NAME

blockmean – filter to block average (*x,y,z*) data by L2 norm

SYNOPSIS

blockmean [*xyz[w]file(s)*] **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–R***xmin/xmax/ymin/ymax*[**r**] [**–C**] [**–D**] [**–E**] [**–F**] [**–H**[**i**][*nrec*]] [**–S**[**w**|**z**]] [**–V**] [**–W**[**io**]] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

blockmean reads arbitrarily located (*x,y,z*) triples [or optionally weighted quadruples (*x,y,z,w*)] from standard input [or *xyz[w]file(s)*] 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 **D_FORMAT** parameter in your .gmtdefaults4 file, or you may choose binary input and/or output using single or double precision storage.

xyz[w]file(s)
3 [or 4] column ASCII file(s) [or binary, see **–b**] 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.

–I
x_inc [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg** **–Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

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

–E
Provide Extended report which includes **s** (the standard deviation of 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.

–F
Block centers have pixel registration. [Default: grid registration.] (Registrations are defined in **GMT** Cookbook Appendix B on grid file formats.) Each block is the locus of points nearest the grid value location. For example, with **–R**10/15/10/15 and and **–I**1: with the **–F** option 10 <= (*x,y*) < 11 is one of 25 blocks; without it 9.5 <= (*x,y*) < 10.5 is one of 36 blocks.

–H
Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

–S Use **–Sz** to report the sum of all *z*-values inside a block, or **–Sw** to report the sum of weights [Default reports mean value]. If **–Sw** is selected and no weights are supplied (i.e., no **–W** given), then the weight sum will equal the number of points inside each block.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W 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 inputonly, **–Wo** for weighted output only. [Default uses unweighted i/o].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 (or 4 if **–Wi** is set)].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 3 (or 4 if **–Wo** is set)]. **–E** adds 3 additional columns.

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

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

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

SEE ALSO

blockmedian(l), *blockmode*(l), *gmtdefaults*(l), *GMT*(l), *nearneighbor*(l), *surface*(l), *triangulate*(l)

NAME

blockmedian – filter to block average (*x,y,z*) data by L1 norm.

SYNOPSIS

blockmedian [*xyz[w]file(s)*] **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–R***xmin*/*xmax*/*ymin*/*ymax*[**r**] [**–C**] [**–E**] [**–F**] [**–H**[**i**][*nrec*]] [**–Q**] [**–T***quartile*] [**–V**] [**–W**[**io**]] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo*]

DESCRIPTION

blockmedian reads arbitrarily located (*x,y,z*) triples [or optionally weighted quadruples (*x,y,z,w*)] from standard input [or *xyz[w]file(s)*] 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 **D_FORMAT** parameter in your .gmtdefaults4 file, or you may choose binary input and/or output using single or double precision storage.

xyz[w]file(s)
3 [or 4] column ASCII file(s) [or binary, see **–b**] 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.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

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

–E 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*]. See **–W** for *w* output.

–F Block centers have pixel registration. [Default: grid registration.] (Registrations are defined in **GMT** Cookbook Appendix B on grid file formats.) Each block is the locus of points nearest the grid value location. For example, with **–R**10/15/10/15 and and **–I**1: with the **–F** option 10 <= (*x,y*) < 11 is one of 25 blocks; without it 9.5 <= (*x,y*) < 10.5 is one of 36 blocks.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

−Q (Quicker) Finds median *z* and (*x, y*) at that *z* [Default finds median *x*, median *y*, median *z*].

−T Sets the *quartile* of the distribution to be returned [Default is 0.5 which returns the median *z*]. Here, 0 < *quartile* < 1.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−W 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 inputonly, **−Wo** for weighted output only. [Default uses unweighted i/o].

−: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 (or 4 if **−Wi** is set)].

−bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 3 (or 4 if **−Wo** is set)]. **−E** adds 3 additional columns.

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f[i|o]g** means **−f[i|o]**0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

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

blockmedian hawaii_b.xyg **−R**198/208/18/25 **−I5m** **−bi**3 > hawaii_5x5.xyg

SEE ALSO

blockmean(l), *blockmode*(l), *GMT*(l), *gmtdefaults*(l), *nearneighbor*(l), *surface*(l), *triangulate*(l)

NAME

blockmode – filter to block average (*x,y,z*) data by mode estimation.

SYNOPSIS

blockmode [*xyz[w]file(s)*] **–I***xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]] **–R***xmin*/*xmax*/*ymin*/*ymax*[**r**] [**–C**] [**–E**] [**–F**] [**–H**[**i**][*nrec*]] [**–Q**] [**–V**] [**–W**[**io**]] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

blockmode reads arbitrarily located (*x,y,z*) triples [or optionally weighted quadruples (*x,y,z,w*)] from standard input [or *xyz[w]file(s)*] 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 **D_FORMAT** parameter in your .gmtdefaults4 file, or you may choose binary input and/or output using single or double precision storage.

xyz[w]file(s)
3 [or 4] column ASCII file(s) [or binary, see **–b**] 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.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

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

–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.

–F Block centers have pixel registration. [Default: grid registration.] (Registrations are defined in **GMT** Cookbook Appendix B on grid file formats.) Each block is the locus of points nearest the grid value location. For example, with **–R**10/15/10/15 and and **–I**1: with the **–F** option 10 <= (*x,y*) < 11 is one of 25 blocks; without it 9.5 <= (*x,y*) < 10.5 is one of 36 blocks.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

−Q (Quicker) Finds mode *z* and mean (*x, y*) [Default finds mode *x*, mode *y*, mode *z*].

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−W 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 inputonly, **−Wo** for weighted output only. [Default uses unweighted i/o].

−: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 (or 4 if **−Wi** is set)].

−bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 3 (or 4 if **−Wo** is set)]. **−E** adds 3 additional columns.

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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:

```
blockmode hawaii_b.xyg −R198/208/18/25 −I5m −bi3 > hawaii_5x5.xyg
```

SEE ALSO

blockmean(l), *blockmedian*(l), *GMT*(l), *gmtdefaults*(l), *nearneighbor*(l), *surface*(l), *triangulate*(l)

NAME

filter1d – Time domain filtering of 1-D time series

SYNOPSIS

filter1d [*infile*] **–F**<*type*><*width*>[*mode*] [**–D***increment*] [**–D**] [**–H**[**i**][*nrec*]] [**–I***ignore_val*] [**–L***lack_width*] [**–N***n_cols/t_col*] [**–Q***q_factor*] [**–S***symmetry_factor*] [**–T***start/stop/int*] [**–V**] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

filter1d is a general time domain filter for multiple column time series data. The user specifies the number of columns of input and which column is the time. (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.

infile Multi-column ASCII (or binary, see **–b**) file holding data values to be filtered.

–F 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 of 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.

OPTIONS

–D *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.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

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

–L 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].

–N Sets number of columns in input and which column contains the independent variable (time). The left-most column is # 0, the right-most is # (*n_cols* - 1). [Default is *n_cols* = 2, *t_col* = 0; i.e., file has t, f(t) pairs].

–Q 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].

–S Checks symmetry of data about window center. Enter a factor between 0 and 1. If ((abs(n_left - n_right)) / (n_left + n_right)) > *factor*, then no output will be given at this point [Default does not check Symmetry].

–T Make evenly spaced timesteps from *start* to *stop* by *int* [Default uses input times].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program.

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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:

```
filter1d cruise.gmtd –T0/1.0e5/2000 –FG10000 –N4/3 –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:

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

SEE ALSO

GMT(l), *sample1d*(l)

NAME

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

SYNOPSIS

fitcircle [*xyfile*] **−L***norm* [**−H**[**i**][*nrec*]] [**−S**] [**−V**] [**−:**[**i**|**o**]] [**−bi**[**s**|**S**|**d**|**D**][*ncol*]] [**−f**[**i**|**o**]*colinfo* **]**

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.

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

OPTIONS

xyfile ASCII [or binary, see **−b**] file containing lon,lat [lat,lon] values in the first 2 columns. If no file is specified, **fitcircle** will read from standard input.

−H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **−Hi** if only input data should have header records [Default will write out header records if the input data have them].

−S 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.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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

D_FORMAT setting.

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:

fitcircle ship.xyg **–L**2

project ship.xyg **–C***ox*/*oy* **–T***px*/*py* **–S** **–F**pz | **sample1d** **–S**–100 **–I**1 > 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

SEE ALSO

GMT(l), *project*(l), *sample1d*(l)

NAME

gmt2rgb – Converting a grdfile, a raw, or a Sun rasterfile to r/g/b grids

SYNOPSIS

gmt2rgb *infile* **–G***template* [**–C***cptfile*] [**–F**] [**–I***xinc*[**m**|**c**][/*yinc*[**m**|**c**]]] [**–L***layer*] [**–R***xmin*/*xmax*/*ymin*/*ymax*[**r**]] [**–V**] [**–W***width*/*height*[/*n_bytes*]

DESCRIPTION

gmt2rgb reads one of three types of input files: (1) A Sun 8-, 24-, or 32-bit rasterfile; we the write out the red, green, and blue components (0-255 range) to separate grid files. Since the rasterfile header is limited you may use the **–R, –F, –I** options to set a complete header record [Default is simply based on the number of rows and columns]. (2) A binary 2-D grdfile; 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 rasterfile. Since raw rastefiles have no header, you have to give the image dimensions via the **–W** option.

infile The (1) Sun rasterfile, (2) 2-D binary grdfile, or (3) raw rasterfile to be converted.

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

OPTIONS

–C name of the color palette table (for 2-D binary input grid only).

–F Will force pixel registration [Default is grid registration].

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–L Output only the specified layer (r, g, or b). [Default outputs all 3 layers].

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W 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**=*/=/4* 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.

EXAMPLES

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

gmt2rgb hawaii_grv.grd **–C**topo.cpt **–G**hawaii_grv_%c.grd

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

gmt2rgb radiation.ras **–L**r **–G**comp_%c.grd

SEE ALSO

gmtdefaults(l), *GMT*(l), *grdedit*(l), *grdimage*(l), *grdmath*(l), *grdview*(l)

NAME

gmtconvert – Converts, Pastes, and/or Extracts columns from ASCII and binary 1-D tables

SYNOPSIS

gmtconvert [*inputfiles*] [**–A**] [**–D**[*template*]] [**–E**] [**–F***cols*] [**–H**[**i**][*nrec*]] [**–L**] [**–I**] [**–M**[**i**|**o**][*flag*]] [**–N**] [**–S**"*search string*")] [**–V**] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

gmtconvert reads its standard input [or inputfiles] and writes out the desired information to standard output. It can do a combination of three things: (1) convert between binary and ASCII data tables, (2) paste corresponding records from multiple files into a single file, (3) extract a subset of the columns, (4) only extract segments whose header matches a text pattern search, (5) just list all multisegment headers and no data records, and (6) extract first and last data record for each segment. Input (and hence output) may have multiple subheaders if **–M** is selected, and ASCII tables may have regular headers as well.

datafile(s)
ASCII (or binary, see **–bi**) file(s) holding a number of data columns.

OPTIONS

–A The records from the input files should be pAsted horizontally, not appended vertically. [Default processes one file at the time]. Note for binary input, all the files you want to paste must have the same number of columns (as set with **–bi**).

–D 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 segment number); this is usually %d but could be %8.8d which gives leading zeros, etc. [Default is gmtconvert_segment_%d.d].

–E Only extract the first and last record for each segment of interest [Default extracts all records].

–F Give a comma-separated list of desired columns or ranges (0 is first column) [Default outputs all columns].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Invert the order of rows, i.e., output the final records in reverse order, starting with the last and ending up with the first input row [Default goes forward].

–L Only output a listing of all multisegment header records and no data records (requires **–M** and ASCII data).

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–N Do not write records that only contain NaNs in every field [Default writes all records].

–S Only output those segments whose header record contains the specified text string [Default output all segments].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program.

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

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

```
gmtconvert test.b –bis4 > test.dat
```

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

```
gmtconvert test.d –M –bo > test.b
```

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

```
gmtconvert table.d –F5,0 | 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

```
gmtconvert results.b –F8,4-6 -bi9s | psxy ...
```

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

```
gmtconvert left.d right.d –S –F1,2 | psxy ...
```

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

```
gmtconvert big_file.d –M –S"RIDGE AXIS" > subset.d
```

SEE ALSO

GMT(l), *minmax*(l)

NAME

gmtdefaults – To list current **GMT** defaults

SYNOPSIS

gmtdefaults –D[**u**|**s**] | **–L**

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 .gmtdefaults4 in your home or current directory (if you do not have this file, run **gmtdefaults** -D > ˜/.gmtdefaults4 to get one with the system default settings), or (3) overrid 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 emphemeral 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 by the contents of the gmt.conf file in **GMT**'s share directory.

–D Print the system **GMT** defaults to standard output. Append **u** for US defaults or **s** for SI defaults. [**–D** alone gives current choice in gmt.conf].

–L Print the user's currently active defaults to standard output.

Your currently active defaults come from the .gmtdefaults4 file in the current working directory, if present; else from the .gmtdefaults4 file in your home directory, if present; else from the system defaults set at the time **GMT** was compiled.

GMT PARAMETERS

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 **MEASURE_UNIT**). You can explicitly specify the unit used for distances and lengths by appending **c** (cm), **i** (inch), **m** (meter), or **p** (points). When no unit is indicated the value will be assumed to be in the unit set by **MEASURE_UNIT**. Note that the printer resolution **DOTS_PR_INCH** is always the number of dots or pixels per inch. Several parameters take only TRUE or FALSE.

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]

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.) [0]

ANNOT_FONT_PRIMARY
Font used for upper annotations, etc. [Helvetica]. Specify either the font number or the font name (case sensitive!). 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

ANNOT_FONT_SIZE_PRIMARY
Font size (> 0) in points for map annotations. [14]

ANNOT_FONT_SECONDARY
Font to use for time axis secondary annotations. See **ANNOT_FONT_PRIMARY** for available fonts [Helvetica].

ANNOT_FONT_SIZE_SECONDARY
Font size (> 0) for time axis secondary annotations in points [16].

ANNOT_OFFSET_PRIMARY
Distance from end of tickmark to start of annotation [0.2**c** (or 0.075**i**)]. A negative offset will place the annotation inside the map border.

ANNOT_OFFSET_SECONDARY
Distance from base of primary annotation to the top of the secondary annotation [0.2**c** (or 0.075**i**)] (Only applies to time axes with both primary and secondary annotations).

BASEMAP_AXES
Sets which axes to draw and annotate. Case sensitive: Upper case means both draw and annotate, lower case means draw axis only. [WESN].

BASEMAP_FRAME_RGB
Color used to draw map boundaries and annotations. Give a red/green/blue triplet, with each element in the 0–255 range. Prepend '+' to replicate this color to the tick-, grid-, and frame-pens. [0/0/0] (black).

BASEMAP_TYPE
Choose between 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 lon and lat directions (e.g., rectangular projections, polar projections).

CHAR_ENCODING

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+)].

COLOR_BACKGROUND

Color used for the background of images (i.e., when z < lowest colortable entry). Give a red/green/blue triplet, with each element in the 0–255 range. [0/0/0] (black)

COLOR_FOREGROUND

Color used for the foreground of images (i.e., when z > highest colortable entry). Give a red/green/blue triplet, with each element in the 0–255 range. [255/255/255] (white)

COLOR_IMAGE

Selects which operator to use when rendering bit-mapped color images. Due to the lack of the colorimage operator in some *PostScript* implementations, **GMT** offers 2 different options:

adobe (Adobe's colorimage definition) [Default].
tiles (Plot image as many individual rectangles).

COLOR_MODEL

Selects if color palette files contain RGB values (r,g,b in 0-255 range), HSV values (h in 0-360, s,v in 0-1 range), or CMYK values (c,m,y,k in 0-100% range). A **COLOR_MODEL** setting in the color palette file will override this setting. Internally, color interpolation takes place directly on the rgb values which can give unexpected hues, whereas interpolation directly on the hsv values better preserves the hues. Prepend the prefix "+" to force interpolation in the selected color system (does not apply to the CMYK system) [rgb].

COLOR_NAN

Color used for the non-defined areas of images (i.e., where z == NaN). Give a red/green/blue triplet, with each element in the 0–255 range. [128/128/128] (gray)

D_FORMAT

Output format (C language printf syntax) to be used when printing double precision floating point numbers. For geographic coordinates, see **OUTPUT_DEGREE_FORMAT**. [%lg].

DEGREE_SYMBOL

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

DOTS_PR_INCH

Resolution of the plotting device (dpi). Note that in order to be as compact as possible, **GMT** *PostScript* output uses integer formats only so the resolution should be set depending on what output device you are using. E.g, using 300 and sending the output to a Linotype 300 phototypesetter (2470 dpi) will not take advantage of the extra resolution (i.e., positioning on the page and line thicknesses are still only done in steps of 1/300 inch; of course, text will look smoother) [300].

ELLIPSOID

The (case sensitive) name of the ellipsoid used for the map projections [WGS-84]. Choose among

WGS-841984	World Geodetic System [Default]	
OSU91A	1991	Ohio State University
OSU86F1986	Ohio State University	
Engelis 1985	Goodard Earth Models	
SGS-85 1985	Soviet Geodetic System	
MERIT-83	1983	United States Naval Observatory
GRS-80 1980	International Geodetic Reference System	
Hughes-1980	1980	Hughes Aircraft Company for DMSP SSM/I grid products

Lerch	1979	For geoid modelling
ATS77	1977	Average Terrestrial System, Canada Maritime provinces
IAG-75	1975	International Association of Geodesy
Indonesian	1974	Applies to Indonesia
WGS-72	1972	World Geodetic System
NWL-10D	1972	Naval Weapons Lab (Same as WGS-72)
South-American	1969	Applies to South America
Fischer-1968	1968	Used by NASA for Mercury program
Modified-Mercury-1968	1968	Same as Fischer-1968
GRS-67	1967	International Geodetic Reference System
International-1967	1967	Worldwide use
WGS-66	1966	World Geodetic System
NWL-9D	1966	Naval Weapons Lab (Same as WGS-66)
Australian	1965	Applies to Australia
APL4.9	1965	Appl. Physics
Kaula	1961	From satellite tracking
Hough	1960	Applies to the Marshall Islands
WGS-60	1960	World Geodetic System
Fischer-1960	1960	Used by NASA for Mercury program
Mercury-1960	1960	Same as Fischer-1960
Modified-Fischer-1960	1960	Applies to Singapore
Fischer-1960-SouthAsia	1960	Same as Modified-Fischer-1960
Krassovsky	1940	Used in the (now former) Soviet Union
War-Office	1926	Developed by G. T. McCaw
International-1924	1924	Worldwide use
Hayford-1909	1909	Same as the International 1924
Helmert-1906	1906	Applies to Egypt
Clarke-1880	1880	Applies to most of Africa, France
Clarke-1880-Arc1950	1880	Modified Clarke-1880 for Arc 1950
Clarke-1880-IGN	1880	Modified Clarke-1880 for IGN
Clarke-1880-Jamaica	1880	Modified Clarke-1880 for Jamaica
Clarke-1880-Merchich	1880	Modified Clarke-1880 for Merchich
Clarke-1880-Palestine	1880	Modified Clarke-1880 for Palestine
Andrae	1876	Applies to Denmark and Iceland
Clarke-1866	1866	Applies to North America, the Philippines
Clarke-1866-Michigan	1866	Modified Clarke-1866 for Michigan
Struve	1860	Friedrich Georg Wilhelm Struve
Clarke-1858	1858	Clarke's early ellipsoid
Airy	1830	Applies to Great Britain
Airy-Ireland	1830	Applies to Ireland in 1965
Modified-Airy	1830	Same as Airy-Ireland
Bessel	1841	Applies to Central Europe, Chile, Indonesia
Bessel-Schwazeck	1841	Applies to Namibia
Bessel-Namibia	1841	Same as Bessel-Schwazeck
Bessel-NGO1948	1841	Modified Bessel for NGO 1948
Everest-1830	1830	India, Burma, Pakistan, Afghanistan, Thailand
Everest-1830-Kalianpur	1830	Modified Everest for Kalianpur (1956)
Everest-1830-Kertau	1830	Modified Everest for Kertau, Malaysia & Singapore
Everest-1830-Timbalai	1830	Modified Everest for Timbalai, Sabah Sarawak
Everest-1830-Pakistan	1830	Modified Everest for Pakistan
Walbeck	1819	First least squares solution by Finnish astronomer
Plessis	1817	Old ellipsoid used in France
Delambre	1810	Applies to Belgium
CPM	1799	Comm. des Poids et Mesures, France

Maupertius	1738	Really old ellipsoid used in France
Sphere	1980	The mean radius in GRS-80 (for spherical/plate tectonics applications)

Note that for some global projections, **GMT** may default to GRS-80 Sphere regardless of ellipsoid actually chosen. A warning will be given when this happens. If a different ellipsoid name than those mentioned here is given, **GMT** will attempt to open a file with that name in the current directory, and read a single record that contains the ellipsoid name, year, major-axis (in m), minor-axis (in m), and flattening (f) from the first record, where the fields must be separated by white-space (not commas). This way a custom ellipsoid (e.g., those used for other planets) may be used. A negative flattening means **GMT** will recalculate flattening from the two radii.

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

FRAME_PEN
Thickness of pen used to draw plain map frame in dpi units or points (append p) [1.25p].

FRAME_WIDTH
Width (> 0) of map borders for fancy map frame [0.2**c** (or 0.075**i**)].

GLOBAL_X_SCALE
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].

GLOBAL_Y_SCALE
Same, but for y-coordinates [1.0].

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

GRID_PEN_PRIMARY
Pen thickness used to draw grid lines in dpi units or points (append p) [0.25p].

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

GRID_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 [bcnsr][bsifd]. The first letter indicates native **GMT** binary, old format NetCDF, COARDS-compliant NetCDF, Surfer format or Sun Raster format. The second letter stands for byte, short, int, float and double, respectively. When /invalid is omitted the appropriate value for the given format is used (NaN or largest negative). When /scale/offset is omitted, /1.0/0.0 is used. [nf].

GRID_PEN_SECONDARY
Pen thickness used to draw grid lines in dpi units or points (append p) [0.5p].

GRIDFILE_SHORTHAND
If TRUE, all gridfile 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].

HEADER_FONT
Font to use when plotting headers. See **ANNOT_FONT_PRIMARY** for available fonts [Helvetica].

HEADER_FONT_SIZE
Font size (> 0) for header in points [36].

HEADER_OFFSET
Distance from top of axis annotations (or axis label, if present) to base of plot header [0.5**c** (or 0.1875**i**)].

HISTORY
If TRUE, passes the history of past common command options via the hidden .gmtcommands4 file [TRUE].

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

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

HSV_MIN_VALUE
Minimum value (0−1) assigned for most negative intensity value [0.3].

HSV_MAX_VALUE
Maximum value (0−1) assigned for most positive intensity value [1.0].

INPUT_CLOCK_FORMAT
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].

INPUT_DATE_FORMAT
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 **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 don't specify years. Examples are yyyyWwwd, yyyy-Www, etc. [yyyy-mm-dd].

INTERPOLANT
Determines if linear (linear), Akima's spline (akima), or natural cubic spline (cubic) should be used for 1-D interpolations in various programs [akima].

IO_HEADER
(* **−H**) Specifies whether input/output ASCII files have header record(s) or not [FALSE].

N_HEADER_RECS
Specifies how many header records to expect if **−H** is turned on [1].

LABEL_FONT
Font to use when plotting labels below axes. See **ANNOT_FONT_PRIMARY** for available fonts [Helvetica].

LABEL_FONT_SIZE
Font size (> 0) for labels in points [24].

LABEL_OFFSET
Distance from base of axis annotations to the top of the axis label [0.3**c** (or 0.1125**i**)].

LINE_STEP
Determines the maximum length (> 0) of individual straight line-segments when drawing arcuate lines [0.025**c** (or 0.01**i**)]

MAP_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.

MAP_SCALE_HEIGHT

Sets the height (> 0) on the map of the map scale bars drawn by various programs [0.2**c** (or 0.075**i**)].

MEASURE_UNIT

Sets the unit length. Choose between cm, inch, m, and point. [cm]. 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.

N_COPIES

(* **−c**) Number of plot copies to make [1].

OBLIQUE_ANNOTATION

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 tickmarks are extended to give a projection equal to the specified tick_length. If bit 5 is set (16), tickmarks 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].

OUTPUT_CLOCK_FORMAT

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 a|A with p|P 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].

OUTPUT_DATE_FORMAT

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 **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 consistant, e.g., you cannot specify months if you don't 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].

OUTPUT_DEGREE_FORMAT

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]. The various terms have the following purpose:

+	means output longitude in the 0 to 360 range [-180/+180]
-	means output longitude in the -360 to 0 range [-180/+180]

D	Use **D_FORMAT** for floating point degrees.
ddd	Fixed format integer degrees
:	delimiter used
mm	Fixed format integer arc minutes
ss	Fixed format integer arc seconds
F	Encode sign using WESN suffix

The default is +D.

PAGE_COLOR

Sets the color of the imaging background, i.e., the paper. Give a red/green/blue triplet, with each element in the 0–255 range. [255/255/255] (white).

PAGE_ORIENTATION

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

PAPER_MEDIA

Sets the physical format of the current plot paper [A4]. The following formats (and their widths and heights in points) are recognized (Additional site-specific formats may be specified in the gmtmedia.d file in **$GMTHOME**/share; 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
note	540	720
letter	612	792
legal	612	1008
11x17	792	1224
ledger	1224	792

For a completely custom format (e.g., for large format plotters) you may also specify Custom_WxH, where W and H are in points. To force the printer to request a manual paper feed, append '-' to the media name, e.g., A3- will require the user to insert a A3 paper into the printer's manual feed slot. To indicate you are making an EPS file, append '+' to the media name. Then,

GMT will attempt to issue a tight bounding box [Default Bounding Box is the paper dimension].

PLOT_CLOCK_FORMAT

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 **OUTPUT_CLOCK_FORMAT** for details. [hh:mm:ss].

PLOT_DATE_FORMAT

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 **OUTPUT_DATE_FORMAT** 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** setting. [yyyy-mm-dd].

PLOT_DEGREE_FORMAT

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 **OUTPUT_DEGREE_FORMAT** 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.

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].

PS_COLOR

Determines whether PostScript output should use RGB, HSV, or CMYK when specifying color [RGB]. Note if HSV is selected it does not apply to images which in that case uses RGB.

PS_IMAGE_COMPRESS

Determines if PostScript images are compressed using the Run-Length Encoding scheme (rle), LZW compression (lzw), or not at all (none) [none].

PS_IMAGE_FORMAT

Determines whether images created in PostScript should use ASCII or binary format. The latter takes up less space and executes faster but may choke some printers, especially those off serial ports. Select ascii or bin [ascii].

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 linewidth is drawn around the end points, and *square* cap where a half square of size equal to the linewidth 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_MITER_LIMIT

Sets the threshold angle (integer in 0-180 range) used for mitered joins. 0 and 180 are special flag values that imply the PostScript default [11] and always bevels, respectively. Other values sets the acute angle used to decide between mitered and bevelled.

PS_VERBOSE

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].

TICK_LENGTH

The length of a tickmark. Normally, tickmarks are drawn on the outside of the map boundaries. To select interior tickmarks, use a negative tick_length [0.2**c** (or 0.075**i**)].

TICK_PEN

The pen thickness to be used for tickmarks in dpi units or points (append p) [0.5p].

TIME_FORMAT_PRIMARY

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.

TIME_FORMAT_SECONDARY

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.

TIME_EPOCH

Specifying this parameter also sets **TIME_SYSTEM** to OTHER. It is a string of the form yyyy-mm-ddT[hh:mm:ss] (Gregorian) or yyyy-Www-ddT[hh:mm:ss] (ISO) indicating the value of the calendar and clock at the origin (zero point) of relative time units (see **TIME_UNIT**).

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 **c** second). **TIME_IS_INTERVAL** can have any of the following three values: (1) OFF [Default]. No adjustment, time is decoded as given. (2) +*n***u**. Activate interval adjustment for input by truncate to previous whole number of *n* units and then center time on the following interval. (3) -*n***u**. 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 1999-12-1T00:00:00.0 (start of December).

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_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
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 **$GMTHOME/**share/time/us.d file and make a similar file. Please submit it to the **GMT** Developers for official inclusion.

TIME_SYSTEM

Determines 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	Julian Date, -4713-11-25T12:00:00 d
MJD	Modified Julian Date, 1858-11-27T00:00:00 d
J2000	2000-01-01T00:00:00 d
S1985	1985-01-01T00:00:00 c
UNIX	1970-01-01T00:00:00 c
RD0001	0001-01-01T00:00:00 c
RATA	0000-12-31T00:00:00 d

or specify OTHER and supply your own **TIME_EPOCH** and **TIME_UNIT** settings [J2000].

TIME_UNIT

This parameter is active only when **TIME_SYSTEM** is set to OTHER, and specifies the units of relative time data. 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 c (second).

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)].

UNIX_TIME

(* **–U**) Specifies if a UNIX system time stamp should be plotted at the lower left corner of the plot [FALSE].

UNIX_TIME_POS

(* **–U**) Sets the position of the UNIX time stamp relative to the current plots lower left corner [-2**c**/-2**c** (or -0.75**i**/-0.75**i**)].

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].

VERBOSE

(* **–V**) Determines if **GMT** programs should display run-time information or run silently [FALSE].

X_AXIS_LENGTH

Sets the default length (> 0) of the x-axis [25**c** (or 9**i**)].

Y_AXIS_LENGTH

Sets the default length (> 0) of the y-axis [15**c** (or 6**i**)].

X_ORIGIN

(* **–X**) Sets the x-coordinate of the origin on the paper for a new plot [2.5**c** (or 1**i**)]. For an overlay, the default offset is 0.

Y_ORIGIN

(* **–Y**) Sets the y-coordinate of the origin on the paper for a new plot [2.5**c** (or 1**i**)]. For an overlay, the default offset is 0.

Y2K_OFFSET_YEAR

When 2-digit years are used to represent 4-digit years (see various **DATE_FORMAT**s), **Y2K_OFFSET_YEAR** gives the first year in a 100-year sequence. For example, if **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].

XY_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].

Y_AXIS_TYPE

Determines if the annotations for a y-axis (for linear projections) should be plotted horizontally (hor_text) or vertically (ver_text) [hor_text].

EXAMPLES

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

gmtdefaults –D > ˜/.gmtdefaults4

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.

BUGS

If you have typographical errors in your .gmtdefaults4 file(s), a warning message will be issued, and the **GMT** defaults for the affected parameters will be used.

SEE ALSO

GMT(l), *gmtset*(l)

NAME

gmtmath – Reverse Polish Notation calculator for data tables

SYNOPSIS

gmtmath [**–A***t_f(t).d*] [**–C***cols*] [**–F***cols*] [**–H**[**i**][*nrec*]] [**–I**] [**–M**[**i**|**o**][*flag*]] [**–N***n_col*/*t_col*] [**–Q**] [**–S**[**f**|**l**]] [**–T***t_min*/*t_max*/*t_inc*|*tfile*] [**–V**] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]** *operand* **[** *operand* **] OPERATOR [** *operand* **] OPERATOR ... = [** *outfile* **]**

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]. When two data tables are on the stack, each element in file A is modified by the corresponding element in file B. However, 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 **–b** to indicate the data domain. If STDIN is given, <stdin> 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**).

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

OPERATORS

Choose among the following operators:
Operatorn_args Returns

ABS	1	abs (A).
ACOS	1	acos (A).
ACOSH	1	acosh (A).
ADD	2	A + B.
AND	2	NaN if A and B == NaN, B if A == NaN, else A.
ASIN	1	asin (A).
ASINH	1	asinh (A).
ATAN	1	atan (A).
ATAN2	2	atan2 (A, B).
ATANH	1	atanh (A).
BEI	1	bei (A).
BER	1	ber (A).
CEIL	1	ceil (A) (smallest integer >= A).
CHICRIT	2	Critical value for chi-squared-distribution, with alpha = A and n = B.
CHIDIST	2	chi-squared-distribution P(chi2,n), with chi2 = A and n = B.
CORRCOEFF	2	Correlation coefficient r(A, B).
COL	1	Places column A on the stack.
COS	1	cos (A) (A in radians).
COSD	1	cos (A) (A in degrees).
COSH	1	cosh (A).
CPOISS	2	Cumulative Poisson distribution F(x,lambda), with x = A and lambda = B.
D2DT2	1	d^2(A)/dt^2 2nd derivative.
D2R	1	Converts Degrees to Radians.
DILOG	1	dilog (A).
DIV	2	A / B.

DDT	1	d(A)/dt 1st derivative.
DUP	1	Places duplicate of A on the stack.
ERF	1	Error function erf (A).
ERFC	1	Complementary Error function erfc (A).
ERFINV	1	Inverse error function of A.
EQ	2	1 if A == B, else 0.
EXCH	2	Exchanges A and B on the stack.
EXP	1	exp (A).
FCRIT	3	Critical value for F-distribution, with alpha = A, n1 = B, and n2 = C.
FDIST	3	F-distribution Q(F,n1,n2), with F = A, n1 = B, and n2 = C.
FLIPUD	1	Reverse order of each column
FLOOR	1	floor (A) (greatest integer <= A).
FMOD	2	A % B (remainder).
GE	2	1 if A >= B, else 0.
GT	2	1 if A > B, else 0.
HYPOT	2	hypot (A, B) = sqrt (A*A + B*B).
I0	1	Modified Bessel function of A (1st kind, order 0).
I1	1	Modified Bessel function of A (1st kind, order 1).
IN	2	Modified Bessel function of A (1st kind, order B).
INT	1	Numerically integrate A.
INV	1	1 / A.
ISNAN	1	1 if A == NaN, else 0.
J0	1	Bessel function of A (1st kind, order 0).
J1	1	Bessel function of A (1st kind, order 1).
JN	2	Bessel function of A (1st kind, order B).
K0	1	Modified Kelvin function of A (2nd kind, order 0).
K1	1	Modified Bessel function of A (2nd kind, order 1).
KN	2	Modified Bessel function of A (2nd kind, order B).
KEI	1	kei (A).
KER	1	ker (A).
LE	2	1 if A <= B, else 0.
LMSSCL	1	LMS scale estimate (LMS STD) of A.
LOG	1	log (A) (natural log).
LOG10	1	log10 (A) (base 10).
LOG1P	1	log (1+A) (accurate for small A).
LOG2	1	log2 (A) (base 2).
LOWER	1	The lowest (minimum) value of A.
LRAND	2	Laplace random noise with mean A and std. deviation B.
LSQFIT	1	Let current table be [A | b]; return least squares solution x = A \ b.
LT	2	1 if A < B, else 0.
MAD	1	Median Absolute Deviation (L1 STD) of A.
MAX	2	Maximum of A and B.
MEAN	1	Mean value of A.
MED	1	Median value of A.
MIN	2	Minimum of A and B.
MODE	1	Mode value (Least Median of Squares) of A.
MUL	2	A * B.
NAN	2	NaN if A == B, else A.
NEG	1	-A.
NEQ	2	1 if A != B, else 0.
NRAND	2	Normal, random values with mean A and std. deviation B.
OR	2	NaN if A or B == NaN, else A.
PLM	3	Associated Legendre polynomial P(-1<A<+1) degree B order C.
POP	1	Delete top element from the stack.

POW	2	A ˆ B.
R2	2	R2 = Aˆ2 + Bˆ2.
R2D	1	Convert Radians to Degrees.
RAND	2	Uniform random values between A and B.
RINT	1	rint (A) (nearest integer).
ROOTS	2	Treats col A as f(t) = 0 and returns its roots
ROTT	2	Rotate A by the (constant) shift B in the t-direction.
SIGN	1	sign (+1 or -1) of A.
SIN	1	sin (A) (A in radians).
SINC	1	sinc (A) (sin (pi*A)/(pi*A)).
SIND	1	sin (A) (A in degrees).
SINH	1	sinh (A).
SQRT	1	sqrt (A).
STD	1	Standard deviation of A.
STEP	1	Heaviside step function H(A).
STEPT	1	Heaviside step function H(t-A).
SUB	2	A - B.
SUM	1	Cumulative sum of A
TAN	1	tan (A) (A in radians).
TAND	1	tan (A) (A in degrees).
TANH	1	tanh (A).
TN	2	Chebyshev polynomial Tn(-1<A<+1) of degree B.
TCRIT	2	Critical value for Student's t-distribution, with alpha = A and n = B.
TDIST	2	Student's t-distribution A(t,n), with t = A, and n = B.
UPPER	1	The highest (maximum) value of A.
XOR	2	B if A == NaN, else A.
Y0	1	Bessel function of A (2nd kind, order 0).
Y1	1	Bessel function of A (2nd kind, order 1).
YN	2	Bessel function of A (2nd kind, order B).
ZCRIT	1	Critical value for the normal-distribution, with alpha = A.
ZDIST	1	Cumulative normal-distribution C(x), with x = A.

SYMBOLS

The following symbols have special meaning:

PI	3.1415926...
E	2.7182818...
T	Table with t-coordinates

OPTIONS

–A 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**).

–C 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.

–F Give a comma-separated list of desired columns or ranges that should be part of the output (0 is first column) [Default outputs all columns].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

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

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–N Select the number of columns and the column number that contains the "time" variable. Columns are numbered starting at 0 [2/0].

–Q Quick mode for scalar calculation. Shorthand for **–Ca –N**1/0 **–T**0/0/1.

–S 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].

–T 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**). 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 Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program.

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input, but see **–F**]

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

BEWARE

(1) The operator **PLM** calculates the associated Legendre polynomial of degree L and order M, and its argument is the cosine of the colatitude which must satisfy -1 <= x <= +1. **PLM** is not normalized.

(2) All derivatives are based on central finite differences, with natural boundary conditions.

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 | **gmtmath** STDIN **SQRT =** | process3

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

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 depth(in m) = 2500 + 350 * sqrt (age) to print the depth anomalies:

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

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

set mode_age = ‘**gmtmath –S –T** ages.d **MODE =**‘

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

gmtmath –Tt.d **T DILOG =** dilog.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 = ‘**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-t0), where H is the Heaviside step function for a given t0 = 1.55. Then, you need a 4-column augmented table loaded with t in column 0 and your observed y(t) in column 3. The calculation becomes

gmtmath –N4/1 **–A**ty.d **-C**0 1 **ADD –C**2 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 are preloaded with t and y(t), respectively, the other columns are zero. If you already have a precalculated table with the augmented matrix [A | b] in a file (say lsqsys.d), the least squares solution is simply

gmtmath –T lsqsys.d **LSQFIT =** solution.d

BEWARE

(1) 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).
(2) The stack depth limit is hard-wired to 100.
(3) All functions expecting a positive radius (e.g., **LOG**, **KEI**, etc.) are passed the absolute value of their argument..br (4) The **DDT** and **D2DT2** functions only work on regularly spaced data.
(5) **ROOTS** must be the last operator on the stack, only followed by **=**.

REFERENCES

Abramowitz, M., and I. A. Stegun, 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, vol. 55, Dover, New York.
Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, 1992, *Numerical Recipes*, 2nd edition, Cambridge Univ., New York.

SEE ALSO

GMT(l), *grd2xyz*(l), *grdedit*(l), *grdinfo*(l), *grdmath*(l), *xyz2grd*(l)

NAME

gmtselect – Select data subsets based on multiple spatial criteria

SYNOPSIS

gmtselect [*infiles*] [**–A***min_area*[*/min_level/max_level*]] [**–C**[**f**]*dist/ptfile*] [**–D***resolution*] [**–F***polygonfile*] [**–H**[**i**][*nrec*]] [**–I**[**cflrs**]] [**–J***parameters*] [**–L***dist/linefile*] [**–M**[**i**|**o**][*flag*]] [**–N***maskvalues*[**o**]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–V**] [**–Z***min/max*]] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

gmtselect is a filter that reads (longitude, latitude) positions from the first 2 columns of *infiles* [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.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

infiles ASCII (or binary, see **–b**) data file(s) to be operated on. If not given, standard input is read.

OPTIONS

–A Ignored unless **–N** is set. Geographical features with an area smaller than *min_area* in km^2 or of hierarchical level that is lower than *min_level* or higher than *max_level* will be ignored [Default is 0/4 (all features)]. See DATABASE INFORMATION in the **pscoast** man-pages for more details.

–C Pass all records whose location is within *dist* of any of the points in the file *ptfile*. If *dist* is zero then the 3rd column of *ptfile* must have each point's individual radius of influence. Distances are in user units; specify **–fg** or a map projection to indicate distances in km. Prepend **f** to indicate you want approximate flat earth distance calculations (faster) rather than exact geodesic distances (slower). However, if the current **ELLIPSOID** is set to Sphere then spherical great circle calculations are used. Note that *ptfile* must be in ASCII regardless of whether **–b** is used.

–D 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**]. 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.

–F 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 **–b** is used.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I 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 **–A, –D, –N**.
z select records NOT within the range specified by **–Z**.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–L Pass all records whose location is within *dist* of any of the line segments in the multiple-segment file *linefile*. If *dist* is zero then the 2nd column of each sub-header in the *ptfile* must have each lines's individual distance value. Distances are in user units; specify **–fg** or a map projection to indicate distances in km. Note that *linefile* must be in ASCII regardless of whether **–b** is used.

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–N 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:

–N*wet/dry*.
–N*ocean/land/lake/island/pond*.
Append **o** to let points exactly on feature boundaries be considered outside the feature [Default is inside]. [Default is s/k/s/k/s (i.e., s/k), which passes all points on dry land].

–Z Pass all records whose 3rd colum (z) lies within the given range. Input file must have at least three columns. To indicate no limit on min or max, specify a hyphen (-).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible). If no map projection is supplied we implicitly set **–Jx**1.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

NOTE

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.

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

```
gmtselect lonlatfile –Jd1d –C300/pts.d –L100/lines.d –Il > subset
```

Here, you must specify a map projection so that knows you are giving it geographical data (otherwise 300 would be interpreted as Cartesian distance in x-y units instead of km).

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

gmtselect data.d **−R**120/121/22/24 **−Dh −Ns/k** > subset

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

gmtselect quakes.d **−F**lonlatpath.d **−fg** > subset

SEE ALSO

gmtdefaults(l), *GMT*(l), *grdlandmask*(l), *pscoast*(l)

NAME

gmtset − To change individual **GMT** default parameters

SYNOPSIS

gmtset [**−G***defaultsfile*] PARAMETER1 [=] *value1* PARAMETER2 [=] *value2* PARAMETER3 [=] *value3* ...

DESCRIPTION

gmtset will adjust individual **GMT** defaults settings in the current directory's .gmtdefaults4 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 600, run the script, and reset to 300 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.

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 **gmtdefaults** man page.

OPTIONS

−G Name of specific .gmtdefaults4 file to modify [Default looks first in current directory, then in your home directory, and finally in the system defaults].

EXAMPLES

To change the dpi to 600, set annotation font to Helvetica, and select grid-crosses of size 0.1 inch, and set annotation offset to 0.2 cm:

gmtset DOTS_PR_INCH 600 **ANNOT_FONT_PRIMARY** Helvetica **GRID_CROSS_SIZE_PRIMARY** 0.1**i ANNOT_OFFSET_PRIMARY** 0.2**c**

SEE ALSO

GMT(l), *gmtdefaults*(l)

NAME

grd2cpt – Read a grdfile and make a color palette file

SYNOPSIS

grd2cpt *grdfile* [**–C***cptmaster*] [**–D**] [**–E***nlevels*] [**–I**] [**–L***minlimit/maxlimit*] [**–M**] [**–N**] [**–Q**[**i**|**o**]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–S***zstart/zstop/zinc*] [**–V**] [**–Z**]

DESCRIPTION

grd2cpt reads a grdfile and writes a color palette (cpt) file to standard output. The cpt file is based on an existing master cptfile 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 resulting cpt file is used with the grdfile and grdimage with a linear projection, the colors will be uniformly distributed in area on the plot. Let z be the data values in the grdfile. Define CDF(Z) = (# of z < Z) / (# of z in grdfile). (NaNs are ignored). These z-values are then normalized to the master cptfile 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 whereever 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 .gmtdefaults4 file or the command line will be used. This default behavior can be overruled using the options **–D**, **–M** or **–N**.

grdfile The 2-D binary grdfile used to derive the color palette table.

OPTIONS

–C 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 cptfile [Default gives a rainbow cpt file].

–D Select the colors for lowest and highest *z*-values in the output cpt file as the back- and foreground colors that will be written to the cpt file [Default uses the colors specified in the master file, or those defined by the parameters **COLOR_BACKGROUND**, **COLOR_FOREGROUND**, and **COLOR_NAN**].

–E Create a linear color table by dividing the grid z-range into *nlevels* equidistant slices.

–I Reverses the sense of color progression in the master cptfile. Also exchanges the foreground and background colors, including those specified by the parameters **COLOR_BACKGROUND** and **COLOR_FOREGROUND**.

–L Limit range of cptfile to *minlimit/maxlimit*, and don't count data outside 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 .gmtdefaults4 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 Selects a logarithmic interpolation scheme [Default is linear]. **–Qi** expects input z-values to be log10(z), assigns colors, and writes out z [Default]. **–Qo** takes log10(z) first, assigns colors, and writes out z.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–S Set steps in cpt file. Calculate entries in cptfile from *zstart* to *zstop* in steps of (*zinc*). [Default chooses arbitrary values by a crazy scheme.]

–V Verbose operation. This will write CDF(Z) estimates to stderr. [Default is silent.]

–Z Will create a continuous color palette. [Default is discontinuous, i.e., constant color intervals]

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):

grd2cpt mydata.grd **–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 cptfile relief, run

grd2cpt mydata.grd **–C***relief* **-L**0/10000 **-S**0/200/20 > mydata.cpt

SEE ALSO

gmtdefaults(l), *GMT*(l), *grdhisteq*(l), *grdinfo*(l), *makecpt*(l)

NAME

grd2xyz – Converting grdfile(s) to ASCII or binary data

SYNOPSIS

grd2xyz *grdfiles* [**–E**[*nodata*]] [**–H**[**i**][*nrec*]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–S**[**r**]] [**–V**] [**–W**[*weight*]] [**–Z**[*flags*]] [**–bo**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

DESCRIPTION

grd2xyz reads one or more binary 2-D grdfiles and writes out xyz-triplets in ASCII [or binary] format to standard output. Modify the precision of the ASCII output format by editing the **D_FORMAT** parameter in your .gmtdefaults4 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) in a number of formats, see **–E** or **–Z** below.

grdfiles Names of 2-D binary grd files to be converted.

OPTIONS

–E Output an ESRI ArcInfo ASCII interchange grid format file. Append *nodata* which will be used wherever the grid value equals NaN [-9999]. Note that all data values are written as integers.

–H Output 1 header record based on information in the first grdfile header. Ignored if binary output is selected. [Default is no header].

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–S Suppress output for nodes whose z-value equals NaN [Default outputs all nodes]. Append **r** to reverse the suppression, i.e., only output the nodes whose z-value equals NaN.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Write out x,y,z,w, where w is the supplied *weight* (or 1 if not supplied) [Default writes x,y,z only].

–Z 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 = xmax, append **x**. For grid periodic in y, skip writing the redundant row at y = ymax 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
c signed 1-byte character
u unsigned 1-byte character
h short 2-byte integer
i 4-byte integer
l long (4- or 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 Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 3]. This option only applies to xyz output; see **−Z** for z table output.

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

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

```
grd2xyz hawaii_grv.grd > hawaii_grv.xyz
```

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

```
grd2xyz raw_data.grd −ZTLf > hawaii_grv.b
```

SEE ALSO

gmtdefaults(l), *GMT*(l), *grdedit*(l), *xyz2grd*(l)

NAME

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

SYNOPSIS

grdblend *blendfile* **−G***grdfile* **−I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **−R***west*/*east*/*south*/*north*[**r**] [**−N***nodata*] [**−Q**] [**−Z***scale*] [**−V**] [**−W**] [**−f***colinfo*]

DESCRIPTION

grdblend reads a listing of gridded files and blend parameters and creates a binary grdfile 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.

blendfile
ASCII file with one record per grid file to include in the blend. Each record must contain the gridfile name, the **−R**-setting for the interior region, and the relative weight *wr*, separated by spaces or tabs. In the combined weighting scheme, this grid will be given weight = zero outside its domain, weight = *wr* inside the interior region, and a 2-D cosine-tapered weight between those end-members in the boundary strip. If the ASCII file is not given **grdblend** will read standard input.

−G *grdfile* is the name of the binary output grdfile.

−I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

−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 wesn. The two shorthands **−Rg −Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **−JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

−N No data. Set nodes with no input grid to this value [Default is NaN].

−Q Create a header-less gridfile suitable for use with **grdraster**. Requires that the output gridfile is a native format (i.e., not netCDF).

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−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].

−Z Scale output values by *scale* before writing to file. [1].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen

TIME_EPOCH), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To create a grdfile from the four gridded files piece_?.grd, make the blendfile like this

```
piece_1.grd     -R<subregion_1> 1
piece_2.grd     -R<subregion_2> 1
piece_3.grd     -R<subregion_3> 1
piece_4.grd     -R<subregion_4> 1
```

Then run

grdblend blend.job **–G**blend.grd **–R**<full_region> **–I**<dx/dy> **-V**

SEE ALSO

GMT(l), *grd2xyz*(l), *grdedit*(l)

NAME

grdclip – Clipping of range in grdfiles.

SYNOPSIS

grdclip *input_file.grd* **–G***output_file.grd* [**–Sa***high/above*] [**–Sb***low/below*] [**–V**]

DESCRIPTION

grdclip will set values < *low* to *below* and/or values > *high* to *above*. Useful when you want all of a continent or an ocean to fall into one color or grayshade in image processing, or clipping of the range of data values is required. *above/below* can be any number or NaN (Not a Number). You must choose at least one of **–Sa** or **–Sb**.

input_file.grd
The input 2-D binary grdfile.

–G *output_file.grd* is the modified output grdfile.

OPTIONS

–Sa Set all data[i] > *high* to *above*.

–Sb Set all data[i] < *low* to *below*.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

To set all values > 70 to NaN and all values < 0 to 0 in file data.grd:

```
grdclip data.grd –Gnew_data.grd –Sa70/NaN –Sb0/0 –V
```

SEE ALSO

GMT(l), *grdlandmask*(l), *grdmask*(l), *grdmath*(l), *grd2xyz*(l), *xyz2grd*(l)

NAME

grdcontour – contouring of 2-D gridded data sets

SYNOPSIS

grdcontour *grdfile* **–C***cont_int* **–J***parameters* [**–A**[**-**|*annot_int*][*labelinfo*]] [**–B**[**p**|**s**]*parameters*] [**–D***dumpfile*] [**–E***azimuth/elevation*] [**–G**[**d**|**f**|**n**|**l**|**L**|**x**|**X**]*params*] [**–K**] [**–L***low/high*] [**–M**[*flag*]] [**–O**] [**–P**] [**–Q***cut*] [**–R***west/east/south/north*[**r**]] [**–S***smoothfactor*] [**–T**[**+**|**-**][*gap/length*][**:***LH*]] [**–U**[*/dx/dy/*][*label*]] [**–V**] [**–W**[**+**][*type*]*pen*] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z**[*factor*[*/shift*]][**p**]] [**–c***copies*] [**–bo**[**s**|**S**|**d**|**D**][*ncol*]]

DESCRIPTION

grdcontour reads a 2-D gridded file and produces a contour map by tracing each contour through the grid. As an option, the x/y/z positions of the contour lines may be dumped to a single multisegment file or many separate files. *PostScript* code is generated and sent to standard output. Various options that affect the plotting are available.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

grdfile 2-D gridded data set to be contoured

–C 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. 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.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

OPTIONS

No space between the option flag and the associated arguments.

–A *annot_int* is annotation interval in data units. Ignored if contour levels are given in a file. [Default is no annotations]. Append **-** to disable all annotations implied by **–C**. 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:

+a*angle*
For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default]. By appending the **u** or **d** we get annotations whose top face the next upper or lower annotation, respectively.

+c*dx*[*/dy*]
Sets the clearance between label and optional text box. Append **c**|**i**|**m**|**p** to specify the unit or % to indicate a percentage of the label font size [15%].

+f*font* Sets the desired font [Default **ANNOT_FONT_PRIMARY**].

+g[*color*]
Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PAGE_COLOR**].

+j*just* Sets label justification [Default is CM].

+k*color*
Sets color of text labels [Default is **COLOR_BACKGROUND**].

+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 boxsets [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

+r*min_rad*
Will not place labels where the line's radius of curvature is less than *min_rad* [Default is 0].

+s*size* Sets the desired font size in points [Default is 9].

+u*unit* Appends *unit* to all line labels. If *unit* starts with a leading hypen (-) then there will be no space between label value and the unit. If no *unit* is appended we use the units listed in the gridfile. [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 hypen (-) then there will be no space between label value and the prefix. [Default is no prefix].

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–D Dump the (x,y,z) coordinates of each contour to separate files, one for each contour segment. The files will be named *dumpfile_cont_segment[_i]*.xyz (or *.b* is **–b** is selected), where *cont* is the contour value and *segment* is a running segment number for each contour interval (for closed contours we append _i.) If the prefix is given as '-' the file names are instead *C#_i* (interior) or *C#_e* (external) plus extension, and # is just a running number. This allows us to make short file names that will work with GNU utilities under DOS. However, when **–M** is used in conjunction with **–D** a single multisegment file is created instead.

–E Sets the viewpoint's azimuth and elevation for perspective view [180/90].

–G Controls the placement of labels along the contours. Choose among five controlling algorithms:

–Gd*dist*[c|i|m|p] or **–GD***dist*[d|e|k|m|n]
For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), **m** (meter), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), or **d** (spherical degree). [Default is 10**c** or 4**i**].

–Gf*ffile.d*
Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the contours. Inexact mathces and points outside the region are skipped.

–Gl|L*line1*[,*line2*,...]
Give *start* and *stop* coordinates for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the contours. 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 may use Z+ and Z- which correspond to the locations of the global max and min locations in the grid, respectively. **–GL** will interpret the point pairs as defining great circles [Default is straight line].

–Gn*n_label*
Specifies the number of equidistant labels for contours line [1]. Upper case **–GN** starts labeling exactly at the start of the line [Default centers them along the line]. **–GN**-1 places one justified label at start, while **–GN**+1 places one justified label at the end of contours. Optionally, append /*min_dist*[c|i|m|p] to enforce that a minimum distance separation between successive labels is enforced.

–Gx|X*xfile.d*
Reads the multi-segment file *xfile.d* and places labels at the intersections between the contours and the lines in*xfile.d*. **–GX** will resample the lines first along great-circle arcs.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Limit range: Do not draw contours for data values below *low* or above *high*.

–M When used in conjunction with **–D** a single multisegment file is created, and each contour section is preceeded by a header record whose first column is *flag* followed by the contour level.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Do not draw contours with less than *cut* number of points [Draw all contours].

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible). [Default is region defined in the grd file].

–S Used to resample the contour lines at roughly every (gridbox_size/*smoothfactor*) interval.

–T Will draw tickmarks pointing in the downward direction every *gap* along the innermost closed contours. Append *gap* and tickmark length or use defaults [0.5**c**/0.1c or 0.2**i**/0.04i]. User may choose to tick only local highs or local lows by specifying **–T+** or **–T-**, respectively. Appending :LH will plot the characters L and H at the center of closed innermost contours (local lows and highs). L and H can be any single character (e.g., LH, -+, etc.) 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 Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W *type*, if present, can be ’a’ for annotated contours or ’c’ for regular contours [Default]. *pen* sets the attributes for the particular line. Default values for annotated contours: width = 3, color = black, texture = solid. Regular contours have default width = 1. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes ‘-’ and dots ‘.’. If the **+** flag is specified then the color of the contour lines are taken from the cpt file (see **–C**).

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z 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.

–c Specifies the number of plot copies. [Default is 1].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program.

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

To contour the file hawaii_grav.grd every 25 mGal on a Mercator map at 0.5 inch/degree, annotate every 50 mGal (using fontsize = 10), using 1 degree tickmarks, and draw 30 minute gridlines:

grdcontour hawaii_grav.grd **–Jm**0.5**i** **–C**25 **–A**50**+f**10 **–B**1**g**30**m** > hawaii_grav.ps

To contour the file image.grd 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:

grdcontour image.grd **–Jx**0.1**c**/50.0**c** **–C**cont.d **–S**4 **–B**20/0.1:."RMS Misfit": **–Wa**5/255/0/0 **–Wc**1/0/0/255**ta** | 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.

SEE ALSO

GMT(l), *gmtdefaults*(l), *psbasemap*(l), *grdimage*(l), *grdview*(l), *pscontour*(l)

NAME

grdcut – Extract a subregion out of a *.grd* file

SYNOPSIS

grdcut *input_file.grd* **–G***output_file.grd* **–R***west*/*east*/*south*/*north*[**r**] [**–V**] [**–f**[**i**|**o**]*colinfo*]

DESCRIPTION

grdcut will produce a new *output_file.grd* file which is a subregion of *input_file.grd.* The subregion is specified with **–R** as in other programs; the specified range must not exceed the range of *input_file.grd.* If in doubt, run **grdinfo** to check range. Complementary to **grdcut** there is **grdpaste**, which will join together two grdfiles along a common edge.

input_file.grd
this is the input *.grd* format file.

–G*output_file.grd*
this is the output *.grd* format file.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible). This defines the subregion to be cut out.

OPTIONS

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

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:

grdcut grav_148_162_8_32.grd **–G**grav_150_160_10_30.grd **–R**150/160/10/30 **–V**

SEE ALSO

grdpaste(l), *grdinfo*(l), *GMT*(l)

NAME

grdedit – Modifying the header in a 2-D grdfile

SYNOPSIS

grdedit *grdfile* [**–A**] [**–D***xname/yname/zname/scale/offset/title/remark*] [**–N***xyzfile*] [**–R***west/east/south/north*[**r**]] [**–S**] [**–T**] [**–V**] [**–:**[**i**|**o**]] [**–bi**[s|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

grdedit reads the header information in a binary 2-D grdfile 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 grdheader.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

grdfile Name of the 2-D grdfile to modify

OPTIONS

–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 gridfiles (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 Give new values for *xname, yname, zname, scale, offset, title,* and *remark*. To leave some of the values untouched, specify = as the new value.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–N Read the ASCII (or binary; see **–bi**) file *xyzfile* and replace the corresponding nodal values in the grid with these *z* values.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible). 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 longitudionally according to the new borders given in **–R**.

–T Make necessary changes in the header to convert a gridline-registred 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 Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 input columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

Let us assume the file data.grd 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

grdedit data.grd **−R**-60/-50/10/30 **−D**=/=/=/=/=/"Gravity Anomalies"/=

The grid world.grd has the limits 0/360/-72/72. To shift the data so that the limits would be -180/180/-72/72, use

grdedit world.grd **−R**-180/180/-72/72 **−S**

The file junk.grd was created prior to **GMT** 3.1 with incompatible **−R** and **−I** arguments. To reset the x- and y-increments we run

grdedit junk.grd **−A**

SEE ALSO

GMT(l), *grd2xyz*(l), *xyz2grd*(l)

NAME

grdfft – Perform mathematical operations on grdfiles in the wavenumber (or frequency) domain

SYNOPSIS

grdfft *in_grdfile* **–G***out_grdfile* [**–A***azimuth*] [**–C***zlevel*] [**–D**[*scale*|**g**]] [**–E**[**x**|**y**][**w**]] [**–F**[**x**|**y**]*params*] [**–I**[*scale*|**g**]] [**–L**] [**–M**] [**–N***stuff*] [**–S***scale*] [**–T***te/rl/rm/rw/ri*] [**–V**]

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 grdfiles are assumed to be in meters. Geographical grids may be used by specifying the **–M** option that scales degrees to meters. If you have grdfiles with dimensions in km, you could change this to meters using **grdedit** or scale the output with **grdmath**.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

in_grdfile
2-D binary grd file to be operated on.

–G Specify the name of the output grd file.

OPTIONS

–A Take the directional derivative in the *azimuth* direction measured in degrees CW from north.

–C Upward (for *zlevel* > 0) or downward (for *zlevel* < 0) continue the field *zlevel* meters.

–D Differentiate the field, i.e., take d(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 * *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 Estimate power spectrum in the radial direction. Place **x** or **y** immediately after **–E** to compute the spectrum in the x or y direction instead. No grdfile is created; f (i.e., frequency or wave number), power[f], and 1 standard deviation in power[f] are written to stdout. Append **w** to write wavelength instead of frequency.

–F Filter the data. Place **x** or **y** immediately after **–F** to filter *x* or *y* direction only; default is isotropic. Choose between a cosine-tapered band-pass or a Gaussian band-pass filter. Cosine-taper: Specify four wavelengths in correct units (see **–M**) 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., **–F**1000000/250000/50000/10000 **–M** 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. **–Fy**1000/250/-/- will highpass *y*, passing wavelengths < 250 and rejecting wavelengths > 1000. Gaussian band-pass: Append two wavelengths in correct units (see **–M**) to design a bandpass filter. At the given wavelenghts 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 **–F**400/- will highpass the data.

–I 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].

–L Leave trend alone. By default, a linear trend will be removed prior to the transform.

–M Map units. Choose this option if your grdfile is a geographical grid and you want to convert degrees into meters. If the data are close to either pole, you should consider projecting the grdfile onto a rectangular coordinate system using **grdproject**.

–N Choose or inquire about suitable grid dimensions for FFT. **–Nf** will force the FFT to use the dimensions of the data. **–Nq** will inQuire about more suitable dimensions. **–N***nx/ny* will do FFT on array size *nx/ny* (Must be >= grdfile size). Default chooses dimensions >= data which optimize speed, accuracy of FFT. If FFT dimensions > grdfile dimensions, data are extended and tapered to zero.

–S Multiply each element by *scale* in the space domain (after the frequency domain operations). [Default is 1.0].

–T Compute the isostatic compensation from the topography load (input grdfile) on an elastic plate of thickness *te*. Also append densities for load, mantle, water, and infill in SI units. If *te* == 0 then the Airy response is returned. **–T** implicitly sets **–L**.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

To upward continue the sea-level magnetic anomalies in the file mag_0.grd to a level 800 m above sealevel:

grdfft mag_0.grd **–C**800 **–V –G**mag_800.grd

To transform geoid heights in m (geoid.grd) on a geographical grid to free-air gravity anomalies in mGal:

grdfft geoid.grd **–Dg –M –V –G**grav.grd

To transform gravity anomalies in mGal (faa.grd) 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:

grdfft faa.grd **–Ig –A**38 **–S**1e6 **–V –G**defl_38.grd

Second vertical derivatives of gravity anomalies are related to the curvature of the field. We can compute these as mGal/m^2 by differentiating twice:

grdfft gravity.grd **–D –D –V –G**grav_2nd_derivative.grd

The first order gravity anomaly (in mGal) due to the compensating surface caused by the topography load topo.grd (in m) on a 20 km thick elastic plate, assumed to be 4 km beneath the observation level can be computed as

grdfft topo.grd **–T**20000/2800/3330/1030/2300 **–S**0.022 **–C**4000 **–G**comp_faa.grd

where 0.022 is the scale needed for the first term in Parker's expansion for computing gravity from topography (= 2 * PI * G * (rhom - rhol)).

SEE ALSO

GMT(l), *grdedit*(l), *grdmath*(l), *grdproject*(l)

NAME

grdfilter – Filter a 2-D gridded file in the space (or time) domain

SYNOPSIS

grdfilter *input_file.grd* **–D***distance_flag* **–F**<*filtertype*><*width*>[*mode*] **–G***output_file.grd* [**–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–T**] [**–V**] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

grdfilter will filter a *.grd* file in the time domain using one of the selected convolution or non-convolution filters and compute distances using Cartesian or Spherical geometries. The output *.grd* file can optionally be generated as a sub**–R**egion of the input and/or with a new **–I**ncrement. 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.

input_file.grd
The file of points to be filtered.

–D 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 two 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 Sets the 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.
(**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 0.0.

–G *output_file.grd* is the output of the filter.

OPTIONS

–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].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

Suppose that north_pacific_dbdb5.grd 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:

```
grdfilter north_pacific_dbdb5.grd –Gfiltered_pacific.grd –Fm600 –D4 –R150/250/10/40 –I0.5 –V
```

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 fiter "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 grdfilter on the resulting Cartesian grid.

SEE ALSO

GMT(l), *grdfft*(l)

NAME

grdgradient – Compute directional derivative or gradient from 2-D grd file representing z(x,y)

SYNOPSIS

grdgradient *in_grdfile* **–G***out_grdfile* [**–A***azim*[*/azim2*]] [**–D**[**c**][**o**][**n**]] [**–E**[**s**|**p**]*azim/elev*[*/ambient/diffuse/specular/shine*]] [**–L***flag*] [**–M**] [**–N**[**e**][**t**] [*amp*][*/sigma*[*/offset*]]] [**–S***slopefile*] [**–V**]

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**).

in_grdfile
2-D grd file from which to compute directional derivative.

–G Name of the output grdfile for the directional derivative.

OPTIONS

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

–A 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. **–A***0/270* illuminates from the north (top) and west (left).

–D 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).

–E 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 **–E***60/30/=/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 picewise 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.)

–L 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).]

–M By default the units of **grdgradient** are in units_of_z/ units_of_dx_and_dy. However, the user may choose this option to convert dx,dy in degrees of longitude,latitude into meters, so that the units of **grdgradient**
are in z_units/meter.

–N 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(abs(*g*
- *offset*)). **–Ne** normalizes using a cumulative Laplace distribution yielding *gn* = *amp* * (1.0 - exp(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) * atan((*g* - *offset*)/ *sigma*) where *sigma* is estimated using the L2 norm of (*g* - *offset*) if it is not given.

–S Name of output grdfile with scalar magnitudes of gradient vectors. Requires **–D**.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

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 **–Ne**0.6.

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 grd.

If you simply need the *x*- or *y*-derivatives of the grid, use **grdmath**.

EXAMPLES

To make a file for illuminating the data in geoid.grd using exp- normalized gradients imitating light sources in the north and west directions:

grdgradient geoid.grd **–A**0/270 **–G**gradients.grd **–Ne**0.6 **–V**

To find the azimuth orientations of seafloor fabric in the file topo.grd:

grdgradient topo.grd **–Dno –G**azimuths.grd **–V**

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)

SEE ALSO

GMT(l), *gmtdefaults*(l), *grdhisteq*(l), *grdimage*(l), *grdview*(l), *grdvector*(l)

NAME

grdhisteq − Histogram equalization for grd files

SYNOPSIS

grdhisteq *in_grdfile* [**−G***out_grdfile*] [**−C***n_cells*] [**−D**] [**−N**[*norm*]] [**−Q**] [**−V**]

DESCRIPTION

grdhisteq allows the user to find the data values which divide a given grdfile 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 grdfile 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 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 grdfile 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 grdfile, run **grdhisteq** and make a grdfile 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 grdfile output with the Gaussian option, you will have a grdfile 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.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

in_grdfile
2-D binary grd file to be equalized.

OPTIONS

−C Sets how many cells (or divisions) of data range to make.

−D Dump level information to standard output.

−G Name of output 2-D grd file. Used with **−N** only.

−N Gaussian output. Use with **−G** to make an output grdfile 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].

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

To find the height intervals that divide the file heights.grd into 16 divisions of equal area:

```
grdhisteq heights.grd −C16 −D > levels.d
```

To make the poorly distributed intensities in the file raw_intens.grd suitable for use with **grdimage** or **grdview**, run

```
grdhisteq raw_intens.grd −Gsmooth_intens.grd −N −V
```

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 grdfile (nx * ny) and the cumulative gaussian distribution function F(x). That is, let N = nx * ny. 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 grdfile. But if N is very large, it is possible for x to be greater than 4. Therefore, with the **grdimage** 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**.

SEE ALSO

gmtdefaults(l), *GMT*(l), *grd2cpt*(l), *grdgradient*(l), *grdimage*(l), *grdmath*(l), *grdview*(l), *makecpt*(l)

NAME

grdimage – Create grayshaded or colored image from a 2-D netCDF grd file

SYNOPSIS

grdimage *grd_z* | *grd_r grd_g grd_b* **–C***cptfile* **–J***parameters* [**–B**[**p**|**s**]*parameters*] [**–Ei**|*dpi*] [**–G**[**f**|**b**]*rgb*] [**–I***intensfile*] [**–K**] [**–M**] [**–O**] [**–P**] [**–Q**] [**–R***west***/***east***/***south***/***north*[**r**]] [**–S***search_radius*] [**–T**[**s**][**o**[*pen*]]] [**–U**[*/dx/dy/*][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–c***copies*]

DESCRIPTION

grdimage reads one 2-D gridded file and produces a gray-shaded (or colored) map by assigning each z-value a gray-shade (or color). Alternatively, **grdimage** reads three 2-D gridded 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 grdfile using **grdgradient** and modified by **grdmath** or **grdhisteq**. Each grid-node in the grdfile is represented as a shaded (or colored) rectangle centered on the grid node. When using map projections, the grid is first resampled on a new rectangular grid (This can be a time-consuming process for large grid files; but see **–T**). A 24-bit true color *PostScript* file is output. The region option can be used to select a map region larger or smaller than that implied by the extent of the grdfile.

grd_z | *grd_r grd_g grd_b*
2-D gridded data set (or red, green, blue grids) to be imaged

–C name of the color palette table (for *grd_z* only).

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)

–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f|s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r|z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d|l|p***pow***|t|T**][*/y-scale*[**d|l|p***pow***|t|T**]] (Linear, log, and power scaling)

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–E Sets the resolution of the projected grid that will be created if a map projection other than Linear or Mercator was selected. 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 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.

–I Gives the name of a grdfile with intensities in the (-1,+1) range. [Default is no illumination].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Force conversion to monochrome image using the (television) YIQ transformation.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Mask out nodes with z = NaN using the colormasking feature in *PostScript* Level 3 (the PS device must support PS Level 3).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible). 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 grdfile will result in a subset of the grid [Default is region given by the grdfile].

–S Set the search radius for the averaging procedure [Default avoids aliasing].

−T 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.

−U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−X −Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

−c Specifies the number of plot copies. [Default is 1].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To gray-shade the file hawaii_grav.grd 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:

grdimage hawaii_grav.grd **−Jl**18/24/1.5**c** **−C**shades.cpt **−B**1 > hawaii_grav_image.ps

To create an illuminated color *PostScript* plot of the gridded data set image.grd, using the intensities provided by the file intens.grd, and color levels in the file colors.cpt, with linear scaling at 10 inch/x-unit, tickmarks every 5 units:

grdimage image.grd **−Jx**10**i** **−C**colors.cpt **−I**intens.grd **−B**5 > image.ps

To create an false color *PostScript* plot from the three gridded files red.grd, green.grd, and blue.grd, with linear scaling at 10 inch/x-unit, tickmarks every 5 units:

grdimage red.grd green.grd blue.grd **−Jx**10**i** **−B**5 > rgbimage.ps

SEE ALSO

GMT(l), *gmt2rgb*(l), *grdcontour*(l), *grdview*(l), *grdgradient*(l), *grdhisteq*(l)

NAME

grdinfo – Get information about the contents of a 2-D grd file

SYNOPSIS

grdinfo *grdfiles* [**–C**] [**–F**] [**–I***dx*[/*dy*]][**–L**[**1**|**2**]] [**–M**] [**–T***dz*] [**–V**] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

grdinfo reads a 2-D binary grd file and reports various statistics for the (*x,y,z*) data in the grdfile(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.

grdfile The name of one or several 2-D grd files.

OPTIONS

No space between the option flag and the associated 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 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).

–L1 Report median and L1 scale of z (L1 scale = 1.4826 * Median Absolute Deviation (MAD)).

–L2 Report mean and standard deviation 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.

–T 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**.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To obtain all the information about the data set in file hawaii_topo.grd:

grdinfo –L1 –L2 –M hawaii_topo.grd

SEE ALSO

GMT(1), *grd2cpt*(1), *grd2xyz*(1), *grdedit*(1)

NAME

grdlandmask – Create "wet-dry" mask grdfile from shoreline data base.

SYNOPSIS

grdlandmask **–G***mask_grd_file*] **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–R***west*/*east*/*south*/*north*[**r**] [**–A***min_area*[/*min_level*/*max_level*]] [**–D***resolution*] [**–F**] [**–N***maskvalues*[**o**]] [**–V**]

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 grdfile may be used in subsequent operations involving **grdmath** to mask out data from land [or water] areas.

–G Name of resulting output mask grd file.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg** **–Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude).

OPTIONS

–A Features with an area smaller than *min_area* in km^2 or of hierarchical level that is lower than *min_level* or higher than *max_level* will be ignored [Default is 0/4 (all features)]. See DATABASE INFORMATION in the **pscoast** man-pages for more details.

–D 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**]. 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.

–F Force pixel registration. [Default is grid registration].

–N 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:

–N*wet/dry*.
–N*ocean/land/lake/island/pond*.

[Default is 0/1/0/1/0 (i.e., 0/1)].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

To set all nodes on land to NaN, and nodes over water to 1, using the high resolution data set, do

grdlandmask **–R**-60/-40/-40/-30 **–Dh** **–I5m** **–N**1/NaN **–G**land_mask.grd **–V**

To make a 1x1 degree global grid with the hierarchical levels of the nodes based on the low resolution data:

grdlandmask −R0/360/-90/90 **−Dl −I**1 **−N**0/1/2/3/4 **−G**levels.grd **−V**

SEE ALSO

GMT(l), *grdmath*(l), *grdclip*(l), *psmask*(l), *psclip*(l), *pscoast*(l)

NAME

grdmask – Create mask grdfiles from xy paths.

SYNOPSIS

grdmask *pathfiles* **–G***mask_grd_file*] **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–R***west*/*east*/*south*/*north*[**r**] [**–A**] [**–F**] [**–H**[**i**][*nrec*]] [**–M**[*flag*]] [**–N***out/edge/in*] [**–S***search_radius*[**m**|**c**|**k**|**K**]] [**–V**] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

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 grdfile 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".

pathfiles
The name of 1 or more ASCII [or binary, see **–b**] files holding the polygon(s) or data points.

–G Name of resulting output mask grd file.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

–A If the input data are geographic (as indicated by **–fi**) then the sides in the polygons will be approximated by great circle arcs. This can be turned off using the **–A** switch.

–F Force pixel registration. [Default is grid registration].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–M Multiple segment file. Segments are separated by a record whose first character is *flag*. [Default is '>'].

–N Sets the values that will be assigned to nodes that are *out*side the polygons, on the *edge*, or *in*side. Values can be any number, including the textstring NaN [Default is 0/0/1].

−S Set nodes depending on their distance from the nearest data point. Nodes within *radius* [0] from a data point are considered inside. Append **m** to indicate minutes or **c** to indicate seconds. Append **k** to indicate km (implies **−R** and **−I** are in degrees, and we will use a fast flat Earth approximation to calculate distance). For more accuracy, use uppercase **K** if distances should be calculated along geodesics. However, if the current **ELLIPSOID** is set to Sphere then spherical great circle calculations are used.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To set all nodes inside and on the polygons coastline_*.xy to 0, and outside points to 1, do

grdmask coastline_*.xy **−R**-60/-40/-40/-30 **−I5m −N**1/0/0 **−G**land_mask.grd **−V**

To set nodes within 50 km of data points to 1 and other nodes to NaN, do

grdmask data.xyz **−R**-60/-40/-40/-30 **−I**5m **−N**NaN/1/1 **−S**50**k −G**data_mask.grd **−V**

SEE ALSO

GMT(l), *grdlandmask*(l), *grdmath*(l), *grdclip*(l), *psmask*(l), *psclip*(l)

NAME

grdmath − Reverse Polish Notation calculator for grd files

SYNOPSIS

grdmath [**−F**] [**−I***xinc*[*unit*][**=**|**+**][**/***yinc*[*unit*][**=**|**+**]]] [**−R***west***/***east***/***south***/***north*[**r**]] [**−V**] [**−bi**[**s**|**S**|**d**|**D**][*ncol*]] [**−f***colinfo*] *operand* [*operand*] **OPERATOR** [*operand*] **OPERATOR** ... **=** *outgrdfile*

DESCRIPTION

grdmath will perform operations like add, subtract, multiply, and divide on one or more grd 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 grd file. When two grd files are on the stack, each element in file A is modified by the corresponding element in file B. However, some operators only require one operand (see below). If no grdfiles are used in the expression then options **−R**, **−I** must be set (and optionally **−F**). The expression **=** *outgrdfile* can occur as many times as the depth of the stack allows.

operand

If *operand* can be opened as a file it will be read as a grd file. If not a file, it is interpreted as a numerical constant or a special symbol (see below).

outgrdfile is a 2-D grd file that will hold the final result.

OPERATORS

Choose among the following operators:

Operatorn_args Returns

ABS	1	abs (A).
ACOS	1	acos (A).
ACOSH	1	acosh (A).
ADD	2	A + B.
AND	2	NaN if A and B == NaN, B if A == NaN, else A.
ASIN	1	asin (A).
ASINH	1	asinh (A).
ATAN	1	atan (A).
ATAN2	2	atan2 (A, B).
ATANH	1	atanh (A).
BEI	1	bei (A).
BER	1	ber (A).
CAZ	2	Cartesian azimuth from grid nodes to stack x,y.
CDIST	2	Cartesian distance between grid nodes and stack x,y.
CEIL	1	ceil (A) (smallest integer >= A).
CHICRIT	2	Critical value for chi-squared-distribution, with alpha = A and n = B.
CHIDIST	2	chi-squared-distribution P(chi2,nn), with chi2 = A and n = B.
CORRCOEFF	2	Correlation coefficient r(A, B).
COS	1	cos (A) (A in radians).
COSD	1	cos (A) (A in degrees).
COSH	1	cosh (A).
CPOISS	2	Cumulative Poisson distribution F(x,lambda), with x = A and lambda = B.
CURV	1	Curvature of A (Laplacian).
D2DX2	1	d^2(A)/dx^2 2nd derivative.
D2DY2	1	d^2(A)/dy^2 2nd derivative.
D2R	1	Converts Degrees to Radians.
DDX	1	d(A)/dx 1st derivative.
DDY	1	d(A)/dy 1st derivative.
DILOG	1	dilog (A).

DIV	2	A / B.
DUP	1	Places duplicate of A on the stack.
ERF	1	Error function erf (A).
ERFC	1	Complementary Error function erfc (A).
ERFINV	1	Inverse error function of A.
EQ	2	1 if A == B, else 0.
EXCH	2	Exchanges A and B on the stack.
EXP	1	exp (A).
EXTREMA	1	Local Extrema: +2/-2 is max/min, +1/-1 is saddle with max/min in x, 0 elsewhere.
FCRIT	3	Critical value for F-distribution, with alpha = A, n1 = B, and n2 = C.
FDIST	3	F-distribution Q(F,n1,n2), with F = A, n1 = B, and n2 = C.
FLIPLR	1	Reverse order of values in each row.
FLIPUD	1	Reverse order of values in each column.
FLOOR	1	floor (A) (greatest integer <= A).
FMOD	2	A % B (remainder).
GE	2	1 if A >= B, else 0.
GT	2	1 if A > B, else 0.
HYPOT	2	hypot (A, B) = sqrt (A*A + B*B).
I0	1	Modified Bessel function of A (1st kind, order 0).
I1	1	Modified Bessel function of A (1st kind, order 1).
IN	2	Modified Bessel function of A (1st kind, order B).
INRANGE	3	1 if B <= A <= C, else 0.
INSIDE	1	1 when inside or on polygon(s) in A, else 0.
INV	1	1 / A.
ISNAN	1	1 if A == NaN, else 0.
J0	1	Bessel function of A (1st kind, order 0).
J1	1	Bessel function of A (1st kind, order 1).
JN	2	Bessel function of A (1st kind, order B).
K0	1	Modified Kelvin function of A (2nd kind, order 0).
K1	1	Modified Bessel function of A (2nd kind, order 1).
KN	2	Modified Bessel function of A (2nd kind, order B).
KEI	1	kei (A).
KER	1	ker (A).
LDIST	1	Compute distance from lines in multi-segment ASCII file A.
LE	2	1 if A <= B, else 0.
LMSSCL	1	LMS scale estimate (LMS STD) of A.
LOG	1	log (A) (natural log).
LOG10	1	log10 (A) (base 10).
LOG1P	1	log (1+A) (accurate for small A).
LOG2	1	log2 (A) (base 2).
LOWER	1	The lowest (minimum) value of A.
LRAND	2	Laplace random noise with mean A and std. deviation B.
LT	2	1 if A < B, else 0.
MAD	1	Median Absolute Deviation (L1 STD) of A.
MAX	2	Maximum of A and B.
MEAN	1	Mean value of A.
MED	1	Median value of A.
MIN	2	Minimum of A and B.
MODE	1	Mode value (Least Median of Squares) of A.
MUL	2	A * B.
NAN	2	NaN if A == B, else A.
NEG	1	-A.
NEQ	2	1 if A != B, else 0.

NRAND	2	Normal, random values with mean A and std. deviation B.
OR	2	NaN if A or B == NaN, else A.
PDIST	1	Compute distance from points in ASCII file A.
PLM	3	Associated Legendre polynomial P(-1<A<+1) degree B order C.
POP	1	Delete top element from the stack.
POW	2	A ^ B.
R2	2	R2 = A^2 + B^2.
R2D	1	Convert Radians to Degrees.
RAND	2	Uniform random values between A and B.
RINT	1	rint (A) (nearest integer).
ROTX	2	Rotate A by the (constant) shift B in x-direction.
ROTY	2	Rotate A by the (constant) shift B in y-direction.
SAZ	2	Sperhical azimuth from grid nodes to stack x,y.
SDIST	2	Spherical (Great circle) distance (in degrees) between grid nodes and stack lon,lat (A, B).
SIGN	1	sign (+1 or -1) of A.
SIN	1	sin (A) (A in radians).
SINC	1	sinc (A) (sin (pi*A)/(pi*A)).
SIND	1	sin (A) (A in degrees).
SINH	1	sinh (A).
SQRT	1	sqrt (A).
STD	1	Standard deviation of A.
STEP	1	Heaviside step function: H(A).
STEPX	1	Heaviside step function in x: H(x-A).
STEPY	1	Heaviside step function in y: H(y-A).
SUB	2	A - B.
TAN	1	tan (A) (A in radians).
TAND	1	tan (A) (A in degrees).
TANH	1	tanh (A).
TCRIT	2	Critical value for Student's t-distribution, with alpha = A and n = B.
TDIST	2	Student's t-distribution A(t,n), with t = A, and n = B.
TN	2	Chebyshev polynomial Tn(-1<t<+1,n), with t = A, and n = B.
UPPER	1	The highest (maximum) value of A.
XOR	2	B if A == NaN, else A.
Y0	1	Bessel function of A (2nd kind, order 0).
Y1	1	Bessel function of A (2nd kind, order 1).
YLM	2	Re and Im normalized surface harmonics (degree A, order B).
YN	2	Bessel function of A (2nd kind, order B).
ZCRIT	1	Critical value for the normal-distribution, with alpha = A.
ZDIST	1	Cumulative normal-distribution C(x), with x = A.

SYMBOLS

The following symbols have special meaning:

PI	3.1415926...
E	2.7182818...
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

OPTIONS

–F Select pixel registration (used with **–R, –I**). [Default is grid registration].

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

BEWARE

(1) The operator **SDIST** calculates spherical distances bewteen 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.
(2) The operator **YLM** calculates the fully normalized spherical harmonics for degree L and order M for all positions in the grid, which is assumed to be in degrees. **YLM** returns 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 consequtive = file.grd calls.
(3) The operator **PLM** calculates the associated Legendre polynomial of degree L and order M, and its argument is the cosine of the colatitude which must satisfy -1 <= x <= +1. Unlike **YLM**, **PLM** is not normalized.
(4) All the derivatives are based on central finite differences, with natural boundary conditions.

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

EXAMPLES

To take log10 of the average of 2 files, use

grdmath file1.grd file2.grd **ADD** 0.5 **MUL LOG10 =** file3.grd

Given the file ages.grd, which holds seafloor ages in m.y., use the relation depth(in m) = 2500 + 350 * sqrt (age) to estimate normal seafloor depths:

grdmath ages.grd **SQRT** 350 **MUL** 2500 **ADD =** depths.grd

To find the angle a (in degrees) of the largest principal stress from the stress tensor given by the three files s_xx.grd s_yy.grd, and s_xy.grd from the relation tan (2*a) = 2 * s_xy / (s_xx - s_yy), use

grdmath 2 s_xy.grd **MUL** s_xx.grd s_yy.grd **SUB DIV ATAN2** 2 **DIV =** direction.grd

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:

grdmath –R0/360/-90/90 **–I**1 8 4 **YML** 1.1 **MUL EXCH** 0.4 **MUL ADD** = harm.grd

To extract the locations of local maxima that exceed 100 mGal in the file faa.grd:

grdmath faa.grd **DUP EXTREMA** 2 **EQ MUL DUP** 100 **GT MUL** 0 **NAN** = z.grd
grd2xyz z.grd **–S** > max.xyz

REMARKS

(1) 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).
(2) Piping of files are not allowed.
(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.

REFERENCES

Abramowitz, M., and I. A. Stegun, 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, vol. 55, Dover, New York.
Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, 1992, *Numerical Recipes*, 2nd edition, Cambridge Univ., New York.

SEE ALSO

GMT(l), *gmtmath*(l), *grd2xyz*(l), *grdedit*(l), *grdinfo*(l), *xyz2grd*(l)

NAME

grdpaste – Paste together two *.grd* files along a common edge.

SYNOPSIS

grdpaste *file_a.grd file_b.grd* **–G***outfile.grd* [**–V**]

DESCRIPTION

grdpaste will combine *file_a.grd* and *file_b.grd* into *outfile.grd* by pasting them together along their common edge. Files *file_a.grd* and *file_b.grd* 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.

file_a.grd
: One of two files to be pasted together.

file_b.grd
: The other of two files to be pasted together.

–G*outfile.grd*
: The name for the combined output.

OPTIONS

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

Suppose file_a.grd is 150E - 180E and 0 - 30N, and file_b.grd is 150E - 180E, -30S - 0, then you can make outfile.grd which will be 150 - 180 and -30S - 30N by:

grdpaste file_a.grd file_b.grd **–G**outfile.grd **–V**

SEE ALSO

GMT(l), *grdcut*(l), *grdinfo*(l), *grdsample*(l)

NAME

grdproject – Forward and Inverse map transformation of 2-D grd files

SYNOPSIS

grdproject *in_grdfile* **–J***parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–A**[**k|m|n|i|c|p**]] [**–C**[*dx/dy*]] [**–D***xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]]] [**–E***dpi*] [**–F**] [**–G***out_grdfile*] [**–I**] [**–Mc|i|m|p**] [**–N***nx/ny*] [**–S***search_radius*] [**–V**]

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. The new nodes are filled based on a simple weighted average of nearby points. Aliasing is avoided by using sensible values for the *search_radius*. 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.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

in_grdfile
2-D binary grd file to be transformed.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude).

OPTIONS

–A Force 1:1 scaling, i.e., output (or input, see **–I**) data are in actual projected meters. To specify other units, append **k** (km), **m** (mile),**n** (nautical mile), **i** (inch), **c** (cm), or **p** (points). Without **–A**, the output (or input, see **–I**) are in the units specified by **MEASURE_UNIT** (but see **–M**).

–C 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 Set the grid spacing for the new grid. Append **m** for minutes, **c** for seconds.

–E Set the resolution for the new grid in dots per inch.

–F Toggle between pixel and gridline registration [Default is same as input].

–G Specify the name of the output netCDF grd file.

–I Do the Inverse transformation, from rectangular to geographical.

–M Append **c**, **i**, or **m** to indicate that cm, inch, or meter should be the projected measure unit [Default is set by **MEASURE_UNIT** in .gmtdefaults4]. Cannot be used with **–A**.

–N Set the number of grid nodes in the new grid.

–S Set the search radius for the averaging procedure [Default avoids aliasing].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

EXAMPLES

To transform the geographical grid dbdb5.grd onto a pixel Mercator grid at 300 dpi, run

grdproject dbdb5.grd **–R**20/50/12/25 **–Jm**0.25**i –E**300 **–F –G**dbdb5_merc.grd

To inversely transform the file topo_tm.grd back onto a geographical grid, use

grdproject topo_tm.grd **–R**-80/-70/20/40 **–Jt**-75/1:500000 **–I –D**5**m –V –G**topo.grd

This assumes, of course, that the coordinates in topo_tm.grd were created with the same projection parameters.
To inversely transform the file topo_utm.grd (which is in UTM meters) back to a geographical grid we specify a one-to-one mapping with meter as the measure unit:

grdproject topo_utm.grd **–R**203/205/60/65 **–Ju**5/1:1 **–I –Mm –V –G**topo.grd

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 interrested in.

SEE ALSO

GMT(l), *gmtdefaults*(l), *mapproject*(l)

NAME

grdreformat – Converting between different grdfile formats.

SYNOPSIS

grdreformat *ingrdfile*[=*id*[*/scale/offset*[*/NaNvalue*]]] *outgrdfile*[=*id*[*/scale/offset*[*/NaNvalue*]]]
[**–N**] [**–R***west*/*east*/*south*/*north*[**r**]] [**–V**]

DESCRIPTION

grdreformat reads a grdfile 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.

ingrdfile

The grdfile to be read. Append format =*id* code if not a standard COARDS-compliant netCDF grdfile. 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.)

outgrdfile

The grdfile to be written. Append format =*id* code if not a standard COARDS-compliant netCDF grdfile. 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.)

OPTIONS

–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 *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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

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 depreciated. 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 automatically recognize any type of netCDF grdfile. These can be in either COARDS-compliant or pre-4.1 format, and contain floating-point or integer data. To read other types of grdfiles, append the =*id* suffix to the filename *ingrdfile*.

id	*GMT 4 netCDF standard formats*
nb	GMT netCDF format (byte) (COARDS-compliant)

ns	GMT netCDF format (short) (COARDS-compliant)
ni	GMT netCDF format (integer) (COARDS-compliant)
nf	GMT netCDF format (float) (COARDS-compliant)
nd	GMT netCDF format (double) (COARDS-compliant)

id	*GMT 3 netCDF legacy formats*
cb	GMT netCDF format (byte) (depreciated)
cs	GMT netCDF format (short) (depreciated)
ci	GMT netCDF format (integer) (depreciated)
cf	GMT netCDF format (float) (depreciated)
cd	GMT netCDF format (double) (depreciated)

id	*GMT native binary formats*
bm	GMT native, C-binary format (bit-mask)
bb	GMT native, C-binary format (byte)
bs	GMT native, C-binary format (short)
bi	GMT native, C-binary format (integer)
bf	GMT native, C-binary format (float)
bd	GMT native, C-binary format (double)

id	*Miscellaneous formats*
rb	SUN rasterfile format (8-bit standard)
rf	GEODAS grid format GRD98 (NGDC)
sf	Golden Software Surfer format 6 (float)
sd	Golden Software Surfer format 7 (double, read-only)

GMT STANDARD NETCDF FILES

The standard format used for gdrfiles 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 Appendix B.

Multi-variable grid files

By default, **GMT** programs will read the first 2-dimensional grid contained in a COARDS-compliant netCDF file. Alternatively, use *ingrdfile***?***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.

3-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: *ingrdfile***?***varname***[***layer***]**. Alternatively, specify the value associated with that layer using parentheses in stead of brackets: *ingridfile***?***varname***(***level***)**. 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.

NATIVE BINARY FILES

For binary native **GMT** files the size of the **GMT** grdheader block is *hsize* = 896 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). For header and grid details, see Appendix B.

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

EXAMPLES

To extract the second layer from a 3-dimensional grid named temp from a COARDS-compliant netCDF file climate.grd:

grdreformat climate.grd?temp[1] temp.grd **-V**

To create a 4-byte native floating point grid from the COARDS-compliant netCDF file data.grd:

grdreformat data.grd ras_data.b4=bf **–V**

To make a 2-byte short integer file, scale it by 10, subtract 32000, setting NaNs to -9999, do

grdreformat values.grd shorts.i2=bs/10/-32000/-9999 **–V**

To create a Sun standard 8-bit rasterfile for a subset of the data file image.grd, assuming the range in image.grd is 0-1 and we need 0-255, run

grdreformat image.grd **–R**-60/-40/-40/-30 image.ras8=rb/255/0 **–V**

To convert etopo2.grd to etopo2.i2 that can be used by **grdraster**, try

grdreformat etopo2.grd etopo2.i2=bs **–N –V**

SEE ALSO

GMT(l), *grdmath*(l)

NAME

grdsample – Resample a grd file onto a new grid

SYNOPSIS

grdsample *in_grdfile* **–G***out_grdfile* [**–F**] [**–I***xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]]] [**–L***flag*] [**–N***nx/ny*] [**–Q**[*value*]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–T**] [**–V**]

DESCRIPTION

grdsample reads a grdfile and interpolates it to create a new grdfile with either: a different registration (**–T**); or, a new grid-spacing (**–I**) or number of nodes (**–N**), and perhaps also a new sub-region (**–R**). Interpolation is bicubic [Default] or bilinear (**–Q**) and uses boundary conditions (**–L**). 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**.

in_grdfile
The name of the input 2-D binary grd file.

–G The name of the output grd file.

OPTIONS

No space between the option flag and the associated arguments.

–F Force pixel registration. [Default is grid registration].

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–L Boundary condition *flag* may be *x* or *y* or *xy* indicating data is periodic in range of x or y or both set by **–R**, 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).]

–Q Quick mode, use bilinear rather than bicubic interpolation [Default]. Optionally, append *value* in the 0 <= *value* <= 1 range. This parameter controls how close to nodes with NaN values the interpolation will go. E.g., a *value* of 0.5 will interpolate about 1/2-way from a non-NaN to a NaN node, whereas 0.1 will go about 90% of the way, etc. [Default is 1, which means none of the four nearby nodes may be NaN]. A *value* of 0 will just return the value of the nearest node instead of interpolating.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–T Translate between grid and pixel registration while keeping **–R** and **–I** the same; if input is grid-registered, output will be pixel-registered and vice-versa. The input file determines **–R**, and **–I** so no other options are necessary (except possibly **–L** or **–Q**).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

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 [**–Q**] 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.

EXAMPLES

To resample the 5 x 5 minute grid in hawaii_5by5_topo.grd onto a 1 minute grid:

grdsample hawaii_5by5_topo.grd **–I**1**m –G**hawaii_1by1_topo.grd

To translate the gridline-registered file surface.grd to pixel registration:

grdsample surface.grd **–T –G**pixel.grd

SEE ALSO

GMT(l), *grdedit*(l), *grdfft*(l), *grdfilter*(l)

NAME

grdtrack – Sampling of a 2-D grdfile along 1-D trackline (a sequence of x,y points)

SYNOPSIS

grdtrack *xyfile* **–G***grdfile* [**–H**[**i**][*nrec*]] [**–L***flag*] [**–M**[**i**|**o**][*flag*]] [**–Q**[*value*] [**–R***west*/*east*/*south*/*north*[**r**]] [**–S**] [**–V**] [**–Z**] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

grdtrack reads a grdfile (or a Sandwell/Smith IMG file) and a table (from file or standard input) with (x,y) positions in the first two columns (more columns may be present). It interpolates the grid at the positions in the table and writes out the table with the interpolated values added as a new column. A bicubic [Default] or bilinear [**–Q**] interpolation is used, requiring boundary conditions at the limits of the region (see **–L**).

xyfile This is an ASCII [or binary, see **–b**] file where the first 2 columns hold the (x,y) positions where the user wants to sample the 2-D data set.

–G *grdfile* is a 2-D binary grd 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 [72.0059773539].

OPTIONS

No space between the option flag and the associated arguments.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–L Boundary condition *flag* may be *x* or *y* or *xy* indicating data is periodic in range of x or y or both set by **–R**, 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).]

–M Multiple segment file. Segment separator is a record beginning with *flag*. [Default is '>'].

–Q Quick mode, use bilinear rather than bicubic interpolation [Default]. Optionally, append *value* in the 0 <= *value* <= 1 range. This parameter controls how close to nodes with NaN values the interpolation will go. E.g., a *value* of 0.5 will interpolate about 1/2-way from a non-NaN to a NaN node, whereas 0.1 will go about 90% of the way, etc. [Default is 1, which means none of the four nearby nodes may be NaN]. A *value* of 0 will just return the value of the nearest node instead of interpolating.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–S Suppress the output of interpolated points that result in NaN values.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–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 Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is one more than input].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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 [**–Q**] 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.

EXAMPLES

To sample the file hawaii_topo.grd 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.grd –H > track_4.xygt
```

To sample the Sandwell/Smith IMG format file topo.8.2.img (2 minute predicted bathymetry on a Mercator grid) along the lon,lat coordinates given in the file cruise_track.xy, try

```
grdtrack cruise_track.xy –Gtopo.8.2.img,1,1 > obs_and_predicted.d
```

SEE ALSO

GMT(l), *surface*(l), *sample1d*(l)

NAME

grdtrend – Fit and/or remove a polynomial trend in a grd file

SYNOPSIS

grdtrend *grdfile* **–N***n_model*[**r**] [**–D***diff.grd*] [**–T***trend.grd*] [**–V**] [**–W***weight.grd*]

DESCRIPTION

grdtrend reads a 2-D gridded file and fits a low-order polynomial trend to these data by [optionally weighted] least-squares. The trend surface is defined by:

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 **–N***n_model*, the number of model parameters to use; thus, **–N***4* fits a bilinear trend, **–N***6* 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.

No space between the option flag and the associated arguments.

grdfile The name of a 2-D binary grd file.

–N [**r**]*n_model* sets the number of model parameters to fit. Append **r** for robust fit.

OPTIONS

No space between the option flag and the associated arguments.

–D Write the difference (input data - trend) to the file *diff.grd*.

–T Write the fitted trend to the file *trend.grd*.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W If *weight.grd* 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.grd*.

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.

EXAMPLES

To remove a planar trend from hawaii_topo.grd and write result in hawaii_residual.grd:

grdtrend hawaii_topo.grd **–N**3 **–D**hawaii_residual.grd

To do a robust fit of a bicubic surface to hawaii_topo.grd, writing the result in hawaii_trend.grd and the weights used in hawaii_weight.grd, and reporting the progress:

grdtrend hawaii_topo.grd **–N**10**r** **–T**hawaii_trend.grd **–W**hawaii_weight.grd **–V**

SEE ALSO

GMT(l), *grdfft*(l), *grdfilter*(l)

NAME

grdvector – Plot vector fields from grdfiles

SYNOPSIS

grdvector *compx.grd compy.grd* **–J***parameters* [**–A**] [**–B**[**p**|**s**]*parameters*] [**–C***cptfile*] [**–E**] [**–G***fill*] [**–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]]] [**–K**] [**–N**] [**–O**] [**–P**] [**–Q***parameters*] [**–R***west*/*east*/*south*/*north*[**r**]] [**–S**[**l**]*scale*] [**–T**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–W***contourpen*] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z**] [**–c***copies*]

DESCRIPTION

grdvector reads two 2-D gridded 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 shorthand for using 2 calls to **grd2xyz** and pasting the output through **psxy –SV**.

compx.grd
Contains the x-component of the vector field.

compy.grd
Contains the y-component of the vector field.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

OPTIONS

No space between the option flag and the associated arguments.

–A Means grdfiles have polar (r, theta) components instead of Cartesian (x, y).

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Use *cptfile* to assign colors based on vector length.

–E Center vectors on grid nodes [Default draws from grid node].

–G Sets color or shade for vector interiors [Default is no fill]. Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–I Only plot vectors at nodes every *x_inc, y_inc* apart (must be multiples of original grid spacing). Append **m** for minutes or **c** for seconds. [Default plots every node].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–N Do NOT clip vectors at map boundaries [Default will clip].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Select vector plot [Default is stick-plot]. Optionally, specify *parameters* which are *arrowwidth/headlength/headwidth* [Default is 0.075**c**/0.3**c**/0.25**c** (or 0.03**i**/0.12**i**/0.1**i**)]. Append n*size* which will cause vectors shorter than *size* to have their appearance scaled by length/*size*.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible). Specify a subset of the grid.

–S Sets scale for vector length in data units per distance measurement unit [1]. Append **c**, **i**, **m**, **p** to indicate the measurement unit (cm, inch, m, point). Prepend **l** to indicate a fixed length for all vectors.

–T Means azimuth should be converted to angles based on the selected map projection.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Set pen attributes used for vector outlines [Default: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes ‘-’ and dots ‘.’.

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z Means the angles provided are azimuths rather than direction (requires **–A**).

–c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To draw the vector field given by the files r.grd and theta.grd on a linear plot with scale 5 cm per data unit, using vector rather than stick plot, and scale vector magnitudes so that 10 units equal 1 inch, run

```
grdvector r.grd theta.grd –Jx5c –A –Q –S10i > gradient.ps
```

SEE ALSO

GMT(1), *grdcontour*(1), *psxy*(1)

NAME

grdview − Create 3-D perspective grayshaded/colored image or mesh from a 2-D grd file

SYNOPSIS

grdview *relief_file* **−J***parameters* [**−B**[**p**|**s**]*parameters*] [**−C***cptfile*] [**−E***view_az/view_el*] [**−G***drapefile* | **−G***grd_r,grd_g,grd_b*] [**−I***intensfile*] [**−Jz**|**Z***parameters*] [**−K**] [**−L**[*flags*]] [**−N***level*[*/color*]] [**−O**] [**−P**] [**−Q***type*[**g**]] [**−R***west*/*east*/*south*/*north*[*/zmin*/*zmax*][**r**]] [**−S***smooth*] [**−T**[**s**][**o**[*pen*]]] [**−U**[*/dx*/*dy*/][*label*]] [**−V**] [**−W***type/pen*] [**−X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**−Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**−Z***zlevel*] [**−c***copies*]

DESCRIPTION

grdview reads a 2-D gridded 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 rasterimage. 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 grd file.

relief_file
2-D gridded data set to be imaged (the relief of the surface).

−J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

−Jc*lon0/lat0/scale* (Cassini)
−Jj*lon0/scale* (Miller)
−Jm*scale* (Mercator - Greenwich and Equator as origin)
−Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
−Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
−Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
−Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
−Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
−Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
−Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
−Ju*zone/scale* (UTM - Universal Transverse Mercator)
−Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

−Ja*lon0/lat0/scale* (Lambert)
−Je*lon0/lat0/scale* (Equidistant)
−Jf*lon0/lat0/horizon/scale* (Gnomonic)
−Jg*lon0/lat0/scale* (Orthographic)
−Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

−Jb*lon0/lat0/lat1/lat2/scale* (Albers)
−Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
−Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

−Jh*lon0/scale* (Hammer)

–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C 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**|**p***dpi/pattern*[:**F***r/g/b*[**B***r/g/b*]].

–E Sets the view point by specifying azimuth and elevation in degrees. [Default is 180/90].

–G 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 gridfiles 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*.

–I Gives the name of a grdfile with intensities in the (-1,+1) range. [Default is no illumination].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Boundary condition *flags* may be *x* or *y* or *xy* indicating data is periodic in range of x or y or both, or *flags* 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).] If no *flags* are set, use bilinear rather than the default bicubic resampling when draping is required.

–N Draws a plane at this z-level. If the optional r/g/b is provided, the frontal facade between the plane and the data perimeter is colored.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Select one of four settings: 1. Specify **m** for mesh plot [Default], and optionally append /r/g/b 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 mask out nodes with z = NaN 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 **g**. Colors are then converted to shades of gray using the (television) YIQ transformation.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible). 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.

–S Smooth the contours before plotting (see **grdcontour**) [Default is no smoothing].

–T 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.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–Wc Draw contour lines on top of surface or mesh (not image). Append pen attributes used for the contours. [Default: width = 3, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–Wm Sets the pen attributes used for the mesh. [Default: width = 1, color = black, texture = solid]. You must also select **–Qm** or **–Qsm** for meshlines to be drawn.

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z Sets the z-level of the basemap [0].

–c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To make a mesh plot from the file hawaii_grav.grd 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

grdview hawaii_grav.grd **–Jl**18/24/1.5**c** **–C**hawaii.cpt **–Jz**0.05**c** **–Qm** **–N**-100 **–E**225/30 **–Wc** > hawaii_grav_image.ps

To create a illuminated color perspective plot of the gridded data set image.grd, 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.grd, and looking from the SE, use

grdview image.grd **–Jx**10.0**c** **–C**color.rgb **–Qs** **–E**135/30 **–I**intens.grd > image3D.ps

To make the same plot using the rastering option with dpi = 50, use

grdview image.grd **–Jx**10.0**c** **–C**color.rgb **–Qi**50 **–E**135/30 **–I**intens.grd > image3D.ps

To create a color *PostScript* perspective plot of the gridded data set magnetics.grd, using the color palette file mag_intens.cpt, draped over the relief given by the file topography.grd, with Mercator map width of 6 inch and tickmarks every 1 degree, with intensities provided by the file topo_intens.grd, and looking from the SE, run

grdview topography.grd **–JM6i** **–G**magnetics.grd **–C**mag_intens.cpt **–Qs** **–E**140/30 **–I**topo_intens.grd > draped3D.ps

Given topo.grd and the Landsat image veggies.ras, first run **gmt2rgb** 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

gmt2rgb veggies.ras **–G**layer_%c.grd
grdview topo.grd **–JM6i** **–Qi** **–E**140/30 **–I**topo_intens.grd **–G**layer_r.grd,layer_g.grd,layer_b.grd > image.ps

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 grdfile(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.

SEE ALSO

GMT(l), *gmt2rgb*(l), *grdcontour*(l), *grdimage*(l), *nearneighbor*(l), *psbasemap*(l), *pscontour*(l), *pstext*(l), *surface*(l)

NAME

grdvolume – Calculating volume under a surface within a contour

SYNOPSIS

grdvolume *grdfile* [**–C***cval* or **–C***low/high/delta*] [**–L***base*] [**–R***west*/*east*/*south*/*north*[**r**]] [**–S**[**k**]] [**–T**] [**–V**[**l**]] [**–Z***fact*[/*delta*]]

DESCRIPTION

grdvolume reads a 2-D binary grd 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. The contour that produced the maximum mean height is reported as well. This feature may be used with **grdfilter** in designing an Optimal Robust Separator [*Wessel*, 1998].

grdfile The name of the input 2-D binary grd file.

OPTIONS

No space between the option flag and the associated arguments.

–C find area and volume inside the *cval* contour. Alternatively, search using all contours from *low* to *high* in steps of *delta*. [Default returns entire area and volume of grid]. The area is measured in the plane of the countour.

–L Also add in the volume from the level of the contour down to *base* [Default base is contour].

–S Convert degrees to meters, append **k** for km [Default is Cartesian].

–T Use curvature minimum rather than maximum height to find best contour value (when contour search is selected with **–C**).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"]. Append **l** to see all the results for each contour level tested (when contour search has been selected).

–Z Optionally subtract *shift* before scaling data by *fact*. [Default is no scaling]. (Numbers in **–C, -L** refer to values after this scaling has occurred).

EXAMPLES

To determine the volume in km^3 under the surface hawaii_topo.grd (height in km), use

grdvolume hawaii_topo.grd **–Sk**

To find the volume between the surface peaks.grd and the contour z = 250, use

grdvolume peaks.grd **–Sk –C**250

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.grd, use

grdvolume peaks.grd **–Sk –C**100/300/10 > results.d

To see the areas and volumes for all the contours in the previous example, use

grdvolume peaks.grd **–Sk –Vl –C**100/300/10 > results.d

SEE ALSO

GMT(l), *grdfilter*(l)

REFERENCES

Wessel, P., 1998, An empirical method for optimal robust regional-residual separation of geophysical data, *Math. Geol., 30(4)*, 391–408.

NAME

makecpt – Make GMT color palette tables

SYNOPSIS

makecpt [**–C***table*] [**–D**] [**–I**] [**–M**] [**–N**] [**–Q[i|o]**] [**–T***z0/z1/dz* | **–T***ztable*] [**–V**] [**–Z**]

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.

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 whereever 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 .gmtdefaults4 file or the command line will be used. This default behavior can be overruled using the options **–D**, **–M** or **–N**.

OPTIONS

–C 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 cptfile [Default gives a rainbow cpt file].

–D Select the colors for lowest and highest *z*-values in the output cpt file as the back- and foreground colors that will be written to the cpt file [Default uses the colors specified in the master file, or those defined by the parameters **COLOR_BACKGROUND**, **COLOR_FOREGROUND**, and **COLOR_NAN**].

–I Reverses the sense of color progression in the master cptfile. Also exchanges the foreground and background colors, including those specified by the parameters **COLOR_BACKGROUND** and **COLOR_FOREGROUND**.

–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 .gmtdefaults4 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 Selects a logarithmic interpolation scheme [Default is linear]. **–Qi** expects input z-values to be log10(z), assigns colors, and writes out z [Default]. **–Qo** takes log10(z) first, assigns colors, and writes out z.

–T Defines the range of the new cptfile by giving the lowest and highest z-value and the interval. Alternatively, give the name of a ASCII file that has one z-value per record. If not given, the existing range in the master cptfile will be used intact.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–Z Creates a continuous cpt file [Default is discontinuous, i.e. constant colors for each interval].

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:

makecpt –C*polar* **–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:

makecpt –T-2/6/1 **–Z** > rainbow.cpt

To make a GEBCO look-alike cpt file for bathymetry, run

```
makecpt −Cgebco > my_gebco.cpt
```

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.

SEE ALSO

GMT(l), *grd2cpt*(l)

NAME

mapproject – Forward and Inverse map transformation of 2-D coordinates

SYNOPSIS

mapproject *infiles* **–J***parameters* **–R***west/east/south/north*[**r**] [**–Ab|B|f|F***lon0/lat0*] [**–C**[*dx/dy*]] [**–Dc|i|m|p**] [**–E**[*datum*]] [**–F**[**k|m|n|i|c|p**]] [**–G**[*x0/y0*][*/unit*]] [**–H**[**i**][*nrec*]] [**–I**] [**–L**line.xy**[/***unit***]] [–M[i|o][***flag***]] [–Q[d|e] [–S] [–T[h]***from***[/***to***]] [–V] [–:[i|o]] [–b[i|o][s|S|d|D][***ncol***]] [–f[i|o]***colinfo*]

DESCRIPTION

mapproject reads (longitude, latitude) positions from *infiles* [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.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

infiles Data file(s) to be transformed. If not given, standard input is read.

–J Selects the 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 m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/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].
The ellipsoid used in the map projections is user-definable by editing the .gmtdefaults4 file in your home directory. 63 commonly used ellipsoids and a spheroid are currently supported, and users may also specify their own ellipsoid parameters (see man gmtdefaults for more details). **GMT** default is WGS-84. Several GMT parameters can affect the projection: **ELLIPSOID**, **INTERPOLANT**, **MAP_SCALE_FACTOR**, and **MEASURE_UNIT**; see the **gmtdefaults** 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:

–Jc*lon0/lat0/scale* or **–JC***lon0/lat0/width* (Cassini).
Give projection center and scale (1:xxxx or UNIT/degree).

–Jc*lon0/lat0/scale* or **–JC***lon0/lat0/width* (Cassini).
Give projection center and scale (1:xxxx or UNIT/degree).

–Jj*lon0/scale* or **–JJ***lon0/width* (Miller Cylindrical Projection).
Give the central meridian and scale (1:xxxx or UNIT/degree).

–Jm*parameters* (Mercator **[C]**).
Specify one of:

–Jm*scale* or **–JM***width*
Give scale along equator (1:xxxx or UNIT/degree).

–Jm*lon0/lat0/scale* or **–JM***lon0/lat0/width*
Give central meridian, standard latitude and scale along parallel (1:xxxx or UNIT/degree).

–Jo*parameters* (Oblique Mercator **[C]**).
Specify one of:

–Joa*lon0/lat0/azimuth/scale* or **–JOa***lon0/lat0/azimuth/width*
Set projection center, azimuth of oblique equator, and scale.

–Job*lon0/lat0/lon1/lat1/scale* or **–JOb***lon0/lat0/lon1/lat1/scale*
Set projection center, another point on the oblique equator, and scale.

–Joc*lon0/lat0/lonp/latp/scale* or **–JOc***lon0/lat0/lonp/latp/scale*
Set projection center, pole of oblique projection, and scale.

Give scale along oblique equator (1:xxxx or UNIT/degree).

–Jq*lon0/scale* or **–JQ***lon0/width* (Equidistant Cylindrical Projection (Plate Carree)).
Give the central meridian and scale (1:xxxx or UNIT/degree).

–Jt*parameters* (Transverse Mercator **[C]**).
Specify one of:

–Jt*lon0/scale* or **–JT***lon0/width*
Give the central meridian and scale (1:xxxx or UNIT/degree).

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

–Ju*zone/scale* or **–JU***zone/width* (UTM - Universal Transverse Mercator **[C]**).
Give the zone number (1-60) and scale (1:xxxx or UNIT/degree).
Zones: prepend - or + to enforce southern or northern hemisphere conventions [northern if south > 0].

–Jy*lon0/lats/scale* or **–JY***lon0/lats/width* (Basic Cylindrical Projections **[E]**).
Give the central meridian, standard parallel, and scale (1:xxxx or UNIT/degree). The standard parallel is typically one of these (but can be any value):
45 - The Peters projection
37.4 - The Trystan Edwards projection
30 - The Behrman projection
0 - The Lambert projection

AZIMUTHAL PROJECTIONS:

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

–Ja*lon0/lat0/scale* or **–JA***lon0/lat0/width* (Lambert **[E]**).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Je*lon0/lat0/scale* or **–JE***lon0/lat0/width* (Equidistant).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Jf*lon0/lat0/horizon/scale* or **–JF***lon0/lat0/horizon/width* (Gnomonic).
lon0/lat0 specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, < 90). Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Jg*lon0/lat0/scale* or **–JG***lon0/lat0/width* (Orthographic).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Js*lon0/lat0/scale* or **–JS***lon0/lat0/width* (General Stereographic **[C]**).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx (true at pole) or *slat*/1:xxxx (true at standard parallel *slat*) or *radius/lat* (*radius* in UNIT from origin to the oblique latitude *lat*).

CONIC PROJECTIONS:

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

–Jd*lon0/lat0/lat1/lat2/scale* or **–JD***lon0/lat0/lat1/lat2/width* (Equidistant)
Give projection center, two standard parallels, and scale (1:xxxx or UNIT/degree).

–Jl*lon0/lat0/lat1/lat2/scale* or **–JL***lon0/lat0/lat1/lat2/width* (Lambert **[C]**)
Give origin, 2 standard parallels, and scale along these (1:xxxx or UNIT/degree).

MISCELLANEOUS PROJECTIONS:

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

–Ji*lon0/scale* or **–JI***lon0/width* (Sinusoidal **[E]**).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jk[**f**|**s**]*lon0/scale* or **–JK**[**f**|**s**]*lon0/width* (Eckert IV (f) and VI (s) **[E]**).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jn*lon0/scale* or **–JN***lon0/width* (Robinson).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jr*lon0/scale* **–JR***lon0/width* (Winkel Tripel).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jv*lon0/scale* or **–JV***lon0/width* (Van der Grinten).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jw*lon0/scale* or **–JW***lon0/width* (Mollweide **[E]**).
Give the central meridian 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.

–Jx*x-scale*[*/y-scale*] or **–JX***width*[*/height*] (Linear, log, and power scaling)
Give *x-scale* in UNIT/x-unit and/or *y-scale* in UNIT/y-unit; or specify *width* and/or *height* in UNIT (*y-scale*=*x-scale* if not specified separately). Use negative scale(s) to reverse the direction of an axis (e.g., to have y be positive down). 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.

p*power* 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 gmtdefaults) can be invoked using **–JXh** (for landscape); **–JXv** (for portrait) will swap the x- and y-axis lengths. The **GMT** default unit for this installation is UNIT. However, you may change this by editing your .gmtdefaults4 file(s) (run gmtdefaults to create one if you do not have it).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

infile(s) input file(s) with 2 or more columns. If no file(s) is given, **mapproject** will read the standard input.

–A[**f|b**] **–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. Upper case **F** or **B** will convert from geodetic to geocentric latitudes and estimate azimuth of geodesics (assuming the current ellipsoid is not a sphere).

–C 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 **MEASURE_UNIT**) unless **–F** is used, in which case the offsets are in meters.

–D Temporarily override **MEASURE_UNIT** and use **c** (cm), **i** (inch), **m** (meter), or **p** (points) instead. Cannot be used with **–F**.

–E 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,1/f*. If *datum* is - or not given we assume WGS-84.

–F Force 1:1 scaling, i.e., output (or input, see **–I**) data are in actual projected meters. To specify other units, append **k** (km), **m** (mile), **n** (nautical mile), **i** (inch), **c** (cm), or **p** (points). Without **–F**, the output (or input, see **–I**) are in the units specified by **MEASURE_UNIT** (but see **–D**).

–G Calculate distances along track OR to the optional point set with **–G***x0/y0*. Append the distance unit; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), **d** (spherical degree), **c** (Cartesian distance using input coordinates) or **C** (Cartesian distance using projected coordinates). The last unit requires **–R** and **–J** to be set. Upper case **E**, **K**, **M**, **N**, or **D** will use exact methods for geodesic distances (Rudoe's method for distances in length units and emplying geocentric latitudes in degree calculations, assuming the current ellipsoid is not a sphere).

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Do the Inverse transformation, i.e. get (longitude,latitude) from (x,y) data.

–L 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; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), **d** (spherical degree), **c** (Cartesian distance using input coordinates) or **C** (Cartesian distance using projected coordinates). The last unit requires **–R** and **–J** to be set. A spherical approximation is used for geographic data.

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–Q 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 Coordinate conversions between datums *from* and *to*. Use **−Th** if 3rd input column has height above ellipsoid [Default assumes height = 0, i.e., on the ellipsoid]. Specify datums using the ID (see **−Qd**) or give *ellipsoid*:*dx,dy,dz*, where *ellipsoid* may be an ellipsoid ID (see **−Qe**) or given as *a,1/f*. If *datum* is - or not given we use 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 **ELLIPSOID** setting is correct for your case.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

−bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f**[**i**|**o**]**g** means **−f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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

mapproject lonlatfile **−R**20/50/12/25 **−Jm**0.5**c** > 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

mapproject tracks.* **−R**-80/-70/20/40 **−Jt**-75/1:500000 **−: −S −Di −bo −bi**2 > 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

mapproject old.dat **−Th**131 > 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 multi-segment ASCII file coastline.xy, run

mapproject quakes.dat **−L**coastline.xy/k > quake_dist.dat

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.

ELLIPSOIDS AND SPHEROIDS

GMT will use ellipsoidal formulae if they are implemented and the user have selected an ellipsoid as the reference shape (see **ELLIPSOID** in **gmtdefaults**). The user needs to be aware of a few potential pitfalls: (1) For some projections, such as Transverse Mercator, Albers, and Lamberts 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 **MAP_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.

SEE ALSO

gmtdefaults(l), *GMT*(l), *project*(l)

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.

NAME

minmax – Find extreme values in data tables

SYNOPSIS

minmax [*files*] [**–C**] [**–EL|l|H|h***col*] [**–H**[**i**][*nrec*]] [**–I***dx*[*/dy*[*/dz...*]] [**–M**[*flag*]] [**–T***dz*] [**–:**[**i|o**]] [**–bi**[**s|S|d|D**][*ncol*]] [**–f**[**i|o**]*colinfo* **]**

DESCRIPTION

minmax 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, **minmax** 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 **–R***w/e/s/n* which can be used directly in the command line for other programs (hence only *dx* and *dy* are neede), 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***zmin/zmax/dz* string for makecpt.

xyzfile ASCII [or binary, see **–b**] file(s) holding a fixed number of data columns.

OPTIONS

–C Report the min/max values per column in separate columns [Default uses <min/max> format].

–E 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.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I 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 behaviour, use **–Ip***dx***.**

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–T Report the min/max of the first column to the nearest multiple of *dz* and output this in the form **–T***zmin/zmax/dz*.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both]. Only works when **–I** is selected.

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i|o**]**g** means **–f**[**i|o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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

D_FORMAT setting.

EXAMPLES

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

```
minmax 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 and use this region to draw a line using psxy, run

```
psxy ‘minmax –I5 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, use

```
minmax junkfile –C –I1/1/1/1
```

SEE ALSO

GMT(l)

NAME

nearneighbor – A "Nearest neighbor" gridding algorithm

SYNOPSIS

nearneighbor [*xyzfile(s)*] **–G***out_grdfile* **–I***xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]] **–N***sectors* **–R***west*/*east*/*south*/*north*[**r**] **–S***search_radius*[**m**|**c**|**k**|**K**] [**–E***empty*] [**–F**] [**–H**[**i**][*nrec*]] [**–L***flag*] [**–V**] [**–W**] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

DESCRIPTION

nearneighbor reads arbitrarily located (x,y,z[,w]) triples [quadruplets] from standard input [or *xyzfile(s)*] 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 ˆ 2), where d = 3 * r / search_radius and r is distance from the node. This weight is modulated by the observation points' weights [if supplied].

xyzfile(s)
3 [or 4, see **–W**] column ASCII file(s) [or binary, see **–b**] holding (x,y,z[,w]) data values. If no file is specified, **nearneighbor** will read from standard input.

–G Give the name of the output grdfile.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–N The circular area centered on each node is divided into several sectors. Average values will only be computed if there is at least one value inside each of the sectors for a given node. Nodes that fail this test are assigned the value NaN (but see **–E**). [Default is quadrant search, i.e., *sectors* = 4]. Note that only the nearest value per sector enters into the averaging, not all values inside the circle.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–S Sets the *search_radius* in same units as the grid spacing; append **m** to indicate minutes or **c** to indicate seconds. Append **k** to indicate km (implies **–R** and **–I** are in degrees, and we will use a fast flat Earth approximation to calculate distance). For more accuracy, use uppercase **K** if distances should be calculated along geodesics. However, if the current **ELLIPSOID** is set to Sphere then spherical great circle calculations are used.

OPTIONS

–E Set the value assigned to empty nodes [NaN].

–F Force pixel registration. [Default is grid registration].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

–L Boundary condition *flag* may be *x* or *y* or *xy* indicating data is periodic in range of x or y or both set by **–R**, or *flag* may be *g* indicating geographical conditions (x and y are lon and lat). [Default is no boundary conditions].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–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 Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 (or 4 if **–W** is set) columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

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, and set empty nodes to -9999:

nearneighbor seaMARCII_bathy.lon_lat_z **–R**242/244/-22/-20 **–I**.5**m** **–E**-9999 **–G**bathymetry.grd **–S**5k **–N**8

To make a global gridded 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 empty nodes to NaN, run

nearneighbor geoid.xyz **–R**0/360/-90/90 **–I**1 **–L**g **–G**geoid.grd **–S**20K **–N**4

SEE ALSO

blockmean(l), *blockmedian*(l), *blockmode*(l), *GMT*(l), *surface*(l), *triangulate*(l)

NAME

project – project data along a line or great circle, generate a profile track, or translate coordinates.

SYNOPSIS

project [*infile*] **–C***cx/cy* [**–A***azimuth*] [**–Dd|g**] [**–E***bx/by*] [**–F***flags*] [**–G***dist*] [**–H**[**i**][*nrec*]] [**–L**[**w**][*l_min/l_max*]] [**–M**[**i|o**][*flag*]] [**–N**] [**–Q**] [**–S**] [**–T***px/py*] [**–V**] [**–W***w_min/w_max*] [**–:**[**i|o**]] [**–b**[**i|o**][**s|S|d|D**][*ncol*]] [**–f**[**i|o**]*colinfo* **]**

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**.

–C *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.

OPTIONS

infile name of ASCII (or binary, see **–bi**) file(s) with 2 or more columns holding (x,y,[z]) data values. If no filenames are given, **project** will read from standard input. If the **–G** option is selected, no input data are read.

–F 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*.

–A *azimuth* defines the azimuth of the projection (Definition 1).

–D Set the location of the Discontinuity in longitude (*r* coordinate). **–Dd** will place the discontinuity at the Dateline, (-180 < *r* < 180); **–Dg** will place it at Greenwich, (0 < *r* < 360). Default usually falls at dateline due to *atan2* calls.

–E *bx/by* defines the end point of the projection path (Definition 2).

–G *dist* Generate mode. No input is read. Create (*r, s, p*) output points every *dist* units of *p*. See **–Q** option.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–L 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**.

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–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.

–T *px/py* sets the position of the roTation pole of the projection. (Definition 3).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Width controls. Project only those points whose *q* coordinate is within *w_min* < *q* < *w_max*.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is given by **–F** or **–G**].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

To generate points every 10km along a great circle from 10N,50W to 30N,10W:

```
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 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:

project c2610.xygmb **–C**-30/-30 **–A**-20 **–W**-10000/0 **–L**-500/500 **–F**pz **–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:

project mydata.xy **–C**5/3 **–A**70 **–F**pq > 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:

project pacific.lonlat **–T**-75/68 **–C**-149:26/-17:37 **–F**pq > pacific.pq

Suppose that pacific_topo.grd is a grdfile of bathymetry, and you want to make a file of flowlines in the hotspot reference frame. If you run:

grd2xyz pacific_topo.grd | **project –T**-75/68 **–C**0/-90 **–F**xyq | **xyz2grd –R***etc* **–I***etc* **–C**flow.grd

then flow.grd is a file in the same area as pacific_topo.grd, but flow contains the latitudes about the pole of the projection. You now can use grdcontour on flow.grd 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.

SEE ALSO

fitcircle(l), *GMT*(l), *mapproject*(l), *grdproject*(l)

NAME

psbasemap – To plot *PostScript* basemaps

SYNOPSIS

psbasemap –B[**p**|**s**]*parameters* **–J***parameters* **–R***west*/*east*/*south*/*north*[/*zmin*/*zmax*][**r**] [**–E***azimuth*/*elevation*] [**–G***fill*] [**–Jz**|**Z***parameters*] [**–K**] [**–L**[**f**][**x**]*lon0*/*lat0*[/*slon*]/*slat*/*length*[**m**|**n**|**k**][:*label*:*just*][**+p***pen*][**+f***fill*]]] [**–O**] [**–P**] [**–U**[/*dx*/*dy*/][*label*]] [**–T**[**f**|**m**][**x**]*lon0*/*lat0*/*size*[/*info*][:w,e,s,n:][+*gint*[/*mint*]]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***zlevel*] [**–c***copies*]

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.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

–B Sets map boundary annotation and tickmark intervals. The format of *tickinfo* is [**p**|**s**]*xinfo*[/*yinfo*[/*zinfo*]][:."Title":][**W**|**w**][**E**|**e**][**S**|**s**][**N**|**n**][**Z**|**z**[+]]. The leading **p** [Default] or **s** selects the primary or secondary annotation information. Each of the *?info* segments are textstrings of the form *info*[:"Axis label":][:="prefix":][:,"unit label":]. The *info* string is made up of one or more concatenated substrings of the form [**which**]*stride*[+-*phase*][**u**]. The optional **which** can be either **a** for annotation tick spacing [Default], **f** for frame tick spacing, and **g** for gridline spacing. If frame interval is not set, it is assumed to be the same as annotation interval. *stride* is the desired stride interval. The optional *phase* shifts the annotation interval by that amount. The optional **u** 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 **PLOT_DATE_FORMAT**), **o** (month, plot with 2 digits), **U** (ISO week, plot using **PLOT_DATE_FORMAT**), **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 **PLOT_DATE_FORMAT**), **d** (day, plot day of month 0-31 or year 1-366, via **PLOT_DATE_FORMAT**), **R** (day, same as **d**, aligned with **TIME_WEEK_START**), **H** (hour, plot using **PLOT_CLOCK_FORMAT**), **h** (hour, plot with 2 digits), **M** (minute, plot using **PLOT_CLOCK_FORMAT**), **m** (minute, plot with 2 digits), **C** (second, plot using **PLOT_CLOCK_FORMAT**), **c** (second, plot with 2 digits). Note for geographic axes **m** and **c** instead mean arc minutes and arc seconds. All entities that are language-specific are under control by **TIME_LANGUAGE**. To specify separate x and y ticks, separate the substrings that apply to the x and y axes with a slash [/] (If a 3-D basemap is selected with **–E** and **–Jz**, a third substring pertaining to the vertical axis may be appended.) For linear/log/power projections (**–Jx**|**X**): Labels for each axis can be added by surrounding them with colons. If the first character in the label is a period, then the label is used as plot title; if it is a comma then the label is appended to each annotation; if it is an equal sign (=) the the prefix is prepended to each annotation (start label/prefix with - to avoid space between annotation and item); else it is the axis label. If the label consists of more than one word, enclose the entire label in double quotes (e.g., :"my label":).

By default, all 4 boundaries are plotted (referred to as **W, E, S, N**). To change the default, append the code for only those axes you want (e.g., **WS** for standard lower-left x- and y-axis system). Upper case (e.g., **W**) means draw axis/tickmarks AND annotate it, whereas lower case (e.g., **w**) will only draw axis/tickmarks. (If a 3-D basemap is selected with **–E** and **–Jz**, append **Z** or **z** to control the appearance of the vertical axis. Append '+' to draw the outline of the cube defined by **–R**. Note that for 3-D views the title, if given, will be suppressed.)

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, or 3. Annotations will then occur at 1, 1–2–5, or 1–2–3–4–...–9, respectively. 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: **ANNOT_MIN_ANGLE**, **ANNOT_MIN_SPACING**, **ANNOT_FONT_PRIMARY**, **ANNOT_FONT_SECONDARY**, **ANNOT_FONT_SIZE_PRIMARY**, **ANNOT_FONT_SIZE_SECONDARY**, **ANNOT_OFFSET_PRIMARY**, **ANNOT_OFFSET_SECONDARY**, **BASEMAP_AXES**, **BASEMAP_FRAME_RGB**, **BASEMAP_TYPE**, **DEGREE_FORMAT**, **FRAME_PEN**, **FRAME_WIDTH**, **GRID_CROSS_SIZE_PRIMARY**, **GRID_PEN_PRIMARY**, **GRID_CROSS_SIZE_SECONDARY**, **GRID_PEN_SECONDARY**, **HEADER_FONT**, **HEADER_FONT_SIZE**, **LABEL_FONT**, **LABEL_FONT_SIZE**, **LINE_STEP**, **OBLIQUE_ANNOTATION**, **PLOT_CLOCK_FORMAT**, **PLOT_DATE_FORMAT**, **TIME_FORMAT_PRIMARY**, **TIME_FORMAT_SECONDARY**, **TIME_LANGUAGE**, **TIME_WEEK_START**, **TICK_LENGTH**, **TICK_PEN**, and **Y_AXIS_TYPE**; see the **gmtdefaults** man page for details.

−J Selects the 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 m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/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].

The ellipsoid used in the map projections is user-definable by editing the .gmtdefaults4 file in your home directory. 63 commonly used ellipsoids and a spheroid are currently supported, and users may also specify their own ellipsoid parameters (see man gmtdefaults for more details). **GMT** default is WGS-84. Several GMT parameters can affect the projection: **ELLIPSOID**, **INTERPOLANT**, **MAP_SCALE_FACTOR**, and **MEASURE_UNIT**; see the **gmtdefaults** 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:

−Jc*lon0/lat0/scale* or **−JC***lon0/lat0/width* (Cassini).
Give projection center and scale (1:xxxx or UNIT/degree).

−Jc*lon0/lat0/scale* or **−JC***lon0/lat0/width* (Cassini).
Give projection center and scale (1:xxxx or UNIT/degree).

−Jj*lon0/scale* or **−JJ***lon0/width* (Miller Cylindrical Projection).
Give the central meridian and scale (1:xxxx or UNIT/degree).

−Jm*parameters* (Mercator **[C]**).
Specify one of:

−Jm*scale* or **−JM***width*
Give scale along equator (1:xxxx or UNIT/degree).

−Jm*lon0/lat0/scale* or **−JM***lon0/lat0/width*
Give central meridian, standard latitude and scale along parallel (1:xxxx or UNIT/degree).

−Jo*parameters* (Oblique Mercator **[C]**).
Specify one of:

–Joa*lon0/lat0/azimuth/scale* or **–JOa***lon0/lat0/azimuth/width*
Set projection center, azimuth of oblique equator, and scale.

–Job*lon0/lat0/lon1/lat1/scale* or **–JOb***lon0/lat0/lon1/lat1/scale*
Set projection center, another point on the oblique equator, and scale.

–Joc*lon0/lat0/lonp/latp/scale* or **–JOc***lon0/lat0/lonp/latp/scale*
Set projection center, pole of oblique projection, and scale.

Give scale along oblique equator (1:xxxx or UNIT/degree).

–Jq*lon0/scale* or **–JQ***lon0/width* (Equidistant Cylindrical Projection (Plate Carree)).
Give the central meridian and scale (1:xxxx or UNIT/degree).

–Jt*parameters* (Transverse Mercator **[C]**).
Specify one of:

–Jt*lon0/scale* or **–JT***lon0/width*
Give the central meridian and scale (1:xxxx or UNIT/degree).

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

–Ju*zone/scale* or **–JU***zone/width* (UTM - Universal Transverse Mercator **[C]**).
Give the zone number (1-60) and scale (1:xxxx or UNIT/degree).
Zones: prepend - or + to enforce southern or northern hemisphere conventions [northern if south > 0].

–Jy*lon0/lats/scale* or **–JY***lon0/lats/width* (Basic Cylindrical Projections **[E]**).
Give the central meridian, standard parallel, and scale (1:xxxx or UNIT/degree). The standard parallel is typically one of these (but can be any value):
45 - The Peters projection
37.4 - The Trystan Edwards projection
30 - The Behrman projection
0 - The Lambert projection

AZIMUTHAL PROJECTIONS:

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

–Ja*lon0/lat0/scale* or **–JA***lon0/lat0/width* (Lambert **[E]**).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Je*lon0/lat0/scale* or **–JE***lon0/lat0/width* (Equidistant).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Jf*lon0/lat0/horizon/scale* or **–JF***lon0/lat0/horizon/width* (Gnomonic).
lon0/lat0 specifies the projection center. *horizon* specifies the max distance from projection center (in degrees, < 90). Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Jg*lon0/lat0/scale* or **–JG***lon0/lat0/width* (Orthographic).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx or *radius/lat*, where *radius* is distance in UNIT from origin to the oblique latitude *lat*.

–Js*lon0/lat0/scale* or **–JS***lon0/lat0/width* (General Stereographic **[C]**).
lon0/lat0 specifies the projection center. Give scale as 1:xxxx (true at pole) or *slat*/1:xxxx (true at standard parallel *slat*) or *radius/lat* (*radius* in UNIT from origin to the oblique latitude *lat*).

CONIC PROJECTIONS:

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

–Jd*lon0/lat0/lat1/lat2/scale* or **–JD***lon0/lat0/lat1/lat2/width* (Equidistant)
Give projection center, two standard parallels, and scale (1:xxxx or UNIT/degree).

–Jl*lon0/lat0/lat1/lat2/scale* or **–JL***lon0/lat0/lat1/lat2/width* (Lambert **[C]**)
Give origin, 2 standard parallels, and scale along these (1:xxxx or UNIT/degree).

MISCELLANEOUS PROJECTIONS:

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

–Ji*lon0/scale* or **–JI***lon0/width* (Sinusoidal **[E]**).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jk[**f**|**s**]*lon0/scale* or **–JK**[**f**|**s**]*lon0/width* (Eckert IV (f) and VI (s) **[E]**).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jn*lon0/scale* or **–JN***lon0/width* (Robinson).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jr*lon0/scale* **–JR***lon0/width* (Winkel Tripel).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jv*lon0/scale* or **–JV***lon0/width* (Van der Grinten).
Give the central meridian and scale along equator (1:xxxx or UNIT/degree).

–Jw*lon0/scale* or **–JW***lon0/width* (Mollweide **[E]**).
Give the central meridian 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.

–Jx*x-scale*[*/y-scale*] or **–JX***width*[*/height*] (Linear, log, and power scaling)
Give *x-scale* in UNIT/x-unit and/or *y-scale* in UNIT/y-unit; or specify *width* and/or *height* in UNIT (*y-scale*=*x-scale* if not specified separately). Use negative scale(s) to reverse the direction of an axis (e.g., to have y be positive down). 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.

p*power* 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 gmtdefaults) can be invoked using **–JXh** (for landscape); **–JXv** (for portrait) will swap the x- and y-axis lengths. The **GMT** default unit for this installation is UNIT. However, you may change this by editing your .gmtdefaults4 file(s) (run gmtdefaults to create one if you do not have it).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

–E Sets the viewpoint's azimuth and elevation (for perspective view) [180/90].

–G Paint inside of basemap. [Default is no fill]. Specify the grey shade (0–255) or color (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 0–100%; or valid color name).

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L 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 [miles if **m** is appended; nautical miles if **n** is appended]. Use **–Lf** to get a "fancy" scale [Default is plain]. The default label equals the distance unit (km, miles, nautical miles) and is justified on top of the scale [t]. Change this by giving your own label (or - to keep the default) and justification (l(eft), r(ight), t(op), b(ottom), and u(unit) - using the label as a unit appended to all distance annotations along the scale). If you want to place a rectangle behind the scale, specify pen and/or fill parameters with the **+p** and **+f** modifiers.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–T 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 what *kind* of rose 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 = * then a north star is plotted instead of the north label. Annotation and two levels of tick intervals for geographic and magnetic directions are 10/5/1 and 30/5/1 degrees, respectively; override these settings by appending +*gints*[/*mints*]. Color and pen attributes are taken from **COLOR_BACKGROUND** and **TICK_PEN**, respectively, while label fonts and sizes follow the usual annotation, label, and header font settings.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z For 3-D projections: Sets the z-level of the basemap [0].

–c Specifies the number of plot copies. [Default is 1].

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.

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

```
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

```
psbasemap –R1/10000/1e20/1e25 –JX25cl/15cl –B2:Wavelength:/a1pf3: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

```
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

```
psbasemap –R0/360/0/1000 –JP6i –B30p/100 > polar.ps
```

CYLINDRICAL MAP PROJECTIONS

Cassini

A 10-cm-wide basemap using the Cassini projection may be obtained by

```
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

```
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

```
psbasemap –R0/360/-90/90 –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

```
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

```
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

psbasemap –R-50/310/-90/90 **–JQ**130/25**c –B**30**g**30:."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

psbasemap –R95/5/108/20**r –Ju**46/1:10000000 **–B**3**g**3:.UTM: > utm.ps

Basic 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 Peters projection. A 9 inch wide global basemap centered on the Pacific is made by

psbasemap –R0/360/-90/90 **–JY**180/45/9**i –B**30**g**30:.Peters: > peters.ps

CONIC MAP PROJECTIONS

Albers [equal-area]

A basemap for middle Europe may be created by

psbasemap –R0/90/25/55 **–Jb**45/20/32/45/0.25**c –B**10**g**10:."Albers Equal-area": > albers.ps

Lambert [conformal]

Another basemap for middle Europe may be created by

psbasemap –R0/90/25/55 **–Jl**45/20/32/45/0.1**i –B**10**g**10:."Lambert Conformal Conic": > lambertc.ps

Equidistant

Yet another basemap of width 6 inch for middle Europe may be created by

psbasemap –R0/90/25/55 **–JD**45/20/32/45/6**i –B**10**g**10:."Equidistant conic": > econic.ps

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:

psbasemap –R0/360/-90/90 **–JA**-80/-30/15**c –B**30**g**30/15**g**15:."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:

psbasemap –R0/360/-90/90 **–JE**125/10/15**c –B**30**g**30/15**g**15:."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:

psbasemap –R0/360/-90/90 **–JF**-100/40/60/6**i –B**30**g**30/15**g**15:."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:

psbasemap –R0/360/-90/90 **–JG**125/10/6**i –B**30**g**30/15**g**15:."Orthographic": > ortho.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

psbasemap –R-45/45/-90/-60 **–Js**0/-90/12**c**/-60 **–B**5**g**5:."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

psbasemap –R100/-45/160/-5**r –JS**130/-30/12**c –B**30**g**30/15**g**15:."General Stereographic View": > stereo2.ps

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

psbasemap –R0/360/-90/90 **–Jh**180/1:200000000 **–B**30**g**30/15**g**15:.Hammer: > hammer.ps

Sinusoidal [equal-area]

To make a sinusiodal world map centered on Greenwich, with a scale along the equator of 0.02 inch/degree, use

psbasemap –R-180/180/-90/90 **–Ji**0/0.02**i –B**30**g**30/15**g**15:."Sinusoidal": > sinus1.ps

To make an interrupted sinusiodal 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:

psbasemap –R-160/-20/-90/90 **–Ji**-90/0.02**i –B**30**g**30/15**g**15**Wesn –K** > sinus_i.ps
psbasemap –R-20/60/-90/90 **–Ji**20/0.02**i –B**30**g**30/15**g**15**wesn –O –K –X**2.8**i** >> sinus_i.ps
psbasemap –R60/200/-90/90 **–Ji**130/0.02**i –B**30**g**30/15**g**15**wEsn –O –X**1.6**i** >> sinus_i.ps

Eckert IV [equal-area]

Pseudo-cylindrical projection typically used for global maps only. Set the central longitude and scale, e.g.,

psbasemap –R0/360/-90/90 **–Jkf**180/0.064**c –B**30**g**30/15**g**15:."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.,

psbasemap –R0/360/-90/90 **–Jks**180/0.064**c –B**30**g**30/15**g**15:."Eckert VI": > eckert6.ps

Robinson

Projection designed to make global maps "look right". Set the central longitude and width, e.g.,

psbasemap –R-180/180/-90/90 **–JN**0/8**i –B**30**g**30/15**g**15:."Robinson": > robinson.ps

Winkel Tripel

Yet another projection typically used for global maps only. You can set the central longitude, e.g.,

```
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 –R0/360/-90/90 –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:

```
psbasemap –R0/360/-90/90 –JV180/7i –B30g30/15g15:."Van der Grinten": > grinten.ps
```

RESTRICTIONS

For some projections, a spherical earth is implicitly assumed. A warning will notify the user if **–V** is set.

BUGS

The **–B** option is somewhat complicated to explain and comprehend. However, it is fairly simple for most applications (see examples).

SEE ALSO

gmtdefaults(l), *GMT*(l)

NAME

psclip – To set up polygonal clip paths

SYNOPSIS

psclip *xyfiles* **–J***parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–B**[**p**|**s**]*parameters*] [**–E***azim*/*elev*] [**–K**] [**–N**] [**–M**[*flag*]] [**–O**] [**–P**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***zlevel*] [**–c***copies*] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

psclip –C [**–K**] [**–O**]

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. As an option (**–N**), the user may choose to reverse the sense of what is the inside and outside of the paths. 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.

xyfiles ASCII [or binary, see **–b**] file(s) with (x,y) values for clip polygons. If no files are given, the standard input is read.

–C Mark end of existing clip path. No input file or projection information are needed.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–E Sets the viewpoint's azimuth and elevation [180/90].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Multiple segment file. Segments are separated by a record whose first character is <*flag*>. [Default is '>'].

–N Invert the sense of what is inside and outside, i.e., use the outside of the polygons for clipping.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z For 3-D projections: Sets the z-level of the polygons [0].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–c Specifies the number of plot copies. [Default is 1].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To make an overlay *PostScript* file that will set up a complex clip area to which subsequent plotting will be confined, run:

psclip my_region.xy **–R**0/40/0/40 **–J**m0.3**i –O –K** > clip_mask_on.ps

To deactivate the clipping in an existing plotfile, run:

psclip –C –O >> complex_plot.ps

BUGS

psclip cannot handle polygons that contain the south or north pole. For such polygons, you should split them into two and make each explicitly contain the polar point. The two clip polygons will combine to give the desired effect.

SEE ALSO

GMT(l), *grdmask*(l), *psbasemap*(l), *psmask*(l)

NAME

pscoast – To plot land-masses, water-masses, coastlines, borders, and rivers

SYNOPSIS

pscoast –J*parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–A***min_area*[/*min_level*/*max_level*]] [**–B**[**p**|**s**]*parameters*] [**–C***fill*] [**–D***resolution*] [**–E***azimuth/elevation*] [**–G***fill*] [**–I***river*[/*pen*]] [**–Jz**|**Z***parameters*] [**–K**] [**–L**[**f**][**x**]*lon0*/*lat0*[/*slon*]/*slat*/*length*[**m**|**n**|**k**][:*label*:*just*][**+p***pen*][**+f***fill*]]] [**–O**] [**–M**[*flag*]] [**–N***border*[/*pen*]] [**–O**] [**–P**] [**–Q**] [**–S***fill*] [**–T**[**f**|**m**][**x**]*lon0*/*lat0*/*size*[/*info*][:w,e,s,n:][+*gint*[/*mint*]]] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–W***pen*] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***zlevel*] [**–c***copies*] [**–bo**[**s**|**S**|**d**|**D**][*ncol*]]

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 datafiles 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. The *PostScript* code is written to standard output.

No space between the option flag and the associated arguments. Use upper case for the option flags and lower case for modifiers.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)

–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude).

OPTIONS

–A Features with an area smaller than *min_area* in km^2 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)]. See DATABASE INFORMATION below for more details.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Set the shade (0–255), color (r/g/b), or pattern (**p**|**P***dpi/pattern*; see **–G**) for lakes [Default is the fill chosen for "wet" areas (**–S**)].

–D Selects the resolution of the data set to use ((**f**)ull, (**h**)igh, (**i**)ntermediate, (**l**)ow, and (**c**)rude). The resolution drops off by 80% between data sets. [Default is **l**].

–E Sets the viewpoint's azimuth and elevation (for perspective view) [180/90].

–G Select painting or clipping of "dry" areas. Append a shade, color, pattern, or **c** for clipping. Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–I Draw rivers. Specify the type of rivers and [optionally] append pen attributes [Default pen: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.
Choose from the list of river types below. Repeat option **–I** as often as necessary.
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
a = All rivers and canals (1-10)
r = All permanent rivers (1-4)
i = All intermittent rivers (5-7)
c = All canals (8-10)

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L 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 [miles if **m** is appended; nautical miles if **n** is appended]. Use **–Lf** to get a "fancy" scale [Default is plain]. The default label equals the distance unit (km, miles, nautical miles) and is justified on top of the scale [t]. Change this by giving your own label (or - to keep the default) and justification (l(eft), r(ight), t(op), b(ottom), and u(unit) - using the label as a unit appended to all distance annotations along the scale). If you want to place a rectangle behind the scale, specify pen and/or fill parameters with the **+p** and **+f** modifiers.

–M Dumps a single multisegment ASCII (or binary, see **–bo**) file to standard output. No plotting occurs. Specify any combination of **–W, –I, –N**. Optionally, you may append the *flag* character that is written at the start of each segment header ['>'].

–N Draw political boundaries. Specify the type of boundary and [optionally] append pen attributes [Default pen: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'. 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 Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Mark end of existing clip path. No projection information is needed.

–S Select painting or clipping of "wet" areas. Append the shade (0–255), color (r/g/b), pattern (see **–G**), or **c** for clipping.

–T 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 what *kind* of rose 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 = * then a north star is plotted instead of the north label. Annotation and two levels of tick intervals for geographic and magnetic directions are 10/5/1 and 30/5/1 degrees, respectively; override these settings by appending +*gints*[/*mints*]. Color and pen attributes are taken from **COLOR_BACKGROUND** and **TICK_PEN**, respectively, while label fonts and sizes follow the usual annotation, label, and header font settings.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Draw coastlines. [Default is no coastlines]. Append pen attributes [Defaults: width = 1, color = black, texture = solid].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z For 3-D projections: Sets the z-level of the coastlines [0].

–c Specifies the number of plot copies. [Default is 1].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program.

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

pscoast **–R**-30/30/-40/40 **–Jm**0.1**i** **–B**5 **–I**1/1**p**/0/0/255 **–I**2/0.25**p**/0/0/255 **–N**1/0.25**tap** **–W**0.25**p**/255/255/255 **–G**0/255/0 **–S**0/0/255 **–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

pscoast –R-30/-10/60/65 **–Jm**1**c –B**5 **–Gp**100/28 > 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

pscoast –R-30/30/-40/40 **–Jm**0.1**i –B**5 **–Gc –P –K** > africa.ps
grdimage –Jm0.1**i** etopo5.grd **–C**colors.cpt **–O –K** >> africa.ps
pscoast –Q –O >> africa.ps

DATABASE INFORMATION

The coastline database is 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.

pscoast will first look for coastline files in directory **$GMTHOME**/share (where **$GMTHOME** is an environmental variable). If the desired file is not found, it will look for the file coastline.conf in the same directory. 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.

BUGS

The options to fill (**−C −G −S**) may not always work if the Azimuthal equidistant projection is chosen (**−Je**|**E**). 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 1970ies-style and do not reflect the recent border rearrangements in Europe. We intend to update these as high-resolution data become available to us.

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.

SEE ALSO

gmtdefaults(l), *GMT*(l), *grdlandmask*(l), *psbasemap*(l)

NAME

pscontour – Contour xyz-data by direct triangulation

SYNOPSIS

pscontour *xyzfile* **–C***cptfile* **–J***parameters* **–R***west/east/south/north*[**r**] [**–A**[**-**][*labelinfo*]] [**–B**[**p**|**s**]*parameters*] [**–D**[*dumpfile*]] [**–E***view_az/view_el*] [**–G**[**d**|**f**|**n**|**l**|**L**|**x**|**X**]*params*] [**–H**[**i**][*nrec*]] [**–I**] [**–K**] [**–L***pen*] [**–M**[*flag*]] [**–N**] [**–O**] [**–P**] [**–S**] [**–T***indexfile*] [**–U**[*/dx/dy/*][*label*]] [**–V**] [**–W**[**+**]*pen*] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–c***copies*] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]]

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), 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.

xyzfile Raw ASCII (or binary, see **–b**) xyz data to be contoured.

–C name of the color palette file. Must have discrete colors if you want to paint the surface (**–I**). Only contours that have annotation flags set will be annotated.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–A Give - to disable all annotations. 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:

+a*angle*
For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default].

+c*dx*[*/dy*]
Sets the clearance between label and optional text box. Append **c**|**i**|**m**|**p** to specify the unit or % to indicate a percentage of the label font size [15%].

+f*font* Sets the desired font [Default **ANNOT_FONT_PRIMARY**].

+g[*color*]
Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PAGE_COLOR**].

+j*just* Sets label justification [Default is CM].

+k*color*
Sets color of text labels [Default is **COLOR_BACKGROUND**].

+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 multisegment header (first scan for an embedded **–L***label* option, if not use the first word following the segment flag).

+Ld Take the Cartesian plot distances along the line as the label; append **c**|**i**|**m**|**p** as the unit [Default is **MEASURE_UNIT**].

+LD Calculate actual map distances; append **d**|**e**|**k**|**m**|**n** 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.

+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 boxsets [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

+r*min_rad*
Will not place labels where the line's radius of curvature is less than *min_rad* [Default is 0].

+s*size* Sets the desired font size in points [Default is 9].

+u*unit* Appends *unit* to all line labels. If *unit* starts with a leading hypen (-) 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 hypen (-) then there will be no space between label value and the prefix. [Default is no prefix].

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–D Dump the (x,y,z) coordinates of each contour to separate files, one for each contour segment. The files will be named *dumpfile_cont_segment[_i]*.xyz, where *cont* is the contour value and *segment* is a running segment number for each contour interval (for closed contours we append _i.) However, when **–M** is used in conjunction with **–D** a single multisegment file is created instead.

–E Sets the view point by specifying azimuth and elevation in degrees. [Default is 180/90].

–G Controls the placement of labels along the contours. Choose among five controlling algorithms:

–Gd*dist*[c|i|m|p] or **–GD***dist*[d|e|k|m|n]
For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), **m** (meter), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), or **d** (spherical degree). [Default is 10**c** or 4**i**].

–Gf*ffile.d*
Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the contours. Inexact mathces and points outside the region are skipped.

–Gl|L*line1*[,*line2*,...]
Give *start* and *stop* coordinates for one or more comma-separated straight line segments. Labels will be placed where these lines intersect the contours. 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]. **–GL** will interpret the point pairs as defining great circles [Default is straight line].

–Gn*n_label*

Specifies the number of equidistant labels for contours line [1]. Upper case **–GN** starts labeling exactly at the start of the line [Default centers them along the line]. **–GN**-1 places one justified label at start, while **–GN**+1 places one justified label at the end of contours. Optionally, append /*min_dist*[c|i|m|p] to enforce that a minimum distance separation between successive labels is enforced.

–Gx|X*xfile.d*

Reads the multi-segment file *xfile.d* and places labels at the intersections between the contours and the lines in*xfile.d*. **–GX** will resample the lines first along great-circle arcs.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Color the triangles using the color palette table.

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Draw the underlying triangular mesh using the specified pen attributes [Default is no mesh]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–M When used in conjunction with **–D** a single multisegment file is created, and each contour section is preceded by a header record whose first column is *flag* followed by the contour level.

–N Do NOT clip contours or image at the boundaries [Default will clip to fit inside region **–R**].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S Skip all input *xyz* points that fall outside the region [Default uses all the data in the triangulation].

–T 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**)].

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Select contouring and set contour pen attributes. If the **+** flag is set then the contour lines are colored according to the cpt file (see **–C**).

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–c Specifies the number of plot copies. [Default is 1].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 input columns]. Use 4-byte integer triplets for node ids (**–T**).

−bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 3 output columns].

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

```
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

```
pscontour temp.xyz −R0/150/0/100 −Jx0.1 −Ctemp.cpt −G −W0.25p > temp.ps
```

BUGS

Sometimes there will appear to be thin lines of the wrong color in the image. This is a round-off problem which may be remedied by using a higher value of **DOTS_PR_INCH** in the .gmtdefaults4 file.

SEE ALSO

GMT(l), *grdcontour*(l), *grdimage*(l), *nearneighbor*(l), *psbasemap*(l), *psscale*(l), *surface*(l), *triangulate*(l)

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.
www.cs.cmu.edu/˜quake/triangle.html

NAME

pshistogram – Bin data and plot histograms

SYNOPSIS

pshistogram *file* **–Jx|X***parameters* **–W***bin_width* [**–A**] [**–B[p|s]***parameters*] [**–C**] [**–E***azimuth***/***elevation*] [**–G***fill*] [**–H[i]**[*nrec*]] [**–Jz|Z***parameters*] [**–I[o]**] [**–K**] [**–L***pen*] [**–O**] [**–P**] [**–Q**] [**–R***xmin***/***xmax***/***ymin***/***ymax***[r]**] [**–S**] [**–T***col*] [**–U**[**/***dx***/***dy***/**][*label*]] [**–V**] [**–X[a|c|r]**[*x-shift***[u]**]] [**–Y[a|c|r]**[*y-shift***[u]**]] [**–Z***type*] [**–c***copies*] [**–bi[s|S|d|D]**[*ncol*]] [**–f[i|o]***colinfo*]

DESCRIPTION

pshistogram reads *file* [or standard input] and examines data column *col* 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.

file ASCII [or binary, see **–b**] datafile. If no file is given, **pshistogram** will read standard input.

–Jx *xscale[/yscale]* (Linear scale(s) in distance unit/data unit).

–W Sets the bin width used for histogram calculations.

OPTIONS

No space between the option flag and the associated arguments.

–A Plot the histogram horizontally from x = 0 [Default is vertically from y = 0].

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Center bin on each value. [Default is left edge].

–E Sets the viewpoint's azimuth and elevation (for perspective view) [180/90].

–G Select filling of bars. [Default is no fill]. Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Inquire about min/max x and y after binning. No plotting is done. Append **o** to output an ASCII table of the resulting x,y data to stdout.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Draw bar outline using the specified pen thickness. [Default is no outline]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Draw a cumulative histogram.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX|x**), or 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 as indicated (however, input/output and plotting formats are flexible). If not given, **pshistogram** will automatically find reasonable values for the region.

–S Draws a stairs-step diagram instead of histogram.

–T Specify which column to use for the histogram data. First column is 0 [0].

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z 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).

–c Specifies the number of plot copies. [Default is 1].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

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:

pshistogram v3206.t **–JX**h **–W**250 **–C –L**0.5**p –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:

pshistogram errors.xy **–W**1 **–R**-10/10/0/0 **–Jx**0.75**c**/0.01**c –B**2:Error:/100:Counts: **–G**0 **–T**1 **–V** > plot.ps

Since no y-range was specified, pshistogram will calculate ymax in even increments of 100.

SEE ALSO

GMT(l), *psbasemap*(l), *psrose*(l), *psxy*(l)

NAME

psimage – To plot images (EPS files or Sun rasterfiles) on maps

SYNOPSIS

psimage *imagefile* [**–W**[**-**]*xlength*[/*ylength*] | **–E***dpi*] [**–C***xpos*/*ypos*[/*justify*]] [**–F***pen*] [**–G**[**f**|**b**]*color*] [**–I**] [**–K**] [**–M**] [**–N***nxrep*[/*nyrep*]] [**–O**] [**–P**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–c***copies*]

DESCRIPTION

psimage reads an Encapsulated *PostScript* file or a 1, 8, 24, or 32-bit Sun rasterfile 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. 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).

imagefile
This must be an Encapsulated *PostScript* (EPS) file or a Sun rasterfile. An EPS file must contain an appropriate BoundingBox. A rasterfile can have a depth of 1, 8, 24, or 32 bits. Old-style, Standard, Run-length-encoded, and RGB Sun rasterfiles are supported. Other raster formats can be converted to Sun format via a variety of public-domain software (e.g., convert, xv).

–E Sets the dpi of the image in dots pr inch, or use **–W**.

–W Sets the size of the image. If not given *ylength* is set to *xlength* * (ny/nx). If *xlength* is negative we use the absolate value and interpolate image to the device resolution using the PostScript image operator. Alternatively, use **–E**.

OPTIONS

No space between the option flag and the associated arguments.

–C Sets position of lower left corner of image, and optionally append justification [0/0/LB].

–F Draws a rectangular frame around the image with the given pen [no frame]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–Gb Sets background color (replace white pixel) for 1-bit image templates. Use - for transparency (and set **–Gf** to the desired color).

–Gf Sets foreground color (replace black pixel) for 1-bit image templates. Use - for transparency (and set **–Gb** to the desired color).

–I Invert image before plotting (1-bit images only). This is what is done when you use **–GP** in other GMT programs.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Convert color image to monochrome grayshades using the (television) YIQ-transformation.

–N Replicate the image *nxrep* in x and *nyrep* in y [Default is 1/1].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−X −Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

−c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To plot the image contained in the 8-bit rasterfile scanned_face.ras, scaling it to 8 by 10 cm, use

psimage scanned_face.ras **−W**8**c**/10**c** > 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, use

psimage tiger.eps **−C**2**i**/1**i**/TR **−W**3**i** > image.ps

To replicate the image template 1_bit.ras over a 5 by 5 inch area, colorize it, and setting each piece to be 1 by 1 cm, use

psimage 1_bit.ras **−Gb**200/150/100 **−Gf**100/50/30 **−N**5**i**/5**i** **−W**1**c** > image.ps

SEE ALSO

GMT(l)

NAME

pslegend – To plot a map legend

SYNOPSIS

pslegend *textfile* **–D**[**x**]*lon*/*lat*/*width*/*height*/*just* **–J***parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–C***dx*/*dy*] [**–F**] [**–G***fill*] [**–K**] [**–L***spacing*] [**–O**] [**–P**] [**–S**[*script*]] [**–U**[*/dx/dy/*][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–c***copies*] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]]

DESCRIPTION

pslegend will make legends that can be overlaid on maps. It reads specific legend-related information from an input file [or stdin]. Because all the elements of the legend can already be created with other tools (**psxy**, **pstext**) we use those tools by creating a batch job of commands that are executed to make the final *PostScript* overlay. Because of this process, the option exists to just output the script which can then be fine-tuned manually. Unless otherwise noted, annotations will be made using the annotation font and size in effect.

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

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 –E –I –L –M –N –S** and **–Z**.

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.

D *offset pen*

The D record results in a horizontal line across the legend. The line starts and stops *offset* units from the frame sides, and is drawn using the specified *pen*.

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 **l** for lines.

H *fontsize font Header*

The H record plots a centered text string using the specified font parameters.

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 using the specified font parameters.

M *slon*|- slat length **f**|**p [–R***w/e/s/n* **–J***param* **]**

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 (append **m** or **n** for miles or nautical miles), and **f**|**p** for fancy or plain scale. If the **–R –J** supplied to **pslegend** is different than the projection needed for the scale, supply the optional **–R –J** settings as well. Note that *length* can have *:label:just* appended, where *label* replaces the default label (unless - is given) and *just* (**l**|**r**|**t**|**b**) dictates where the label is placed [Default is **t**]. Use **u** to treat the label as distance units appended to each annotation.

N *ncolums*
Change the number of columns in the legend [1].

S *dx1 symbol size fill pen dx2 text*
Plots the selected symbol with specified size, fill, and outline. The symbol is centered at *dx1* from the left margin of the column, with the explanatory text starting *dx2* from the margin. Use - if no fill is required. Two **psxy** symbols front (f) and vector (v) require special attention. You must prepend the length of the desired item to the rest of the symbol argument; this will be used internally to set the correct fault or vector length and will be stripped before passing the arguments to **psxy**.

> <paragraph mode header for pstext>
Start a new text paragraph by specifying all the parameters needed (see **pstext –M** description). Note that **pslegend** knows what all those values should be, so normally you can leave the entire record (after >) blank. 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 -.

T *paragraph-text*
One or more of these T records with text must follow after the **>** 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.

–D
Positions the legend and specifies its size. The *just* is a 2-char justification string (see **pstext**) that relates the given position to a point on the rectangular legend box. If you want to specify the position in projected units (i.e., inches or cm), use **–Dx**.

–J
Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–C Sets the clearance between the legend frame and the internal items [0.15**c**/0.15**c** (or 0.05**i**/0.05**i**)].

–F Draws a border around the legend using **FRAME_PEN**.

–G Specify the grey shade (0−255) or color (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 0−100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Sets the linespacing factor in units of the current annotation font size [1.1].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S Instead of writing the *PostScript* plot [Default], output the **GMT** script used to make the legend to standard output, or optionally to the file *script*.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

−V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−X −Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

−c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To add an example of a legend to a Mercator plot (map.ps) with the given specifications, use

```
pslegend −R-10/10/-10/10 −JM4i −G255 −D0/0/4i/3i/BL << EOF >> map.ps
G -0.15i
H 24 Times-Roman My Map Legend
G 0.05i
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 t 0.15i yellow 0.25p 0.3i This triangle is yellow
S 0.1i h 0.15i green 0.25p 0.3i This hexagon is green
S 0.1i d 0.15i blue 0.25p 0.3i This diamond is blue
S 0.1i - 0.15i - 0.25tap 0.3i A contour
V 0 1p
D 0.2i 1p
N 1
I SOEST_logo.ras 3 CT
L 9 4 R Smith et al., @%5%J. Geophys. Res., 99@%%, 2000
G 0.5i
>
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 need to adjust the box height to get the right size.
G -0.3i
M 5 5 1000:km:l f
EOF
```

SEE ALSO

GMT(l), *psbasemap*(l), *pstext*(l), *psxy*(l)

NAME

psmask – To clip or mask areas of no data on a map

SYNOPSIS

psmask [*xyzfile*] **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–J***parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–B**[**p**|**s**]*parameters*] [**–D***dumpfile*] [**–E***azimuth*/*elevation*] [**–F**] [**–G***fill*] [**–H**[**i**][*nrec*]] [**–K**] [**–M**[*flag*]] [**–N**] [**–O**] [**–P**] [**–S***search_radius*[**m**|**c**|**k**|**K**]] [**–T**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–c***copies*] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]]

psmask –C [**–K**] [**–O**]

DESCRIPTION

psmask reads a (*x*,*y*,*z*) file [or standard input] and uses this information to find out which grid cells are reliable. Only gridcells which have one or more data points are considered reliable. As an option, you may specify a radius of influence. Then, all gridcells 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.

xyzfile File with (*x*,*y*,*z*) values (e.g., that was used to run **surface**). If no file is given, standard input is read. For binary files, see **–b**.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Mark end of existing clip path. No input file is needed. Implicitly sets **–O**.

–D Dumps out the resulting clipping polygons to disk. Ignored if **–T** is set. If no dumpprefix is given we use mask (Files will be called mask_*.d).

–E Sets the viewpoint's azimuth and elevation for perspective plots [180/90].

–F Force pixel registration. [Default is grid registration].

–G Paint the clip polygons [or tiles] with selected fill [Default is no fill]. Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Multiple segment file(s). Segments are separated by a special record. For ASCII files the first character must be *flag* [Default is '>']. For binary files all fields must be NaN and **–b** must set the number of output columns explicitly. By default the **–M** setting applies to both input and output. Use **–Mi** and **–Mo** to give separate settings.

–N Invert the sense of the test, i.e. clip regions where there is data coverage.

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S 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 **m** to indicate minutes or **c** to indicate seconds. Append **k** to indicate km (implies **–R** and **–I** are in degrees, and we will use a fast flat Earth approximation to calculate distance). For more accuracy, use uppercase **K** if distances should be calculated along geodesics. However, if the current **ELLIPSOID** is set to Sphere then spherical great circle calculations are used.

–T Plot tiles instead of clip polygons. Use **–G** to set tile color or pattern.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–c Specifies the number of plot copies. [Default is 1].

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:

psmask africa_grav.xyg **–R**20/40/20/40 **–I5m –J**M10**i –O –K** > mask.ps

The same example but this time we use white tiling:

psmask africa_grav.xyg **–R**20/40/20/40 **–I5m –J**M10**i –T –O –K –G**255 > mask.ps

SEE ALSO

GMT(l), *grdmask*(l), *surface*(l), *psbasemap*(l), *psclip*(l)

NAME

psrose – Plot (length, azimuth) as windrose diagram or polar histogram (sector or rose diagram).

SYNOPSIS

psrose *file* [**–A***sector_width*[**r**]] [**–B**[**p**|**s**]*parameters*] [**–C**[*mode_file*]] [**–E***azimuth*/*elevation*] [**–G***fill*] [**–H**[**i**][*nrec*]] [**–I**] [**–K**] [**–M**[*parameters*] [**–O**] [**–P**] [**–R***r0*/*r1*/*az_0*/*az_1*] [**–S***radial_scale*[**n**]] [**–T**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–W***pen*] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***scale*] [**–c***copies*] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]]

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.

file Name of ASCII [or binary, see **–b**] data file. If no file is given, psrose will read standard input.

OPTIONS

No space between the option flag and the associated arguments.

–A Gives the sector width in degrees for sector and rose diagram. [Default 0 means windrose diagram]. Append **r** to draw rose diagram instead of sector diagram.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details. Remember that "x" here is radial distance and "y" is azimuth. The ylabel may be used to plot a figure caption.

–C Plot vectors showing the principal directions given in the *modes* file. If no file is given, compute and plot mean direction.

–E Sets the viewpoint's azimuth and elevation [180/90]

–G Selects shade or color for sector infill [Default is no fill]. Specify the grey shade (0–255) or color (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 0–100%; or valid color name).

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Inquire. Computes statistics needed to specify useful **–R**. No plot is generated.

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Specify new arrow attributes tailwidth/headlength/headwidth/r/g/b to change the appearance of arrows (Only if **–C** is set). [Default is 0.075**c**/0.3**c**/0.25**c**/0/0/0 (or 0.03**i**/0.12**i**/0.1**i**/0/0/0)].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–R Specifies the 'region' of interest in (r,azimuth) space. r0 is 0, r1 is max length in units. For azimuth, specify -90/90 for half circle plot or 0/360 for full circle.

–S Specifies radius of circle. Append **n** 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 Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Set pen attributes for sector outline or rose plot. [Default is no outline]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z Multiply the data radii by *scale*. E.g., use **–Z**0.001 to convert your data from m to km [Default is no scaling].

–c Specifies the number of plot copies. [Default is 1].

–: Input file has (azimuth,radius) pairs rather than the expected (radius,azimuth).

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

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 (200/240/255) outlined by a solid red pen (width = 0.75 points), draw the mean azimuth, and shown in Portrait orientation, use:

psrose fault_segments.az_r **–R**0/150/-90/90 **–B**50g25:"Fault length":/g30:."Rose diagram": **–S3i** **–A**10**r** **–G**200/240/255 **–W**0.75**p**/255/0/0 **–Z**0.001 **–C** **–P** **–T** **–:** | lpr

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:

psrose lines.az_r **–R**0/500/0/360 **–S**5**c** **–Bg**100/**g**45:."Windrose diagram": **–W**0.5**p** **–Uc** | lpr

BUGS

No default radial scale and grid settings for polar histograms. User must run **psrose –I** to find max length in binned data set.

SEE ALSO

GMT(l), *gmtdefaults*(l), *pshistogram*(l)

NAME

psscale – Create grayscale or colorscale for maps

SYNOPSIS

psscale **–D***xpos*/*ypos*/*length*/*width*[**h**] [**–A**] [**–B**[**p**|**s**]*parameters*] [**–C***cpt_file*] [**–E**[**b**|**f**][*length*]] [**–I**[*max_intens*[*low_i*/*high_i*]]] [**–K**] [**–L**[**i**][*gap*]] [**–M**] [**–N***dpi*] [**–O**] [**–P**] [**–Q**] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***zfile*] [**–c***copies*]

DESCRIPTION

psscale plots grayscales or colorscales 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 r/g/b values) **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.

–D Defines the position of the center/top (for horizontal scale) or center/left (for vertical scale) and the dimensions of the scale. Append *h* to get a horizontal scale [Default is vertical].

OPTIONS

No space between the option flag and the associated arguments.

–A Place annotations above (instead of below) horizontal scalebars and to the left (instead of the right) of vertical scalebars.

–B Set annotation, tick, and gridline interval for the colorbar. The xaxis label will plot beneath a horizontal bar (or vertically to the right of a vertical bar). As an option, use the yaxis label to plot the data unit to the right of a horizontal bar (and above a vertical bar). If no values are provided, the default is to annotate every color level (which may be overridden by ULB flags in the cpt file). Note that since vertical labels will be plotted as a column of individual characters, no octal escape characters embedded in the label are allowed. Text with such characters will be plotted horizontally (relative to the color scale). By default, labels are generated from the numerical entries. To specify custom text labels for intervals, you must append ;*label* to each z-slice in the cpt file.

–C *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. As for **grdview**, **psscale** can understand pattern specifications in the cpt file.

–E Add sidebar triangles for **b**ack- and/or **f**oreground colors. Add **f** or **b** for only one sidebar triangle [Default gives both]. Optionally, append triangle height [Default is half the barwidth].

–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].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L 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.

–M Force a monochrome graybar using the (television) YIQ transformation.

–N Effective dots-per-inch for the rectangular image making up the color scale [300].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–Q Select logarithmic scale and power of ten annotations. All z-values in the cpt file will be converted to p = log10(z) and only integer p values will be annotated using the 10^p format [Default is linear scale].

–S Do not separate different colour intervals with black lines.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z 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**).

–c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To append a vertical colorscale (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

psscale –D6.5**i**/2**i**/7.5**c**/1.25**c –O –C**colors.cpt **–I –E –B**5:BATHYMETRY:/:m: >> map.ps

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.

SEE ALSO

GMT(l), *makecpt*(l), *grd2cpt*(l)

NAME

pstext – To plot text on maps

SYNOPSIS

pstext *textfile* **–J***parameters* **–R***west*/*east*/*south*/*north*[**r**] [**–B**[**p**|**s**]*parameters*] [**–C***dx/dy*] [**–D**[**j**]*dx*/*dy*[**v**[*color*]] [**–E***azimuth*/*elevation*] [**–G***color*] [**–H**[**i**][*nrec*]] [**–Jz**|**Z***parameters*] [**–K**] [**–L**] [**–M**[*flag*]] [**–N**] [**–O**] [**–P**] [**–S***pen*] [**–U**[/*dx*/*dy*/][*label*]] [**–V**] [**–W**[*color*][**o**|**O**|**c**|**C**[*pen*]]] [**–X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**–Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**–Z***zlevel*] [**–c***copies*] [**–:**[**i**|**o**]]

DESCRIPTION

pstext plots textstrings 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, and @# toggles small caps 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 **CHAR_ENCODING** must be set to and extended character set in your .gmtdefaults4 file in order to use the accented characters. Using the **–W** option, a colored 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**)).

textfile This file contains 1 or more records with (*x, y, size, angle, fontno, justify, text*). If no file is given, **pstext** will read standard input. *size* is text size in points, *angle* is measured in degrees counter-clockwise from horizontal, *fontno* sets the font type, *justify* sets the alignment. If *fontno* is not an integer, then it is taken to be a textstring with the desired fontname. See the **gmtdefaults** man page for names and numbers of available fonts (or run **pstext –L**). The alignment refers to the part of the textstring 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.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)

–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Sets the clearance between the text and the surrounding box [15%]. Only used if **–W** is specified. Append the unit you want (**c**m, **i**nch, **m**eter. or **p**oint; if not given we consult **MEASURE_UNIT**) or % for a percentage of the font size.

–D Offsets the text from the projected (*x,y*) point by *dx,dy* [0/0]. Use **-Dj** to offset the text away from the point instead (i.e. the text's justification will determine the direction of the shift). 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.

–E Sets the viewpoint's azimuth and elevation (for perspective view) [180/90]. (Not implemented for paragraph mode).

–G Sets the gray-shade (0-255) or color (r/g/b, each 0-255) used for drawing the text. [Default is black]

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–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 size angle fontno justify linespace parwidth parjust), where *x y size angle fontno justify* are defined above, 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 as well as three more: @;*r/g/b*; changes the font color (@;; resets it), @:*size*: changes the font size (@:: resets it), and @_ toggles underline on/off. Separate paragraphs with a blank line.

–N Do NOT clip text at map boundaries [Default will clip].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S Draw text outline. Append pen attributes. (Not implemented for paragraph mode).

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Paint a rectangle beneath the text string. Set color [Default is no fill]. Append **o** to draw rectangle outline, add a *pen* to specify pen attributes [width = 1, color = black, texture = solid]. Choose upper case **O** to get a rounded rectangle. Choose lower case **c** to get a concave rectangle (only in paragraph mode). Choose upper case **C** to get a convex rectangle (only in paragraph mode).

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z For 3-D projections: Sets the z-level of the basemap [0]. (Not implemented for paragraph mode).

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–c Specifies the number of plot copies. [Default is 1].

EXAMPLES

To plot the outlines of the textstrings stored in the file text.d on a Mercator plot with the given specifications, use

```
pstext text.d –R-30/30/-10/20 –Jm0.1i –P –B5 –S0.5p > plot.ps
```

To add a typeset figure caption for a 3-inch wide illustration, use

```
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
```

BUGS

In paragraph mode, the presence of composite characters and other escape sequences may lead to unfortunate word splitting.

The **−N** option does not adjust the BoundingBox information so you may have to post-process the PostScript outout with epstool or ps2epsi to obtain a correct BoundingBox.

SEE ALSO

GMT(l), *psbasemap*(l)

NAME

pswiggle − Plot anomaly along track on a map

SYNOPSIS

pswiggle *xyz_files* **−J***parameters* **−R***west***/***east***/***south***/***north*[**r**] **−Z***scale* [**−A***azimuth*] [**−B**[**p**|**s**]*parameters*] [**−C***center*] [**−D***gap*] [**−E***azimuth/elevation*] [**−G***color*] [**−H**[**i**][*nrec*]] [**−Jz**|**Z***parameters*] [**−I***fix_az*] [**−K**] [**−M**[*flag*]] [**−N**] [**−O**] [**−P**] [**−S**[**x**]*lon0*/*lat0*/*length*[*units*]] [**−T***track_pen*] [**−U**[*/dx/dy/*][*label*]] [**−V**] [**−W***wiggle_pen*] [**−X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**−Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**−c***copies*] [**−:**[**i**|**o**]] [**−bi**[**s**|**S**|**d**|**D**][*ncol*]] [**−f***colinfo*]

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.

files List one or more file-names. If no files are given, **pswiggle** will read standard input.

−J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

−Jc*lon0/lat0/scale* (Cassini)
−Jj*lon0/scale* (Miller)
−Jm*scale* (Mercator - Greenwich and Equator as origin)
−Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
−Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
−Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
−Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
−Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
−Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
−Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
−Ju*zone/scale* (UTM - Universal Transverse Mercator)
−Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

−Ja*lon0/lat0/scale* (Lambert)
−Je*lon0/lat0/scale* (Equidistant)
−Jf*lon0/lat0/horizon/scale* (Gnomonic)
−Jg*lon0/lat0/scale* (Orthographic)
−Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

−Jb*lon0/lat0/lat1/lat2/scale* (Albers)
−Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
−Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–Z Gives anomaly scale in data-units/distance-unit.

OPTIONS

No space between the option flag and the associated arguments.

–A Sets the preferred positive azimuth. Positive wiggles will "gravitate" towards that direction.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Subtract *center* from the data set before plotting [0].

–D Means there is a data gap if 2 consecutive points are more than *gap* distance units apart. For longitude/latitude data *gap* is in km, else it is in the user's units.

–E Sets the viewpoint's azimuth and elevation [180/90].

–G Set fill of positive wiggles. [Default is black] Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I Set a fixed azimuth projection for wiggles [Default uses track azimuth, but see **–A**].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–M Multiple segment file. Segments are separated by a record whose first character is *flag*. [Default is '>'].

–N Paint negative wiggles instead of positive [Default].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S 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

–T Draw track [Default is no track]. Append pen attributes to use [Defaults: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Draw wiggle outline [Default is no outline]. Append pen attributes to use [Defaults: width = 1, color = black, texture = solid].

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 input columns].

–c Specifies the number of plot copies. [Default is 1].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

EXAMPLES

To plot the magnetic anomaly stored in the file track.xym along track @ 1000 nTesla/cm (after removing a mean value of 32000 Tesla), 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

pswiggle track.xym **–R**-20/10/-80/-60 **–JS**0/90/15**c** **–Z**1000 **–B**5 **–P** **–G**255/0/0 **–T**0.25**p**/0/0/255 **–S**1000 **–V** > track_xym.ps

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** **–Fb**5 **–N**4/0 **––D_FORMAT**=%.12lg > smoothed.xyz and plot this data set instead.

SEE ALSO

GMT(l), *filter1d*(l), *psbasemap*(l), *splitxyz*(l)

NAME

psxy − Plot lines, polygons, and symbols on maps

SYNOPSIS

psxy *files* **−J***parameters* **−R***west*/*east*/*south*/*north*[**r**] [**−A**] [**−B**[**p**|**s**]*parameters*] [**−C***cptfile*] [**−D***dx*/*dy*] [**−E**[**x**|**y**|**X**|**Y**][*cap*][/*pen*]] [**−G***fill*] [**−H**[**i**][*nrec*]] [**−K**] [**−L**] [**−N**] [**−M**[*flag*]] [**−O**] [**−P**] [**−S**[*symbol*][*size*]] [**−U**[/*dx*/*dy*/][*label*]] [**−V**] [**−W**[*pen*]] [**−X**[**a**|**c**|**r**][*x-shift*[**u**]]] [**−Y**[**a**|**c**|**r**][*y-shift*[**u**]]] [**−:**[**i**|**o**]] [**−c***copies*] [**−bi**[**s**|**S**|**d**|**D**][*ncol*]] [**−f***colinfo*]

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. Multiple segment files may be plotted using the **−M** option. If **−S** is not selected, a line connecting the data points will be drawn instead. To explicitly close polygons, use **−L**. Select a shade 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 color and outline/no outline, respectively. The *PostScript* code is written to standard output.

files List one or more file-names. If no files are given, **psxy** will read standard input.

−J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

−Jc*lon0/lat0/scale* (Cassini)
−Jj*lon0/scale* (Miller)
−Jm*scale* (Mercator - Greenwich and Equator as origin)
−Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
−Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
−Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
−Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
−Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
−Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
−Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
−Ju*zone/scale* (UTM - Universal Transverse Mercator)
−Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

−Ja*lon0/lat0/scale* (Lambert)
−Je*lon0/lat0/scale* (Equidistant)
−Jf*lon0/lat0/horizon/scale* (Gnomonic)
−Jg*lon0/lat0/scale* (Orthographic)
−Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

−Jb*lon0/lat0/lat1/lat2/scale* (Albers)
−Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
−Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–A Suppress drawing line segments as great circle Arcs. [Default draws great circle arcs].

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C Give a color palette file. When used with **–S**, lets 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, **psxy** expects the user to supply a multisegment line or polygon file (requires **–M**) and will look for **–Z***val* strings in each multisegment header. The *val* will control the color of either the line or polygon (if **–L** is set) via the cpt file.

–D Offset the plot symbol locations by the given amounts *dx/dy* [Default is no offset]. Only applies to symbols.

–E 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 [0.25c (or 0.1i)]. Pen attributes for error bars may also be set. [Defaults: width = 1, color = black, texture = solid]. 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% quartile), the 25% quartile, the 75% quartile, and the maximum (100% quartile) values. The 25-75% box may be filled by using **–G**.

–G Select color for filling of symbols or select color or pattern for filling of polygons [Default is no fill].
Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual

patterns.
Note when **–M** is chosen, **psxy** will search for **–G** and **–W** strings in all the subheaders and let any values thus found over-ride the command line settings.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Force closed polygons: connect the endpoints of the line-segment(s) and draw polygons. Also, in concert with **–C**, **–M**, and **–Z** settings in the headers will use the implied color for polygon fill [Default is polygon pen color].

–M Multiple segment file. Segments are separated by a record whose first character is *flag*. [Default is '>'].

–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 Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–S Plot symbols. If present, size is symbol size in the unit set in .gmtdefaults4 (unless c, i, m, 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 **–b**. Optionally, append c, i, m, p to indicate that the size information in the input data is in units of cm, inch, meter, or point, respectively. [Default is **MEASURE_UNIT**]. Note: if you give both size and symbol via the input file you must use **MEASURE_UNIT** to indicate the units used for the symbol size.
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 corresponding lowercase symbols all are circumscribed by the circle. Choose between these symbol codes:

–S- x-dash. *size* is the length of a short horizontal line segment.

–Sa st**a**r. *size* is diameter of circumscribing circle.

–Sb **b**ar 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* = 0. Append **b***base* to change this value.

–Sc **c**ircle. *size* is diameter of circle.

–Sd **d**iamond. *size* is diameter of circumscribing circle.

–Se **e**llipse. 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 if which we assume the ellipse axes are given in the same units as **–R**.

–Sf **f**ront. **–Sf***gap/size*[*dir*][*type*][:*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. Append *dir* to plot symbols on the **l**eft or **r**ight side of the front [Default is centered]. Append *type* to specify which symbol to plot: **b**ox, **c**ircle, **f**ault, **s**lip, or **t**riangle. [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 [Default is 0].

–Sh **h**exagon. *size* is diameter of circumscribing circle.

–Si **i**nverted triangle. *size* is diameter of circumscribing circle.

–Sk **k**ustom symbol. Append <name>/*size*, and we will look for a definition file called <name>.def in (1) the current directory or (2) in **$GMTHOME**/share/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 **l**etter 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 **ANNOT_FONT_PRIMARY**].

–Sg octa**g**on. *size* is diameter of circumscribing circle.

–Sn pe**n**tagon. *size* is diameter of circumscribing circle.

–Sp **p**oint. No size needs to be specified (1 pixel is used).

–Sq **q**uoted line, i.e., lines with annotations such as contours. Append [**d|f|n|l|x**]*info*[:*labelinfo*]. The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

d*dist*[c|i|m|p] or **D***dist*[d|e|k|m|n]

For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), **m** (meter), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), or **d** (spherical degree). [Default is 10**c** or 4**i**].

f*ffile.d* Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact mathces and points outside the region are skipped.

l|L*line1*[,*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].

n*n_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*[c|i|m|p] to enforce that a minimum distance separation between successive labels is enforced.

x|X*xfile.d*

Reads the multi-segment 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.

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:

+a*angle*

For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default].

+c*dx*[/*dy*]

Sets the clearance between label and optional text box. Append **c|i|m|p** to specify the unit or % to indicate a percentage of the label font size [15%].

+f*font* Sets the desired font [Default **ANNOT_FONT_PRIMARY**].

+g[*color*]

Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PAGE_COLOR**].

+j*just* Sets label justification [Default is CM].

+k*color*
Sets color of text labels [Default is **COLOR_BACKGROUND**].

+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 multisegment header (first scan for an embedded **–L***label* option, if not use the first word following the segment flag).

+Ld Take the Cartesian plot distances along the line as the label; append **c**|**i**|**m**|**p** as the unit [Default is **MEASURE_UNIT**].

+LD Calculate actual map distances; append **d**|**e**|**k**|**m**|**n** 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.

+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 boxsets [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

+r*min_rad*
Will not place labels where the line's radius of curvature is less than *min_rad* [Default is 0].

+s*size* Sets the desired font size in points [Default is 9].

+u*unit* Appends *unit* to all line labels. If *unit* starts with a leading hypen (-) 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 hypen (-) then there will be no space between label value and the prefix. [Default is no prefix].

–Sr **r**ectangle. No size needs to be specified, but the x- and y-dimensions must be found in columns 3 and 4.

–Ss **s**quare. *size* is diameter of circumscribing circle.

–St **t**riangle. *size* is diameter of circumscribing circle.

–Sv **v**ector. Direction (in degrees counter-clockwise from horizontal) and length must be found in columns 3 and 4. *size*, if present, will be interpreted as arrowwidth/headlength/headwidth [Default is 0.075**c**/0.3**c**/0.25**c** (or 0.03**i**/0.12**i**/0.1**i**)]. By default arrow attributes remains invariant to the length of the arrow. To have the size of the vector scale down with decreasing size, append **n***norm*, where vectors shorter than *norm* will have their attributes scaled by length/*norm*. To center vector on balance point, use **–Svb**; to align point with the vector head, use **–Svh**; to align point with the vector tail, use **–Svt** [Default]. To give the head point's coordinates instead of direction and length, use **–Svs**. Upper case **B, H, T, S** will draw a double-headed vector [Default is single head].

–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.)

–Sw pie **w**edge. 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. *size* is diameter of circumscribing circle.

–Sy y-dash. *size* is the length of a short vertical line segment.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Set pen attributes for lines or the outline of symbols. [Defaults: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is the required number of columns given the chosen settings].

–c Specifies the number of plot copies. [Default is 1].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

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

```
psxy DSDP.xy –R150/154/18/23 –Jm5c –Sc0.25c –Gred –B1g15m | lpr
```

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

```
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

```
psxy trench.xy –R150/200/20/50 –Jm0.15i –Sf0.8i/0.1ilt –Gwhite –W –B10 | lpr br
```

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

```
psxy misc.d –R0/100/-50/100 –JX6i –S –Ccpt –B20 > t.ps
```

BUGS

The **–N** option does not adjust the BoundingBox information so you may have to post-process the PostScript outout with epstool or ps2epsi to obtain a correct BoundingBox.

psxy 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.

CUSTOM SYMBOLS

psxy and **psxyz** allows users to define and plot their own custom symbols. This is done by encoding the symbol using a simple plotting code described below. Put all the plotting 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 **–Sk***tsunami/size*. The definition file can contain any number of plot code records, as well as blank lines and comment lines (starting with #). **psxy** and **psxyz** will look for the definition files in (1) the current directory and (2) the **$GMTHOME**/share/custom, 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. With the exception of the circle, these can only be painted, not filled with a pattern (use the freeform polygons to include a triangle that can be filled with a pattern). Generate freeform polygons by starting with an anchor point (append [**–W***pen*] and [**–G***fill*] to indicate pen and fill attributes):

x0 *y0* **M**

and draw a straight line from the current point to the next point with

x *y* **D**

or add an arc by using

xc *yc* *r* *dir1* *dir2* **A**

When a record other than the **D** or **A** is encountered, the polygon is closed and considered complete. The optional pen and fill setting hardwires particular values for this feature. If not present the polygon's characteristics are determined by the command line settings for pen and fill. To deactivate fill or outline for any given polygon, give **–G–** or **–W–**. To add other geometric shapes to your custom symbol, add any number of the following plot code records (each accepts the optional [**–W***pen*] and [**–G***fill*] at the end):

star:	*x*	*y*	*size*	**a**		
circle:	*x*	*y*	*size*	**c**		
cross:	*x*	*y*	*size*	**x**		
diamond:	*x*	*y*	*size*	**d**		
ellipse:	*x*	*y*	*dir*	*major*	*minor*	**e**
hexagon:	*x*	*y*	*size*	**h**		
invtriangle:	*x*	*y*	*size*	**i**		
letter:	*x*	*y*	*size*	*string*	**l**	
octagon:	*x*	*y*	*size*	**g**		
pentagon:	*x*	*y*	*size*	**n**		
rect:	*x*	*y*	*xwidth*	*ywidth*	**r**	
square:	*x*	*y*	*size*	**s**		
triangle:	*x*	*y*	*size*	**t**		

wedge:	*x*	*y*	*radius*	*dir1*	*dir2*	**w**

When designing your symbol, the *x, y* and other dimensions are relative to a symbol of size 1, and all the dimensions will be scaled by the actual symbol size chosen at run-time. To design a symbol, make a grid paper with **psbasemap –R**-0.5/0.5/-0.5/0.5 **–JX**4**i –Ba**0.1g0.05 **–P** > grid.ps and draw your symbol, centering it on (0,0). For examples of symbols, see the set supplied with **GMT** in **$GMTHOME**/share/custom.

SEE ALSO

GMT(l), *psbasemap*(l), *psxyz*(l)

NAME

psxyz – Plot lines, polygons, and symbols in 3-D

SYNOPSIS

psxyz *files* **–J***parameters* **–Jz|Z***parameters* **–R***west***/***east***/***south***/***north*[**/***zmin***/***zmax*][**r**] [**–B**[**p|s**]*parameters*] [**–C***cptfile*] [**–E***azimuth***/***elevation*] [**–G***fill*] [**–H**[**i**][*nrec*]] [**–K**] [**–L**] [**–M**[*flag*]] [**–N**] [**–O**] [**–P**] [**–Q**] [**–S**[*symbol*][*size*]] [**–U**[*/dx/dy/*][*label*]] [**–V**] [**–W**[*pen*]] [**–X**[**a|c|r**][*x-shift*[**u**]]] [**–Y**[**a|c|r**][*y-shift*[**u**]]] [**–Z***zlevel*] [**–:**[**i|o**]] [**–c***copies*] [**–bi**[**s|S|d|D**][*ncol*]] [**–f***colinfo*]

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. Multiple segment files may be plotted using the **–M** option. If no symbols are selected, a line will be drawn. To explicitly close polygons, use **–L**. Select a shade 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 color and outline/no outline, respectively. The *PostScript* code is written to standard output.

files List one or more file-names. If no files are given, **psxyz** will read standard input.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–Jz Sets the vertical scaling (for 3-D maps). Same syntax as **–Jx**.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

No space between the option flag and the associated arguments.

–B Sets map boundary annotation and tickmark intervals; see the **psbasemap** man page for all the details.

–C 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 a multisegment line or polygon file (requires **–M**) where each segment header contains a **–Z***val* string. The *val* will control the color of the line or polygon (use **–L**) via the cpt file.

–E Sets the viewpoint's azimuth and elevation [180/90].

–G Select color for filling of symbols or select color or pattern for filling of polygons [Default is no fill].
Specify the grey shade (0–255) or color (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 0–100%; or valid color name). Alternatively, specify **–Gp***dpi/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 **–GP** for inverse video, or append **:F***color*[**B**[*color*]] to specify fore- and background colors (use *color* = - for transparency). See **GMT** Cookbook & Technical Reference Appendix E for information on individual patterns.
Note when **–M** is chosen, **psxyz** will search for **–G** and **–W** strings in all the subheaders and let any found values over-ride the command line settings.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–K More *PostScript* code will be appended later [Default terminates the plot system].

–L Force closed polygons: will connect the endpoints of the line-segment(s) and draw polygons.

–M Multiple segment file. Segments are separated by a record whose first character is *flag*. [Default is '>'].

–N Do NOT skip symbols that fall outside map border [Default plots points inside border only].

–O Selects Overlay plot mode [Default initializes a new plot system].

–P Selects Portrait plotting mode [**GMT** Default is Landscape, see **gmtdefaults** to change this].

–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 Plot symbols. *size* is symbol size in the unit set in .gmtdefaults4 (unless **c**, **i**, **m**, 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 **–b**. Optionally, append c, i, m, p to indicate that the size information in the input data is in units of cm, inch, meter, or point, respectively. [Default is **MEASURE_UNIT**]. The uppercase symbols **A, C, D, G, H, I, N, S, T** are normalized to have the same area as a circle of given size, while the corresponding lowercase symbols are circumscribed by the circle. Choose between:

–S- x-dash. *size* is the length of a short horizontal (x-dir) line segment.

–Sa st**a**r. *size* is diameter of circumscribing circle.

–Sb (**b**)ar 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* = 0. Append b*base* to change this value.

–Sc (**c**)ircle. *size* is diameter of circle.

–Sd (**d**)iamond. *size* is diameter of circumscribing circle.

–Se **e**llipse. Direction (in degrees counterclockwise 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 if which we assume the ellipse axes are given in the same units as **–R**.

–Sf **f**ront. **–Sf***gap/size*[*dir*][*type*][:*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. Append *dir* to plot symbols on the **l**eft or **r**ight side of the front [Default is centered]. Append *type* to specify which symbol to plot: **b**ox, **c**ircle, **f**ault, **s**lip, or **t**riangle. [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].

–Sg octa**g**on. *size* is diameter of circumscribing circle.

–Sh **h**exagon. *size* is diameter of circumscribing circle.

–Si **i**nverted triangle. *size* is diameter of circumscribing circle.

–Sk **k**ustom symbol. Append <name>/*size*, and we will look for a definition file called <name>.def in (1) the current directory or (2) in **$GMTHOME**/share/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 in the **psxy** man page.

–Sl **l**etter 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 **ANNOT_FONT_PRIMARY**].

–Sn pe**n**tagon. *size* is diameter of circumscribing circle.

–So c(**o**)lumn (3-D) extending from *base* to z. *size* sets base width (Use *xsize/ysize* if not the same). Append **u** if *size* is in x-units [Default is plot-distance units]. By default, *base* = 0. Append b*base* to change this value. The facet colors will be modified to simulate shading. Use **–SO** to disable such 3-D illumination.

–Sp (**p**)oint. No size needs to be specified.

–Sq **q**uoted 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|f|n|l|x**]*info*[:*labelinfo*]. The required argument controls the placement of labels along the quoted lines. Choose among five controlling algorithms:

d*dist*[c|i|m|p] or **D***dist*[d|e|k|m|n]
For lower case **d**, give distances between labels on the plot in your preferred measurement unit **c** (cm), **i** (inch), **m** (meter), or **p** (points), while for upper case **D**, specify distances in map units and append the unit; choose among **e** (m), **k** (km), **m** (mile), **n** (nautical mile), or **d** (spherical degree). [Default is 10**c** or 4**i**].

f*ffile.d* Reads the ascii file *ffile.d* and places labels at locations in the file that matches locations along the quoted lines. Inexact mathces and points outside the region are skipped.

l|L*line1*[,*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].

n*n_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*[c|i|m|p] to enforce that a minimum distance separation between successive labels is enforced.

x|X*xfile.d*
Reads the multi-segment 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.

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:

+a*angle*
For annotations at a fixed angle, **+an** for line-normal, or **+ap** for line-parallel [Default].

+c*dx*[/*dy*]
Sets the clearance between label and optional text box. Append **c|i|m|p** to specify the unit or % to indicate a percentage of the label font size [15%].

+f*font* Sets the desired font [Default **ANNOT_FONT_PRIMARY**].

+g[*color*]
Selects opaque text boxes [Default is transparent]; optionally specify the color [Default is **PAGE_COLOR**].

+j*just* Sets label justification [Default is CM].

+k*color*
Sets color of text labels [Default is **COLOR_BACKGROUND**].

+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 multisegment header (first scan for an embedded **–L***label* option, if not use the first word following the segment flag).

+Ld Take the Cartesian plot distances along the line as the label; append **c**|**i**|**m**|**p** as the unit [Default is **MEASURE_UNIT**].

+LD Calculate actual map distances; append **d**|**e**|**k**|**m**|**n** 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.

+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 boxsets [Default is no outline]; optionally specify pen for outline [Default is width = 0.25p, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

+r*min_rad*
Will not place labels where the line's radius of curvature is less than *min_rad* [Default is 0].

+s*size* Sets the desired font size in points [Default is 9].

+u*unit* Appends *unit* to all line labels. If *unit* starts with a leading hypen (-) 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 hypen (-) then there will be no space between label value and the prefix. [Default is no prefix].

–Sr **r**ectangle. No size needs to be specified, but the x- and y-dimensions must be found in columns 4 and 5.

–Ss (**s**)quare. *size* is diameter of circumscribing circle.

–St (**t**)riangle. *size* is diameter of circumscribing circle.

–Su c(**u**)be (3-D). *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 (**v**)ector. Direction and length must be found in columns 4 and 5. *size* means *arrowwidth/headlength/headwidth* in [[Default is 0.075**c**/0.3**c**/0.25**c** (or 0.03**i**/0.12**i**/0.1**i**)]. By default arrow attributes remains invariant to the length of the arrow. To have the size of the vector scale down with decreasing size, append **n***norm*, where vectors shorter than *norm* will have their attributes scaled by length/*norm*. To center vector on balance point, use **–Svb**; to align point with the vector head, use **–Svh**; to align point with the vector tail, use **–Svt** [Default]. To give the head point's coordinates instead of direction and length, use **–Svs**. Upper case **B, H, T, S** will draw a double-headed vector [Default is single head].

–SV Same as **–Sv**, except azimuth 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.)

–Sw pie **w**edge. 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 (**x**)cross. *size* is diameter of circumscribing circle.

–Sy y-dash. *size* is the length of a short horizontal (y-dir) line segment.

–Sz zdash. *size* is the length of a short vertical (z-dir) line segment.

–U Draw Unix System time stamp on plot. User may specify where the lower left corner of the stamp should fall on the page relative to lower left corner of plot. Optionally, append a label, or **c** (which will plot the command string.). The **GMT** parameters **UNIX_TIME** and **UNIX_TIME_POS** can affect the appearance; see the **gmtdefaults** man page for details.

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Set pen attributes for lines or the outline of symbols. [Defaults: width = 1, color = black, texture = solid]. *pen* is a comma delimetered list of *width*, *color* and *texture*, each of which is optional. *width* can be indicated as a measure (points, centimeters, inches) or as **faint**, **thin**[**ner**|**nest**], **thick**[**er**|**est**], **fat**[**ter**|**test**], or **obese**. *color* specifies a grey shade (0–255) or color (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 0–100%; or valid color name). *texture* is a combination of dashes '-' and dots '.'.

–X –Y Shift origin of plot by (*x-shift,y-shift*), and opetionally append units (**c**, **i**, **m**, **p**). Prepend **a** for absolute coordinates; the default (**r**) will reset plot origin. Give **c** to center plot using current page size.

–Z For 3-D projections: Sets the z-level of the basemap [0].

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is the required number of columns given the settings].

–c Specifies the number of plot copies. [Default is 1].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

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:

psxyz heights.xyz **–R**0/10/0/10/0/100 **–Jx**1.25**c** **–Jz**0.125**c** **–So**1.25**c** **–G**0/0/255 **–B**2:XLABEL:/2:YLABEL:/10:ZLABEL::."3-D PLOT":15 **–E**135/30 **–Uc** **–W** **–P** > heights.ps

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 outout with epstool or ps2epsi to obtain a correct BoundingBox.

SEE ALSO

GMT(l), *psbasemap*(l), *psxy*(l)

NAME

sample1d – Resampling of 1-D data sets

SYNOPSIS

sample1d *infile* [**–Fl|a|c**] [**–H**[**i**][*nrec*]] [**–I***xinc*] [**–M**[**i|o**][*flag*]] [**–N***knotfile*] [**–S***xstart*] [**–T***x_col*] [**–V**] [**–b**[**i|o**][**s|S|d|D**][*ncol*]] [**–f**[**i|o**]*colinfo* **]**

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.

infile This is a multi-column ASCII [of binary, see **–b**] file with one column containing the independent variable (which must be monotonically in/de-creasing) and the remaining columns holding misc. data values. If no file is provided, **sample1d** reads from standard input.

OPTIONS

No space between the option flag and the associated arguments.

–F Choose from **l** (Linear), **a** (Akima spline), and **c** (natural cubic spline) [Default is **–Fa**]. You may change the default interpolant; see **INTERPOLANT** in your .gmtdefaults4 file.

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I *xinc* defines the sampling interval. [Default is the separation between the first and second abscissa point in the *infile*]

–M Multiple segment file. Segments are separated by a record whose first character is *flag*. [Default is '>'].

–N *knotfile* is an optional ASCII file with the x locations where the data set will be resampled in the first column

–S For equidistant sampling, *xstart* indicates the location of the first output value. [Default is the smallest even multiple of *xinc* inside the range of *infile*]

–T Sets the column number of the independent variable [Default is 0 (first)].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 (or at least the number of columns implied by **–T**)].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i|o**]**g** means **–f**[**i|o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

To resample the file profiles.tdgmb, which contains (time,distance,gravity,magnetics,bathymetry) records, at 1km equidistant intervals using Akima's spline, use

sample1d profiles.tdgmb **−I**1 **−Fa −T**1 > 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

sample1d depths.dt **−N**grav_pos.dg **−Fc** > new_depths.dt

SEE ALSO

GMT(l), *filter1d*(l)

NAME

spectrum1d – compute auto– [and cross–] spectra from one [or two] timeseries.

SYNOPSIS

spectrum1d [*x[y]file*] **–S***segment_size*] [**–C**[**xycnpago**]] [**–D***dt*] [**–N***name_stem*] [**–V**] [**–W**] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

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.

REQUIRED ARGUMENTS

x[y]file ASCII (or binary, see **–bi**) file holding X(t) [Y(t)] samples in the first 1 [or 2] columns. If no file is specified, **spectrum1d** will read from standard input.

–S *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.

OPTIONS

–C Read the first two columns of input as samples of two timeseries, 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.

–D *dt* Set the spacing between samples in the timeseries [Default = 1].

–N *name_stem* Supply the name stem to be used for output files [Default = "spectrum"].

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–W Write Wavelength rather than frequency in column 1 of the output file[s] [Default = frequency, (cycles / *dt*)].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 2 output columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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

```
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 | spectrum1d –S256 –D1.5 –Ndata –C
```

SEE ALSO

GMT(l), *grdfft*(l)

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.

NAME

splitxyz − filter to divide (x,y,z[,distance,heading]) data into (x,y,z) track segments.

SYNOPSIS

splitxyz [*xyz[dh]file*] **−C***course_change* [**−A***azimuth*/*tolerance*] [**−D***minimum_distance*] [**−F***xy_filter*/*z_filter*] [**−G***gap_distance*] [**−H**[**i**][*nrec*]] [**−M**] [**−N***namestem*] [**−Q***flags*] [**−S**] [**−V**] [**−Z**] [**−:**[**i**|**o**]] [**−b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**−f**[**i**|**o**]*colinfo* **]**

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. The output is always in the ASCII format; input may be ASCII or binary (see **−b**).

xyz[dh]file(s)
3 (but see **−Z**) [or 5] column ASCII file [or binary, see **−b**] 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 file is specified, **splitxyz** will read from standard input.

−C Terminate a segment when a course change exceeding *course_change* degrees of heading is detected.

OPTIONS

−A 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].

−D Do not write a segment out unless it is at least *minimum_distance* units long [0]

−F 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].

−G Do not let a segment have a gap exceeding *gap_distance*; instead, split it into two segments. [Default ignores gaps].

−H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **−Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

−M Use Map units. Then x,y are in degrees of longitude, latitude, and distances in kilometers. [Default: distances are cartesian in same units as x,y].

−N Create Named output files, writing each segment to a separate file in the working directory named *namestem*.profile#, where # increases consecutively from 1. [Default writes entire output to stdout, separating segments by sub-headings that start with > marks].

−Q 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 **−Q***xyzdh* (**−Q***xydh* if **−Z** is set)].

–S d and h is 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, according to **–M** option]

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–Z Data have x,y only (no z-column).

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2, 3, or 5 input columns as set by **–S, –Z**].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 1-5 output columns as set by **–Q**].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

Suppose you want to make a wiggle plot of magnetic anomalies on segments oriented approximately east–west from a cruise called cag71 in the region –R300/315/12/20. You want to use a 100km low–pass filter to smooth the tracks and a 500km high–pass filter to detrend the magnetic anomalies. Try this:

```
gmtlist cag71 –R300/315/12/20 –Fxyzdh | splitxyz –A90/15 –F100/-500 –M -D100 –S –V | pswiggle
–R300/315/12/20 –Jm0.6 –Ba5f1:.cag71: –T1 –W3 –G200 –Z200 > cag71_wiggles.ps
```

MGD-77 users: For this application we recommend that you extract d, h 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*.profile# (when gap exceeds 100 km). Try this:

```
splitxyz survey.bin –Nsurvey –V –G100 –D100 –: –M –bi3
```

SEE ALSO

GMT(l), *mgd77list*(l), *pswiggle*(l)

NAME

surface – adjustable tension continuous curvature surface gridding algorithm

SYNOPSIS

surface [*xyzfile*] **–G***outputfile.grd* **–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]] **–R***west*/*east*/*south*/*north*[**r**] [**–A***aspect_ratio*] [**–C***convergence_limit*] [**–H**[**i**][*nrec*]] [**–Ll***lower*] [**–Lu***upper*] [**–N***max_iterations*] [**–Q**] [**–S***search_radius*[**m**]] [**–T***tension_factor*[**i**|**b**]] [**–V**] [**–Z***over-relaxation_factor*] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

DESCRIPTION

surface reads randomly-spaced (x,y,z) triples from standard input [or *xyzfile*] and produces a binary grdfile 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 preprocess 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 grdfile with values, or simply be the minimum/maximum input data values.

xyzfile 3 column ASCII file [or binary, see **–b**] holding (x,y,z) data values. If no file is specified, **surface** will read from standard input.

–G Output file name. Output is a binary 2-D *.grd* file.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

–A 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.]

–C 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.]

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

–L Impose limits on the output solution. **l***lower* sets the lower bound. *lower* can be the name of a grdfile with lower bound values, a fixed value, **d** to set to minimum input value, or **u** for unconstrained [Default]. **u***upper* sets the upper bound and can be the name of a grdfile with upper bound values, a fixed value, **d** to set to maximum input value, or **u** for unconstrained [Default].

–N 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**.

–S Search radius. Enter *search_radius* in same units as x,y data; append **m** to indicate minutes. 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 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 **–T***tension_factor***i** to set interior tension, and **–T***tension_factor***b** to set boundary tension. If you do not append **i** or **b**, both will be set to the same value. [Default = 0 for both gives minimum curvature solution.]

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"]. **–Vl** will report the convergence after each iteration; **–V** will report only after each regional grid is converged.

–Z 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 **–Vl** option. [Default = 1.4 converges quickly and is almost always stable.]

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 input columns].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

EXAMPLES

To grid 5 by 5 minute gravity block means from the ASCII data in hawaii_5x5.xyg, using a *tension_factor* = 0.25, a *convergence_limit* = 0.1 milligal, writing the result to a file called hawaii_grd.grd, and monitoring each iteration, try:

surface hawaii_5x5.xyg **−R**198/208/18/25 **−I5m −G**hawaii_grd.grd **−T**0.25 **−C**0.1 **−VL**

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 **D_FORMAT** in your .gmtdefaults4 file prior to running **blockm***, or choose binary input and/or output using single or double precision storage.

SEE ALSO

blockmean(l), *blockmedian*(l), *blockmode*(l), *GMT*(l), *nearneighbor*(l), *triangulate*(l)

REFERENCES

Smith, W. H. F, and P. Wessel, 1990, Gridding with continuous curvature splines in tension, *Geophysics*, 55, 293−305.

NAME

trend1d − Fit a [weighted] [robust] polynomial [or Fourier] model for y = f(x) to xy[w] data.

SYNOPSIS

trend1d −Fxymrw −N[f]*n_model***[r]** [*xy[w]file*] [**−C***condition_number*] [**−H[i]**[*nrec*]] [**−I**[*confidence_level*]] [**−V**] [**−W**] [**−:[i|o]**] [**−b[i|o][s|S|d|D]**[*ncol*]] [**−f[i|o]***colinfo* **]**

DESCRIPTION

trend1d reads x,y [and w] values from the first two [three] columns on standard input [or *xy[w]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.

REQUIRED ARGUMENTS

−F 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.

−N 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 **−N***3***r**.

OPTIONS

xy[w]file
ASCII [or binary, see **−b**] file containing x,y [w] values in the first 2 [3] columns. If no file is specified, **trend1d** will read from standard input.

−C 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.].

−H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **−Hi** if only input data should have header records [Default will write out header records if the input data have them].

−I 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 Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

−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.]

−: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

−bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 (or 3 if **−W** is set) columns].

−bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 1-5 columns as given by **−F**].

−f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **−f[i|o]g** means **−f[i|o]**0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

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), ... 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 + ... 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.

EXAMPLES

To remove a linear trend from data.xy by ordinary least squares, use:

trend1d data.xy **–F**xr **–N**2 > detrended_data.xy

To make the above linear trend robust with respect to outliers, use:

trend1d data.xy **–F**xr **–N**2**r** > 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:

trend1d data.xy **–Nf**20**r –I –V**

SEE ALSO

GMT(l), *gmtmath*(l), *grdtrend*(l), *trend2d*(l)

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.

NAME

trend2d – Fit a [weighted] [robust] polynomial model for z = f(x,y) to xyz[w] data.

SYNOPSIS

trend2d –Fxyzmrw –N*n_model*[**r**] [*xyz[w]file*] [**–C***condition_number*] [**–H**[**i**][*nrec*]][**–I**[*confidence_level*]] [**–V**] [**–W**] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

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 **–N***n_model*, the number of model parameters to use; thus, **–N***4* fits a bilinear trend, **–N***6* 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.

–F 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.

–N 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 **–N***4***r**.

OPTIONS

xyz[w]file

ASCII [or binary, see **–b**] file containing x,y,z [w] values in the first 3 [4] columns. If no file is specified, **trend2d** will read from standard input.

–C 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.].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I 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 Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–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.]

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 (or 4 if **–W** is set) input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is 1-6 columns as set by **–F**].

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

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 **–N***n_model***r** (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.

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

EXAMPLES

To remove a planar trend from data.xyz by ordinary least squares, use:

trend2d data.xyz **–F**xyr **–N**2 > detrended_data.xyz

To make the above planar trend robust with respect to outliers, use:

trend2d data.xzy **–F**xyr **–N**2**r** > detrended_data.xyz

To find out how many terms (up to 10) in a robust interpolant are significant in fitting data.xyz, use:

trend2d data.xyz **–N**10**r** **–I** **–V**

SEE ALSO

GMT(l), *grdmath*(l), *grdtrend*(l), *trend1d*(l)

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.

NAME

triangulate – Perform optimal Delauney triangulation and gridding

SYNOPSIS

triangulate *infiles* [**–Dx**|**y**] [**–E***empty*] [**–G***grdfile*] [**–H**[**i**][*nrec*]] [**–I***xinc*[*unit*][**=**|**+**][/*yinc*[*unit*][**=**|**+**]]] [**–J***parameters*] [**–M**[**i**|**o**][*flag*]] [**–R***west*/*east*/*south*/*north*[**r**]] [**–V**] [**–Z**] [**–:**[**i**|**o**]] [**–b**[**i**|**o**][**s**|**S**|**d**|**D**][*ncol*]] [**–f**[**i**|**o**]*colinfo* **]**

DESCRIPTION

triangulate reads one or more ASCII [or binary] files (or standard input) containing x,y[,z] and performs Delauney triangulation, i.e., it find how the points should be connected to give the most equilateral triangulation possible. If a map projection 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 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). This choice is made during the **GMT** installation.

infiles Data files with the point coordinates in ASCII (or binary; see **–b**). If no files are given the standard input is read.

OPTIONS

–D Take either the *x*- or *y*-derivatives of surface represented by the planar facets (only used when **–G** is set).

–E Set the value assigned to empty nodes when **–G** is set [NaN].

–G Use triangulation to grid the data onto an even grid (specified with **–I, –R**). 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).

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them].

–I *x_inc* [and optionally *y_inc*] sets the grid size for optional grid output (see **–G**). Append **m** to indicate minutes or **c** to indicate seconds.

–J Selects the map projection. Scale is UNIT/degree, 1:xxxxx, or width in UNIT (upper case modifier). UNIT is cm, inch, or m, depending on the **MEASURE_UNIT** setting in .gmtdefaults4, but this can be overridden on the command line by appending **c**, **i**, or **m** to the scale/width value. For map height, max dimension, or min dimension, append **h**, **+**, or **-** to the width, respectively.
More details can be found in the **psbasemap** man pages.

CYLINDRICAL PROJECTIONS:

–Jc*lon0/lat0/scale* (Cassini)
–Jj*lon0/scale* (Miller)
–Jm*scale* (Mercator - Greenwich and Equator as origin)
–Jm*lon0/lat0/scale* (Mercator - Give meridian and standard parallel)
–Joa*lon0/lat0/azimuth/scale* (Oblique Mercator - point and azimuth)
–Job*lon0/lat0/lon1/lat1/scale* (Oblique Mercator - two points)
–Joc*lon0/lat0/lonp/latp/scale* (Oblique Mercator - point and pole)
–Jq*lon0/scale* (Equidistant Cylindrical Projection (Plate Carree))
–Jt*lon0/scale* (TM - Transverse Mercator, with Equator as y = 0)
–Jt*lon0/lat0/scale* (TM - Transverse Mercator, set origin)
–Ju*zone/scale* (UTM - Universal Transverse Mercator)
–Jy*lon0/lats/scale* (Basic Cylindrical Projection)

AZIMUTHAL PROJECTIONS:

–Ja*lon0/lat0/scale* (Lambert)
–Je*lon0/lat0/scale* (Equidistant)
–Jf*lon0/lat0/horizon/scale* (Gnomonic)
–Jg*lon0/lat0/scale* (Orthographic)
–Js*lon0/lat0/*[*slat/*]*scale* (General Stereographic)

CONIC PROJECTIONS:

–Jb*lon0/lat0/lat1/lat2/scale* (Albers)
–Jd*lon0/lat0/lat1/lat2/scale* (Equidistant)
–Jl*lon0/lat0/lat1/lat2/scale* (Lambert)

MISCELLANEOUS PROJECTIONS:

–Jh*lon0/scale* (Hammer)
–Ji*lon0/scale* (Sinusoidal)
–Jk[**f**|**s**]*lon0/scale* (Eckert IV (f) and VI (s))
–Jn*lon0/scale* (Robinson)
–Jr*lon0/scale* (Winkel Tripel)
–Jv*lon0/scale* (Van der Grinten)
–Jw*lon0/scale* (Mollweide)

NON-GEOGRAPHICAL PROJECTIONS:

–Jp[**a**]*scale*[*/origin*][**r**|**z**] (Polar coordinates (theta,r))
–Jx*x-scale*[**d**|**l**|**p***pow*|**t**|**T**][*/y-scale*[**d**|**l**|**p***pow*|**t**|**T**]] (Linear, log, and power scaling)

–M Output triangulation network as multiple line segments separated by a record whose first character is *flag* [>]. To plot, use **psxy** with the **–M** option (see Examples).

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

–V Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–Z Controls whether binary data file has two or three columns [2]. Ignored if **–b** is not set.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 2 input columns].

–bo Selects binary output. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of desired columns in your binary output file. [Default is same as input]. Node ids are stored as binary 4-byte integer triplets. **–bo** is ignored if **–M** is selected.

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

ASCII FORMAT PRECISION

The ASCII output formats of numerical data are controlled by parameters in your .gmtdefaults4 file. Longitude and latitude are formatted according to **OUTPUT_DEGREE_FORMAT**, whereas other values are formatted according to **D_FORMAT**. 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 **D_FORMAT** setting.

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

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

triangulate samples.xyz **–bo** **–R**0/30/0/30 **–I**2 **–G**surf.grd > samples.ijk

To draw the optimal Delauney triangulation network based on the same file using a 15-cm-wide Mercator map, use

triangulate samples.xyz **–M** **–R**-100/-90/30/34 **–JM**15**c** | psxy **–M** **–R**-100/-90/30/34 **–JM**15**c** **–W**0.5**p** **–B**1 > network.ps

SEE ALSO

GMT(l), *pscontour*(l)

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. www.cs.cmu.edu/~quake/triangle.html

NAME

xyz2grd – Converting an ASCII or binary table to grd file format

SYNOPSIS

xyz2grd *xyzfile* **–G***grdfile* **–I***xinc*[*unit*][=|+][/*yinc*[*unit*][=|+]] **–R***west*/*east*/*south*/*north*[**r**] [**–A**[**n**|**z**|**u**|**l**]] [**–D***xname*/*yname*/*zname*/*scale*/*offset*/*title*/*remark*] [**–E**[*nodata*]] [**–F**] [**–H**[**i**][*nrec*]] [**–N***nodata*] [**–S**[*zfile*]] [**–V**] [**–Z**[*flags*]] [**–:**[**i**|**o**]] [**–bi**[**s**|**S**|**d**|**D**][*ncol*]] [**–f***colinfo*]

DESCRIPTION

xyz2grd reads a z or xyz table and creates a binary grdfile. **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 average 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.)

[xy]zfile

ASCII [or binary] file holding z or (x,y,z) values. xyz triplets do not have to be sorted (for binary triplets, see **–b**). 1-column z tables must be sorted and the **–Z** must be set).

–G *grdfile* is the name of the binary output grdfile.

–I *x_inc* [and optionally *y_inc*] is the grid spacing. Append **m** to indicate minutes or **c** to indicate seconds. If one of the units **e**, **k**, **i**, or **n** is appended instead, the increment will be assumed to be in meter, km, miles, or nautical miles, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on **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. 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 increment; 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 Appendix B for details.

–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 wesn. The two shorthands **–Rg –Rd** stand for global domain (0/360 or -180/+180 in longitude respectively, with -90/+90 in latitude). For calendar time coordinates you may either give relative time (relative to the selected **TIME_EPOCH** and in the selected **TIME_UNIT**; append **t** to **–JX**|**x**), or 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 as indicated (however, input/output and plotting formats are flexible).

OPTIONS

–A 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. Append **l** or **u** to find the lowest (minimum) or upper (maximum) value at each node, respectively. [Default (no **–A** option) will calculate mean value]. Ignored if **–Z** is given.

–D Give values for *xname, yname, zname, scale, offset, title,* and *remark*. To leave some of these values untouched, specify = as the value.

–E Convert an ESRI ArcInfo ASCII interchange grid format file to a **GMT** grid. Append *nodata* which is a data value that should be set to NaN in the grid [Default is to read the optional 6th record in the file and get *nodata*]. The values normally given by **–R**, **–I**, and **–F** are determined from the ESRI header instead.

–F Force pixel registration [Default is grid registration].

–H Input file(s) has Header record(s). Number of header records can be changed by editing your .gmtdefaults4 file. If used, **GMT** default is 1 header record. Use **–Hi** if only input data should have header records [Default will write out header records if the input data have them]. Not used with binary data.

–N 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 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 Selects verbose mode, which will send progress reports to stderr [Default runs "silently"].

–Z 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 = ymax or **B**(ottom) if first row is y = ymin. 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. For gridline registered grids: If data are periodic in x but the incoming data do not contain the (redundant) column at x = xmax, append **x**. For data periodic in y without redundant row at y = ymax, append **y**. Append **s***n* to skip the first *n* number of bytes (probably a header). If the byte-order needs to be swapped, append **w**. Select one of several data types (all binary except **a**):

a	ASCII representation
c	signed 1-byte character
u	unsigned 1-byte character
h	short 2-byte integer
i	4-byte integer
l	long (4- or 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.

–: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

–bi Selects binary input. Append **s** for single precision [Default is **d** (double)]. Uppercase **S** (or **D**) will force byte-swapping. Optionally, append *ncol*, the number of columns in your binary file if it exceeds the columns needed by the program. [Default is 3 input columns]. This option only applies to xyz input files; see **–Z** for z tables.

–f Special formatting of input and output columns (time or geographical data). Specify **i**(nput) or **o**(utput) [Default is both input and output]. Give one or more columns (or column ranges) separated by commas. Append **T** (Absolute calendar time), **t** (time relative to chosen **TIME_EPOCH**), **x** (longitude), **y** (latitude), or **f** (floating point) to each column or column range item. Shorthand **–f**[**i**|**o**]**g** means **–f**[**i**|**o**]0**x**,1**y** (geographic coordinates).

GRID VALUES PRECISION

Regardless of the precision of the input data, GMT programs that create gridded files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and futhermore 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 loose 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.

EXAMPLES

To create a grdfile from the ASCII data in hawaii_grv.xyz, use

xyz2grd hawaii_grv.xyz **–D**degree/degree/mGal/1/0/"Hawaiian Gravity"/"GRS-80 Ellipsoid used" **–G**hawaii_grv_new.grd **–R**198/208/18/25 **–I**5**m –V**

To create a grdfile from the raw binary (3-column, single-precision) scanline-oriented data raw.b, use

xyz2grd raw.b **–D**m/m/m/1/0/=/= **–G**raw.grd **–R**0/100/0/100 **–I**1 **–V –Z –b**3

To make a grdfile 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

xyz2grd topo30. **–D**m/m/m/1/0/=/= **–G**ustopo.grd **–R**234/294/24/50 **–I**30c **–N**-9999 **–V –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

xyz2grd floats.bin **–S**new_floats.bin **–V –Zf**

SEE ALSO

GMT(l), *grd2xyz*(l), *grdedit*(l)

NAME

GMT − The Generic Mapping Tools data processing and display software package

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 24bit color. Linear, log10, and power scaling is supported in addition to 25 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.

SYNOPSIS

GMT is also a wrapper script that can start any of the programs:

GMT module module-options

where module is the name of a **GMT** program and the options are those that pertain to that particular program.

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.

blockmean	L2 (x,y,z) data filter/decimator
blockmedian	L1 (x,y,z) data filter/decimator
blockmode	Mode (x,y,z) data filter/decimator
filter1d	Filter 1-D data sets (time series)
fitcircle	Finds the best-fitting great circle to a set of points
gmt2rgb	Convert Sun rasterfile or grid to r, g, b grids
gmtconvert	Convert between ASCII and binary 1-D tables
gmtdefaults	List the current default settings
gmtmath	Mathematical operations on data tables
gmtset	Set individual default parameters
gmtselect	Extract data subsets based on spatial criteria
grdfilter	Filter 2-D data sets in the space domain
grd2cpt	Make a color palette table from a grdfile
grd2xyz	Conversion from 2-D gridded file to table data
grdblend	Blend several partially over-lapping grdfiles onto one grid
grdclip	Limit the z-range in gridded data
grdcontour	Contouring of 2-D gridded data
grdcut	Cut a sub-region from a grd file
grdedit	Modify header information in a 2-D gridded file
grdfft	Operate on grdfiles in the frequency domain
grdgradient	Compute directional gradient from grdfiles
grdhisteq	Histogram equalization for grdfiles
grdimage	Produce images from 2-D gridded data
grdinfo	Get information about grd files
grdlandmask	Create mask grdfile from shoreline data base
grdmask	Reset nodes outside a clip path to a constant
grdmath	Mathematical operations on grdfiles
grdpaste	Paste together grdfiles along a common edge
grdproject	Project gridded data onto a new coordinate system
grdreformat	Converting between different grdfile formats
grdsample	Resample a 2-D gridded data set onto a new grid
grdtrend	Fits polynomial trends to grdfiles
grdtrack	Sampling of 2-D data set along 1-D track

grdvector	Plot vector fields from grdfiles
grdview	3-D perspective imaging of 2-D gridded data
grdvolume	Volume calculations from 2-D gridded data
makecpt	Make **GMT** color palette tables
mapproject	Forward or inverse map projections of table data
minmax	Find extreme values in data tables
nearneighbor	Nearest-neighbor gridding scheme
project	Project data onto lines/great circles
psbasemap	Create a basemap plot
psclip	Use polygon files to define clipping paths
pscoast	Plot coastlines and filled continents on maps
pscontour	Contour xyz-data by triangulation
pshistogram	Plot a histogram
psimage	Plot images (EPS or Sun raster files) on maps
pslegend	Plot legend on maps
psmask	Create overlay to mask out regions on maps
psrose	Plot sector or rose diagrams
psscale	Plot grayscale or colorscale on maps
pstext	Plot textstrings on maps
pswiggle	Draw time-series along track on maps
psxy	Plot symbols, polygons, and lines on maps
psxyz	Plot symbols, polygons, and lines in 3-D
sample1d	Resampling of 1-D table data sets
spectrum1d	Compute various spectral estimates from time-series
splitxyz	Split xyz-files into several segments
surface	A continuous curvature gridding algorithm
trend1d	Fits polynomial or Fourier trends to y = f(x) data
trend2d	Fits polynomial trends to z = f(x,y) data
triangulate	Perform optimal Delaunay triangulation and gridding
xyz2grd	Convert equidistant xyz data to a 2-D grd file

SEE ALSO

Look up the individual man pages for details. By default, these pages are in **$GMTHOME**/man/manl, depending on your installation settings. Information is also available on the **GMT** home page gmt.soest.hawaii.edu or locally under **$GMTHOME**/www/gmt/gmt_services.html

REFERENCES

Wessel, P., and W. H. F. Smith, 2005, The Generic Mapping Tools (GMT) version 4.1 Technical Reference & Cookbook, SOEST/NOAA.

Wessel, P., and W. H. F. Smith, 1998, New, Improved Version of Generic Mapping Tools Released, EOS Trans., AGU, 79 (47), p. 579.

Wessel, P., and W. H. F. Smith, 1995, New Version of the Generic Mapping Tools Released, EOS Trans., AGU, 76 (33), p. 329.

Wessel, P., and W. H. F. Smith, 1995, New Version of the Generic Mapping Tools Released, http://www.agu.org/eos_elec/95154e.html, Copyright 1995 by the American Geophysical Union.

Wessel, P., and W. H. F. Smith, 1991, Free Software Helps Map and Display Data, EOS Trans., AGU, 72 (41), p. 441.

NAME

pslib 4.0 – A *PostScript* based plotting library

DESCRIPTION

pslib was created to make the generation of *PostScript* page description code easier. It is a library that contains a series of tools that can be used to create plots. The resulting *PostScript* code is ASCII text and can be edited using any text editor. Thus, it is fairly easy 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. **pslib** is written in C but now includes FORTRAN bindings (thanks to John Goff, WHOI) and can therefore be called from both C and FORTRAN programs. To use this library, you must link your plotting program with pslib.a. **pslib** is the core of the **GMT** graphics programs. **pslib** output conforms to the Adobe Encapsulated *PostScript* File Specification Version 3.0 (EPSL), and may be used as EPS files and inserted into, say, a Word document on a Mac. See Appendix F in the Technical Reference for detailed instructions.

Before any **pslib** calls can be issued, the plotting system must be initialized. This is done by calling **ps_plotinit**, which defines macros, sets up the plot-coordinate system, scales, and [optionally] opens a file where all the *PostScript* code will be written. Normally, the plot code is written to *stdout*. The measure unit for sizes and positions can be set to be centimeter (c), inch (i), or meter (m). When all plotting is done, you must terminate the plotting system by calling **ps_plotend**.

pslib uses the direct color model where red, green, and blue are given separately, each must be in the range from 0-255. If red < 0 then no fill operation takes place. 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 linewidth and pattern. **pslib** 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.

A wide variety of output devices that support *PostScript* exist, including laserwriters (color or monochrome) and workstations running *PostScript* based window systems like Sun's OpenWindows. xnews (part of OpenWindows) or ghostscript (public domain) can be used to create rasterfiles at a user-defined resolution (DPI), making it possible to render *PostScript* on a Versatec and other non-*PostScript* raster devices. Regular Sun rasterfiles created under NeWS from *PostScript* files can be sent to a variety of color hardcopy units. Check the devices available on your network.

FUNCTION CALLS

The following is a list of available functions and a short description of what they do and what parameters they expect. All floating point variables are expected to be **double** (i.e., 8 bytes), whereas all integers are assumed to be 4 bytes long. All plotting functions are declared as functions returning an int. Currently, the return value is undefined.

void **ps_arc** (*x, y, radius, angle1, angle2, status*)
double *x, y, radius, angle1, angle2*;
int *status*;

Draws a circular arc centered on (*x,y*) from angle *angle1* to *angle2*. Angles must be given in decimal degrees. If angle1 > angle2, a negative arc is drawn. *status* is a value from 0 through 3. 1 means set new anchor point, 2 means stroke the circle, 3 means both, 0 means none of the above.

void **ps_axis** (*xpos, ypos, length, startval, stopval, tickval, label, annotpointsize, side*)
double *xpos, ypos, length, startval, stopval, tickval*;
double *annotpointsize, side*;
char **label*;

Plots an axis with tickmarks, annotation, and label. *xpos, ypos*, and *length* are in inches (or cm or meters), *annotpointsize* in points (72 points = 1 inch), else data units are used. *side* can be 0, 1, 2, or 3, which selects lower x-axis, right y-axis, upper x-axis, or left y-axis, respectively. labelpointsize = 1.5 * *annotpointsize*. A negative *tickval* will reverse the sense of positive direction, e.g., to have the y-axis be positive down.

void **ps_circle** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;

int *rgb[3], outline*;

Plots a circle and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern.

void **ps_clipoff** ()

Resets the clip path to what it was before the last call to **ps_clipon**.

void **ps_clipon** (*xarray, yarray, npoints, rgb, flag*)
double *xarray[], yarray[]*;
int *npoints, rgb[3], flag*;

Sets up a user-definable clip path. Plotting outside this polygon will be clipped until **ps_clipoff** is called. If *red* >= 0 the inside of the path 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* = 1 means this is the first path in a multi-segment clip path. *flag* = 2 means this is the last segment. Thus, for a single path, *flag* = 3.

void **ps_colorimage** (*xpos, ypos, xlength, ylength, buffer, nx, ny, depth*)
double *xpos, ypos, xlength, ylength*;
unsigned char *buffer[]*;
int *nx, ny, depth*;

Plots a 1-, 2-, 4-, 8-, or 24-bit deep image. This functions sets up a call to the *PostScript* colorimage or image operators. *xpos, ypos, xlength, ylength* specify the position of lower left corner and size (in inches) of image. The pixel values are stored in *buffer*, an unsigned character array in scanline orientation with gray shade or r/g/b values (0 - 255) where 0 is black, 255 is white. *buffer[0]* is 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. If *-nx* is passed then the first three bytes of *buffer* holds the r/g/b color for pixels that are to be masked out using the *PostScript* Level 3 Color Mask method. See the Adobe Systems *PostScript* Reference Manual for more details.

void **ps_colortiles** (*x0, y0, xlength, ylength, buffer, nx, ny*)
double *x0, y0, xlength, ylength*;
int *nx, ny*;
unsigned char *buffer[]*;

Plots a true color image based on individual color tiles. *x0, y0* is the location of the lower left corner of the image in inches. *xlength, ylength* is the image size in inches. *buffer* contains rgb triplets stored as rgbrgbrgb... *nx, ny* is the image size in pixels.

void **ps_command** (*text*)
char **text*;

Writes a raw *PostScript* command to the *PostScript* output file, e.g. "1 setlinejoin".

void **ps_comment** (*text*)
char **text*;

Writes a comment (*text*) to the *PostScript* output file, e.g. "Start of graph 2".

void **ps_cross** (*xcenter, ycenter, diameter*)
double *xcenter, ycenter, diameter*;

Plots a cross at the specified point using current pen-width and -pattern that fits inside a circle of given diameter.

void **ps_diamond** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a diamond and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_ellipse** (*xcenter, ycenter, angle, major, minor, rgb, outline*)
double *xcenter, ycenter, angle, major, minor*;
int *rgb[3], outline*;

Plots a ellipse with its major semiaxis rotated by *angle* degrees and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern.

void **ps_encode_font** (*font_no*)
int *font_no*;

Will reencode this font using the current encoding vector if it is not StandardEncoding.

void **ps_epsimage** (*xpos, ypos, xlength, ylength, buffer, size, nx, ny, ox, oy*)
double *xpos, ypos, xlength, ylength*;
unsigned char *buffer[]*;
int *size, nx, ny, ox, oy*;

Plots an Encapsulated PostScript (EPS) image. The EPS file is stored in *buffer* and has *size* bytes. This functions simply includes the image in the *PostScript* output stream within an appropriate wrapper. Specify position of lower left corner and size (in inches) of image. *nx,ny,ox,oy* refers to the width, height and origin (lower left corner) of the BoundingBox.

void **ps_flush** ()

Flushes the output buffer.

void **ps_hexagon** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a hexagon and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_image** (*xpos, ypos, xlength, ylength, buffer, nx, ny, bits*)
double *xpos, ypos, xlength, ylength*;
unsigned char *buffer[]*;
int *nx, ny, bits*;

Obsolete, simply passes arguments to **ps_colorimage**.

void **ps_imagefill** (*x, y, n, image, imagefile, invert, dpi, outline, f_rgb, b_rgb*)
double *x[], y[], x0, y0*;
int *n, image, invert, dpi, outline, f_rgb[3], b_rgb[3]*;
char *imagefile*;

Similar to **ps_polygon**, but fills the area with an image pattern rather than a color or grayshade. *x* and *y* hold the arrays of *n* points. 90 predefined patterns are available (See **GMT** Appendix E). *image* gives the image number (1-90). If set to 0, *imagefile* must be the name to the user's image, which must be stored as a Sun 1-, 8-, or 24-bit rasterfile.
1-bit images only: (i) The set pixels (1) are colored using the RGB combination in *f_rgb*, while the unset pixels (0) are painted with *b_rgb*. Set the f_rgb[0] to -1 to make set pixels transparent. Set b_rgb[0] to -1 to make the unset pixels transparent. (ii) If *invert* is TRUE (1), the set and unset pixels are interchanged before plotting.
The unit size of the image is controlled by *dpi* (in dots-per-inch). If set to zero, the image is plotted at the device resolution. If *outline* is TRUE, the current penwidth is used to draw the polygon outline.

void **ps_bitimage** (*xpos, ypos, xlength, ylength, buffer, nx, ny, invert, f_rgb, b_rgb*)
double *xpos, ypos, xlength, ylength*;
unsigned char *buffer[]*;
int *nx, ny, invert, f_rgb[3], b_rgb[3]*;

Plots a 1-bit image using the given foreground color *f_rgb* and background color *b_rgb*. Specify position of lower left corner and size (in inches) of image. *buffer* is an unsigned character array with 8 pixels per byte. *nx,ny* refers to the number of pixels in image. The rowlength of *buffer* must be an integral number of 8. *buffer*[0] is upper left corner. *buffer* values are stored as columns, starting at the lower left corner and ending at the upper right corner. If *invert* is 0 then the bits that are 1 are painted with the foreground color, while bits that are 0 are painted with the backgound color. If *invert* is 1, foreground and background are switched. To get a partly transparent image, set the first index of the foreground or background color to -1, i.e. f_rgb[0]=-1 or b_rgb[0]=1. See the Adobe Systems *PostScript* Reference Manual for more details.

void **ps_itriangle** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots an inverted and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_line** (*xarray, yarray, npoints, type, close, split*)
double *xarray[], yarray[]*;
int *npoints, type, close, split*;

Draw a continuous line from the positions in the x-y arrays. If *close* == 1, the first and last point will automatically be closed by the *PostScript* driver. If this is the first segment in a multi-segment path, set *type* == 1. To end the segments and have the line(s) drawn, set *type* == 2. Thus, for a single segment, *type* must be 3. The line is drawn using the current pen-width. Only if *split* is TRUE may ps_line use multiple strokes to draw lines longer that MAX_PATH. ps_polygon will call ps_line with *split* = FALSE since the path must be continuous. If *split* is FALSE and the pathlength exceeds MAX_PATH a warning will be issued.

unsigned char ***ps_load_image** (*fp, header*)
FILE **fp*;
struct imageinfo **header*;

Reads the image contents of the EPS file or Sun rasterfile pointed to by the open filepointer *fp*. The routine can handle Encapsulated *PostScript* files or 1-, 8-, 24-, or 32-bit rasterfiles in old, standard, run-length encoded, or RGB-style Sun format.

void **ps_octagon** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a octagon and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_patch** (*xarray, yarray, npoints, rgb, outline*)
double *xarray[], yarray[]*;
int *npoints, rgb[3], outline*;

Identical to **ps_polygon** except polygon must be < 20 points long and there will be no attempt to shorten the path by discarding unnecessary intermediate points along straight segments. Primarily used when painting large number of small polygons and not waste output space.

void **ps_pentagon** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a pentagon and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_pie** (*xcenter, ycenter, radius, azimuth1, azimuth2, rgb, outline*)
double *xcenter, ycenter, radius, azimuth1, azimuth2*;
int *rgb[3], outline*;

Plots a sector of a circle and paints it with the specified RGB combination. If *outline* == 1, the outline will be drawn using current pen-width and -pattern.

void **ps_plot** (*xabs, yabs, kpen*)
double *xabs, yabs*;
int *kpen;*

Absolute move (*kpen*=3) or draw (*kpen*=2), using current linewidth. Use (*kpen*=-2) to make sure the line is stroked.

void **ps_plotend** (*last_page*)
int *last_page*;

Terminates the plotting sequence and closes plot file (if other than *stdout*). If *last_page* == 1, then a *PostScript* showpage command is issued, which initiates the printing process on hardcopy devices.

void **ps_plotinit** (*plotfile, overlay, mode, xoff, yoff, xscl, yscl, ncopies, dpi, unit, pagesize, rgb, encoding, eps*)
char **plotfile, *encoding;*
int *overlay, mode, ncopies, dpi, unit*;
double *xoff, yoff, xscl, yscl*;
int *pagesize[2], rgb[3]*; **struct EPS *** *eps*;

Initializes the plotting. If *plotfile* == NULL (or ""), then output is sent to *stdout*, else output is sent to *plotfile*. *overlay* should be 1 only if you plan to append it to some existing *PostScript* file. *mode* contains three flags in the three lowest bits. The lowest bit controls the plot orientation and can be 0 (Landscape) or 1 (Portrait). The next bit, if set to 1, will re-encode the fonts to include European accented characters using the now-obsolete **GMT** 3.4 encoding. To use the ISOLatin1 encoding set the 5th bit to 1. The third bit controls the format used to write PostScript images: 0 means binary, 1 means hexadecimal. Most printers needs the latter while some can handle binary which are 50% smaller and therefore execute faster. *xoff,yoff* are used to move the origin from the default position in the lower left corner. *xscl,yscl* are used to scale the entire plot (Usually set to 1.0, 1.0). Set *ncopies* to get more than 1 copy. *dpi* sets the hardcopy resolution in dots pr units. For optimum plot quality and processing speed, choose *dpi* to match the intended plotter resolution. Examples are 300 for most laserwriters, 2540 for Linotype-300, and ~85 for Sun screens. When in doubt, use 300. *unit* can be any of 0 (CM), 1 (INCH), or 2 (M), telling the plot system what units are used for distance and sizes. Note that, regardless of choice of unit, dpi is still in dots-pr-inch. *pagesize* means the physical width and height of the plotting media in points, (typically 612 by 792 for Letter or 595 by 842 for A4 laserwriter plotters. The *rgb* array holds the color of the page (usually white = 255,255,255). The *encoding* is the name of a character encoding scheme to be used, e.g., Standard, ISOLatin1, ISO-8859-2, etc. The EPS structure is defined in the pslib.h include file and contains information that will make up the comments header of a EPS file. Programmers who plan to call pslib routines should read the comments in pslib.h first. Note that the FORTRAN binding does not expect this last argument.

void **ps_plotr** (*xrel, yrel, kpen*)
double *xrel, yrel*;
int *kpen*;

Move (*kpen* = 3) or draw (*kpen* = 2) relative to current point (see **ps_plot**). Use (*kpen*=-2) to make sure the line is stroked.

void **ps_point** (*xcenter, ycenter, diameter*)
double *xcenter, ycenter, diameter*;

Plots a point using current pen with given diameter. Note the linecap setting must first be set to 1 for this function to work.

void **ps_polygon** (*xarray, yarray, npoints, rgb, outline*)
double *xarray[], yarray[]*;
int *npoints, rgb[3], outline*;

Creates a colored polygon from the positions in the x-y arrays. Polygon will automatically be closed by the *PostScript* driver. If *outline* == 0, no outline is drawn. If *outline* == 1, the outline is drawn using current penwidth.

void **ps_rect** (*x1, y1, x2, y2, rgb, outline*)
double *x1, y1, x2, y2*;
int r*ed, green, blue, outline;*

Plots a colored rectangle. (*x1,y1)* and (*x2,y2*) are any two corners on a diagonal. If *outline* == 1, the outline will be drawn using current pen-width and -pattern.

void **ps_rotatetrans** (*x, y, angle*)
double *x, y, angle*;

Rotates the coordinate system by *angle* degrees, then translates origin to (*x,y*).

void **ps_segment** (*x0, y0, x1, y1*)
double *x0, y0, x1, y1;*

Draws a line segment between the two points using current pen attributes.

void **ps_setdash** (*pattern, offset*)
char **pattern;*
int *offset;*

Changes the current dashpattern. The character string *pattern* is set to the desired pattern. E.g., "4 2" and *offset* = 1 will plot like:

```
    x ---- ---- ----
```

etc, where x is starting point (The x is not plotted). That is, the line is made up of a repeating pattern of a 4 units long line and a 2 unit long gap, starting 1 unit after the x. To reset to solid line, specify *pattern* = NULL ("") and *offset* = 0. Units are in dpi units.

void **ps_setfont** (*fontnr*)
int *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 set to 0.

void **ps_setformat** (*n_decimals*)
int *n_decimals*;

Sets 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.

void **ps_setline** (*linewidth*)
int *linewidth*;

Changes the current linewidth in DPI units. 0 gives thinnest line, but the use of 0 is implementation-dependent (Works fine on most laserwriters).

void **ps_setlinecap** (*cap*)
int *cap*;

Changes the current linecap. 0 gives butt cap [Default], 1 gives round, and 2 gives square.

void **ps_setlinejoin** (*join*)
int *join*;

Changes the current linejoin. 0 gives mitered [Default], 1 gives round, and 2 gives bevel joins.

void **ps_setmiterlimit** (*limit*)
int *limit*;

Changes the current miter limit. 0 gives default miter, other values are the cutoff-, acute- angle when mitering takes place.

void **ps_setpaint** (*rgb*)
int *rgb[3]*;

Changes the current RGB setting for pens and text.

void **ps_square** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a square and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_star** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a star and fills it with the specified color. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_text** (*x, y, pointsize, text, angle, justify, form*)
double *x, y, pointsize, angle*;
char **text*;
int *justify, form*;

The *text* is plotted starting at (*x,y*), and will make an *angle* with the horizontal. The point (*x,y*) maps onto different points of the textstring by giving various values for *justify*. It is used as follows:

```
9------------10----------- 11
|                    |
5           6          7
|                    |
1------------ 2------------ 3
```

The box represents the textstring. E.g., to plot a textstring with its center of gravity at (*x,y*), you must use *justify* == 6. If *justify* is negative, then all leading and trailing blanks are stripped before plotting. Certain character sequences (flags) have special meaning to ps_text. @~ toggles between current font and the Mathematical Symbols font. @%*no*% sets font to *no*; @%% resets to starting 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. Give fontsize in points (72 points = 1 inch). Normally, the text is typed using solid characters. To draw outline characters, set *form* == 1. If *pointsize* is negative it means that the current point has already been set before **ps_text** was called and that (*x,y*) should be ignored.

void **ps_textbox** (*x, y, pointsize, text, angle, justify, outline, dx, dy, rgb*)
double *x, y, angle, pointsize, dx, dy*;
char **text*;
int *justify, outline, rgb[3]*;

This function is used in conjugation with **ps_text** when a box surrounding the text string is desired. Taking most of the arguments of **ps_text**, the user must also specify the color of the resulting rectangle, and whether its outline should be drawn. More room between text and rectangle can be obtained by setting *dx* and *dy* accordingly.

void **ps_transrotate** (*x, y, angle*)
double *x, y, angle*;

Translates the origin to (*x,y*), then rotates the coordinate system by *angle* degrees.

void **ps_triangle** (*xcenter, ycenter, diameter, rgb, outline*)
double *xcenter, ycenter, diameter*;
int *rgb[3], outline*;

Plots a triangle and paints it with the specified RGB combination. If *outline* == 1, the outline will be drawn using current pen-width and -pattern. The symbol will fit inside a circle of given diameter.

void **ps_vector** (*xtail, ytail, xtip, ytip, tailwidth, headlength, headwidth, headshape, rgb, outline*)
double *xtail, ytail, xtip, ytip, tailwidth, headlength, headwidth, headshape*;
int *rgb[3], outline*;

Draws a vector of size and appearance as specified by the various parameters. *headshape* 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 gives a triangular head, 1.0 gives an arrow shaped head. If *outline* == 1, the outline will be drawn using current penwidth. Add 8 to *outline* for a double-headed vector.

void **ps_words** (*x, y, text, n_words, line_space, par_width, par_just, font, font_size, angle, rgb, justify, draw_box, x_off, y_off, x_gap, y_gap, boxpen_width, boxpen_texture, boxpen_offset, boxpen_rgb, vecpen_width, vecpen_texture, vecpen_offset, vecpen_rgb, boxfill_rgb*)
double *x, y, line_space, par_width, angle, x_off, y_off, x_gap, y_gap*;
int *n_words, font, font_size, justify, draw_box, boxpen_width, boxpen_offset*;
int *boxpen_rgb[3], vecpen_width, vecpen_offset, vecpen_rgb[3], boxfill_rgb[3]*;
char ***text, *boxpen_texture, *vecpen_texture*;

Typesets paragraphs of text. *text* is an array of the words to typeset, using the given line-spacing and paragraph width. The whole text block is positioned at *x, y* which is the anchor point on the box as indicated by *justify* (see ps_text). The whole block is then shifted by *x_off, y_off.* Inside the box, text is justified left, centered, right, or justified as governed by *par_just* (lcrj). *draw_box* contains 4 bit flags pertaining to the surrounding outline box. If on, the first (lowest) bit draws the box outline. The second bit fills the box interior. The third bit makes the outline box have rounded corners (unless *x_gap, y_gap*, which specifies the padding between the text and the box, are zero), while the forth bit draws a line from the original *x, y* point to the shifted position. The escape sequences described for ps_text applies here, as well as two additional commands: @;*r/g/b*; changes the font color (@;; resets it), and @:*size*: changes the font size (@:: resets it).

AUTHOR

Paul Wessel, School of Ocean and Earth Science and Technology, 1680 East-West Road, Honolulu, Hawaii 96822, (808) 956-4778, Internet address: pwessel@hawaii.edu.

BUGS

Caveat Emptor: The author is **not** responsible for any disasters, suicide attempts, or ulcers caused by correct **or** incorrect use of **pslib**. If you find bugs, please report them to the author by electronic mail. Be sure to provide enough detail so that I can recreate the problem.

RESTRICTIONS

Due to the finite memory of some output devices like Laserwriters, certain restrictions due to limitations of the *PostScript* interpreter apply: For now, the arrays passed to **ps_clipon** and **ps_polygon** must contain less than about 1350 points. Also, the buffer array passed to **ps_image** must be able to fit in the available memory. Check the specifications of the hardcopy device you are using. Note that some Raster Image

Processors (RIPs) do not support direct color so that the colors you get may not be exactly the ones you wanted. This is a limitation of the RIP, not the underlying *PostScript* code generated by **pslib**.

REFERENCES

Adobe Systems Inc., 1990, *PostScript* language reference manual, 2nd edition, Addison-Wesley, (ISBN 0-201-18127-4).