triangulate

triangulate - Optimal (Delaunay) triangulation and gridding of Cartesian table data

Synopsis

triangulate [*table*] [**-C***slpfile*] [**-Dx**|**y**] [**-E***empty*] [**-G***grdfile*] [**-I***increment*] [**-J***parameters*] [**-M**] [**-N**] [**-Q**[**n**]] [**-R***region*] [**-S**] [**-V**[*level*]] [**-Z**] [**-b**binary] [**-d**nodata] [**-e**regexp] [**-f**flags] [**-h**headers] [**-i**flags] [**-r**] [**-:**[**i**|**o**]]

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

Description

triangulate reads one or more ASCII [or binary] files (or standard input) containing x,y[,z] and performs Delaunay triangulation, i.e., it find how the points should be connected to give the most equilateral triangulation possible. If a map projection (give **-R** and **-J**) is chosen then it is applied before the triangulation is calculated. By default, the output is triplets of point id numbers that make up each triangle and is written to standard output. The id numbers refer to the points position (line number, starting at 0 for the first line) in the input file. As an option, you may choose to create a multiple segment file that can be piped through psxy to draw the triangulation network. If **-G -I** are set a grid will be calculated based on the surface defined by the planar triangles. The actual algorithm used in the triangulations is either that of Watson [1982] [Default] or Shewchuk [1996] (if installed; type **triangulate -** to see which method is selected). This choice is made during the GMT installation. Furthermore, if the Shewchuk algorithm is installed then you can also perform the calculation of Voronoi polygons and optionally grid your data via the natural nearest neighbor algorithm.

Required Arguments

None.

Optional Arguments

table

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

-C*slpfile*

Read a slope grid (in radians) and compute the propagated uncertainty in the bathymetry using the CURVE algorithm [Zambo et al, 20xx]. Requires the **-G** option to specify the output grid. Note that the *slpgrid* sets the domain for the output grid so **-R**, **-I**, [**-r**] are not required. Cannot be used in conjunction with **-D**, **-F**, **-M**, **-N**, **-Q**, **-S** and **-T**.

-Dx|y

Take either the *x*- or *y*-derivatives of surface represented by the planar facets (only used when **-G** is set).

-E*empty*

Set the value assigned to empty nodes when **-G** is set [NaN].

-G*grdfile*

Use triangulation to grid the data onto an even grid (specified with **-R -I**). Append the name of the output grid file. The interpolation is performed in the original coordinates, so if your triangles are close to the poles you are better off projecting all data to a local coordinate system before using **triangulate** (this is true of all gridding routines) or instead select **sphtriangulate**. For natural nearest neighbor gridding you must add **-Qn**.

-I*xinc*[*unit*][**+e**|**n**][/*yinc*[*unit*][**+e**|**n**]]

x_inc [and optionally *y_inc*] is the grid spacing. Optionally, append a suffix modifier. **Geographical (degrees) coordinates**: Append **m** to indicate arc minutes or **s** to indicate arc seconds. If one of the units **e**, **f**, **k**, **M**, **n** or **u** is appended instead, the increment is assumed to be given in meter, foot, km, Mile, nautical mile or US survey foot, respectively, and will be converted to the equivalent degrees longitude at the middle latitude of the region (the conversion depends on PROJ_ELLIPSOID). If *y_inc* is given but set to 0 it will be reset equal to *x_inc*; otherwise it will be converted to degrees latitude. **All coordinates**: If **+e** 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 **+n** to the supplied integer argument; the increment is then recalculated from the number of nodes and the domain. The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid; see GMT File Formats for details. Note: if **-R***grdfile* is used then the grid spacing has already been initialized; use **-I** to override the values.

-J*parameters* (more …)

Select map projection.

-M

Output triangulation network as multiple line segments separated by a segment header record.

-N

Used in conjunction with **-G** to also write the triplets of the ids of all the Delaunay vertices [Default only writes the grid].

-Q[**n**]

Output the edges of the Voronoi cells instead [Default is Delaunay triangle edges]. Requires **-R** and is only available if linked with the Shewchuk [1996] library. Note that **-Z** is ignored on output. Optionally, append **n** for combining the edges into closed Voronoi polygons.

-R*xmin*/*xmax*/*ymin*/*ymax*[**+r**][**+u***unit*] (more …)

Specify the region of interest.

-S

Output triangles as polygon segments separated by a segment header record. Requires Delaunay triangulation.

-T

Output edges or polygons even if gridding has been selected with the **-G** option [Default will not output the triangulation or Voronoi polygons is gridding is selected].

-V[*level*] (more …)

Select verbosity level [c].

-Z

Controls whether we read (x,y) or (x,y,z) data and if z should be output when **-M** or **-S** are used [Read (x,y) only].

-bi[*ncols*][**t**] (more …)

Select native binary input. [Default is 2 input columns].

-bo[*ncols*][*type*] (more …)

Select native binary output. [Default is same as input]. Node ids are stored as double triplets.

-d[**i**|**o**]*nodata* (more …)

Replace input columns that equal *nodata* with NaN and do the reverse on output.

-e[**~**]*"pattern"* **|** **-e**[**~**]/*regexp*/[**i**] (more …)

Only accept data records that match the given pattern.

-f[**i**|**o**]*colinfo* (more …)

Specify data types of input and/or output columns.

-h[**i**|**o**][*n*][**+c**][**+d**][**+r***remark*][**+r***title*] (more …)

Skip or produce header record(s).

-i*cols*[**+l**][**+s***scale*][**+o***offset*][,*…*] (more …)

Select input columns and transformations (0 is first column).

-r (more …)

Set pixel node registration [gridline]. (Only valid with **-G**).

-:[**i**|**o**] (more …)

Swap 1st and 2nd column on input and/or output.

-^ or just **-**

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

-+ or just **+**

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

-? or no arguments

Print a complete usage (help) message, including the explanation of all options, then exits.

ASCII Format Precision

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

Grid Values Precision

Regardless of the precision of the input data, GMT programs that create grid files will internally hold the grids in 4-byte floating point arrays. This is done to conserve memory and furthermore most if not all real data can be stored using 4-byte floating point values. Data with higher precision (i.e., double precision values) will lose that precision once GMT operates on the grid or writes out new grids. To limit loss of precision when processing data you should always consider normalizing the data prior to processing.

Examples

To triangulate the points in the file samples.xyz, store the triangle information in a binary file, and make a grid for the given area and spacing, use

```
gmt triangulate samples.xyz -bo -R0/30/0/30 -I2 -Gsurf.nc >
```

To draw the optimal Delaunay triangulation network based on the same file using a 15-cm-wide Mercator map, use

```
gmt triangulate samples.xyz -M -R-100/-90/30/34 -JM15c | gm
    -R-100/-90/30/34 -JM15c -W0.5p -B1 > network.ps
```

To instead plot the Voronoi cell outlines, try

```
gmt triangulate samples.xyz -M -Q -R-100/-90/30/34 -JM15c |
    gmt psxy -R-100/-90/30/34 -JM15c -W0.5p -B1 > cells.ps
```

To combine the Voronoi outlines into polygons and paint them according to their ID, try

```
gmt triangulate samples.xyz -M -Qn -R-100/-90/30/34 -JM15c
    gmt psxy -R-100/-90/30/34 -JM15c -W0.5p+cf -L -B1 -Ccol
```

To grid the data using the natural nearest neighbor algorithm, try

```
gmt triangulate samples.xyz -Gnnn.nc -Qn -R-100/-90/30/34 -
```

Notes

The uncertainty propagation for bathymetric grids requires both horizontal and vertical uncertainties and these are weighted given the local slope. See the references for more details.

See Also

gmt, greenspline, nearneighbor, pscontour, sphdistance, sphinterpolate, sphtriangulate, surface

References

Watson, D. F., 1982, Acord: Automatic contouring of raw data, *Comp. & Geosci.*, **8**, 97-101.

Shewchuk, J. R., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, First Workshop on Applied Computational Geometry (Philadelphia, PA), 124-133, ACM, May 1996.

Shewchuk’s Homepage