for agency and vessel codes). The ".mgd77" or ".nc" extensions will automatically be appended, if needed (use **-I** to ignore certain file types). Cruise files will be looked for first in the current directory and second in all directories listed in **\$MGD77_HOME/mgd77_paths.txt** [If **\$MGD77_HOME** is not set it will default to **\$GMT_SHAREDIR/mgd77**].

-Jparameters (more ...) Select map projection.

-R[unit]west/east/south/north[/zmin/zmax][r] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+ -]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

2.14.4 Optional Arguments

-A[c][size][,spacing] Append **c** to annotate using the MGD77 cruise ID [Default uses the filename prefix]. Optional *size* is the font size in points. The leg annotation font is controlled by **FONT_LABEL**. By default, each leg is annotated every time it enters the map region. Alternatively, append *,spacing* to place this label every *spacing* units apart along the track. Append one of the units **k** (km), **n** (nautical mile), **d** (day), or **h** (hour).

-B[pls]parameters (more ...) Set map boundary intervals.

-Cfgle

Select procedure for along-track distance calculation: **f** Flat Earth distances. **g** Great circle distances [Default]. **e** Geodesic distances on current GMT ellipsoid.

-Dstartdate Do not plot data collected before *startdate* (yyyy-mm-ddBD(T)[hh:mm:ss]) [Default is first day].

- Dbstopdate** Do not plot data collected after *stopdate* (yyyy-mm-ddBD(T)[hh:mm:ss]). [Default is last day].
- F** Do not apply the error bit flags if present in a MGD77+ file [Default will apply these flags upon reading the data].
- Gdltgap** Let successive point separations exceeding **dgap** (km) or **tgap** (minutes) indicate a break in the track where we should not draw a line [no gaps recognized]. Repeat to use both types of gap checking.
- Iignore** Ignore certain data file formats from consideration. Append **alcit** to ignore MGD77 ASCII, MGD77+ netCDF, or plain table files, respectively. The option may be repeated to ignore more than one format. [Default ignores none].
- K (more ...)** Do not finalize the PostScript plot.
- Ltrackticks** To put time/distance log-marks on the track. E.g. **a500ka24ht6h** means (**a**)nnotate every 500 km (**k**) and 24 **h**(ours), with (**t**)ickmarks every 500 km and 6 hours. Alternatively you may use the modifiers **d** (days) and **n** (nautical miles).
- O (more ...)** Append to existing PostScript plot.
- P (more ...)** Select “Portrait” plot orientation.
- Sastartdist[u]** Do not plot data that are less than *startdist* meter along track from port of departure. Append **k** for km, **m** for miles, or **n** for nautical miles [Default is 0 meters].
- Sbstopdist[u]** Do not plot data that are more than *stopdist* meter along track from port of departure. Append **k** for km, **m** for miles, or **n** for nautical miles [Default is end of track].
- TT|tdms,mc,mfs,mf,mfc** Controls the attributes of the three kinds of markers (**T** for the first time marker in a new day, **t** for additional time markers in the same day, and **d** for distance markers). For each of these you can specify the 5 comma-separated attributes *markersize*, *markercolor*, *markerfontsize*, *markerfont*, and *markerfontcolor*. Repeat the **-T** option for each marker type.
- U[just/dx/dy/][clabel] (more ...)** Draw GMT time stamp logo on plot.
- W[-|+][pen]** Append *pen* used for the trackline. [Defaults: width = default, color = black, style = solid]. A leading + will use the lookup color (via **-C**) for both symbol fill and outline pen color, while a leading - will set outline pen color and turn off symbol fill.
- X[alcflr][x-shift[u]]**
- Y[alcflr][y-shift[u]] (more ...)** Shift plot origin.
- V[level] (more ...)** Select verbosity level [c].
- ccopies (more ...)** Specify number of plot copies [Default is 1].
- p[x|y|z]azim/elev[zlevel][+wlon0/lat0[z0]][+vx0/y0] (more ...)** Select perspective view.
- t[transp] (more ...)** Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.14.5 Examples

To generate a Mercator plot of the track of the cruise 01010007 in the area 70W to 20E, 40S to 20N, using a Mercator scale of 0.1inch/degree, label the tracks with 10 points characters, annotate the boundaries every 10 degrees, draw gridlines every 5 degrees, and mark the track every day and 1000 km, with ticks every 6 hours and 250 km, and send the plot to the default printer, enter the following command:

```
gmt mgd77track 01010007 -R70W/20E/40S/20N -Jm0.1 -B10g5 -A10 \
-La1da1000kf6hf250k \ | lpr
```

2.14.6 See Also

mgd77info, *psbasemap*, *mgd77list*

2.14.7 References

The Marine Geophysical Data Exchange Format - MGD77, see <http://www.ngdc.noaa.gov/mgg/dat/geodas/docs/mgd77.txt>

2.15 dimfilter

dimfilter - Directional filtering of 2-D gridded files in the space (or time) domain

2.15.1 Synopsis

```
dimfilter input_file.nc -Ddistance_flag -F<filtertype><width>[mode] -Goutput_file.nc
-N<filtertype><n_sectors> [ -Qcols ] [ -Lxinc[unit][=|+][yinc[unit][=|+]] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -T ] [ -V[level] ] [ -f[ilo]colinfo ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.15.2 Description

dimfilter will filter a *.nc* file in the space (or time) domain by dividing the given filter circle into *n_sectors*, applying one of the selected primary convolution or non-convolution filters to each sector, and choosing the final outcome according to the selected secondary filter. It computes distances using Cartesian or Spherical geometries. The output *.nc* file can optionally be generated as a subregion of the input and/or with a new **-Increment**. In this way, one may have “extra space” in the input data so that there will be no edge effects for the output grid. If the filter is low-pass, then the output may be less frequently sampled than the input. **-Q** is for the error analysis mode and only requires the total number of columns in the input file, which contains the filtered depths. Finally, one should know that **dimfilter** will not produce a smooth output as other spatial filters do because it returns a minimum median out of *N* medians of *N* sectors. The output can be rough unless the input data is noise-free. Thus, an additional filtering (e.g., Gaussian via **grdfilter**) of the DiM-filtered data is generally recommended.

2.15.3 Required Arguments

input_file.nc The data grid to be filtered.

-Ddistance_flag Distance *flag* tells how grid (x,y) relates to filter *width*, as follows:

flag = 0: grid (x,y) same units as *width*, Cartesian distances. *flag* = 1: grid (x,y) in degrees, *width* in kilometers, Cartesian distances. *flag* = 2: grid (x,y) in degrees, *width* in km, dx scaled by cos(middle y), Cartesian distances.

The above options are fastest because they allow weight matrix to be computed only once. The next three options are slower because they recompute weights for each latitude.

flag = 3: grid (x,y) in degrees, *width* in km, dx scaled by cosine(y), Cartesian distance calculation.

flag = 4: grid (x,y) in degrees, *width* in km, Spherical distance calculation.

-F<filtertype><width>[mode] Sets the primary filter type. Choose among convolution and non-convolution filters. Append the filter code followed by the full diameter *width*. Available convolution filters are:

(b) Boxcar: All weights are equal.

(c) Cosine Arch: Weights follow a cosine arch curve.

(g) Gaussian: Weights are given by the Gaussian function.

Non-convolution filters are:

(m) Median: Returns median value.

(p) Maximum likelihood probability (a mode estimator): Return modal value. If more than one mode is found we return their average value. Append - or + to the filter width if you rather want to return the smallest or largest of the modal values.

-N<filtertype><n_sectors> Sets the secondary filter type and the number of bow-tie sectors. *n_sectors* must be integer and larger than 0. When *n_sectors* is set to 1, the secondary filter is not effective. Available secondary filters are:

(l) Lower: Return the minimum of all filtered values.

(u) Upper: Return the maximum of all filtered values.

(a) Average: Return the mean of all filtered values.

(m) Median: Return the median of all filtered values.

(p) Mode: Return the mode of all filtered values.

-Goutput_file.nc *output_file.nc* is the output of the filter.

2.15.4 Optional Arguments

-I *x_inc* [and optionally *y_inc*] is the output Increment. Append **m** to indicate minutes, or **c** to indicate seconds. If the new *x_inc*, *y_inc* are NOT integer multiples of the old ones (in the input data), filtering will be considerably slower. [Default: Same as input.]

-R *west*, *east*, *south*, and *north* defines the Region of the output points. [Default: Same as input.]

-T Toggle the node registration for the output grid so as to become the opposite of the input grid [Default gives the same registration as the input grid].

- Qcols cols** is the total number of columns in the input text table file. For this mode, it expects to read depths consisted of several columns. Each column represents a filtered grid with a filter width, which can be obtained by **grd2xyz -Z**. The outcome will be median, MAD, and mean. So, the column with the medians is used to generate the regional component and the column with the MADs is used to conduct the error analysis.
- V[level] (more ...)** Select verbosity level [c].
- f[ilo]colinfo (more ...)** Specify data types of input and/or output columns.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.15.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “z”. See *grdreformat* and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

2.15.6 Geographical And Time Coordinates

When the output grid type is netCDF, the coordinates will be labeled “longitude”, “latitude”, or “time” based on the attributes of the input data or grid (if any) or on the **-f** or **-R** options. For example, both **-f0x -f1t** and **-R90w/90e/0t/3t** will result in a longitude/time grid. When the x, y, or z coordinate is time, it will be stored in the grid as relative time since epoch as specified by *TIME_UNIT* and *TIME_EPOCH* in the *gmt.conf* file or on the command line. In addition, the **unit** attribute of the time variable will indicate both this unit and epoch.

2.15.7 Examples

Suppose that `north_pacific_dbdb5.nc` is a file of 5 minute bathymetry from 140E to 260E and 0N to 50N, and you want to find the medians of values within a 300km radius (600km full width) of the output points, which you choose to be from 150E to 250E and 10N to 40N, and you want the output values every 0.5 degree. To prevent the medians from being biased by the sloping plane, you want to divide the filter circle into 6 sectors and to choose the lowest value among 6 medians. Using spherical distance calculations, you need:

```
gmt dimfilter north_pacific_dbdb5.nc -Gfiltered_pacific.nc -Fm600 -D4 \
-N16 -R150/250/10/40 -I0.5 -V
```

Suppose that `cape_verde.nc` is a file of 0.5 minute bathymetry from 32W to 15W and 8N to 25N, and you want to remove small-length-scale features in order to define a swell in an area extending from 27.5W to 20.5W and 12.5N to 19.5N, and you want the output value every 2 minute. Using cartesian distance calculations, you need:

```
gmt dimfilter cape_verde.nc -Gt.nc -Fm220 -N18 -D2 -R-27.5/-20.5/12.5/19.5 -I2m -V
gmt grdfilter t.nc -Gcape_swell.nc -Fg50 -D2 -V
```

Suppose that you found a range of filter widths for a given area, and you filtered the given bathymetric data using the range of filter widths (e.g., `f100.nc f110.nc f120.nc f130.nc`), and you want to define a regional trend using the range of filter widths, and you want to obtain median absolute deviation (MAD) estimates at each data point. Then, you will need to do:

```
gmt grd2xyz f100.nc -Z > f100.d
gmt grd2xyz f110.nc -Z > f110.d
gmt grd2xyz f120.nc -Z > f120.d
gmt grd2xyz f130.nc -Z > f130.d
paste f100.d f110.d f120.d f130.d > depths.d
gmt dimfilter depths.d -Q4 > output.z
```

2.15.8 Limitations

When working with geographic (lat, lon) grids, all three convolution filters (boxcar, cosine arch, and gaussian) will properly normalize the filter weights for the variation in gridbox size with latitude, and correctly determine which nodes are needed for the convolution when the filter “circle” crosses a periodic (0-360) boundary or contains a geographic pole. However, the spatial filters, such as median and mode filters, do not use weights and thus should only be used on Cartesian grids (or at very low latitudes) only. If you want to apply such spatial filters you should project your data to an equal-area projection and run `dimfilter` on the resulting Cartesian grid.

2.15.9 Script Template

The `dim.template.sh` is a skeleton shell script that can be used to set up a complete DiM analysis, including the MAD analysis.

2.15.10 Reference

Kim, S.-S., and Wessel, P. (2008), Directional Median Filtering for Regional-Residual Separation of Bathymetry, *Geochem. Geophys. Geosyst.*, **9**, Q03005, doi:10.1029/2007GC001850.

2.15.11 See Also

gmt, grdfilter

2.16 gmtgravmag3d

gmtgravmag3d - Compute the gravity/magnetic effect of a body by the method of Okabe

2.16.1 Synopsis

```
gmtgravmag3d [ -Cdensity ] [ -D ] [ -Ethickness ] [ -Fxy_file ] [ -Goutputgrid.nc ]
[ -Hf_declf_dip/m_int/m_declm_dip ] [ -Lz_observation ] -V[level] [ -Sradius ] [ -T[[d]xyz_file/vert_file[/m]]|[r|s]raw_file] [ -Zlevel ] [ -V[level] ] [ -fg]
```

Note: No space is allowed between the option flag and the associated arguments.

2.16.2 Description

gmtgravmag3d will compute the gravity or magnetic anomaly of a body described by a set of triangles. The output can either be along a given set of xy locations or on a grid. This method is not particularly fast but allows computing the anomaly of arbitrarily complex shapes.

2.16.3 Required Arguments

- Cdensity* Sets body density in SI. This option is mutually exclusive with -H.
- Hf_declf_dip/m_int/m_declm_dip* Sets parameters for computing a magnetic anomaly. Use *f_declf_dip* to set the geomagnetic declination/inclination in degrees. *m_int/m_declm_dip* are the body magnetic intensity declination and inclination.
- Fxy_file* Provide locations where the anomaly will be computed. Note this option is mutually exclusive with -G.
- Goutputgrid.nc* Output the gravity or magnetic anomaly at nodes of this grid file.
- R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

2.16.4 Optional Arguments

- V[*level*] (*more ...*) Select verbosity level [*c*].
- E[*thickness*] give layer thickness in m [Default = 0 m]. Use this option only when the triangles describe a non-closed surface and you want the anomaly of a constant thickness layer.
- L[*z_observation*] sets level of observation [Default = 0]. That is the height (*z*) at which anomalies are computed.
- Sradius* search radius in km. Triangle centroids that are further away than *radius* from current output point will not be taken into account. Use this option to speed up computation at expenses of a less accurate result.

- T***[[d]xyz_file/vert_file[/m]][r|s]raw_file* Give either names of *xyz*[*m*] and *vertex* files or of a *raw* or *stl* file defining a close surface. In the first case append a *d* immediately after **-T** and optionally a */m* after the vertex file name. In the second case append a *r* or a *s* immediately after **-T** and before the file name. A *vertex* file is a file with *N* rows (one per triangle) and 3 columns with integers defining the order by which the points in the *xyz* file are to be connected to form a triangle. The output of the program *triangulate* comes in this format. The optional */m* instructs the program that the *xyzm* file has four columns and that the fourth column contains the magnetization intensity (plus signal), which needs not to be constant. In this case the third argument of the **-H** option is ignored. A *raw* format (selected by the **r** flag) is a file with *N* rows (one per triangle) and 9 columns corresponding to the *x,y,x* coordinates of each of the three vertices of each triangle. Alternatively, the **s** flag indicates that the surface file is in the ASCII STL (Stereo Lithographic) format. These two types of files are used to provide a closed surface.
- Z***[level]* level of reference plane [Default = 0]. Use this option when the triangles describe a non-closed surface and the volume is defined from each triangle and this reference level. An example will be the water depth to compute a Bouguer anomaly.
- fg** Geographic grids (dimensions of longitude, latitude) will be converted to meters via a “Flat Earth” approximation using the current ellipsoid parameters.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.16.5 Grid Distance Units

If the grid does not have meter as the horizontal unit, append **+uunit** to the input file name to convert from the specified unit to meter. If your grid is geographic, convert distances to meters by supplying **-fg** instead.

2.16.6 Examples

Suppose you ...

```
gmt gmtgravmag3d ...
```

2.16.7 See Also

gmt, *grdgravmag3d*

2.16.8 Reference

Okabe, M., Analytical expressions for gravity anomalies due to polyhedral bodies and translation into magnetic anomalies, *Geophysics*, 44, (1979), p 730-741.

2.17 gravfft

gravfft - Compute gravitational attraction of 3-D surfaces in the wavenumber (or frequency) domain

2.17.1 Synopsis

```
gravfft ingrid [ ingrid2 ] -Goutfile [ -Cn/wavelength/mean_depth/tbw ] [ -Ddensity ] [ -En_terms ]
[ -F[flg|v|nle] ] [ -Iw|b|c|t |k ] -N[flq|s|nx|ny][+ald|h |l][+e|n|m][+twidth][+w[suffix]][+z[p]] [ -Q ] [
-Te|rl|rm|rw[+m]] [ -V[level] ] [ -Zm[zl]] [ -fg ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.17.2 Description

gravfft can be used into two main modes. First one computes the gravity/geoid response of a bathymetry file. It will take the 2-D forward FFT of a bathymetry grid and compute its gravity/geoid response using full Parker's method applied to the chosen model. The available models are the "loading from top", or elastic plate model, and the "loading from below" which accounts for the plate's response to a sub-surface load (appropriate for hot spot modeling - if you believe them). In both cases, the model parameters are set with **-T** and **-Z** options. Second mode computes the admittance or coherence between two grids. The output is the average in the radial direction. Optionally, the model admittance may also be calculated. The horizontal dimensions of the grdfiles are assumed to be in meters. Geographical grids may be used by specifying the **-fg** option that scales degrees to meters. If you have grids with dimensions in km, you could change this to meters using *grdedit* or scale the output with *grdmath*. Given the number of choices this program offers, is difficult to state what are options and what are required arguments. It depends on what you are doing.

2.17.3 Required Arguments

ingrid 2-D binary grid file to be operated on. (See GRID FILE FORMATS below). For cross-spectral operations, also give the second grid file *ingrd2*.

-Goutfile Specify the name of the output grid file or the 1-D spectrum table (see **-E**). (See GRID FILE FORMATS below).

2.17.4 Optional Arguments

-Cn/wavelength/mean_depth/tbw Compute only the theoretical admittance curves of the selected model and exit. *n* and *wavelength* are used to compute ($n * wavelength$) the total profile length in meters. *mean_depth* is the mean water depth. Append dataflags (one or two) of *tbw* in any order. *t* = use "from top" model, *b* = use "from below" model. Optionally specify *w* to write wavelength instead of frequency.

-Ddensity Sets density contrast across surface. Used, for example, to compute the gravity attraction of the water layer that can later be combined with the free-air anomaly to get the Bouguer anomaly. In this case do not use **-T**. It also implicitly sets **-N+h**.

-En_terms Number of terms used in Parker expansion (limit is 10, otherwise terms depending on *n* will blow out the program) [Default = 3]

-F[flg|v|nle] Specify desired geopotential field: compute geoid rather than gravity

- f** = Free-air anomalies (mGal) [Default].
- g** = Geoid anomalies (m).
- v** = Vertical Gravity Gradient (VGG; 1 Eovtos = 0.1 mGal/km).
- e** = East deflections of the vertical (micro-radian).
- n** = North deflections of the vertical (micro-radian).

-Iw|b|c|t|k Use *ingrd2* and *ingrd1* (a grid with topography/bathymetry) to estimate admittance/coherence and write it to stdout (**-G** ignored if set). This grid should contain gravity or geoid for the same region of *ingrd1*. Default computes admittance. Output contains 3 or 4 columns. Frequency (wavelength), admittance (coherence) one sigma error bar and, optionally, a theoretical admittance. Append dataflags (one to three) from **w|b|c|t**. **w** writes wavelength instead of wavenumber, **k** selects km for wavelength unit [m], **c** computes coherence instead of admittance, **b** writes a forth column with “loading from below” theoretical admittance, and **t** writes a forth column with “elastic plate” theoretical admittance.

-N[flq|slnx|ny][+a|+d|h|l][+e|n|m][+twidth][+w[*suffix*]][+z[*p*]] Choose or inquire about suitable grid dimensions for FFT and set optional parameters. Control the FFT dimension:

- Nf** will force the FFT to use the actual dimensions of the data.
- Nq** will inQUIRE about more suitable dimensions, report those, then continue.
- Ns** will present a list of optional dimensions, then exit.
- Nnx/ny** will do FFT on array size *nx/ny* (must be \geq grid file size). Default chooses dimensions \geq data which optimize speed and accuracy of FFT. If FFT dimensions $>$ grid file dimensions, data are extended and tapered to zero.

Control detrending of data: Append modifiers for removing a linear trend:

- +d**: Detrend data, i.e. remove best-fitting linear trend [Default].
- +a**: Only remove mean value.
- +h**: Only remove mid value, i.e. $0.5 * (\max + \min)$.
- +l**: Leave data alone.

Control extension and tapering of data: Use modifiers to control how the extension and tapering are to be performed:

- +e** extends the grid by imposing edge-point symmetry [Default],
- +m** extends the grid by imposing edge mirror symmetry
- +n** turns off data extension.

Tapering is performed from the data edge to the FFT grid edge [100%]. Change this percentage via **+twidth**. When **+n** is in effect, the tapering is applied instead to the data margins as no extension is available [0%].

Control writing of temporary results: For detailed investigation you can write the intermediate grid being passed to the forward FFT; this is likely to have been detrended, extended by point-symmetry along all edges, and tapered. Append **+w[*suffix*]** from which output file name(s) will be created (i.e., *ingrid_prefix.ext*) [tapered], where *ext* is your file extension. Finally, you may save the complex grid produced by the forward FFT by appending **+z**. By default we write the real and imaginary components to *ingrid_*real* ext* and *ingrid_imag.ext*. Append **p** to save instead the polar form of magnitude and phase to files *ingrid_mag.ext* and *ingrid_phase.ext*.

- Q** Writes out a grid with the flexural topography (with z positive up) whose average was set by **-Z zm** and model parameters by **-T** (and output by **-G**). That is the “gravimetric Moho”. **-Q** implicitly sets **-N+h**
- S** Computes predicted gravity or geoid grid due to a subplate load produced by the current bathymetry and the theoretical model. The necessary parameters are set within **-T** and **-Z** options. The number of powers in Parker expansion is restricted to 1. See an example further down.
- Tte/rl/rm/rw[+m]** Compute the isostatic compensation from the topography load (input grid file) on an elastic plate of thickness te . Also append densities for load, mantle, and water in SI units. Give average mantle depth via **-Z**. If the elastic thickness is $> 1e10$ it will be interpreted as the flexural rigidity (by default it is computed from te and Young modulus). Optionally, append **+m** to write a grid with the Moho’s geopotential effect (see **-F**) from model selected by **-T**. If $te = 0$ then the Airy response is returned. **-T+m** implicitly sets **-N+h**
- Z zm [zl]** Moho [and swell] average compensation depths. For the “load from top” model you only have to provide zm , but for the “loading from below” don’t forget zl .
- V[*level*]** (*more ...*) Select verbosity level [c].
- fg** Geographic grids (dimensions of longitude, latitude) will be converted to meters via a “Flat Earth” approximation using the current ellipsoid parameters.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.17.5 Grid File Formats

By default GMT writes out grid as single precision floats in a COARDS-complaint netCDF file format. However, GMT is able to produce grid files in many other commonly used grid file formats and also facilitates so called “packing” of grids, writing out floating point data as 1- or 2-byte integers. To specify the precision, scale and offset, the user should add the suffix `=id[/scale/offset[/nan]]`, where *id* is a two-letter identifier of the grid type and precision, and *scale* and *offset* are optional scale factor and offset to be applied to all grid values, and *nan* is the value used to indicate missing data. When reading grids, the format is generally automatically recognized. If not, the same suffix can be added to input grid file names. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information.

When reading a netCDF file that contains multiple grids, GMT will read, by default, the first 2-dimensional grid that can find in that file. To coax GMT into reading another multi-dimensional variable in the grid file, append `?varname` to the file name, where *varname* is the name of the variable. Note that you may need to escape the special meaning of `?` in your shell program by putting a backslash in front of it, or by placing the filename and suffix between quotes or double quotes. The `?varname` suffix can also be used for output grids to specify a variable name different from the default: “*z*”. See [grdreformat](#) and Section *grid-file-format* of the GMT Technical Reference and Cookbook for more information, particularly on how to read splices of 3-, 4-, or 5-dimensional grids.

2.17.6 Grid Distance Units

If the grid does not have meter as the horizontal unit, append **+uunit** to the input file name to convert from the specified unit to meter. If your grid is geographic, convert distances to meters by supplying **-fg** instead.

2.17.7 Considerations

netCDF COARDS grids will automatically be recognized as geographic. For other grids geographical grids were you want to convert degrees into meters, select **-fg**. If the data are close to either pole, you should consider projecting the grid file onto a rectangular coordinate system using *grdproject*.

2.17.8 Plate Flexure

The FFT solution to elastic plate flexure requires the infill density to equal the load density. This is typically only true directly beneath the load; beyond the load the infill tends to be lower-density sediments or even water (or air). Wessel [2001] proposed an approximation that allows for the specification of an infill density different from the load density while still allowing for an FFT solution. Basically, the plate flexure is solved for using the infill density as the effective load density but the amplitudes are adjusted by a factor $A = \sqrt{((r_m - r_i)/(r_m - r_l))}$, which is the theoretical difference in amplitude due to a point load using the two different load densities. The approximation is very good but breaks down for large loads on weak plates, a fairly uncommon situation.

2.17.9 Examples

To compute the effect of the water layer above the bat.grd bathymetry using 2700 and 1035 for the densities of crust and water and writing the result on water_g.grd (computing up to the fourth power of bathymetry in Parker expansion):

```
gmt gravfft bat.grd -D1665 -Gwater_g.grd -E4
```

Now subtract it to your free-air anomaly faa.grd and you will get the Bouguer anomaly. You may wonder why we are subtracting and not adding. After all the Bouguer anomaly pretends to correct the mass deficiency presented by the water layer, so we should add because water is less dense than the rocks below. The answer relies on the way gravity effects are computed by the Parker's method and practical aspects of using the FFT.

```
gmt grdmath faa.grd water_g.grd SUB = bouguer.grd
```

Want an MBA anomaly? Well compute the crust mantle contribution and add it to the sea-bottom anomaly. Assuming a 6 km thick crust of density 2700 and a mantle with 3300 density we could repeat the command used to compute the water layer anomaly, using 600 (3300 - 2700) as the density contrast. But we now have a problem because we need to know the mean moho depth. That is when the scale/offset that can be appended to the grid's name comes in hand. Notice that we didn't need to do that before because mean water depth was computed directly from data. (notice also the negative sign of the offset due to the fact that z is positive up):

```
gmt gravfft bat.grd=nf/1/-6000 -D600 -Gmoho_g.grd
```

Now, subtract it to the sea-bottom anomaly to obtain the MBA anomaly. That is:

```
gmt grdmath water_g.grd moho_g.grd SUB = mba.grd
```

To compute the Moho gravity effect of an elastic plate `bat.grd` with $T_e = 7$ km, density of 2700, over a mantle of density 3300, at an average depth of 9 km

```
gmt gravfft bat.grd -Gelastic.grd -T7000/2700/3300/1035+m -Z9000
```

If you add now the sea-bottom and Moho's effects, you will get the full gravity response of your isostatic model. We will use here only the first term in Parker expansion.

```
gmt gravfft bat.grd -D1665 -Gwater_g.grd -E1
gmt gravfft bat.grd -Gelastic.grd -T7000/2700/3300/1035+m -Z9000 -E1
gmt grdmath water_g.grd elastic.grd ADD = model.grd
```

The same result can be obtained directly by the next command. However, PAY ATTENTION to the following. I don't yet know if it's because of a bug or due to some limitation, but the fact is that the following and the previous commands only give the same result if `-E1` is used. For higher powers of bathymetry in Parker expansion, only the above example seems to give the correct result.

```
gmt gravfft bat.grd -Gmodel.grd -T7000/2700/3300/1035 -Z9000 -E1
```

And what would be the geoid anomaly produced by a load at 50 km depth, below the a region whose bathymetry is given by `bat.grd`, a Moho at 9 km depth and the same densities as before?

```
gmt gravfft topo.grd -Gswell_geoid.grd -T7000/2700/3300/1035 -Fg -Z9000/50000 -S -E1
```

To compute the admittance between the `topo.grd` bathymetry and `faa.grd` free-air anomaly grid using the elastic plate model of a crust of 6 km mean thickness with 10 km effective elastic thickness in a region of 3 km mean water depth:

```
gmt gravfft topo.grd faa.grd -It -T10000/2700/3300/1035 -Z9000
```

To compute the admittance between the `topo.grd` bathymetry and `geoid.grd` geoid grid with the "loading from below" (LFB) model with the same as above and sub-surface load at 40 km, but assuming now the grids are in geographic and we want wavelengths instead of frequency:

```
gmt gravfft topo.grd geoid.grd -Ibw -T10000/2700/3300/1035 -Z9000/40000 -fg
```

To compute the gravity theoretical admittance of a LFB along a 2000 km long profile using the same parameters as above

```
gmt gravfft -C400/5000/3000/b -T10000/2700/3300/1035 -Z9000/40000
```

2.17.10 References

Luis, J.F. and M.C. Neves. 2006, The isostatic compensation of the Azores Plateau: a 3D admittance and coherence analysis. *J. Geothermal Vulc. Res.* Volume 156, Issues 1-2, Pages 10-22, <http://dx.doi.org/10.1016/j.jvolgeores.2006.03.010> Wessel. P., 2001, Global distribution of seamounts inferred from gridded Geosat/ERS-1 altimetry, *J. Geophys. Res.*, 106(B9), 19,431-19,441, <http://dx.doi.org/10.1029/2000JB000083>

2.17.11 See Also

gmt, grdfft, grdmath, grdproject

2.18 grdgravmag3d

grdgravmag3d - Compute the gravity effect of a grid by the method of Okabe

2.18.1 Synopsis

```
grdgravmag3d grdfile_top [grdfile_bot] [ -Cdensity ] [ -D ] [ -Fxy_file ] [ -Goutgrid ]  
[ -Iinc[unit][=|+][yinc[unit][=|+]] ] [ -Lz_obs ] [ -Q[nn_pad][pad_dist][<w/e/s/n>] ] [ -R  
[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ] [ -Zlevel ] [ -fg ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.18.2 Description

grdgravmag3d will compute the gravity anomaly of a body described by one or (optionally) two grids. The output can either be along a given set of xy locations or on a grid. This method is not particularly fast but allows computing the anomaly of arbitrarily complex shapes.

2.18.3 Required Arguments

***grdfile_top* [*grdfile_bot*]** Grid file whose gravity effect is going to be computed. If two grids are provided then the gravity/magnetic effect of the volume between them is computed.

-C*density* Sets body density in SI.

-F*xy_file* Provide locations where the anomaly will be computed. Note this option is mutually exclusive with **-G**.

-G*outgrid* Output the gravity anomaly at nodes of this grid file.

2.18.4 Optional Arguments

-I*inc[unit][=|+][yinc[unit][=|+]]* *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If **=** is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending **+** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R*grdfile*** is used then the grid spacing has already been initialized; use **-I** to override the values.

-L*z_obs* Sets level of observation [Default = 0]. That is the height (*z*) at which anomalies are computed.

-Q[*nn_pad*][*pad_dist*][<w/e/s/n>]

Extend the domain of computation with respect to output -R region. **-Qnn_pad** artificially extends the width of the outer rim of cells to have a fake width of $n_pad * dx/[dy]$.

-Qpad_dist extend the region by west-pad, east+pad, etc.

-Qregion Same syntax as **-R**.

-R[unit]xmin/xmax/ymin/ymax[r] (*more ...*) Specify the region of interest. Note: this overrides the source grid region (Default: use same region as input)

-V[level] (*more ...*) Select verbosity level [c].

-Zlevel level of reference plane [Default = 0]. Use this option when the triangles describe a non-closed surface and the volume is defined from each triangle and this reference level. An example will be the water depth to compute a Bouguer anomaly.

-fg Geographic grids (dimensions of longitude, latitude) will be converted to meters via a “Flat Earth” approximation using the current ellipsoid parameters.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.18.5 Grid Distance Units

If the grid does not have meter as the horizontal unit, append **+uunit** to the input file name to convert from the specified unit to meter. If your grid is geographic, convert distances to meters by supplying **-fg** instead.

2.18.6 Examples

Suppose you want to compute the gravity effect of the phantom “Sandy Island” together with its not phantom seamount

```
gmt grdgravmag3d sandy_bat.grd -C1700 -Z-4300 -M -I1m -Gsandy_okb.grd -V
```

2.18.7 See Also

gmt, *gmtgravmag3d*

2.18.8 Reference

Okabe, M., Analytical expressions for gravity anomalies due to polyhedral bodies and translation into magnetic anomalies, *Geophysics*, 44, (1979), p 730-741.

2.19 grdredpol

grdredpol - Compute the Continuous Reduction To the Pole, AKA differential RTP.

2.19.1 Synopsis

```
grdredpol anom_grd -Grtp_grd [-Cdec/dip] [-Edec_grd/dipgrd] [-F<m/n>] [-Mm|r] [-N] [-Wwin_width] [-V[level]] [-Tyear] [-Zfiltergrd] [-V[level]] [-n[blc|ln][+a][+bBC][+tthreshold]]
```

Note: No space is allowed between the option flag and the associated arguments.

2.19.2 Description

grdredpol will take a *.nc* file with a magnetic anomaly and compute the reduction to the pole (RTP) anomaly. This anomaly is the one that would have been produced if the bodies were magnetized vertically and the anomalies were observed at the geomagnetic pole. Standard RTP procedure assumes the direction of magnetization to be uniform throughout the causative body, and the geomagnetic field to be uniform in direction throughout the study region. Although these assumptions are reasonable for small areas, they do not hold for large areas.

In the method used here computations are carried out in both the frequency and the space domains. The idea is that a large area may be decomposed in small size windows where both the ambient field and the magnetization vector change by a very small amount. Inside each of those windows, or bins, a set of filter coefficients are calculated and reconstructed for each individual point the component filter using a first order Taylor series expansion.

2.19.3 Required Arguments

anom_grd The anomaly grid to be converted.

-Grtp_grd is the filename for output grdf file with the RTP solution

2.19.4 Optional Arguments

-Cdec/dip Use this (constant) declination and inclination angles for both field and magnetization. This option consists in the classical RTP procedure.

-Edip_grd/dec_grd Get magnetization DIP & DEC from these grids [default: use IGRF]. Note that these two grids do not need to have the same resolution as the anomaly grid. They can be coarser.

-Fm/n The filter window size in terms of row/columns. The default value is 25x25.

-Mm|r Set boundary conditions. m|r stands for mirror or replicate edges (Default is zero padding).

-N Do NOT use Taylor expansion.

-Rwest/east/south/north defines the Region of the output points. [Default: Same as input.]

-Tyear Decimal year used by the IGRF routine to compute the declination and inclination at each point [default: 2000]

-Wwidth The size of the moving window in degrees [5].

-Zfilter_grd Write the filter file to disk.

-V[level] (*more ...*) Select verbosity level [c].

-n[blc|lln][+a][+bBC][+c][+tthreshold] (*more ...*) Select interpolation mode for grids.

2.19.5 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see **-n**) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values values so they do not exceed the input min/max values by adding **+c** to your **-n** option.

2.19.6 Examples

Suppose that *anom.grd* is a file with the magnetic anomaly reduced to the 2010 epoch and that the *dec.grd* and *dip.grd* contain the magnetization declination and inclination respectively for an area that encloses that of the *anom.grd*, compute the *RTP* using bins of 2 degrees and a filter of 45 coefficients.

```
gmt grdredpol anom.grd -Grtp.grd -W2 -F45/45 -T2010 -Edec.grd/dip.grd -V
```

To compute the same *RTP* but now with the field and magnetization vectors collinear and computed from IGRF :

```
gmt grdredpol anom.grd -Grtp.grd -W2 -F45/45 -T2010 -V
```

2.19.7 Reference

Luis, J.L. and Miranda, J.M. (2008), Reevaluation of magnetic chrons in the North Atlantic between 35N and 47N: Implications for the formation of the Azores Triple Junction and associated plateau. *JGR*, VOL. **113**, B10105, doi:10.1029/2007JB005573

2.20 grdseamount

grdseamount - Compute synthetic seamount (Gaussian or cone, circular or elliptical) bathymetry

2.20.1 Synopsis

```
grdseamount [ intable ] -Ixinc[unit][=+][yinc[unit][=+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -A[out/in] ] [ -C ] [ -E ] [ -Ggrdfile ] [ -L[cut] ] [ -Nnorm ] [ -Sscale ] [ -T[flattening] ] [ -Zlevel ] [ -V[level] ] [ -bi[ncols][type][w][+L+B] ] [ -fg ] [ -icols[I][sscale][offset][,...] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.20.2 Description

grdseamount will compute the bathymetry for synthetic seamounts given their shape parameters. We read a list with seamount locations and sizes and can evaluate either Gaussian or conical shapes, which may be circular or elliptical, and optionally truncated. Various scaling options are available to modify the result, including an option to add in a background depth. The input must contain *lon*, *lat*, *radius*, *height* for each seamount. For elliptical features (**-E**) we expect *lon*, *lat*, *azimuth*, *semi-major*, *semi-minor*, *radius*, *height* instead. If flattening is specified (**-T**) then a final column with flattening is expected.

2.20.3 Required Arguments

-I*inc*[*unit*][=|+][*yinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[*unit*]*xmin/xmax/ymin/ymax*[*r*] (*more ...*) Specify the region of interest.

2.20.4 Optional Arguments

-A[*out/in*] Build a mask grid, append outside/inside values [1/NaN]. Here, height is ignored and **-L**, **-N** and **-Z** are disallowed

-C Cone model [Default is Gaussian].

-E Elliptical data format [Default is Circular]. Read *lon*, *lat*, *azimuth*, *major*, *minor*, *height* (m) for each seamount.

-G*outgrid* Sets name of output gridfile.

-L[*cut*] List area, volume, and mean height for each seamount; NO grid is created. Optionally, append the noise-floor cutoff level [0].

-N*norm* Normalize grid so maximum grid height equals *norm*.

-S*scale* Sets optional scale factor for radii [1].

-T[*flattening*] Seamounts are to be truncated. Append *flattening*, otherwise we expect it in last input column [no truncation].

-Z*level* Add in background depth [0].

-bi[*ncols*][*type*] (*more ...*) Select binary input. [Default is 3 input columns]. This option only applies to xyz input files; see **-Z** for z tables.

- fg** Geographic grids (dimensions of longitude, latitude) will be converted to km via a “Flat Earth” approximation using the current ellipsoid parameters.
- h[ilo][n][+c][+d][+rremark][+rtitle] (*more ...*)** Skip or produce header record(s). Not used with binary data.
- icol[s][sscale][offset][,...] (*more ...*)** Select input columns (0 is first column).
- V[level] (*more ...*)** Select verbosity level [c].
- r (*more ...*)** Set pixel node registration [gridline].
- :[ilo] (*more ...*)** Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.20.5 Examples

To be added.

2.20.6 See Also

gmt.conf, *gmt*, *grdmath*, *gravfft*

2.21 pssegy

pssegy - Plot a SEGY file on a map

2.21.1 Synopsis

```
pssegy SEGyfile -Jparameters -R[unit]xmin/xmax/ymin/ymax[r] -Ddeviation -F[rgb|gray] -W [ -Bbias ] [ -Cclip ] [ -Eerror ] [ -I ] [ -K ] [ -Lnsamp ] [ -Mntrace ] [ -N ] [ -O ] [ -P ] [ -Q<mode><value> ] [ -Sheader ] [ -Tfilename ] [ -U[just/dx/dy]/[c|label] ] [ -V[level] ] [ -X[alc|fr]/[x-shift][u] ] [ -Y[alc|fr]/[y-shift][u] ] [ -p[x|y|z]/[azim/elev]/[zlevel]/[+wlon0/lat0]/[z0]/[+vx0/y0] ] [ -t[transp] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.21.2 Description

pssegy reads a native (IEEE) format SEGY file and produces a PostScript image of the seismic data. The *imagemask* operator is used so that the seismic data are plotted as a 1-bit deep bitmap in a single (user-specified) color or gray shade, with a transparent background. The bitmap resolution is taken from

the current GMT defaults. The seismic traces may be plotted at their true locations using information in the trace headers (in which case order of the traces in the file is not significant). Standard GMT geometry routines are used so that in principle any map projection may be used, however it is likely that the geographic projections will lead to unexpected results. Beware also that some parameters have non-standard meanings.

Note that the order of operations before the seismic data are plotted is $\text{deviation} * [\text{clip}]([\text{bias}] + [\text{normalize}](\text{sample value}))$. Deviation determines how far *in the plot coordinates* a $[\text{normalized}][\text{biased}][\text{clipped}]$ sample value of 1 plots from the trace location.

The SEG Y file should be a disk image of the tape format (ie 3200 byte text header, which is ignored, 400 byte binary reel header, and 240 byte header for each trace) with samples as native $\text{real} * 4$ (IEEE real on all the platforms to which I have access)

2.21.3 Required Arguments

SEG Yfile Seismic data set to be imaged

-Jparameters (*more ...*) Select map projection.

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in $[\text{+}]dd:mm[:ss.xxx][\text{W|E|S|N}]$ format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-Ddeviation gives the deviation in X units of the plot for 1.0 on the scaled trace.

-F[*rgb*|*gray*] Fill trace (variable area, defaults to filling positive). *rgb* or *gray* gives the color with which the imagemask is filled.

-W Draw wiggle trace.

You *must* specify at least one of **-W** and **-F**.

2.21.4 Optional Arguments

-A Flip the default byte-swap state (default assumes data have a bigendian byte-order).

-Cclip Sample value at which to clip data (clipping is applied to both positive and negative values).

-Error Allow *error* difference between requested and actual trace locations when using **-T** option.

-I Fill negative rather than positive excursions.

-K (*more ...*) Do not finalize the PostScript plot.

-L Override number of samples per trace in reel header (program attempts to determine number of samples from each trace header if possible to allow for variable length traces).

- M** Override number of traces specified in reel header. Program detects end of file (relatively) gracefully, but this parameter limits number of traces that the program attempts to read.
- N** Normalize trace by dividing by rms amplitude over full trace length.
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- Q**<mode><value>
 - Can be used to change 4 different settings:
 - Qb**<bias> to bias scaled traces (-B-0.1 subtracts 0.1 from values)
 - Qu**<redvel> to apply reduction velocity (-ve removes reduction already present)
 - Qx**<mult> to multiply trace locations by *mult*
 - Qy**<dy> to override sample interval in reel header.
- Shheader** Read trace locations from trace headers: header is either c for CDP, o for offset, or b<num> to read a long starting at byte <num> in the header (first byte corresponds to num=0). Default has location given by trace number.
- Tfilename** Plot only traces whose location corresponds to a list given in *filename*. Order in which traces are listed is not significant - the entire space is checked for each trace.
- Uredvel** Apply reduction velocity by shifting traces *upwards* by redvel/offsetl. Negative velocity removes existing reduction. Units should be consistent with offset in trace header and sample interval.
- V[level]** (*more ...*) Select verbosity level [c].
- X[alclflr][x-shift[u]]**
- Y[alclflr][y-shift[u]]** (*more ...*) Shift plot origin.
- Z** Do not plot traces with zero rms amplitude.
- p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0]** (*more ...*) Select perspective view.
- t[transp]** (*more ...*) Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.21.5 Examples

To plot the SEG Y file `wal.segy` with normalized traces plotted at true offset locations, clipped at ± 3 and with wiggle trace and positive variable area shading in black, use

```
gmt pssegy wal.segy -JX5i/-5i -R0/100/0/10 -D1 -C3 -N -So -W -Fblack > segy.ps
```

To plot the SEGY file `wal.segy` with traces plotted at true `cdp*0.1`, clipped at `+/-3`, with bias `-1` and negative variable area shaded red, use

```
gmt pssegy wal.segy -JX5i/-5i -R0/100/0/10 -D1 -C3 -Sc -X0.1 -Fred -B-1 -I > segy.ps
```

2.21.6 See Also

gmt, *pssegyz*

2.22 pssegyz

pssegyz - Create imagedmasked postscript from SEGY file

2.22.1 Synopsis

pssegyz *SEGYfile* **-J***parameters* **-Jz***Zparameters* **-R***west/east/south/north**[/zmin/zmax]***[r]** **-D***deviation* **-F***[rgb|gray]* **-W** **[-B***bias* **]** **[-C***clip* **]** **[-I]** **[-K]** **[-L***nsamp* **]** **[-M***ntrace* **]** **[-N]** **[-O]** **[-P]** **[-Q***<mode><value>* **]** **[-S***header_x/header_y* **]** **[-U***[just/dx/dy]***[c***label* **]** **[-V***[level]* **]** **[-X****[a***lc***f***r***]****[x-***shift***[u]** **]** **[-Y****[a***lc***f***r***]****[y-***shift***[u]** **]** **[-Z]** **[-p****[x***l***y***z***]***azim/elev***[/zlevel]****[+w***lon0/lat0***[/z0]****[+v***x0/y0* **]** **[-t***[transp]* **]**

Note: No space is allowed between the option flag and the associated arguments.

2.22.2 Description

pssegyz reads a native (IEEE) format SEGY file and produces a PostScript image of the seismic data. The *imagemask* operator is used so that the seismic data are plotted as a 1-bit deep bitmap in a single (user-specified) color or gray shade, with a transparent background. The bitmap resolution is taken from the current GMT defaults. The seismic traces may be plotted at their true locations using information in the trace headers (in which case order of the traces in the file is not significant). Standard GMT geometry routines are used so that in principle any map projection may be used, however it is likely that the geographic projections will lead to unexpected results. Beware also that some parameters have non-standard meanings, and a couple of the options for **pssegy** are not available in **pssegyz**.

Note that the order of operations before the seismic data are plotted is *deviation*[clip]([bias]+[normalize](sample value))*. Deviation determines how far *in the plot coordinates* a *[normalized][biased][clipped]* sample value of 1 plots from the trace location.

The SEGY file should be a disk image of the tape format (ie 3200 byte text header, which is ignored, 400 byte binary reel header, and 240 byte header for each trace) with samples as native *real*4* (IEEE real on all the platforms to which I have access)

2.22.3 Required Arguments

SEGYfile Seismic data set to be imaged

-J*parameters* (*more ...*) Select map projection.

- R[*unit*]***west/east/south/north*[*zmin/zmax*][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.
- Ddeviation** gives the deviation in X units of the plot for 1.0 on the scaled trace, This may be a single number (applied equally in X and Y directions) or devX/devY
- F[*rgb|gray*]** Fill trace (variable area, defaults to filling positive). *rgb* or *gray* gives the color with which the imagemask is filled.
- W** Draw wiggle trace.
- You *must* specify at least one of **-W** and **-F**.

2.22.4 Optional Arguments

- A** Flip the default byte-swap state (default assumes data have a bigendian byte-order).
- Cclip** Sample value at which to clip data (clipping is applied to both positive and negative values).
- I** Fill negative rather than positive excursions.
- K** (*more ...*) Do not finalize the PostScript plot.
- Lnsamp** Override number of samples per trace in reel header (program attempts to determine number of samples from each trace header if possible to allow for variable length traces).
- Mntrace** Override number of traces specified in reel header. Program detects end of file (relatively) gracefully, but this parameter limits number of traces that the program attempts to read.
- N** Normalize trace by dividing by rms amplitude over full trace length.
- O** (*more ...*) Append to existing PostScript plot.
- P** (*more ...*) Select “Portrait” plot orientation.
- Q***<mode><value>*
- Can be used to change 4 different settings:** **-Qb***<bias>* to bias scaled traces (**-B**-0.1 subtracts 0.1 from values)
- Qu***<redvel>* to apply reduction velocity (**-ve** removes reduction already present)
- Qx***<mult>* to multiply trace locations by *mult*
- Qy***<dy>* to override sample interval in reel header.
- Shheader_x|header_y** Read trace locations from trace headers: header is either *c* for CDP, *o* for offset, *b**<num>* to read a long starting at byte *<num>* in the header (first byte corresponds to num=0), or a number to fix the location. First parameter for x, second for y. Default has X and Y given by trace number.

- Uredvel** Apply reduction velocity by shifting traces *upwards* by redvel/loffsetl. Negative velocity removes existing reduction. Units should be consistent with offset in trace header and sample interval.
- V[level]** (*more ...*) Select verbosity level [c].
- X[alcflr][x-shift[u]]**
- Y[alcflr][y-shift[u]]** (*more ...*) Shift plot origin.
- Z** Do not plot traces with zero rms amplitude.
- p[x|y|z]azim/elev[/zlevel][+wlon0/lat0[/z0]][+vx0/y0]** (*more ...*) Select perspective view.
- t[transp]** (*more ...*) Set PDF transparency level in percent.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.22.5 Examples

To plot the SEGY file wal.segy with normalized traces plotted at true offset locations, clipped at +-3 and with wiggle trace and positive variable area shading in black, use

```
pssegyz wal.segy -JX5i/-5i -D1 -Jz0.05i -E180/5 -R0/100/0/10/0/10 \  
-C3 -N -So -W -Fblack > segy.ps
```

2.22.6 Bugs

Variable area involves filling four-sided figures of distressing generality. I *know* that some of the more complex degenerate cases are not dealt with correctly or at all; the incidence of such cases increases as viewing angles become more oblique, and particularly as the viewing elevation increases. Wiggle-trace plotting is not affected.

2.22.7 See Also

gmt, *pssegy*

2.23 segy2grd

segy2grd - Converting SEGY data to a GMT grid

2.23.1 Synopsis

```
segy2grd segyfile -Ggrdfile -Ixinc[unit][=+][yinc[unit][=+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -A[nlz] ] [ -Dxname/yname/zname/scale/offset/title/remark ] [ -M[flags] ] [ -Nnodata ] [ -S[zfile] ] [ -V[level] ] [ -bi[ncols][type][w][+L+B] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.23.2 Description

segy2grd reads an IEEE SEG Y file and creates a binary grid file. Either a simple mapping (equivalent to `xyz2grd -Z`) or a more complicated averaging where a particular grid cell includes values from more than one sample in the SEG Y file can be done. **segy2grd** will report if some of the nodes are not filled in with data. Such unconstrained nodes are set to a value specified by the user [Default is NaN]. Nodes with more than one value will be set to the average value.

2.23.3 Required Arguments

segyfile is an IEEE floating point SEG Y file. Traces are all assumed to start at 0 time/depth.

-G*grdfile* *grdfile* is the name of the binary output grid file.

-I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds.

-R[*unit*]*west*/*east*/*south*/*north*[*zmin*/*zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin*/*zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

2.23.4 Optional Arguments

-A[*nlz*] Add up multiple values that belong to the same node (same as **-Az**). Append **n** to simply count the number of data points that were assigned to each node. [Default (no **-A** option) will calculate mean value]. Not used for simple mapping.

-D*xname*/*yname*/*zname*/*scale*/*offset*/*title*/*remark* Give values for *xname*, *yname*, *zname*, *scale*, *offset*, *title*, and *remark*. To leave some of these values untouched, specify = as the value.

-M[*flags*] Fix number of traces to read in. Default tries to read 10000 traces. **-M0** will read number in binary header, **-Mn** will attempt to read only *n* traces.

-N*nodata* No data. Set nodes with no input sample to this value [Default is NaN].

-S[*zfile*] set variable spacing *header* is c for cdp, o for offset, b<number> for 4-byte float starting at byte number If **-S** not set, assumes even spacing of samples at the dx, dy supplied with **-I**

- L Override number of samples in each trace
- X applies scalar *x-scale* to coordinates in trace header to match the coordinates specified in -R
- Y Specifies sample interval as *s_int* if incorrect in the SEG Y file
- V[*level*] (*more ...*) Select verbosity level [c].
- r (*more ...*) Set pixel node registration [gridline].
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version Print GMT version and exit.
- show-datadir Print full path to GMT share directory and exit.

2.23.5 Examples

To create a grid file from an even spaced SEG Y file test.segy, try

```
gmt segy2grd test.segy -I0.1/0.1 -Gtest.nc -R198/208/18/25 -V
```

Note that this will read in 18-25s (or km) on each trace, but the first trace will be assumed to be at X=198

To create a grid file from the SEG Y file test.segy, locating traces according to the CDP number, where there are 10 CDPs per km and the sample interval is 0.1, try

```
gmt segy2grd test.segy -Gtest.nc -R0/100/0/10 -I0.5/0.2 -V -X0.1 -Y0.1
```

Because the grid interval is larger than the SEG Y file sampling, the individual samples will be averaged in bins

2.23.6 See Also

gmt, grd2xyz, grdedit, pssegy

2.24 backtracker

backtracker - Generate forward and backward flowlines and hotspot tracks

2.24.1 Synopsis

```
backtracker [ table ] -Erot_filellon/lat/angle [ -A[young/old] ] [ -Dflb ] [ -Fdrift.txt ] [ -Lflbstep ] [ -Nupper_age ] [ -Qfixed_age ] [ -Sfilestem ] [ -Tzero_age ] [ -V[level] ] [ -W[alt] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icol[s][l][sscale][offset][,...] ] [ -ocol[s][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.24.2 Description

backtracker reads (longitude, latitude, age) positions from *infile*s [or standard input] and computes rotated (x,y,t) coordinates using the specified rotation parameters. It can either calculate final positions [Default] or create a sampled track (flowline or hotspot track) between the initial and final positions. The former mode allows additional data fields after the first 3 columns which must have (longitude,latitude,age). See option **-:** on how to read (latitude,longitude,age) files.

2.24.3 Required Arguments

-Erofile Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

```
lon lat tstart [tstop] angle [ khat a b c d e f g df ]
```

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)*[a b d; b c e; d e f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternative 1: Give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPlates rotation database. We return an error if the rotation cannot be found. Alternative 2: Specify *lon/lat/angle*, i.e., the longitude, latitude, and opening angle (all in degrees and separated by /) for a single total reconstruction rotation.

2.24.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-A[*young/old*] Used in conjunction with **-Lbif** to limit the track output to those sections whose predicted ages lie between the specified *young* and *old* limits. If **-LBIF** is used instead then the limits apply to the stage ids (id 1 is the youngest stage). If no limits are specified then individual limits for each record are expected in columns 4 and 5 of the input file.

-Dflb Set the direction to go: **-Df** will go backward in time (from younger to older positions), while **-Db** will go forward in time (from older to younger positions) [Default]. Note: For **-Db** you are specifying the age at the given location, whereas for **-Df** you are not; instead you specify the age at the reconstructed point.

-Fdrift.txt Supply a file with lon, lat, age records that describe the history of hotspot motion for the current hotspot. The reconstructions will only use the 3rd data input column (i.e., the age) to obtain the location of the hotspot at that time, via an interpolation of the hotspot motion history. This adjusted location is then used to reconstruct the point or path [No drift].

-Lflbstep Specify a sampled path between initial and final position: **-Lf** will draw particle flowlines, while **-Lb** will draw backtrack (hotspot track) paths. Append sampling interval in km. If *step* < 0 then only the rotation times will be returned. When **-LF** or **-LB** is used, the third output column

- will contain the stage id (1 is youngest) [Default is along-track predicted ages]. You can control the direction of the paths by using **-D**.
- Nupper_age** Set the maximum age to extend the oldest stage rotation back in time [Default is no extension].
 - Qfixed_age** Assign a fixed age to all positions. Only lon, lat input is expected [Default expects longitude, latitude, age]. Useful when the input are points defining isochrons.
 - Sfilestem** When **-L** is set, the tracks are normally written to *stdout* as a multisegment file. Specify a *filestem* to have each track written to *filestem.#*, where # is the track number. The track number is also copied to the 4th output column.
 - Tzero_age** Set the current time [Default is 0 Ma].
 - V[level] (more ...)** Select verbosity level [c].
 - W[alt]** Rotates the given input (lon,lat,t) and calculates the confidence ellipse for the projected point. The input point *must* have a time coordinate that exactly matches a particular total reconstruction rotation time, otherwise the point will be skipped. Append **t** or **a** to output time or angle, respectively, after the projected lon, lat. After these 2-3 items, we write azimuth, major, minor (in km) for the 95% confidence ellipse. See **-D** for the direction of rotation.
 - bi[ncols][type] (more ...)** Select binary input. [Default is 3 input columns].
 - bo[ncols][type] (more ...)** Select binary output. [Default is same as input].
 - h[ilo][n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
 - icol[s][l][sscale][offset][,...] (more ...)** Select input columns (0 is first column).
 - ocol[s][,...] (more ...)** Select output columns (0 is first column).
 - :[ilo] (more ...)** Swap 1st and 2nd column on input and/or output.
 - ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
 - ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
 - ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
 - version** Print GMT version and exit.
 - show-datadir** Print full path to GMT share directory and exit.

2.24.5 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.24.6 Examples

To backtrack the (x,y,t) points in the file seamounts.d to their origin (presumably the hotspot), using the DC85.d Euler poles, run

```
gmt backtracker seamounts.d -Db -EDC85.d > newpos.d
```

To project flowlines forward from the (x,y,t) points stored in several 3-column, binary, double precision files, run

```
gmt backtracker points.gmt \* -Df -EDC85.d -Lf25 -bo -bi3 > lines.b
```

This file can then be plotted with **psxy**. To compute the predicted Hawaiian hotspot track from 0 to 80 Ma every 1 Ma, given a history of hotspot motion file (HIdrift.txt) and a set of total reconstruction rotations for the plate (PAC_APM.d), try

```
echo 204 19 80 | gmt backtracker -Df -EPAC_APM.d -Lb1 > path.d
```

2.24.7 See Also

gmt, *project*, *grdrotater*, *grdspotter*, *mapproject*, *hotspotter*, *originator*

2.24.8 References

Wessel, P., 1999, "Hotspotting" tools released, EOS Trans. AGU, 80 (29), p. 319.

2.25 grdpmodeler

grdpmodeler - Evaluate a plate model on a geographic grid

2.25.1 Synopsis

```
grdpmodeler agegrdfile -Erot_file -Goutgrdfile -Saldir|w|x|y|X|Y [ -Fpolygonfile ] [ -Tage ]
[ -V[level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -
icols[I][sscale][ooffset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.25.2 Description

grdpmodeler reads a geographical age grid and a plate motion model and evaluates one of several model predictions. Optionally, the user may supply a clipping polygon in multiple-segment format; then, only the part of the grid inside the polygon is used to determine the model prediction; the remainder of the grid is set to NaN.

2.25.3 Required Arguments

ingrdfile Name of a grid file in geographical (lon, lat) coordinates with ages in Myr.

-Erotfile Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

lon lat tstart [tstop] angle [khat a b c d e f g df]

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)*[a b d; b c e; d e f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternatively, give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPlates rotation database. We return an error if the rotation cannot be found.

-Goutgrdfile Name of output grid. This is the grid with the model predictions given the specified rotations.

-Saldlr|w|lx|ly|X|Y Type of model prediction. Choose from **a** for plate motion azimuth, **d** for distance between current locations and their origin at the ridge (in km), **r** for plate motion rate (in mm/yr), **w** for plate rotation rate (degree/Myr), **x** change in longitude relative to location of crust formation, **y** change in latitude relative to location of crust formation, **X** longitude of crust formation, and **Y** latitude of crust formation.

2.25.4 Optional Arguments

-Fpolygonfile Specify a multisegment closed polygon file that describes the inside area of the grid where the model should be evaluated; the outside will be set to NaN [Default evaluates model on the entire grid].

-Tage Use a fixed age for model evaluation (i.e., override the ages in the age grid). This lets you evaluate the model at a snapshot in time.

-V[level] (more ...) Select verbosity level [c].

-bi[ncols][type] (more ...) Select binary input. [Default is 2 input columns].

-h[ilo][n][+c][+d][+rremark][+rtitle] (more ...) Skip or produce header record(s).

-icols[l][sscale][offset][,...] (more ...) Select input columns (0 is first column).

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.25.5 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.25.6 Examples

We will use a grid with Pacific crust ages (*pac_age.nc*), a plate motion model (*Pac_APM.d*), and a polygon that contains the outline of the present Pacific plate (*pac_clip_path.d*). To evaluate the plate motion azimuths at the present time for the Pacific, try

```
gmt grdpmodeler pac_age.nc -EPac_APM.d -V -Fpac_clip_path.d \
  -Gpac_dir_0.nc -Sa -T0
```

To determine the changes in latitude since crust formation for the entire Pacific, try

```
gmt grdpmodeler pac_age.nc -EPac_APM.d -V -Fpac_clip_path.d \
  -Gpac_dlat.nc -Sy
```

To determine the plate motion rates in effect when the Pacific crust was formed, try

```
gmt grdpmodeler pac_age.nc -EPac_APM.d -V -Fpac_clip_path.d \
  -Gpac_vel.nc -Sr
```

To determine how far the crust has moved since formation, try

```
gmt grdpmodeler pac_age.nc -EPac_APM.d -V -Fpac_clip_path.d \
  -Gpac_dist.nc -Sd
```

2.25.7 See Also

backtracker, *grdrotater*, *grdspotter*, *hotspotter*, *originator*, *rotconverter*

2.26 grdrotater

grdrotater - Finite rotation reconstruction of geographic grid

2.26.1 Synopsis

```
grdrotate ingrdfile -Erot file/lon/lat/angle -Goutgrdfile [ -Fpolygonfile ] [ -N ] [ -R [unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ -Tage ] [ -V [level] ] [ -b[ilo][ncol][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][...] ] [ -n[blc][ln][+a][+bBC][+tthreshold] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.26.2 Description

grodrotater reads a geographical grid and reconstructs it given a total reconstruction rotation. Optionally, the user may supply a clipping polygon in multiple-segment format; then, only the part of the grid inside the polygon is used to determine the return grid region. The outline of the projected region is returned on stdout provided the rotated region is not the entire globe.

2.26.3 Required Arguments

ingrdfile Name of a grid file in geographical (lon, lat) coordinates.

-Erofile Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

```
lon lat tstart [tstop] angle [ khat a b c d e f g df ]
```

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)^* [a b d; b c e; d e f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternative 1: Give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPlates rotation database. We return an error if the rotation cannot be found. Alternative 2: Specify *lon/lat/angle*, i.e., the longitude, latitude, and opening angle (all in degrees and separated by /) for a single total reconstruction rotation.

-Goutgrdfile Name of output grid. This is the grid with the data reconstructed according to the specified rotation.

2.26.4 Optional Arguments

-Epolygonfile Specify a multisegment closed polygon file that describes the inside area of the grid that should be projected [Default projects entire grid].

-N Do Not output the rotated polygon outline [Default will write it to stdout].

-R[unit]west/east/south/north[/zmin/zmax][r] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]-dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

-S Skip the rotation of the grid, just rotate the polygon outline (requires **-F**).

- Tage** Sets the desired age of reconstruction when **-E** is given.
- V[level] (more ...)** Select verbosity level [c].
- bi[ncols][type] (more ...)** Select binary input. [Default is 2 input columns].
- bo[ncols][type] (more ...)** Select binary output. [Default is same as input].
- h[i]o[n][+c][+d][+rremark][+rtitle] (more ...)** Skip or produce header record(s).
- :[i]o (more ...)** Swap 1st and 2nd column on input and/or output.
- i[cols][l][sscale][ooffset][,...] (more ...)** Select input columns (0 is first column).
- n[b|c|l|n][+a][+bBC][+c][+tthreshold] (more ...)** Select interpolation mode for grids.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.26.5 Consequences of grid resampling

Resample or sampling of grids will use various algorithms (see **-n**) that may lead to possible distortions or unexpected results in the resampled values. One expected effect of resampling with splines is the tendency for the new resampled values to slightly exceed the global min/max limits of the original grid. If this is unacceptable, you can impose clipping of the resampled values so they do not exceed the input min/max values by adding **+c** to your **-n** option.

2.26.6 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.26.7 Examples

To rotate the data defined by grid *topo.nc* and the polygon outline *clip_path.d*, using a total reconstruction rotation with pole at (135.5, -33.0) and a rotation angle of 37.3 degrees and bicubic interpolation, try

```
gmt grdrotater topo.nc -E135.5/-33/37.3 -V -Fclip_path.d -Grot_topo.nc > rot_clip_path.d
```

To rotate the entire grid *faa.nc* back to 32 Ma using the rotation file *rotations.txt* and a bilinear interpolation, try

```
gmt grdrotater faa.nc -Erotations.txt -T32 -V -Grot_faa.nc -nl > rot_faa_path.d
```

To just see how the outline of the grid large.nc will plot after the same rotation, try

```
gmt grdrotater large.nc -Erotations.txt -T32 -V -S \ | psxy -Rg -JH180/6i -B30 -W0.5p \ | gv -
```

Let say you have rotated gridA.nc and gridB.nc, restricting each rotation to nodes inside polygons polyA.d and polyB.d, respectively, using rotation A = (123W,22S,16,4) and rotation B = (108W, 16S, -14.5), yielding rotated grids rot_gridA.nc and rot_gridB.nc. To determine the region of overlap between the rotated grids, we use grdmath:

```
gmt grdmath 1 rot_gridA.nc ISNAN SUB 1 rot_gridB.nc ISNAN SUB 2 EQ = overlap.nc
```

The grid overlap.nc now has 1s in the regions of overlap and 0 elsewhere. You can use it as a mask or use grdcontour **-D** to extract a polygon (contour).

2.26.8 See Also

backtracker, grdspotter, hotspotter, originator, rotconverter

2.27 grdspotter

grdspotter - Create CVA image from a gravity or topography grid

2.27.1 Synopsis

```
grdspotter [grdfile] -Erotfile -GCVAGrid -Ixinc[unit][=+][yinc[unit][=+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -Aagegrid ] [ -DDIgrid ] [ -LIDgrid ] [ -M ] [ -Nupper_age ] [ -PPAGrid ] [ -QIDinfo ] [ -S ] [ -Ttufixed_val ] [ [ -V[level] ] [ -Wn_try ] ] [ -Zz_min[/z_max[/z_inc]] ] [ -r ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.27.2 Description

grdspotter reads a grid file with residual bathymetry or gravity and calculates flowlines from each node that exceeds a minimum value using the specified rotations file. These flowlines are then convolved with the volume of the prism represented by each grid node and added up to give a Cumulative Volcano Amplitude grid (CVA).

2.27.3 Required Arguments

grdfile Data grid to be processed, typically residual bathymetry or free-air anomalies.

-Erotfile Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

```
lon lat tstart [tstop] angle [ khat a b c d e f g df ]
```

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction

rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)*[a b d; b c e; d e f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternatively, give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPlates rotation database. We return an error if the rotation cannot be found.

-G Specify name for output CVA grid file.

-I*inc*[*unit*][=|+][*lyinc*[*unit*][=|+]] *x_inc* [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *ly_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][**r**] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][**W**|**E**|**S**|**N**] format. Append **r** if lower left and upper right map coordinates are given instead of *w/e/s/n*. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a *z-range* (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

2.27.4 Optional Arguments

-Aagegrid Supply a crustal age grid that is co-registered with the input data grid. These ages become the upper ages to use when constructing flowlines [Default extend flowlines back to oldest age found in the rotation file; but see **-N**].

-DDIgrid Use flowlines to determine the maximum CVA encountered along each flowline and create a Data Importance (DI) grid with these values at the originating nodes.

-LIDgrid Supply a co-registered grid with seamount chain IDs for each node. This option requires that you also use **-Q**.

-M Do not attempt to keep all flowlines in memory when using **-D** and/or **-P**. Should you run out of memory you can use this option to compute flowlines on-the-fly. It will be slower as we no longer

- can reuse the flowlines calculated for the CVA step. Cannot be used with **-W** or the multi-slice mode in **-Z**.
- Nupper_age** Set the upper age to assign to nodes whose crustal age is unknown (i.e., NaN) [no upper age]. Also see **-A**.
 - PPAgrid** Use flowlines to determine the flowline age at the CVA maximum for each node and create a Predicted Age (PA) grid with these values at the originating nodes.
 - QIDinfo** Either give (1) a single ID to use or (2) the name of a file with a list of IDs to use [Default uses all IDs]. Each line would be TAG ID [w e s n]. The *w/e/s/n* zoom box is optional; if specified it means we only trace the flowline if inside this region [Default uses region set by **-R**]. Requires **-L**.
 - S** Normalize the resulting CVA grid to percentages of the CVA maximum. This also normalizes the DI grid (if requested).
 - Ttufixed_val** Selects ways to adjust ages; repeatable. Choose from **-Tt** to truncate crustal ages given via the **-A** option that exceed the upper age set with **-N** [no truncation], or **-Tufixed_val** which means that after a node passes the test implied by **-Z**, we use this *fixed_val* instead in the calculations. [Default uses individual node values].
 - V[level] (more ...)** Select verbosity level [c].
 - Wn_try** Get *n_try* bootstrap estimates of the maximum CVA location; the longitude and latitude results are written to stdout [Default is no bootstrapping]. Cannot be used with **-M**.
 - Zz_min[/z_max[/z_inc]]** Ignore nodes with *z*-values lower than *z_min* [0] and optionally larger than *z_max* [Inf]. Give *z_min/z_max/z_inc* to make separate CVA grids for each *z*-slice [Default makes one CVA grid]. Multi-slicing cannot be used with **-M**.
 - r (more ...)** Set pixel node registration [gridline].
 - ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
 - + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
 - ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
 - version** Print GMT version and exit.
 - show-datadir** Print full path to GMT share directory and exit.

2.27.5 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.27.6 Examples

To create a CVA image from the Pacific topography grid `Pac_res_topo.nc`, using the DC85.d Euler poles, and only output a grid for the specified domain, run

```
gmt grdspotter Pac_res_topo.nc -EDC85.d -GCVA.nc -R190/220/15/25 -I2m -N145 -Tt -V
```

This file can then be plotted with *grdimage*.

2.27.7 See Also

gmt, *grdimage*, *project*, *mapproject*, *backtracker*, *hotspotter*, *originator*

2.27.8 References

Wessel, P., 1999, "Hotspotting" tools released, EOS Trans. AGU, 80 (29), p. 319.

Wessel, P., 2008, Hotspotting: Principles and properties of a plate tectonic Hough transform, *Geochem. Geophys. Geosyst.* 9(Q08004): doi:10.1029/2008GC002058.

2.28 hotspotter

hotspotter - Create CVA image from seamount locations

2.28.1 Synopsis

```
hotspotter [table] -Erotfile -GCVAGrid -Ixinc[unit][=|+][yinc[unit][=|+]] -R[unit]xmin/xmax/ymin/ymax[r] [ -Nupper_age ] [ -S ] [ -T ] [ -V[level] ] [ -bi[ncols][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[l][sscale][offset][,...] ] [ -ocols[,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.28.2 Description

hotspotter reads (longitude, latitude, amplitude, radius, age) records from *infile*s [or standard input] and calculates flowlines using the specified stage or total reconstruction rotations. These flowlines are convolved with the shape of the seamount (using a Gaussian shape given amplitude and radius = 6 sigma) and added up to give a Cumulative Volcano Amplitude grid (CVA). See option **-:** on how to read (latitude,longitude,...) files.

2.28.3 Required Arguments

table One or more ASCII (or binary, see **-bi**[ncols][type]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Erotfile Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

```
lon lat tstart [tstop] angle [ khat a b c d e f g df ]
```

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)^* [a \ b \ d; \ b \ c \ e; \ d \ e \ f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternatively, give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPLates rotation database. We return an error if the rotation cannot be found.

-GCVAggrid Specify name for output grid file.

-I $_{xinc}$ [unit][=|+][$_{yinc}$ [unit][=|+]] *x_{inc}* [and optionally *y_{inc}*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates:** Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on *PROJ_ELLIPSOID*). If *y_{inc}* is given but set to 0 it will be reset equal to *x_{inc}*; otherwise it will be converted to degrees latitude. **All coordinates:** If = is appended then the corresponding max *x* (*east*) or *y* (*north*) may be slightly adjusted to fit exactly the given increment [by default the increment may be adjusted slightly to fit the given domain]. Finally, instead of giving an increment you may specify the *number of nodes* desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see *App-file-formats* for details. Note: if **-Rgrdfile** is used then the grid spacing has already been initialized; use **-I** to override the values.

-R[unit]west/east/south/north[/zmin/zmax][r] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]-dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension.

2.28.4 Optional Arguments

-Dfactor Modify the sampling interval along flowlines. Default [0.5] gives approximately 2 points within each grid box. Smaller factors gives higher resolutions at the expense of longer processing time.

-Nupper_age Set the upper age to assign seamounts whose crustal age is unknown (i.e., NaN) [no upper age].

-S Normalize the resulting CVA grid to percentages of the CVA maximum.

-T Truncate seamount ages exceeding the upper age set with **-N** [no truncation].

- V[*level*]** (*more ...*) Select verbosity level [*c*].
- bi[*ncols*][*type*]** (*more ...*) Select binary input. [Default is 5 input columns].
- V[*level*]** (*more ...*) Select verbosity level [*c*].
- icols[*l*][*sscale*][*offset*][*,...*]** (*more ...*) Select input columns (0 is first column).
- ocols[*,...*]** (*more ...*) Select output columns (0 is first column).
- r** (*more ...*) Set pixel node registration [*gridline*].
- :[*ilo*]** (*more ...*) Swap 1st and 2nd column on input and/or output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.28.5 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.28.6 Examples

To create a CVA image from the Pacific (x,y,z,r,t) data in the file *seamounts.d*, using the DC85.d Euler poles, run

```
gmt hotspotter seamounts.d -EDC85.d -GCVA.nc -R130/260/-66/60 -I10m -N145 -T -V
```

This file can then be plotted with *grdimage*.

2.28.7 See Also

gmt, *grdimage*, *grdrotater*, *grdspotter*, *project*, *mapproject*, *backtracker*, *originator*

2.28.8 References

Wessel, P., 1999, "Hotspotting" tools released, EOS Trans. AGU, 80 (29), p. 319.

Wessel, P., 2008, Hotspotting: Principles and properties of a plate tectonic Hough transform, *Geochem. Geophys. Geosyst.* 9(Q08004): doi:10.1029/2008GC002058.

2.29 originator

originator - Associate seamounts with nearest hotspot point sources

2.29.1 Synopsis

```
originator [ table ] -E[+]rotfile -F[+]hs_file [ -Dd_km ] [ -L[flag] ] [ -Nnupper_age ] [ -Qr/t ] [ -S[n_hs] ] [ -T ] [ -V[level] ] [ -Wmaxdist ] [ -Z ] [ -bi[ncols][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ] [ -icols[I][sscale][ooffset][,...] ] [ -:[ilo] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.29.2 Description

originator reads (longitude, latitude, height, radius, crustal_age) records from *infile*s [or standard input] and uses the given Absolute Plate Motion (APM) stage or total reconstruction rotation file and the list of hotspot locations to determine the most likely origin (hotspot) for each seamount. It does so by calculating flowlines back in time and determining the closest approach to all hotspots. The output consists of the input records with four additional fields added for each of the *n_hs* closest hotspots. The four fields are the hotspot id (e.g., HWI), the stage id of the flowline segment that came closest, the pseudo-age of the seamount, and the closest distance to the hotspot (in km). See option **-:** on how to read (latitude, longitude,height, radius, crustal_age) files.

2.29.3 Required Arguments

-E*rotfile* Give file with rotation parameters. This file must contain one record for each rotation; each record must be of the following format:

```
lon lat tstart [tstop] angle [ khat a b c d e f g df ]
```

where *tstart* and *tstop* are in Myr and *lon lat angle* are in degrees. *tstart* and *tstop* are the ages of the old and young ends of a stage. If *tstop* is not present in the record then a total reconstruction rotation is expected and *tstop* is implicitly set to 0 and should not be specified for any of the records in the file. If a covariance matrix **C** for the rotation is available it must be specified in a format using the nine optional terms listed in brackets. Here, $\mathbf{C} = (g/khat)*[a b d; b c e; d e f]$ which shows **C** made up of three row vectors. If the degrees of freedom (*df*) in fitting the rotation is 0 or not given it is set to 10000. Blank lines and records whose first column contains # will be ignored. You may prepend a leading + to the filename to indicate you wish to invert the rotations. Alternatively, give the filename composed of two plate IDs separated by a hyphen (e.g., PAC-MBL) and we will instead extract that rotation from the GPlates rotation database. We return an error if the rotation cannot be found. Prepend + if you want to invert the rotations prior to use.

-F*file* Give file with hotspot locations. This file must contain one record for each hotspot to be considered; each record must be of the following format:

```
lon lat hs_abbrev hs_id r t_off t_on create fit plot name
```

E.g., for Hawaii this may look like

```
205 20 HWI 1 25 0 90 Y Y Y Hawaii
```

Most applications only need the first 4 columns which thus represents the minimal hotspot information record type. The abbreviation may be maximum 3 characters long. The id must be an

integer from 1-32. The positional uncertainty of the hotspot is given by *r* (in km). The *t_off* and *t_on* variables are used to indicate the active time-span of the hotspot. The create, fit, and plot indicators are either Y or N and are used by some programs to indicate if the hotspot is included in the ID-grids used to determine rotations, if the hotspot chain will be used to determine rotations, and if the hotspot should be included in various plots. The name is a 32-character maximum text string with the full hotspot name. Blank lines and records whose first column contains # will be ignored. Prepend + if we should look for hotspot drift tables whose name must be *hs_abbrev_drift.txt*. Such files may be located in the current directory, the same directory as *hs_file*, or in the directories pointed to by GMT_DATADIR. If found then we interpolate to get hotspot location as a function of time [fixed].

2.29.4 Optional Arguments

table One or more ASCII (or binary, see **-bi**[*ncols*][*type*]) data table file(s) holding a number of data columns. If no tables are given then we read from standard input.

-Dd_km Sets the flowline sampling interval in km. [Default is 5].

-L[flag] Output closest approach for nearest hotspot only (ignores **-S**). Choose **-Lt** for (*time, dist, z*) [Default], **-Lw** for (*omega, dist, z*), and **-Ll** for (lon, lat, time, dist, z). Normally, *dist* is in km; use upper case modifiers **TWL** to get *dist* in spherical degrees.

-Nupper_age Set the maximum age to extend the oldest stage back in time [no extension].

-Qr/t Input files only has (*x,y,z*); specify constant values for *r,t* that will be implied for each record.

-S[n_hs] Set the number of closest hotspots to report [Default is 1].

-T Truncate seamount ages exceeding the upper age set with **-N** [no truncation].

-V[level] (*more ...*) Select verbosity level [c].

-Wmaxdist Only report those seamounts whose flowlines came within *maxdist* to any hotspot [Default reports all seamounts].

-Z Use the hotspot ID number rather than the name tag in output records.

-bi[ncols][type] (*more ...*) Select binary input. [Default is 5 input columns].

-V[level] (*more ...*) Select verbosity level [c].

-icol[s][l][sscale][offset][,...] (*more ...*) Select input columns (0 is first column).

-:[ilo] (*more ...*) Swap 1st and 2nd column on input and/or output.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.29.5 Geodetic versus Geocentric Coordinates

All spherical rotations are applied to geocentric coordinates. This means that incoming data points and grids are considered to represent geodetic coordinates and must first be converted to geocentric coordinates. Rotations are then applied, and the final reconstructed points are converted back to geodetic coordinates. This default behavior can be bypassed if the ellipsoid setting *PROJ_ELLIPSOID* is changed to Sphere.

2.29.6 Examples

To find the likely (hotspot) origins of the seamounts represented by the (x,y,z,r,tc) points in the file seamounts.d, using the DC85.d Euler poles and the pac_hs.d list of possible hotspots, and report the 2 most likely hotspot candidates for each seamount, run

```
gmt originator seamounts.d -S2 -EDC85.d -Fpac_hs.d > origins.d
```

To determine the predicted age of a seamount, distances to the closest hotspot, and echo the observed age given its location, observed age, and a rotation model, try

```
echo "1.55 -8.43 52.3" | gmt originator -FONeill_2005_hotspots.txt \  
-EOMS2005_APM_fixed.txt -Q1/120 -Lt
```

where 52.3 Ma is observed age. The output is 70 -95.486 52.3. To repeat the same exercise with a moving hotspot model, try

```
echo "1.55 -8.43 52.3" | gmt originator -FONeill_2005_hotspots.txt \  
-EOMS2005_APM_smooth.txt -Q1/120 -Lt
```

Now the output is 80 -213.135 52.3. Negative distances means the closest approach was east of the hotspot.

2.29.7 See Also

gmt, gmdrotater, grdspotter, project, mapproject, backtracker, hotspotter

2.29.8 References

Wessel, P., 1999, "Hotspotting" tools released, EOS Trans. AGU, 80 (29), p. 319.

2.30 rotconverter

rotconverter - Manipulate total reconstruction and stage rotations

2.30.1 Synopsis

```
rotconverter [ +/- ] rotA [ +/- rotB ] [ +/- rotC ] ... [ -A ] [ -D ] [ -E[fact] ] [ -Fout ] [ -G ] [ -N ] [ -S ] [ -T ] [ -W ] [ -V[level] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.30.2 Description

rotconverter reads one or more plate motion models (stage or total reconstruction rotations) stored in the given files. If more than one plate motion model is given we will add or subtract them in the order they were listed. The minus sign means we should first transpose the rotation and then add it to the previous rotation. If a file cannot be opened we will attempt to decode the file name as a single rotation whose parameters are separated by slashes.

2.30.3 Required Arguments

rotX Name of a file with a plate motion model. Separate several files with desired operator (+ or -). The very first file may also have a leading minus operator to imply a transpose. We also recognize filenames of the form A-B, where both A and B are uppercase plate abbreviations as used by GPlates, to indicate we should look up the rotation between the two plates in the GPlates rotation file (e.g., PAC-MBL). If any of the specified rotation models cannot be opened as a file, we will try to decode the file name as *lon/lat/tstart[/tstop]/angle* for a single rotation given on the command line. The *tstop* argument is required for stage poles only. For a single total reconstruction rotation without any time information, give *lon/lat/angle* only.

2.30.4 Optional Arguments

- A Indicate that times are actually just opening angles [times in Myr].
- D Report longitudes use the -180/+180 range [Default is 0/360].
- E[*fact*] Scale opening angles by *fact* on output [0.5]. Typically used to get half-rates needed for flow-lines. Requires stage pole output (see -F).
- F*out* Specify the output format for rotations. The *out* flag must be either **t** or **s** for total reconstruction or stage rotations, respectively. [Default is -F**t** (output contains total reconstruction rotations)].
- G Output final rotations in the Plates4 format used by GPlates [Default is spotter format].
- N Place all output poles in the northern hemisphere [Default reports positive rotation angles].
- S Place all output poles in the southern hemisphere [Default reports positive rotation angles].
- T Transpose the final result, i.e., change the sign of the rotation angles.
- W Ensure all output rotations have negative opening angles [Default reports positive rotation angles].
- V[*level*] (*more ...*) Select verbosity level [c]. Report statistics of extracted rotations.
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- ++ or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.30.5 Limitations

Note that only one of **-N**, **-S**, and **-W** can be used at the same time.

2.30.6 Examples

To convert the total reconstruction rotations in the file `model_total_reconstruction.APM` to stage poles, run

```
gmt rotconverter model_total_reconstruction.APM -Fs > model_stages.APM
```

To obtain Nazca motion relative to Pacific hotspots by adding the motion of Nazca relative to a fixed Pacific to the Pacific-Hotspot reference model `DC85_stages.d`, and report the result as total reconstruction reconstruction poles in the northern hemisphere, try

```
gmt rotconverter DC85_stages.APM + Pac_Naz_stages.RPM -N -Ft > \
Naz_HS_total_reconstruction.APM
```

To add the final rotations `ROT(150.1, 70.5, -20.3)` and `ROT(145.0, 40.0, 11.4)`, try

```
gmt rotconverter 150.1/70.5/-20.3 + 145/40/11.4
```

which prints out `157.32, -80.44, 11.97`.

To make stage rotations suitable for generating flowlines (fracture zones) from a model of relative plate motions `PL1-PL2.RPM`, assuming symmetric spreading, try

```
gmt rotconverter PL1-PL2.RPM -E -Fs > PL1-PL2_half.RPM
gmt rotconverter - PL1-PL2.RPM -E -Fs > PL2-PL1_half.RPM
```

To compute rotations for India relative to a fixed Africa using the plate circuit India-Central Indian Basin-Antarctica-Africa, based on the `GPlates` rotations database, try

```
gmt rotconverter IND-CIB CIB-ANT ANT-AFR > India_Africa.RPM
```

2.30.7 See Also

backtracker, *grdrotater*, *grdspotter*, *hotspotter*, *originator*

2.31 x2sys_binlist

`x2sys_binlist` - Create bin index listing from track data files

2.31.1 Synopsis

```
x2sys_binlist track(s) -TTAG [ -D ] [ -E ] [ -V[level] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.31.2 Description

x2sys_binlist reads one or more track data files and produces a multisegment ASCII track bin-index file (tbf) with the track name in the header and one data record per bin crossed; these records contain *lon*, *lat*, *index*, *flags*[, *dist*], where *lon*, *lat* are the coordinates of the center of the bin, the *index* is the 1-D number of the bin, and *flags* is a bitflag that describes which data fields were available in this bin. The optional *dist* requires **-D**. The input files can be of any format, which must be described and passed with the **-T** option. The bin-index listing is a crude representation of where the track goes and is used by the data archivist to build an x2sys track data base for miscellaneous track queries, such as when needing to determine which tracks should be compared in a crossover analysis. You must run **x2sys_init** to initialize the tag before you can run the indexing.

2.31.3 Required Arguments

tracks Can be one or more ASCII, native binary, or COARDS netCDF 1-D data files. To supply the data files via a text file with a list of tracks (one per record), specify the name of the track list after a leading equal-sign (e.g., =tracks.lis). If the names are missing their file extension we will append the suffix specified for this *TAG*. Track files will be searched for first in the current directory and second in all directories listed in **\$X2SYS_HOME/TAG/TAG_paths.txt** (if it exists). [If **\$X2SYS_HOME** is not set it will default to **\$GMT_SHAREDIR/x2sys**]. (Note: MGD77 files will also be looked for via **MGD77_HOME/mgd77_paths.txt** and *.gmt files will be searched for via **\$GMT_SHAREDIR/mgg/gmtfile_paths**).

-TTAG Specify the x2sys *TAG* which tracks the attributes of this data type.

2.31.4 Optional Arguments

-D Calculate the length of trackline segments per bin [Default skips this step]. The length fragments are given as the 5th output column (after the *flags*). The length units are obtained via the *TAB* setting (see **x2sys_init**).

-E Convert geographic data to a cylindrical equal-area projection prior to binning. Basically, we apply the projection **-JYlon0/37:04:17.166076/360**, where *lon0* is the mid-longitude of the region. Requires **-D**, geographical data, and a global region (e.g., **-Rg** or **-Rd**). This option is useful for statistics related to trackline density but should not be used when preparing bin-index files for the x2sys track data bases.

-V[level] (*more ...*) Select verbosity level [c].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

++ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.31.5 Examples

To create a bin index file from the MGD77 file 01030061.mgd77 using the settings associated with the tag MGD77, do

```
gmt x2sys_binlist 01030061.mgd77 -TMGD77 > 01030061.tbf
```

To create a track bin index file of all MGD77+ files residing in the current directory using the settings associated with the tag MGD77+ and calculate track distances, run

```
gmt x2sys_binlist *.nc -TMGD77+ -D > all.tbf
```

2.31.6 See Also

x2sys_cross, *x2sys_datalist*, *x2sys_get*, *x2sys_init*, *x2sys_put*, *x2sys_report*, *x2sys_solve*

2.32 x2sys_cross

x2sys_cross - Calculate crossovers between track data files

2.32.1 Synopsis

```
x2sys_cross track(s) -TTAG [ -Acombi.lis ] [ -C[runtimes] ] [ -Ilalc ] [ -Jparameters ] [ -Qeli ] [ -Slulhspeed ] [ -V[level] ] [ -Wsize ] [ -Z ] [ -bo[ncols][type][w][+L+B] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.32.2 Description

x2sys_cross is used to determine all intersections between (“external cross-overs”) or within (“internal cross-overs”) tracks (Cartesian or geographic), and report the time, position, distance along track, heading and speed along each track segment, and the crossover error (COE) and mean values for all observables. The names of the tracks are passed on the command line. By default, **x2sys_cross** will look for both external and internal COEs. As an option, you may choose to project all data using one of the map-projections prior to calculating the COE.

2.32.3 Required Arguments

tracks Can be one or more ASCII, native binary, or COARDS netCDF 1-D data files. To supply the data files via a text file with a list of tracks (one per record), specify the name of the track list after a leading equal-sign (e.g., =tracks.lis). If the names are missing their file extension we will append the suffix specified for this *TAG*. Track files will be searched for first in the current directory and second in all directories listed in **\$X2SYS_HOME/TAG/TAG_paths.txt** (if it exists). [If **\$X2SYS_HOME** is not set it will default to **\$GMT_SHAREDIR/x2sys**]. (Note: MGD77 files will also be looked for via **MGD77_HOME/mgd77_paths.txt** and *.gmt files will be searched for via **\$GMT_SHAREDIR/mgg/gmtfile_paths**).

-TTAG Specify the *x2sys TAG* which tracks the attributes of this data type.

2.32.4 Optional Arguments

- Acombi.lis** Only process the pair-combinations found in the file *combi.lis* [Default process all possible combinations among the specified files]. The file *combi.lis* created by `x2sys_get -L` option
- C[runtimes]** Compute and append the processing run-time for each pair to the progress message. Append a filename to save these run-times to file. The idea here is to use the knowledge of run-times to split the main process in a number of sub-processes that can each be launched in a different processor of your multi-core machine. See the MATLAB function *split_file4coes.m* that lives in the `x2sys` supplement source code.
- Ilalc** Sets the interpolation mode for estimating values at the crossover. Choose among:
 - l** Linear interpolation [Default].
 - a** Akima spline interpolation.
 - c** Cubic spline interpolation.
- Jparameters (more ...)** Select map projection.
- Qeli** Append **e** for external COEs only, and **i** for internal COEs only [Default is all COEs].
- R[unit]west/east/south/north[/zmin/zmax][r]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in `[+/-]dd:mm[:ss.xxx][W|E|S|N]` format. Append **r** if lower left and upper right map coordinates are given instead of *w/e/s/n*. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option limits the COEs to those that fall inside the specified domain.
- Sllulhspeed** Defines window of track speeds. If speeds are outside this window we do not calculate a COE. Specify
 - Sl** sets lower speed [Default is 0].
 - Su** sets upper speed [Default is Infinity].
 - Sh** does not limit the speed but sets a lower speed below which headings will not be computed (i.e., set to NaN) [Default calculates headings regardless of speed].
- V[level] (more ...)** Select verbosity level [*c*].
- Wsize** Give the maximum number of data points on either side of the crossover to use in the spline interpolation [3].
- Z** Report the values of each track at the crossover [Default reports the crossover value and the mean value].
- bo[ncols][type] (more ...)** Select binary output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just `-`).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.32.5 Remarks

The COEs found are printed out to standard output in ASCII format (unless **-bo** is set). When ASCII is chosen, the output format depends on whether or not old-style XOVER output (**-L**) has been selected [See the `x_over(1)` man page for more details]. If ASCII, then the first record contains the name of the tag used, the second records specifies the exact command line used for this run, and the third record contains the names of each column. For each track pair, there will be a segment header record containing the two file names and their start/stop/dist information (start/stop is absolute time or NaN if unavailable while dist is the total track length), whereas subsequent records have the data for each COE encountered. The fields written out are x, y, time along track #1 and #2, distance along track #1 and #2, heading along track #1 and #2, velocity along track #1 and #2, and then pairs of columns for each selected observable. These are either pairs of (COE, average value) for each data type (or track-values #1 and #2; see **-Z**). It is recommended that the Akima spline is used instead of the natural cubic spline, since it is less sensitive to outliers that tend to introduce wild oscillations in the interpolation.

2.32.6 Sign Convention

If `track_a` and `track_b` are passed on the command line, then the COE value is Value (`track_a`) - Value (`track_b`).

2.32.7 Precision And Format

The output format of individual columns are controlled by `FORMAT_FLOAT_OUT` except for geographic coordinates (`FORMAT_GEO_OUT`) and absolute calendar time (`FORMAT_DATE_OUT`, `FORMAT_CLOCK_OUT`). Make sure these are set to give you enough significant digits to achieve the desired precision.

2.32.8 Examples

To compute all internal crossovers in the gmt-formatted file `c2104.gmt`, and using the tag `GMT`, try

```
gmt x2sys_cross c2104.gmt -TGMT > c2104.d
```

To find the crossover locations with bathymetry between the two MGD77 files `A13232.mgd77` and `A99938.mgd77`, using the `MGD77` tag, try

```
gmt x2sys_cross A13232.mgd77 A99938.mgd77 -Qe -TMGD77 > crossovers.d
```

2.32.9 References

Wessel, P. (2010), Tools for analyzing intersecting tracks: the x2sys package. *Computers and Geosciences*, **36**, 348-354.

Wessel, P. (1989), XOVER: A cross-over error detector for track data, *Computers and Geosciences*, **15**(3), 333-346.

2.32.10 See Also

gmt, *x2sys_binlist*, *x2sys_init*, *x2sys_datalist*, *x2sys_get*, *x2sys_list*, *x2sys_put*, *x2sys_report*, *x2sys_solve*, *x_over* (1)

2.33 x2sys_datalist

x2sys_datalist - Extract content of track data files

2.33.1 Synopsis

```
x2sys_datalist track(s) -TTAG [ -A ] [ -E ] [ -Fname1,name2,... ) ] [ -I[list] ] [ -L[corrtable] ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -S ] [ [ -V[level] ] [ -bo[ncols][type][w][+L+B] ] [ -h[ilo][n][+c][+d][+rremark][+rtitle] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.33.2 Description

x2sys_datalist reads one or more files and produces a single ASCII [or binary] table. The files can be of any format, which must be described and passed with the **-T** option. You may limit the output to a geographic region, and insist that the output from several files be separated by a multiple segment header. Only the named data fields will be output [Default selects all columns].

2.33.3 Required Arguments

tracks Can be one or more ASCII, native binary, or COARDS netCDF 1-D data files. To supply the data files via a text file with a list of tracks (one per record), specify the name of the track list after a leading equal-sign (e.g., =tracks.lis). If the names are missing their file extension we will append the suffix specified for this *TAG*. Track files will be searched for first in the current directory and second in all directories listed in **\$X2SYS_HOME/TAG/TAG_paths.txt** (if it exists). [If **\$X2SYS_HOME** is not set it will default to **\$GMT_SHAREDIR/x2sys**]. (Note: MGD77 files will also be looked for via **MGD77_HOME/mgd77_paths.txt** and *.gmt files will be searched for via **\$GMT_SHAREDIR/mgg/gmtfile_paths**).

-TTAG Specify the x2sys *TAG* which tracks the attributes of this data type.

2.33.4 Optional Arguments

- A Eliminate COEs by distributing the COE between the two tracks in proportion to track weight. These (dist, adjustment) spline knots files for each track and data column are called *track.column.adj* and are expected to be in the $\$X2SYS_HOME/TAG$ directory. The adjustments are only applied if the corresponding adjust file can be found [No residual adjustments]
- E Enhance ASCII output by writing GMT segment headers between data from each track [no segment headers].
- F*name1,name2,...*) Give a comma-separated sub-set list of column names defined in the definition file. [Default selects all data columns].
- I[*list*] Name of ASCII file with a list of track names (one per record) that should be excluded from consideration [Default includes all tracks].
- L[*corrtable*] Apply optimal corrections to columns where such corrections are available. Append the correction table to use [Default uses the correction table *TAG_corrections.txt* which is expected to reside in the $\$X2SYS_HOME/TAG$ directory]. For the format of this file, see CORRECTIONS below.
- R[*unit*]*west/east/south/north*[*zmin/zmax*][*r*] *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append *r* if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option limits the COEs to those that fall inside the specified domain.
- S Suppress output records where all the data columns are NaN [Default will output all records].
- V[*level*] (*more ...*) Select verbosity level [*c*].
- bo[*ncols*][*type*] (*more ...*) Select binary output.
- h[ilo][*n*][+*c*][+*d*][+*rremark*][+*rtitle*] (*more ...*) Skip or produce header record(s).
- ^ or **just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use **just -**).
- + or **just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or **no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.33.5 Examples

To extract all data from the old-style MGG supplement file `c2104.gmt`, recognized by the tag `GMT`:

```
gmt x2sys_datalist c2104.gmt -TGMT > myfile
```

To make `lon, lat,` and `depth` input for **blockmean** and **surface** using all the files listed in the file `tracks.lis` and define by the tag `TRK`, but only the data that are inside the specified area, and make output binary, run

```
gmt x2sys_datalist =tracks.lis -TTRK -Fon,lat,depth -R40/-30/25/35 -bo > alltopo_bin.xyz
```

2.33.6 Corrections

The correction table is an ASCII file with coefficients and parameters needed to carry out corrections. This table is usually produced by **x2sys_solve**. Comment records beginning with `#` are allowed. All correction records are of the form

trackID observation correction

where *trackID* is the track name, *observation* is one of the abbreviations for an observed field contained in files under this TAG, and *correction* consists of one or more white-space-separated *terms* that will be **subtracted** from the observation before output. Each *term* must have this exact syntax:

factor[**function*](*scale*)(*abbrev*[*-origin*])^{*power*}]

where terms in brackets are optional (the brackets themselves are not used but regular parentheses must be used exactly as indicated). No spaces are allowed except between *terms*. The *factor* is the amplitude of the basis function, while the optional *function* can be one of `sin`, `cos`, or `exp`. The optional *scale* and *origin* can be used to translate the argument (before giving it to the optional function). The argument *abbrev* is one of the abbreviations for columns known to this TAG. However, it can also be one of the three auxiliary terms **dist** (for along-track distances), **azim** for along-track azimuths, and **vel** (for along-track speed); these are all sensitive to the `-C` and `-N` settings used when defining the TAB; furthermore, **vel** requires **time** to be present in the data. If *origin* is given as **T** it means that we should replace it with the value of *abbrev* for the very first record in the file (this is usually only done for *time*). If the first data record entry is NaN we revert *origin* to zero. Optionally, raise the entire expression to the given *power*, before multiplying by *factor*. The following is an example of fictitious corrections to the track `ABC`, implying the `z` column should have a linear trend removed, the field **obs** should be corrected by a strange dependency on latitude, **weight** needs to have 1 added (hence correction is given as `-1`), and **fuel** should be reduced by a linear distance term:

```
ABC z 7.1 1e-4*((time-T))
```

```
ABC obs 0.5*exp(-1e-3(lat))^1.5
```

```
ABC weight -1
```

```
ABC fuel 0.02*((dist))
```

2.33.7 See Also

blockmean, *gmt*, *surface*, *x2sys_init*, *x2sys_datalist*, *x2sys_get*, *x2sys_list*, *x2sys_put*, *x2sys_report*, *x2sys_solve*

2.34 x2sys_get

x2sys_get - Get track listing from the x2sys track index databases

2.34.1 Synopsis

```
x2sys_get -TTAG [ -C ] [ -Fflags ] [ -G ] [ -L+[list] ] [ -Nflags ] [ [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -V[level] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.34.2 Description

x2sys_get will return the names of the track data files in the x2sys data base for this TAG that match the given requirements. You may choose a specific region and optionally ask only for tracks that meet certain data criteria. Finally, you may select an option to list all possible pairs that might generate crossovers.

2.34.3 Required Arguments

-TTAG Specify the x2sys TAG which tracks the attributes of this data type.

2.34.4 Optional Arguments

- C** Instead of reporting the track names, just output the coordinates of the center of each bin that has at least one track with the specified data.
- D** Only report the track names [Default adds the availability of data for each field].
- Fflags** Give a comma-separated list of column names (as described in the definition file) that should be present. [Default selects all data columns].
- G** Report data flags (Y or N) for the entire track rather than just for the portion that is inside the region set by **-R** [Default].
- L+[list]** Crossover mode. Return a list of track pairs that should be checked for possible crossovers. The list is determined from the bin-index data base on the assumption that tracks occupying the same bin are very likely to intersect. By default we return all possible pairs in the data base. Append the name of a file with a list of tracks if you want to limit the output to those pairs that involve at least one of the track names in your list. The output is suitable for the **-A** option in **x2sys_cross**. By default, only external crossover pairs are listed. Use **-L+** to include internal pairs in the list.
- Nflags** Give a comma-separated list of column names (as described in the definition file) that must be absent.
- R[unit]west/east/south/north[/zmin/zmax][r]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual

rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin, zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option limits the tracks to those that fall at least partly inside the specified domain.

- V[level] (more ...)** Select verbosity level [c].
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.34.5 Examples

To find all the tracks associated with the tag MGD77, restricted to occupy a certain region in the south Pacific, and have at least free air anomalies and bathymetry, try

```
gmt x2sys_get -V -TMGD77 -R180/240/-60/-30 -Ffaa,depth
```

To find all the tracks associated with the tag MGD77 that have depth but not twt, try

```
gmt x2sys_get -V -TMGD77 -Fdepth -Nwt
```

To find all the pairs associated with the tag MGD77 that might intersect each other, but only those pairs which involves tracks in your list new.lis, try

```
gmt x2sys_get -V -TMGD77 -Lnew.lis > xpairs.lis
```

2.34.6 Note

The tracks that are returned all have the requested data (**-F**) within the specified region (**-R**). Furthermore, the columns of Y and N for other data types also reflect the content of the track portion within the selected region, unless **-G** is set.

2.34.7 See Also

x2sys_binlist, x2sys_cross, x2sys_datalist, x2sys_init, x2sys_list, x2sys_put, x2sys_report, x2sys_solve

2.35 x2sys_init

x2sys_init - Initialize a new x2sys track database

2.35.1 Synopsis

x2sys_init TAG **-D**defile [**-C**clflg] [**-E**suffix] [**-F**] [**-G**dlg] [**-I**dx[/dy]] [**-N**dlsunit] [**-R**[unit]xmin/xmax/ymin/ymax[r]] [**-V**[level]] [**-W**tdgap]

Note: No space is allowed between the option flag and the associated arguments.

2.35.2 Description

x2sys_init is the starting point for anyone wishing to use **x2sys**; it initializes a set of data bases that are particular to one kind of track data. These data, their associated data bases, and key parameters are given a short-hand notation called an **x2sys** TAG. The TAG keeps track of settings such as file format, whether the data are geographic or not, and the binning resolution for track indices. Running **x2sys_init** is a prerequisite to running any of the other **x2sys** programs, such as **x2sys_binlist**, which will create a crude representation of where each data track go within the domain and which observations are available; this information serves as input to **x2sys_put** which updates the track data base. Then, **x2sys_get** can be used to find which tracks and data are available inside a given region. With that list of tracks you can use **x2sys_cross** to calculate track crossovers, use **x2sys_report** to report crossover statistics or **x2sys_list** to pull out selected crossover information that **x2sys_solve** can use to determine track-specific systematic corrections. These corrections may be used with **x2sys_datalist** to extract corrected data values for use in subsequent work. Because you can run **x2sys_init** you *must* set the environmental parameter **X2SYS_HOME** to a directory where you have write permission, which is where **x2sys** can keep track of your settings.

2.35.3 Required Arguments

TAG The unique name of this data type **x2sys** TAG.

-Ddefile Definition file prefix for this data set [See DEFINITION FILES below for more information]. Specify full path if the file is not in the current directory.

2.35.4 Optional Arguments

-Cclflg Select procedure for along-track distance calculation when needed by other programs:

c Cartesian distances [Default, unless **-G** is set].

f Flat Earth distances.

g Great circle distances [Default if **-G** is set].

e Geodesic distances on current GMT ellipsoid.

-Esuffix Specifies the file extension (suffix) for these data files. If not given we use the definition file prefix as the suffix (see **-D**).

-F Force creating new files if old ones are present [Default will abort if old TAG files are found].

-Gdlg Selects geographical coordinates. Append **d** for discontinuity at the Dateline (makes longitude go from -180 to + 180) or **g** for discontinuity at Greenwich (makes longitude go from 0 to 360 [Default]). If not given we assume the data are Cartesian.

- Idx[*dy*]** *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds for geographic data. These spacings refer to the binning used in the track bin-index data base.
- Ndlsunit** Sets the units used for distance and speed when requested by other programs. Append **d** for distance or **s** for speed, then give the desired *unit* as **c** (Cartesian userdist or userdist/usertime), **e** (meters or m/s), **f** (feet or feet/s), **k** (km or kms/hr), **m** (miles or miles/hr), **n** (nautical miles or knots) or **u** (survey feet or survey feet/s). [Default is **-Ndk -Nse** (km and m/s) if **-G** is set and **-Ndc** and **-Nsc** otherwise (Cartesian units)].
- R[*unit*]*west/east/south/north*[*zmin/zmax*][**r**]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+]*dd*:*mm*:*ss*.*xxx*[**W**|**E**|**S**|**N**] format. Append **r** if lower left and upper right map coordinates are given instead of *w/e/s/n*. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append *zmin/zmax*. In case of perspective view (**-p**), a *z*-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option bases the statistics on those COE that fall inside the specified domain.
- V[*level*]** (*more ...*) Select verbosity level [*c*].
- Wtldgap** Give **t** or **d** and append the corresponding maximum time gap (in user units; this is typically seconds [Infinity]), or distance (for units, see **-N**) gap [Infinity] allowed between the two data points immediately on either side of a crossover. If these limits are exceeded then a data gap is assumed and no COE will be determined.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- ++ or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.35.5 Definition Files

These *.def files contain information about the data file format and have two sections: (1) header information and (2) column information. All header information starts with the character # in the first column, immediately followed by an upper-case directive. If the directive takes an argument it is separated by white-space. You may append a trailing # comments. Five directives are recognized:

ASCII states that the data files are in ASCII format.

BINARY states that the data files are native binary files.

NETCDF states that the data files are COARDS-compliant 1-D netCDF files.

SKIP takes an integer argument which is either the number of lines to skip (when reading ASCII files) or the number of bytes to skip (when reading native binary files). Not used with netCDF files.

GEO indicates that these files are geographic data sets, with periodicities in the x -coordinate (longitudes). Alternatively, use **-G**.

MULTISEG means each track consists of multiple segments separated by a GMT segment header (alternatively, use **-m** when defining the system TAG). Not used with netCDF files.

The column information consists of one line per column in the order the columns appear in the data file. For each column you must provide seven attributes:

name type NaN NaN-proxy scale offset offormat

name is the name of the column variable. It is expected that you will use the special names *lon* (or x if Cartesian) and *lat* (or y) for the two required coordinate columns, and *time* when optional time data are present.

type is always **a** for ASCII representations of numbers, whereas for binary files you may choose among **c** for signed 1-byte character (-127,+128), **u** for unsigned byte (0-255), **h** for signed 2-byte integers (-32768,+32767), **i** for signed 4-byte integers (-2,147,483,648,+2,147,483,647), **f** for 4-byte floating points and **d** for 8-byte double precision floating points. For netCDF, simply use **d** as netCDF will automatically handle type-conversions during reading.

NaN is Y if certain values (e.g. -9999) are to be replaced by NAN, and N otherwise.

NaN-proxy is that special value (e.g., -9999).

scale is used to multiply the data after reading.

offset is used to add to the scaled data.

offormat is a C-style format string used to print values from this column.

If you give **-** as the *offormat* then GMT's formatting machinery will be used instead (i.e., *FORMAT_FLOAT_OUT*, *FORMAT_GEO_MAP*, *FORMAT_DATE_MAP*, *FORMAT_CLOCK_MAP*). Some file formats already have definition files premade. These include *mgd77* (for plain ASCII MGD77 data files), *mgd77+* (for enhanced MGD77+ netCDF files), *gmt* (for old *mgg* supplement binary files), *xy* (for plain ASCII x , y tables), *xyz* (same, with one z -column), *geo* (for plain ASCII longitude, latitude files), and *geoz* (same, with one z -column).

2.35.6 Examples

If you have a large set of track data files you can organize them using the *x2sys* tools. Here we will outline the steps. Let us assume that your track data file format consist of 2 header records with text information followed by any number of identically formatted data records with 6 columns (*lat*, *lon*, *time*, *obs1*, *obs2*, *obs3*) and that files are called *.*trk*. We will call this the "line" format. First, we create the *line.def* file:

| # Define file for the line format | | | | | | |
|-----------------------------------|------|-----|-------------------------|-------|--------|---------|
| # SKIP 2 | | | # Skip 2 header records | | | |
| # GEO | | | # Data are geographic | | | |
| #name | type | NaN | NaN-proxy | scale | offset | oformat |
| lat | a | N | 0 | 1 | 0 | %9.5f |
| lon | a | N | 0 | 1 | 0 | %10.5f |
| time | a | N | 0 | 1 | 0 | %7.1f |
| obs1 | a | N | 0 | 1 | 0 | %7.2f |
| obs2 | a | N | 0 | 1 | 0 | %7.2f |
| obs3 | a | N | 0 | 1 | 0 | %7.2f |

Next we create the TAG and the TAG directory with the databases for these line track files. Assuming these contain geographic data and that we want to keep track of the data distribution at a 1 x 1 degree resolution, with distances in km calculated along geodesics and with speeds given in knots, we may run

```
gmt x2sys_init LINE -V -G -Dline -Rg -Ce -Ndk -Nsn -I1/1 -Etrk
```

where we have selected LINE to be our x2sys tag. When x2sys tools try to read your line data files they will first look in the current directory and second look in the file TAG_paths.txt for a list of additional directories to examine. Therefore, create such a file (here LINE_paths.txt) and stick the full paths to your data directories there. All TAG-related files (definition files, tag files, and track data bases created) will be expected to be in the directory pointed to by \$X2SYS_HOME/TAG (in our case \$X2SYS_HOME/LINE). Note that the argument to **-D** must contain the full path if the *.def file is not in the current directory. **x2sys_init** will copy this file to the \$X2SYS_HOME/TAG directory where all other x2sys tools will expect to find it.

Create tbf file(s): Once the (empty) TAG databases have been initialized we go through a two-step process to populate them. First we run **x2sys_binlist** on all our track files to create one (or more) multisegment track bin-index files (tbf). These contain information on which 1 x 1 degree bins (or any other blocksize; see **-I**) each track has visited and which observations (in your case obs1, obs2, obs3) were actually observed (not all tracks may have all three kinds of observations everywhere). For instance, if your tracks are listed in the file tracks.lis we may run this command:

```
gmt x2sys_binlist -V -TLINE :tracks.lis > tracks.tbf
```

Update index data base: Next, the track bin-index files are fed to **x2sys_put** which will insert the information into the TAG databases:

```
gmt x2sys_put -V -TLINE tracks.tbf
```

Search for data: You may now use **x2sys_get** to find all the tracks within a certain sub-region, and optionally limit the search to those tracks that have a particular combination of observables. E.g., to find all the tracks which has both obs1 and obs3 inside the specified region, run

```
gmt x2sys_get -V -TLINE -R20/40/-40/-20 -Fobs1,obs3 > tracks.tbf
```

MGD77[+] or GMT: Definition files already exist for MGD77 files (both standard ASCII and enhanced netCDF-based MGD77+ files) and the old *.gmt files manipulated by the mgg supplements; for these data sets the **-C** and **-N** will default to great circle distance calculation in km and speed in m/s. There are also definition files for plain x,y[,z] and lon,lat[,z] tracks. To initiate new track databases to be used with MGD77 data from NGDC, try

```
gmt x2sys_init MGD77 -V -Dmgd77 -Emgd77 -Rd -Gd -Nsn -I1/1 -Wt900 -Wd5
```

where we have chosen a 15 minute (900 sec) or 5 km threshold to indicate a data gap and selected knots as the speed; the other steps are similar.

Binary files: Let us pretend that your line files actually are binary files with a 128-byte header structure (to be skipped) followed by the data records and where *lon*, *lat*, *time* are double precision numbers while the three observations are 2-byte integers which must be multiplied by 0.1. Finally, the first two observations may be -32768 which means there is no data available. All that is needed is a different line.def file:

| # Define file for the binary line format | | | | | | |
|--|------|-----|-----------------------|-------|--------|----------|
| # BINARY | | | # File is now binary | | | |
| # SKIP 128 | | | # Skip 128 bytes | | | |
| # GEO | | | # Data are geographic | | | |
| #name | type | NaN | NaN-proxy | scale | offset | offormat |
| lon | d | N | 0 | 1 | 0 | %10.5f |
| lat | d | N | 0 | 1 | 0 | %9.5f |
| time | d | N | 0 | 1 | 0 | %7.1f |
| obs1 | h | Y | -32768 | 0.1 | 0 | %6.1f |
| obs2 | h | Y | -32768 | 0.1 | 0 | %6.1f |
| obs3 | h | N | 0 | 0.1 | 0 | %6.1f |

The rest of the steps are identical.

COARDS 1-D netCDF files: Finally, suppose that your line files actually are netCDF files that conform to the COARDS convention, with data columns named *lon*, *lat*, *time*, *obs1*, *obs2*, and *obs3*. All that is needed is a different line.def file:

| # Define file for the netCDF COARDS line format | | | | | | |
|---|------|-----|-----------------------|-------|--------|----------|
| # NETCDF | | | # File is now netCDF | | | |
| # GEO | | | # Data are geographic | | | |
| #name | type | NaN | NaN-proxy | scale | offset | offormat |
| lon | d | N | 0 | 1 | 0 | %10.5f |
| lat | d | N | 0 | 1 | 0 | %9.5f |
| time | d | N | 0 | 1 | 0 | %7.1f |
| obs1 | d | N | 0 | 1 | 0 | %6.1f |
| obs2 | d | N | 0 | 1 | 0 | %6.1f |
| obs3 | d | N | 0 | 1 | 0 | %6.1f |

Note we use no scaling or NAN proxies since those issues are usually handled internally in the netCDF format description.

2.35.7 See Also

x2sys_binlist, *x2sys_datalist*, *x2sys_get*, *x2sys_list*, *x2sys_put*, *x2sys_report*, *x2sys_solve*

2.36 x2sys_list

x2sys_list - Extract subset from crossover data base

2.36.1 Synopsis

```
x2sys_list -Ccolumn -TTAG [ coedbase.txt ] [ -Aasymm_max ] [ -E ] [ -FacdhiInNtTvwxxyz ] [ -I[list] ] [ -L[corrtable] ] [ -Nnx_min ] [ -Qeli ] [ -R[unit]xmin/xmax/ymin/ymax[r] ] [ -Strack ] [ -V[level] ] [ -W[list] ] [ -bo[ncols][type][w][+L+B] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.36.2 Description

x2sys_list will read the crossover ASCII data base *coedbase.txt* (or *stdin*) and extract a subset of the crossovers based on the other arguments. The output may be ASCII or binary.

2.36.3 Required Arguments

- Ccolumn** Specify which data column you want to process. Crossovers related to this column name must be present in the crossover data base.
- TTAG** Specify the x2sys *TAG* which tracks the attributes of this data type.

2.36.4 Optional Arguments

coedbase.txt The name of the input ASCII crossover error data base as produced by **x2sys_cross**. If not given we read standard input instead.

- Aasymm_max** Specifies maximum asymmetry in the distribution of crossovers relative to the mid point in time (or distance, if not time is available). Asymmetry is computed as $(n_right - n_left)/(n_right + n_left)$, referring to the number of crossovers that falls in the left or right half of the range. Symmetric distributions will have values close to zero. If specified, we exclude tracks whose asymmetry exceeds the specify cutoff in absolute value [1, i.e., include all].
- E** Enhance ASCII output by writing GMT segment headers with name of the two tracks and their total number of cross-overs [no segment headers].
- FacdhiInNtTvwxzy** Specify your desired output using any combination of *acdhiInNtTvwxzy*, in any order. Do not use space between the letters, and note your selection is case-sensitive. The output will be ASCII (or binary, **-bo**) columns of values. Description of codes: **a** is the angle (< 90) defined by the crossing tracks, **c** is crossover value of chosen observation (see **-C**), **d** is distance along track, **h** is heading along track, **i** is the signed time interval between the visit at the crossover of the two tracks involved, **I** is same as **i** but is unsigned, **n** is the names of the two tracks, **N** is the id numbers of the two tracks, **t** is time along track in *dateTclock* format (NaN if not available), **T** is elapsed time since start of track along track (NaN if not available), **v** is speed along track, **w** is the composite weight, **x** is *x*-coordinate (or longitude), **y** is *y*-coordinate (or latitude), and **z** is observed value (see **-C**) along track. If **-S** is not specified then **d,h,n,N,t,T,v** results in two output columns each: first for track one and next for track two (in lexical order of track names); otherwise, they refer to the specified track only (except for **n,N** which then refers to the other track). The sign convention for **c,i** is track one minus track two (lexically sorted). Time intervals will be returned according to the **TIME_UNIT** GMT defaults setting.
- I[list]** Name of ASCII file with a list of track names (one per record) that should be excluded from consideration [Default includes all tracks].
- L[corrtable]** Apply optimal corrections to the chosen observable. Append the correction table to use [Default uses the correction table *TAG_corrections.txt* which is expected to reside in the **\$X2SYS_HOME/TAG** directory]. For the format of this file, see **x2sys_solve**.
- Nnx_min** Only report data from pairs that generated at least *nx_min* crossovers between them [use all pairs].

- Qeli** Append **e** for external crossovers or **i** for internal crossovers only [Default is all crossovers].
- R[*unit*]*west/east/south/north*[/*zmin/zmax*][**r**]** *west*, *east*, *south*, and *north* specify the region of interest, and you may specify them in decimal degrees or in [+*-*]dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of *w/e/s/n*. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin*, *zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option bases the statistics on those COE that fall inside the specified domain.
- Strack** Name of a single track. If given we restrict output to those crossovers involving this track [Default output is crossovers involving any track pair].
- V[*level*] (*more ...*)** Select verbosity level [*c*].
- W[*list*]** Name of ASCII file with a list of track names and their relative weights (one track per record) that should be used to calculate the composite crossover weight (output code **w** above). [Default sets weights to 1].
- bo[*ncols*][*type*] (*more ...*)** Select binary output.
- ^ or just -** Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).
- + or just +** Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.
- ? or no arguments** Print a complete usage (help) message, including the explanation of options, then exits.
- version** Print GMT version and exit.
- show-datadir** Print full path to GMT share directory and exit.

2.36.5 Examples

To find all the magnetic crossovers associated with the tag MGD77 from the file COE_data.txt, restricted to occupy a certain region in the south Pacific, and return location, time, and crossover value, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -R180/240/-60/-30 -Cmag -Fxytz > mag_coe.txt
```

To find all the faa crossovers globally that involves track 12345678 and output time since start of the year, using a binary double precision format, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -Cfaa -S12345678 -FTz -bod > faa_coe.b
```

2.36.6 See Also

x2sys_binlist, *x2sys_cross*, *x2sys_datalist*, *x2sys_get*, *x2sys_init*, *x2sys_put*, *x2sys_report*, *x2sys_solve*

2.37 x2sys_merge

x2sys_merge - Merge an updated COEs table (smaller) into the main table (bigger)

2.37.1 Synopsis

x2sys_merge -*Amain_COElist.d* -*Mnew_COElist.d*

Note: No space is allowed between the option flag and the associated arguments.

2.37.2 Description

x2sys_merge will read two crossovers data base and output the contents of the main one updated with the COEs in the second one. The second file should only contain updated COEs relatively to the first one. That is, it MUST NOT contain any new two tracks intersections (This point is NOT checked in the code). This program is useful when, for any good reason like file editing NAV correction or whatever, one had to recompute only the COEs between the edited files and the rest of the database.

2.37.3 Required Arguments

-*Amain_COElist.d* Specify the file *main_COElist.d* with the main crossover error data base.

-*Mnew_COElist.d* Specify the file *new_COElist.d* with the newly computed crossover error data base.

2.37.4 Optional Arguments

2.37.5 Examples

To update the main COE_data.txt with the new COEs estimations saved in the smaller COE_fresh.txt, try

```
gmt x2sys_merge -ACOE_data.txt -MCOE_fresh.txt > COE_updated.txt
```

2.37.6 See Also

x2sys_binlist, *x2sys_cross*, *x2sys_datalist*, *x2sys_get*, *x2sys_init*, *x2sys_list*, *x2sys_put*, *x2sys_report*

2.38 x2sys_put

x2sys_put - Update track index database from track bin file

2.38.1 Synopsis

x2sys_put [*info.tbf*] -**TTAG** [-**D**] [-**F**] [-**V**[*level*]]

Note: No space is allowed between the option flag and the associated arguments.

2.38.2 Description

x2sys_put accepts a track bin-index file created by **x2sys_binlist** and adds this information about the data tracks to the relevant data base. You may chose to overwrite existing data with new information for older tracks (**-F**) and even completely remove information for certain tracks (**-D**). The x2sys *TAG* must match the tag encoded in the *info.tbf* file. To inquire about tracks in the data base, use **x2sys_get**.

2.38.3 Required Arguments

info.tbf Name of a single track bin file. If not given, *stdin* will be read.

-TTAG Specify the x2sys *TAG* which tracks the attributes of this data type.

2.38.4 Optional Arguments

-D Delete all tracks found in the track bin file [Default will try to add them as new track entries].

-F Replace any existing database information for these tracks with the new information in the track bin file [Default refuses to process tracks already in the database].

-V[level] (*more ...*) Select verbosity level [c].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.38.5 Examples

To add the information stored in the track bin-index file *latest.tbf* to the track data bases associated with the tag *MGD77*, and replace any exiting information for these tracks, try

```
gmt x2sys_put latest.tbf -F -V -TMGD77
```

2.38.6 X2sys Databases

The **x2sys_put** utility adds new information to the x2sys data bases. These consists of two files: The first file contains a listing of all the tracks that have been added to the system; it is named *TAG_tracks.d* and is in ASCII format. The second file is named *TAG_index.b* and is in native binary format. It contains information on which tracks cross each of the bins, and what data sets were observed while crossing the bin. The bins are defined by the **-R** and **-I** options passed to **x2sys_init** when the *TAG* was first initiated. Both data base files are stored in the **\$X2SYS_HOME/TAG** directory. Do not attempt to edit these files by hand.

2.38.7 See Also

x2sys_binlist, *x2sys_get*

2.39 x2sys_report

x2sys_report - Report statistics from crossover data base

2.39.1 Synopsis

x2sys_report **-Ccolumn** **-TTAG** [*coedbase.txt*] [**-A**] [**-I**[*list*]] [**-L**[*corrtable*]] [**-Nnx_min**] [**-Qeli**]
[[**-R**[*unit*]*xmin/xmax/ymin/ymax*[*r*]] [**-Strack**] [**-V**[*level*]]

Note: No space is allowed between the option flag and the associated arguments.

2.39.2 Description

x2sys_report will read the input crossover ASCII data base *coedbase.txt* (or *stdin*) and report on the statistics of crossovers (*n*, *mean*, *stdev*, *rms*, *weight*) for each track. Options are available to let you exclude tracks and limit the output.

2.39.3 Required Arguments

coedbase.txt The name of the input ASCII crossover error data base as produced by **x2sys_cross**. If not given we read standard input instead.

-Ccolumn Specify which data column you want to process. Crossovers related to this column name must be present in the crossover data base.

-TTAG Specify the *x2sys TAG* which tracks the attributes of this data type.

2.39.4 Optional Arguments

-A Eliminate COEs by distributing the COE between the two tracks in proportion to track weight and producing (dist, adjustment) spline knots files for each track (for the selected *column*). Such adjustments may be used by **x2sys_datalist**. The adjustment files are called *track.column.adj* and are placed in the **\$X2SYS_HOME/TAG** directory. For background information on how these adjustments are designed, see *Mittal* [1984].

-I[*list*] Name of ASCII file with a list of track names (one per record) that should be excluded from consideration [Default includes all tracks].

-L[*corrtable*] Apply optimal corrections to the chosen observable. Append the correction table to use [Default uses the correction table *TAG_corrections.txt* which is expected to reside in the **\$X2SYS_HOME/TAG** directory]. For the format of this file, see **x2sys_solve**.

-Nnx_min Only report data from tracks involved in at least *nx_min* crossovers [all tracks].

-Qeli Append **e** for external crossovers or **i** for internal crossovers only [Default is external].

-R[*unit*]*west/east/south/north*[/*zmin/zmax*][**r**] *west, east, south, and north* specify the region of interest, and you may specify them in decimal degrees or in [+]*-*dd:mm[:ss.xxx][W|E|S|N] format. Append **r** if lower left and upper right map coordinates are given instead of w/e/s/n. The two shorthands **-Rg** and **-Rd** stand for global domain (0/360 and -180/+180 in longitude respectively, with -90/+90 in latitude). Alternatively, specify the name of an existing grid file and the **-R** settings (and grid spacing, if applicable) are copied from the grid. Using **-Runit** expects projected (Cartesian) coordinates compatible with chosen **-J** and we inversely project to determine actual rectangular geographic region. For perspective view (**-p**), optionally append */zmin/zmax*. In case of perspective view (**-p**), a z-range (*zmin, zmax*) can be appended to indicate the third dimension. This needs to be done only when using the **-Jz** option, not when using only the **-p** option. In the latter case a perspective view of the plane is plotted, with no third dimension. For Cartesian data just give *xmin/xmax/ymin/ymax*. This option bases the statistics on those COE that fall inside the specified domain.

-Strack Name of a single track. If given we restrict output to those crossovers involving this track [Default output is crossovers involving any track pair].

-V[*level*] (*more ...*) Select verbosity level [*c*].

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just **-**).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.39.5 Examples

To report statistics of all the external magnetic crossovers associated with the tag MGD77 from the file COE_data.txt, restricted to occupy a certain region in the south Pacific, try

```
gmt x2sys_report COE_data.txt -V -TMGD77 -R180/240/-60/-30 -Cmag >
mag_report.txt
```

To report on the faa crossovers globally that involves track 12345678, try

```
gmt x2sys_report COE_data.txt -V -TMGD77 -Cfaa -S2345678 > faa_report.txt
```

2.39.6 References

Mittal, P. K. (1984), Algorithm for error adjustment of potential field data along a survey network, *Geophysics*, **49**(4), 467-469.

2.39.7 See Also

x2sys_binlist x2sys_cross x2sys_datalist x2sys_get x2sys_init x2sys_list x2sys_put x2sys_solve

2.40 x2sys_solve

x2sys_solve - Determine least-squares systematic correction from crossovers

2.40.1 Synopsis

```
x2sys_solve -Ccolumn -TTAG -Emode [ COE_list.d ] [ -V[level] ] [ -W[u] ] [ -bi[ncols][type][w][+L+B] ]
```

Note: No space is allowed between the option flag and the associated arguments.

2.40.2 Description

x2sys_solve will use the supplied crossover information to solve for systematic corrections that can then be applied per track to improve data quality. Several systematic corrections can be solved for using a least-squares approach. Note: Only one data column can be processed at the time.

2.40.3 Required Arguments

COE_list.d Name of file with the required crossover columns as produced by **x2sys_list**. NOTE: If **-bi** is used then the first two columns are expected to hold the integer track IDs; otherwise we expect those columns to hold the text string names of the two tracks. If no file is given we will read from *stdin*.

-TTAG Specify the x2sys TAG which tracks the attributes of this data type.

-Ccolumn Specify which data column you want to process. Needed for proper formatting of the output correction table and must match the same option used in **x2sys_list** when preparing the input data.

-Emode The correction type you wish to model. Choose among the following functions $f(p)$, where p are the m parameters per track that we will fit simultaneously using a least squares approach:

c will fit $f(p) = a$ (a constant offset); records must contain cruise ID1, ID2, COE.

d will fit $f(p) = a + b * d$ (linear drift; d is distance; records must contain cruise ID1, ID2, d1, d2, COE.

g will fit $f(p) = a + b \sin(y)^2$ (1980-1930 gravity correction); records must contain cruise ID1, ID2, latitude y , COE.

h will fit $f(p) = a + b \cos(H) + c \cos(2H) + d \sin(H) + e \sin(2H)$ (magnetic heading correction); records must contain cruise ID1, ID2, heading H , COE.

s will fit $f(p) = a * z$ (a unit scale correction); records must contain cruise ID1, ID2, $z1$, $z2$.

t will fit $f(p) = a + b * (t - t0)$ (linear drift; $t0$ is the start time of the track); records must contain cruise ID1, ID2, $t1-t0$, $t2-t0$, COE.

2.40.4 Optional Arguments

-V[level] (*more ...*) Select verbosity level [c].

-W Means that each input records has an extra column with the composite weight for each crossover record. These are used to obtain a weighted least squares solution [no weights]. Append **u** to report unweighted mean/std [Default, report weighted stats].

-bi[ncols][type] (more ...) Select binary input.

-^ or just - Print a short message about the syntax of the command, then exits (NOTE: on Windows use just -).

-+ or just + Print an extensive usage (help) message, including the explanation of any module-specific option (but not the GMT common options), then exits.

-? or no arguments Print a complete usage (help) message, including the explanation of options, then exits.

--version Print GMT version and exit.

--show-datadir Print full path to GMT share directory and exit.

2.40.5 Notes

Most of the model corrections in **-E** involve a constant offset. Because crossovers are differences between values, any absolute level will cancel out and hence the constant offsets we obtain are relative to an undetermined absolute level. To obtain a solvable solution we add the constraint that the sum of all constant offsets equal zero. If the tracks form clusters in which no tracks from one cluster cross any track from another cluster then these are two independent data sets and require they own constraint equation for their offsets. We determine the number of clusters and automatically add the required constraint equations. If you need a particular reference track to have a particular offset (e.g., 0) then you can subtract the offset you found from every track correction and add in the desired offset.

2.40.6 Examples

To fit a simple bias offset to faa for all tracks under the MGD77 tag, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -Cfaa -Fnc > faa_coe.txt
gmt x2sys_solve faa_coe.txt -V -TMGD77 -Cfaa -Ec > coe_table.txt
```

To fit a faa linear drift with time instead, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -Cfaa -FnTc > faa_coe.txt
gmt x2sys_solve faa_coe.txt -V -TMGD77 -Cfaa -Et > coe_table.txt
```

To estimate heading corrections based on magnetic crossovers associated with the tag MGD77 from the file COE_data.txt, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -Cmag -Fnhc > mag_coe.txt
gmt x2sys_solve mag_coe.txt -V -TMGD77 -Cmag -Eh > coe_table.txt
```

To estimate unit scale corrections based on bathymetry crossovers, try

```
gmt x2sys_list COE_data.txt -V -TMGD77 -Cdepth -Fnz > depth_coe.txt
gmt x2sys_solve depth_coe.txt -V -TMGD77 -Cdepth -Es > coe_table.txt
```

2.40.7 See Also

x2sys_binlist, x2sys_cross, x2sys_datalist, x2sys_get, x2sys_init, x2sys_list, x2sys_put, x2sys_report

B

backtracker, **390**
blockmean, **1**
blockmedian, **3**
blockmode, **5**

D

dimfilter, **367**

F

filter1d, **8**
fitcircle, **10**

G

gmt, **28**
gmt.conf, **14**
gmt2kml, **43**
gmt5syntax, **48**
gmt_shell_functions.sh, **12**
gmtcolors, **48**
gmtconnect, **60**
gmtconvert, **62**
gmtdefaults, **66**
gmtget, **67**
gmtgravmag3d, **371**
gmtinfo, **68**
gmtlogo, **70**
gmtmath, **71**
gmtselect, **79**
gmtset, **83**
gmtsimplify, **85**
gmtspatial, **87**
gmtswitch, **90**
gmtvector, **92**
gmtwhich, **95**
gravfft, **372**
grd2cpt, **96**
grd2rgb, **99**
grd2xyz, **101**

grdblend, **103**
grdclip, **106**
grdcontour, **108**
grdcut, **113**
grdedit, **116**
grdffft, **118**
grdfilter, **122**
grdgradient, **126**
grdgravmag3d, **377**
grdhisteq, **129**
grdimage, **131**
grdinfo, **135**
grdlandmask, **137**
grdmask, **140**
grdmath, **142**
grdpaste, **151**
grdpmodeler, **393**
grdproject, **152**
grdraSTER, **155**
grdredpol, **379**
grdreformat, **157**
grdrotater, **395**
grdsample, **161**
grdseamount, **381**
grdspotter, **398**
grdtrack, **163**
grdtrend, **168**
grdvector, **170**
grdview, **173**
grdvolume, **177**
greenspline, **179**
gshhg, **311**

H

hotspotter, **401**

I

img2grd, **313**
isogmt, **184**

K

kml2gmt, 185

M

makecpt, 186
mapproject, 188
mgd77convert, 334
mgd77info, 336
mgd77list, 338
mgd77magref, 346
mgd77manage, 350
mgd77path, 355
mgd77sniffer, 357
mgd77track, 364

N

nearneighbor, 193

O

originator, 403

P

project, 195
ps2raster, 199
psbasemap, 203
psclip, 211
pscoast, 213
pscontour, 218
pscoupe, 317
pshistogram, 223
psimage, 226
pslegend, 228
pslib, 233
psmask, 244
psmeca, 322
pspolar, 326
psrose, 247
psscale, 250
pssegy, 383
pssegyz, 386
pstext, 253
psvelo, 329
pswiggle, 257
psxy, 259
psxyz, 268

R

rotconverter, 406

S

sample1d, 276
segy2grd, 388

spectrum1d, 278
sph2grd, 282
sphdistance, 285
sphinterpolate, 288
sphtriangulate, 290
splitxyz, 292
surface, 295

T

trend1d, 298
trend2d, 300
triangulate, 303

X

x2sys_binlist, 408
x2sys_cross, 410
x2sys_datalist, 413
x2sys_get, 415
x2sys_init, 417
x2sys_list, 422
x2sys_merge, 424
x2sys_put, 425
x2sys_report, 427
x2sys_solve, 428
xyz2grd, 306