Mapping and modeling Earth Science Data

Segment II: Gridding with GMT

Thorsten Becker

University of Southern California, Los Angeles

Universita di Roma TRE, June 2012

Contents

- Gridding with GMT
- Manipulating grids

Going from x-y-z data

(that can be irregularly distributed)

to regular, gridded (pixelated) representations

GMT default binary format

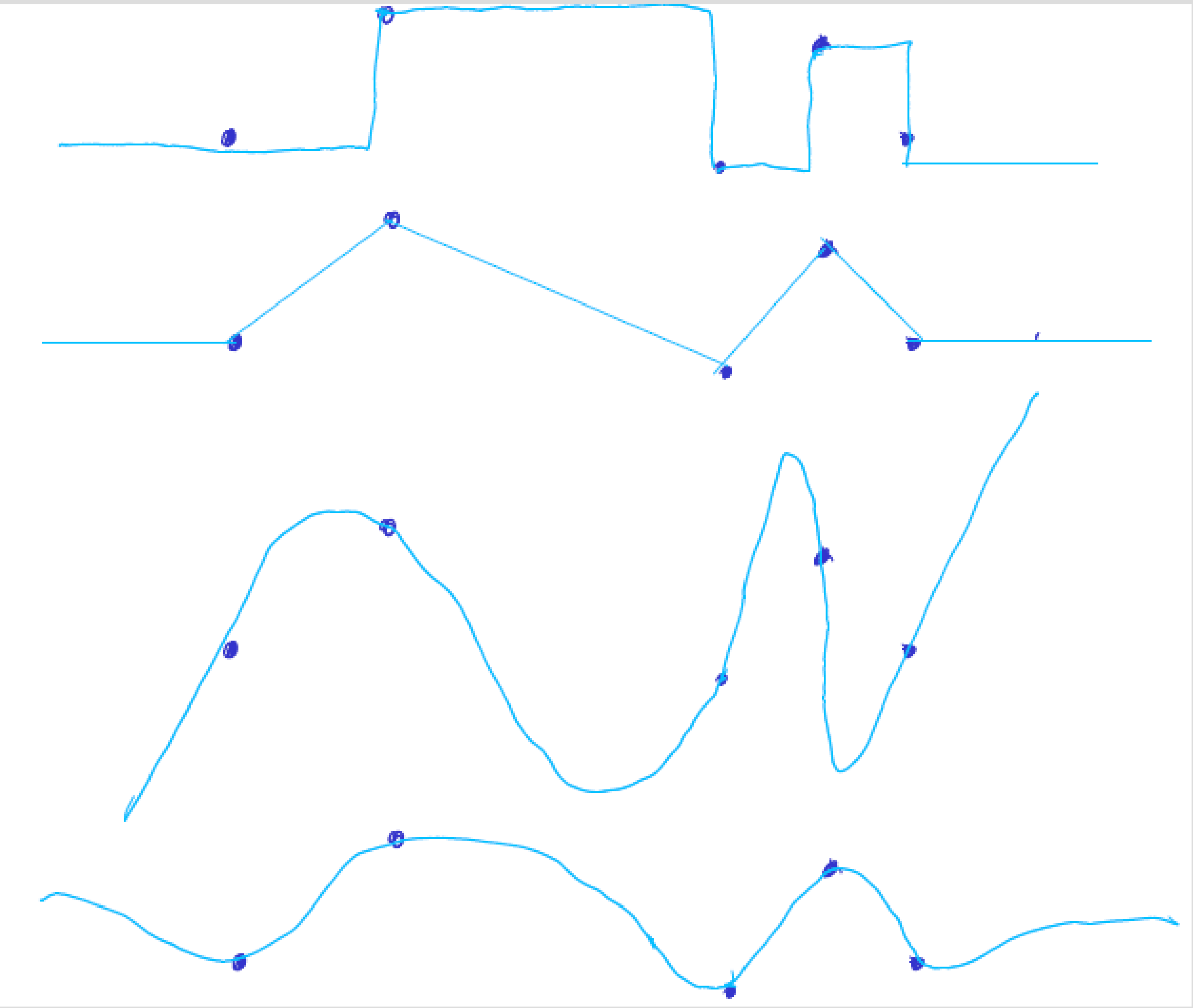
- NetCDF grd
- Single precision internal representation of floats, i.e. might lose precision of numerical data, particularly if not normalized
- Some meta information accessible via `grdinfo`

Regularly binned binary (grd) vs. xyz (lon-lat-value)

- `xyz2grd`
  go from regularly space lon-lat-z or x-y-z data to a NetCDF grd
  - Used -Rxmin/xmax/ymin/ymax and -Idx/dy to specify grid
  - By default, will use NaN for missing values
  - Can also use z only if ordering of data is specified
- `grd2xyz`
  go from grd to x-y-z or z
- Other binary formats can usually be converted via x-y-z intermediate steps

Interpolation

- Really, a modeling problem



Gridding examples

```
examples/gmt/gridding/

interpolate_data
interpolate_data_nearneighor
interpolate_data_sphi
interpolate_data_surface
interpolate_data_triangulate
```

Surface interpolation

- Splines in tension

Terminal

File Edit View Terminal Tabs Help

Terminal | Terminal | Terminal | Terminal

```
SURFACE(1)                    Generic Mapping Tools                    SURFACE(1)

NAME
       surface - adjustable tension continuous curvature surface gridding algorithm

SYNOPSIS
       surface  [  xyzfile  ]  -Goutputfile.grd -Ixinc[unit][=|+][/yinc[unit][=|+]] -Rwest/east/south/north[r] [ -Aaspect_ratio ] [ -Cconver-
       gence_limit ] [ -H[i][nrec] ] [ -Llower ] [ -Luupper ] [ -Nmax_iterations ] [ -Q ] [ -Ssearch_radius[m] ] [ -Ttension_factor[i|b] ] [
       -V[l] ] [ -Zover-relaxation_factor ] [ -:[i|o] ] [ -bi[s|S|d|D[ncol]|c[var1/...]] ] [ -fcolinfo ]

DESCRIPTION
       surface  reads  randomly-spaced  (x,y,z)  triples  from  standard input [or xyzfile] and produces a binary grid file of gridded values
       z(x,y) by solving:

            (1 - T) * L (L (z)) + T * L (z) = 0

       where T is a tension factor between 0 and 1, and L indicates the Laplacian operator.  T = 0 gives  the  "minimum  curvature"  solution
       which  is  equivalent to SuperMISP and the ISM packages.  Minimum curvature can cause undesired oscillations and false local maxima or
       minima (See Smith and Wessel, 1990), and you may wish to use T > 0 to suppress these effects.  Experience suggests T  ~  0.25  usually
       looks  good  for potential field data and T should be larger (T ~ 0.35) for steep topography data.  T = 1 gives a harmonic surface (no
       maxima or minima are possible except at control data points).  It is recommended that the user pre-process the  data  with  blockmean,
       blockmedian,  or  blockmode  to  avoid spatial aliasing and eliminate redundant data.  You may impose lower and/or upper bounds on the
       solution.  These may be entered in the form of a fixed value, a grid with values, or simply be the minimum/maximum input data  values.

       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.  Note that the smallest grid dimension must be at least 4.

       -I     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 c to indicate arc seconds.  If one of the units e, k, i, or n is appended instead, the
```

GMT `surface`

- Splines in tension
- Dial between
  - `T = 0` (default) smoothest solution
  - `T = 1` no local extrema outside data solution
- `-A` sets the aspect ratio for the grid, e.g. depending on mid latitude of the region (but this is not accurate for large, spherical coordinate grids!)
- `-Lld -Lud` will limit the solution to be within range of data

Triangulate

- Linear interpolation in Delaunay triangulation

Terminal

File Edit View Terminal Tabs Help

Terminal | Terminal | Terminal | Terminal

TRIANGULATE(1) Generic Mapping Tools TRIANGULATE(1)

NAME

triangulate - Perform optimal Delaunay triangulation and gridding of Cartesian data [method]

SYNOPSIS

triangulate infiles [**-Dx**|**y**] [**-E**empty] [**-F**] [**-G**grdfile] [**-H**[**i**][nrec]] [**-I**xinc[unit][=|+][/yinc[unit][=|+]]] [**-J**parameters] [**-Q**] [**-R**west/east/south/north[**r**]] [**-V**] [**-Z**] [**-:**[**i**|**o**]] [**-b**[**i**|**o**][**s**|**S**|**d**|**D**[ncol]|**c**[var1/...]]] [**-f**[**i**|**o**]colinfo] [**-m**[**i**|**o**][flag]]

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 **tri-angulate** - to see which method is selected). 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].

-F Force pixel node registration [Default is gridline registration]. (Node registrations are defined in **GMT** Cookbook Appendix B on grid file formats.) Only valid with **-G**).

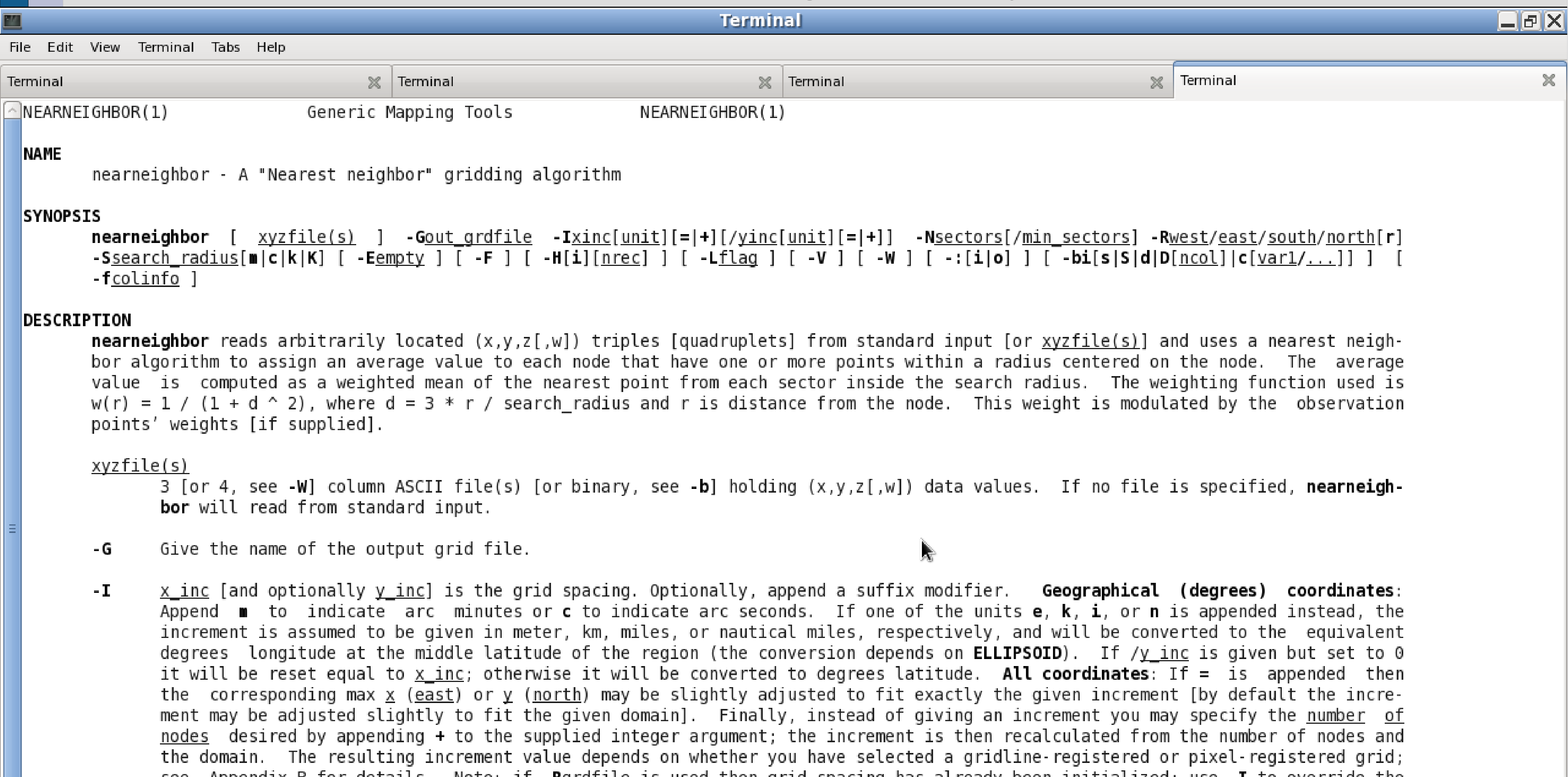
-G Use triangulation to grid the data onto an even grid (specified with **-R -I**). Append the name of the output grid file. The

GMT `triangulate`

- Assuming Cartesian coordinates, perform Delaunay triangulation
- Interpolate linearly using triangles, i.e. derivatives of interpolated scalar within are constant

Nearneighbor

- Local information, weighted by distance



```
NEARNEIGHBOR(1)          Generic Mapping Tools          NEARNEIGHBOR(1)

NAME
       nearneighbor - A "Nearest neighbor" gridding algorithm

SYNOPSIS
       nearneighbor  [  xyzfile(s)  ]  -Gout_grdfile  -Ixinc[unit][=|+][/yinc[unit][=|+]]  -Nsectors[/min_sectors] -Rwest/east/south/north[r]
       -Ssearch_radius[m|c|k|K] [ -Eempty ] [ -F ] [ -H[i][nrec] ] [ -Lflag ] [ -V ] [ -W ] [ -:[i|o] ] [ -bi[s|S|d|D[ncol]|c[var1/...]] ]  [
       -fcolinfo ]

DESCRIPTION
       nearneighbor reads arbitrarily located (x,y,z[,w]) triples [quadruplets] from standard input [or xyzfile(s)] and uses a nearest neigh-
       bor algorithm to assign an average value to each node that have one or more points within a radius centered on the node.  The  average
       value  is  computed as a weighted mean of the nearest point from each sector inside the search radius.  The weighting function used is
       w(r) = 1 / (1 + d ^ 2), where d = 3 * r / 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, nearneigh-
              bor will read from standard input.

       -G     Give the name of the output grid file.

       -I     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 c to indicate arc seconds.  If one of the units e, k, i, or n is appended instead, the
              increment is assumed to be given 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.  All coordinates: If =  is  appended  then
              the  corresponding max x (east) or y (north) may be slightly adjusted to fit exactly the given increment [by default the incre-
              ment may be adjusted slightly to fit the given domain].  Finally, instead of giving an increment you may specify the number  of
              nodes  desired by appending + to the supplied integer argument; the increment is then recalculated from the number of nodes and
              the domain.  The resulting increment value depends on whether you have selected a gridline-registered or pixel-registered grid;
              see  Appendix B for details.  Note: if -Rgrdfile is used then grid spacing has already been initialized; use -I to override the
```

sphinterpolate

- A true spherical combination of surface and triangulate

Terminal

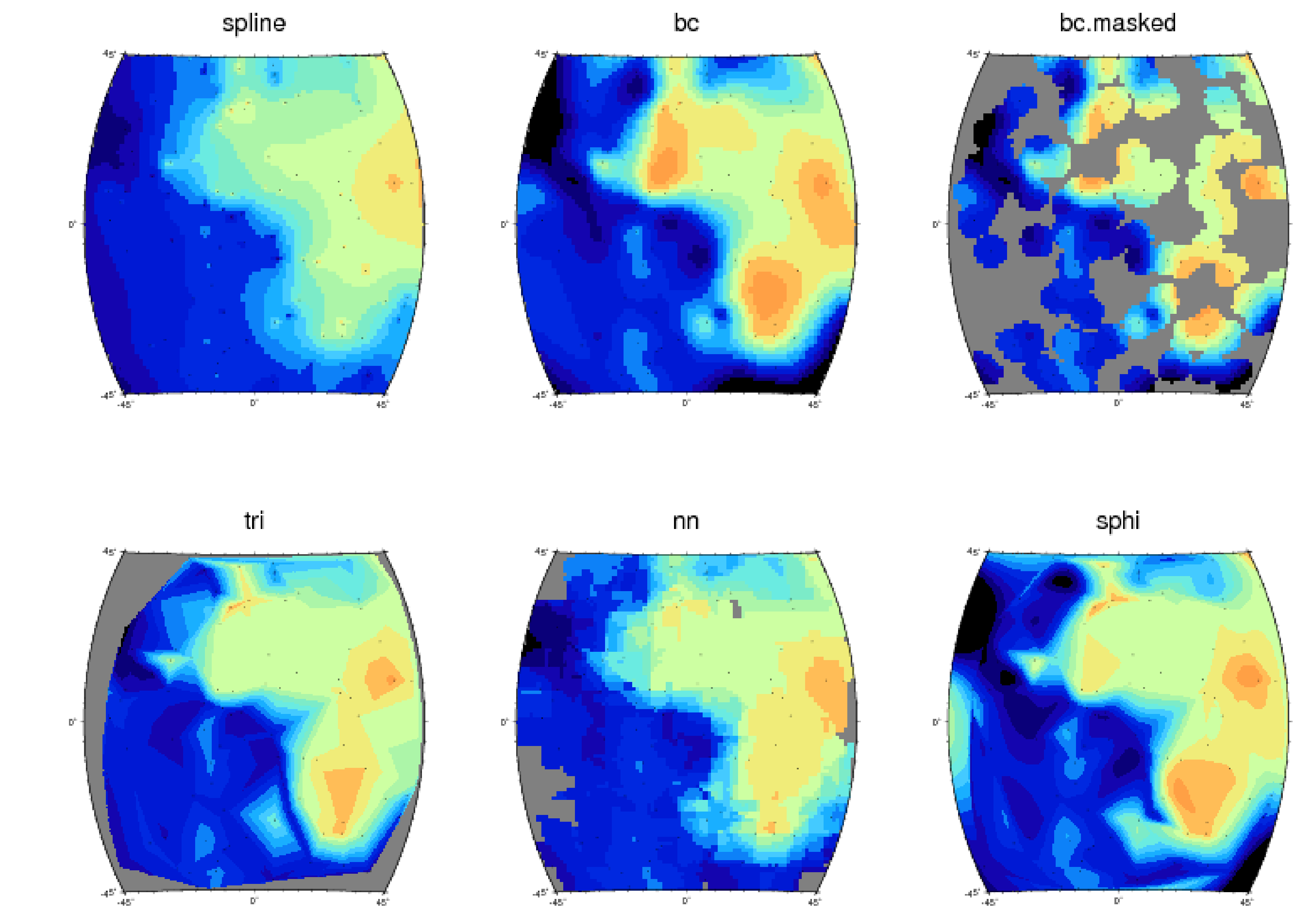
File Edit View Terminal Help

```
SPHINTERPOLATE(1)          Generic Mapping Tools          SPHINTERPOLATE(1)

NAME
       sphinterpolate - Gridding in tension of spherical data

SYNOPSIS
       sphinterpolate  infiles  -Ggrdfile  [  -F  ]  [  -H[i][nrec]  ]  [  -Ixinc[unit][=|+][/yinc[unit][=|+]]  ]  [  -Qmode[/options]  ]  [
       -Rwest/east/south/north[r] ] [ -V ] [ -Z ] [ -:[i|o] ] [ -b[i|o][s|S|d|D[ncol]|c[var1/...]] ][ -m[i|o][flag] ]

DESCRIPTION
       sphinterpolate reads one or more ASCII [or binary] files (or standard input) containing lon, lat, f and performs a Delaunay triangula-
       tion  to  set up a spherical interpolation in tension.  The final grid is saved to the specified file.  Several options may be used to
       affect the outcome, such as choosing local versus global gradient estimation or optimize the tension selection to satisfy one of  four
       criteria.

       infiles
              Data  files with the (lon, lat, f) coordinates in ASCII (or binary; see -b).  If no files are given the standard input is read.

       -G     Name of the output grid to hold the interpolation.

OPTIONS
       -F     Force pixel node registration [Default is gridline registration].  (Node registrations are defined in GMT Cookbook  Appendix  B
              on grid file formats.)

       -H     Input  file(s)  has  header  record(s).  If used, the default number of header records is N_HEADER_RECS.  Use -Hi if only input
              data should have header records [Default will write out header records if the input data have  them].  Blank  lines  and  lines
              starting with # are always skipped.

       -I     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 c to indicate arc seconds.  If one of the units e, k, i, or  n  is  appended  instead,  the
              increment  is assumed to be given 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.  All coordinates: If = is appended then
              the corresponding max x (east) or y (north) may be slightly adjusted to fit exactly the given increment [by default the  incre-
              ment  may be adjusted slightly to fit the given domain].  Finally, instead of giving an increment you may specify the number of
```

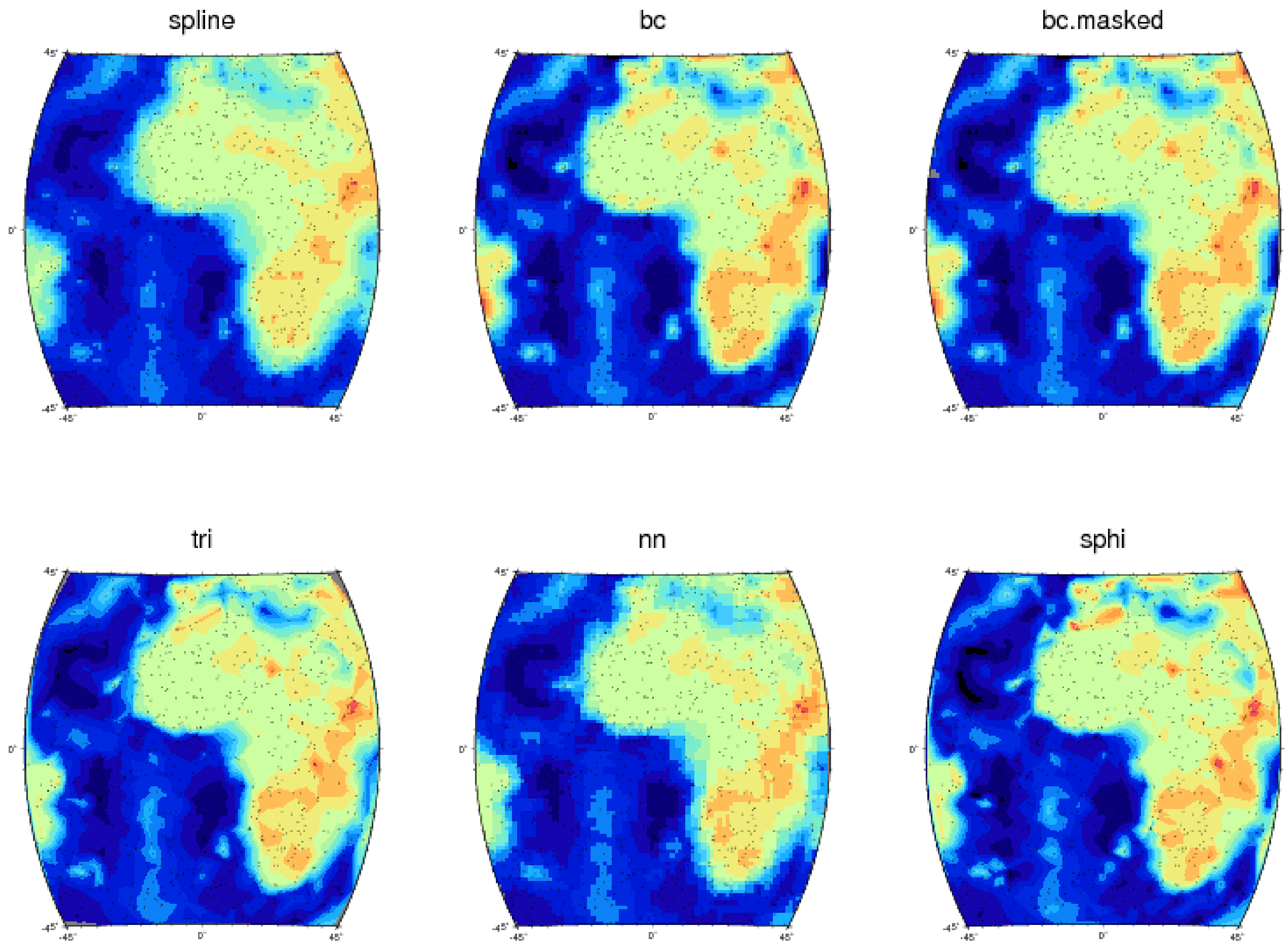
Not all GMT tools are spherical coordinate based

- `surface` works well enough for regional data (remember to adjust -A), and (away from poles) will not do too poorly globally, broadly speaking
- For true spherical, should/can use `sphinterpolate`

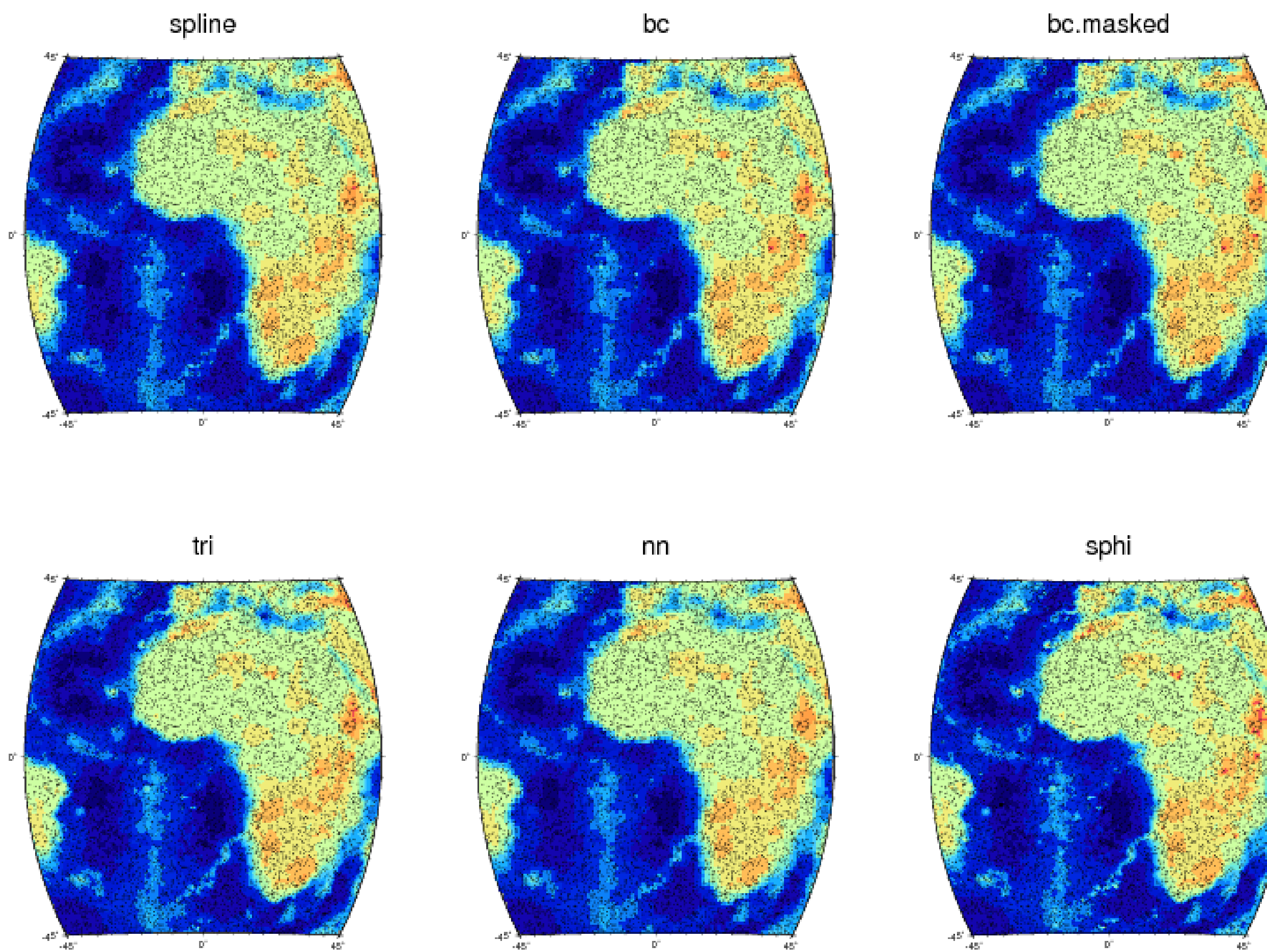
Comparison of methods, N=100



Comparison of methods, N=1000



Comparison of methods, N=10000



Interpolation

- There is no right answer, only trade-offs
  - Optimally smooth vs. no local extrema
  - No prior assumptions about data vs. imposing *a priori* constraints
- There are other tools (such a kriging) which can be used for interpolation, but GMT tools will be a good start, broadly speaking
- Always make sure you know (by plotting) your actual data distribution before interpolation
- Mask out uncertain regions

Once we have a grd

- `grdtrack` : interpolate existing grd to lon/lat
  - -Ql: linear interpolation
  - -Qb: B-spline
  - -Qc: bicubic
  - -Qn: nearest-neighbor
- `Grdsample:` to refine grid sampling (only for plotting)
  - Don't use to coarsen

Operating on grids

- grdmath


```
GRDMATH(1)                    Generic Mapping Tools                    GRDMATH(1)

NAME
       grdmath - Reverse Polish Notation calculator for grid files

SYNOPSIS
       grdmath   [   -F   ]   [   -Ixinc[unit][=|+][/yinc[unit][=|+]]   ]   [  -M  ]  [  -N  ]  [  -Rwest/east/south/north[r]  ]  [  -V  ]  [
       -bi[s|S|d|D[ncol]|c[var1/...]] ] [ -fcolinfo ] operand [ operand ] OPERATOR [ operand ] OPERATOR ... = outgrdfile

DESCRIPTION
       grdmath will perform operations like add, subtract, multiply, and divide on one or more grid files or constants using  Reverse  Polish
       Notation  (RPN)  syntax (e.g., Hewlett-Packard calculator-style).  Arbitrarily complicated expressions may therefore be evaluated; the
       final result is written to an output grid file.  When two grids are on the stack, each element in file A is  modified  by  the  corre-
       sponding  element  in file B.  However, some operators only require one operand (see below).  If no grid files are used in the expres-
       sion 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 grid file.  If not a file, it is interpreted as a numerical constant or
              a special symbol (see below).

       outgrdfile
              The name of a 2-D grid file that will hold the final result.  (See GRID FILE FORMATS below).

       OPERATORS
              Choose among the following 145 operators. "args" are the number of input and output arguments.

              Operator  args Returns

              ABS       1 1  abs (A).
              ACOS      1 1  acos (A).
              ACOSH     1 1  acosh (A).
```

Operating on grids

- `grdfilter` Filtering of data, e.g. convolution with Gaussian to smooth

Terminal

File Edit View Terminal Tabs Help

Terminal | Terminal | Terminal | Terminal

```
GRDFILTER(1)                 Generic Mapping Tools                 GRDFILTER(1)

NAME
       grdfilter - Filter a 2-D grid file in the space (or time) domain

SYNOPSIS
       grdfilter  input_file  -Ddistance_flag -F<filtertype><width>[mode] -Goutput_file [ -Ixinc[unit][=|+][/yinc[unit][=|+]] ] [ -Ni|p|r ] [
       -Rwest/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  isotropic  filters  and
       compute  distances  using  Cartesian or Spherical geometries.  The output .grd file can optionally be generated as a subOPT(R)egion of
       the input and/or with a new -Increment.  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
              The grid file of points to be filtered.  (See GRID FILE FORMATS below).

       -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  three  options  are  slower
              because they recompute weights for each latitude.

              flag = 3:  grid (x,y) in degrees, width in km, dx scaled by cosine(y), Cartesian distance calculation.
```

Operating on grids

- `grdfft` Fourier transform, e.g. to compute spatial power spectra

Terminal

File Edit View Terminal Help

```
GRDFFT(1)                   Generic Mapping Tools                   GRDFFT(1)

NAME
       grdfft - Perform mathematical operations on grid files in the wavenumber (or frequency) domain

SYNOPSIS
       grdfft  in_grdfile  -Gout_grdfile [ -Aazimuth ] [ -Czlevel ] [ -D[scale|g] ] [ -E[x|y][w] ] [ -F[x|y]params ] [ -I[scale|g] ] [ -L ] [
       -M ] [ -Nstuff ] [ -Sscale ] [ -Tte/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 grid are assumed to be in meters.  Geographical grids may be used by specifying the -M option that scales
       degrees  to  meters.   If you have grids with dimensions in km, you could change this to meters using grdedit or scale the output with
       grdmath.

       in_grdfile
              2-D binary grid file to be operated on.  (See GRID FILE FORMATS below).

       -G     Specify the name of the output grid file.  (See GRID FILE FORMATS below).

OPTIONS
       No space between the option flag and the associated arguments.

       -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
```

Operating on grids

- `shana/shsyn` Spherical harmonics analysis for global grids

Terminal

File Edit View Terminal Tabs Help

Terminal | Terminal | Terminal | Terminal

```
> shana -h

shana [l_max, 31] [input mode, 2] [integration mode, 0] [damping, 0]
         calculates spherical harmonic expansion for norm=1
         real theoretical physics spherical harmonic coefficients
         as in Dahlen and Tromp (1998)

         l_max:      maximum order of expansion
         input mode: choose different forms of possible input

                     name<.grd>: read from GMT grd-file
                                 if name is vec_p, look for vec_p.grd and vec_t.grd to
                                 expand vector field with phi theta components

                     something like "shana 31 mygrid" where mygrid.grd is a netcdf/GMT grid file
                     is likely what you want

                     1: read an ASCII file with points (lon, lat, d) for
                         contour delta function integration (e.g. for slabs)
                     5: read an ASCII file with spotted points (lon, lat, z) for A matrix output
                         where A is m by n with n the spherical harmonic coefficients evaluated at m data locations (2m for vectors, u_phi and u_theta)
                         (A matrix will be in binary single precision, FORTRAN style), e.g. for later least squares inversion
                     7: same as 5, but ascii output, regular, C style
                     50: same as 5 but output for poloidal/toroidal vector field expansion
                     70: same as 7 but output for poloidal/toroidal vector field expansion

                     6: read an ASCII file with spotted points (lon, lat, z) for spotted SH analysis with delta functions
```