Appendix A: GMT supplemental programs

These programs are for the most part neither written nor supported by us. They provide extensions of GMT that are needed for particular rather than general applications. The software is provided in a separate archive (GMT_supplements.tar.Z; see Appendix D). Questions and/or bug reports for this software should be addressed to the person(s) listed in the README file associated with the particular program. The contents of the supplemental archive may change without notice; at this writing it contains these directories:

dbase	Programs to extract data from global gridded data sets such as those avaiable from NGDC.
gmtsun	For Sun workstations only. Contains shell-scripts for rasterization of PostScript on Sun workstations and provides an OpenWindows Defaults-editor for GMT (xgmtdefaults).
meca	Programs for plotting focal mechanisms and error ellipses.
mex	Mex files for Matlab® to read/write grd-files.
mgg	Programs to maintain, access, extract data from, and plot marine geophysical data files (MGD-77[†]).
xgrid	Presents an X11 editor for grd-files.

Appendix B: GMT file formats

1. ASCII tables

These files have N records which have M fields each. The first data record may be preceded by 1 or more header records. When using such files, make sure to use the –H option and set the parameter N_HEADER_RECS in the .gmtdefaults file (System default is 1 header record if –H is set). Fields within a record must be separated by spaces or tabs (a few programs may handle commas). Most programs can read multicolumn files, but require that the *x* [and *y*] variable(s) be stored in the 1st [and 2nd] column (There are, however, some exceptions to this rule, such as *filter1d* and *sample1d*). When dealing with time- or (*x,y*)-series it is usually convenient to have each profile in separate files. However, this may sometimes prove impractical due to large numbers of profiles. An example is files of digitized lineations where the number of individual features may range into the thousands. One file per feature would in this case be unreasonable and furthermore clog up the directory. GMT provides a mechanism for keeping more than one profile in a file. Such files are called multiple segment files and are identical to the ones just outlined except that they have subheaders interspersed with data records that signal the start of a segment. The subheaders may be of any format, but all must have the same character in the first column. When using such files, you must specify the –M option. The unique character is by default '>', but you can override that by appending your chosen character to the –M option. E.g., –MH will look for subheaders starting with H, whereas –M'*' will check for asterisks (The quotes are necessary since * has special meaning to *UNIX*).

Note that some programs now support binary tables to speed up input-ouput for i/o-intensive tasks like gridding and preprocessing.

[†] Available on CD-ROM from National Geophysical Data Center (contact pws@mail.ngdc.noaa.gov)

2. 2-D grdfile.

The default 2-D binary netCDF grd-file in GMT has several attributes. The *grdedit* utility program will allow you to edit parts of the header of an existing grdfile. The following attributes are contained within the header record:

Header record:

char title[80];	Descriptive title of the data set
char command[320];	Command line that produced the grdfile
char remark[160];	Any additional comments
char x_units[80];	Units of the x-dimension
char y_units[80];	Units of the y-dimension
char z_units[80];	Units of the z-dimension
int nx;	Number of nodes in the x-dimension
int ny;	Number of nodes in the y-dimension
double x_min;	Minimum x-value of region
double x_max;	Maximum x-value of region
double y_min;	Minimum y-value of region
double y_max;	Maximum y-value of region
double z_min;	Minimum z-value in data set
double z_max;	Maximum z-value in data set
double x_inc;	Node spacing in x-dimension
double y_inc;	Node spacing in y-dimension
double z_scale_factor;	Factor to multiply z-values after read
double z_add_offset;	Offset to add to scaled z-values
int node_offset;	0 for grid line registration, 1 for pixel registration

Data record:

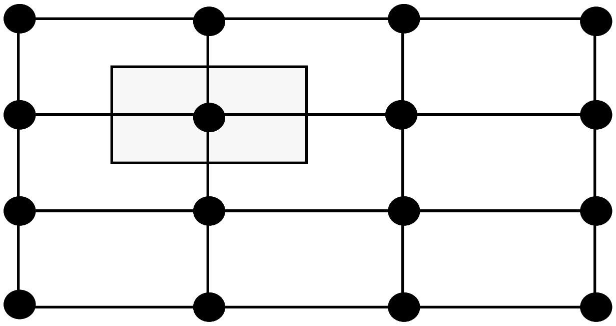
float z[nx*ny]	z-values stored in scanline format

GMT version 3 also allows other formats to be read. In addition to the default netcdf format it can use binary floating points and short integers, as well as 8-bit Sun rasterfiles (colormap ignored). Additional formats may be used by supplying read/write functions and linking these with the GMT libraries. The source file gmt_customio.c has the information that programmers will need to augment GMT to read custom grdfiles. We anticipate that the number of pre-programmed formats will increase as enterprising users implement what they need.

Scanline format means that the data are stored in rows (y = constant) going from the "top" (*y* = y_max (north)) to the "bottom" (*y* = y_min (south)). Data within each row are ordered from "left" (*x* = x_min (west)) to "right" (*x* = x_max (east)). The node_offset signals how the nodes are laid out. The grid is always defined as the intersections of all x (x = x_min, x_min + x_inc, x_min + 2*x_inc, ..., x_max) and y (y = y_min, y_min + y_inc, y_min + 2*y_inc, ..., y_max) lines. The two scenarios differ in which area each data point represents.

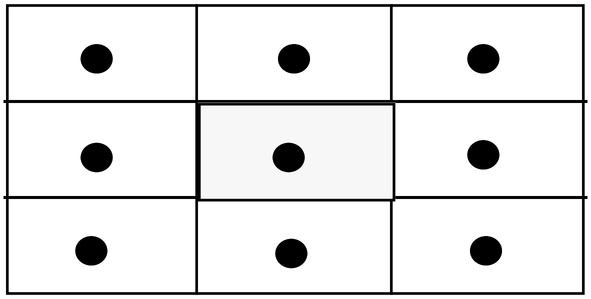
1) Grid line registration.

In this registration, the nodes are centered on the grid line intersections and the data points represent the average value in a cell of dimensions (x_inc * y_inc) centered on the nodes:



2) Pixel registration

Here, the nodes are centered in the grid cells, i.e., the areas between grid lines, and the data points represent the average values within each cell:



Thus, inspecting the figures we see that in the case of grid registration the number of nodes are related to region and grid spacing by

nx = (x_max - x_min) / x_inc + 1 = 4
ny = (y_max - y_min) / y_inc + 1 = 4

while for pixel registration we find

nx = (x_max - x_min) / x_inc = 3
ny = (y_max - y_min) / y_inc = 3

The default registration in GMT is grid line registration. Most programs can handle both types, and for some programs like *grdimage* a pixel registered file makes more sense. Utility programs like *grdsample* and *grdproject* will allow you to convert from one format to the other.

3. Icon files

The Sun monochrome icon file format consists of a header followed by a series of hexadecimal numbers that represents the bit-pattern. Bits are scanline oriented. Most icons have dimensions of 64 by 64, but other sizes will be accepted as well.

Bit-patterns created with Sun's *Icon Editor* or *icontool* are automatically in the correct format. If you use other means of creating the bit-pattern, make sure the output file looks like the example below. The following is an example of one of the predefined patterns supplied with GMT. It was created with the icontool under *sunview* and it corresponds to pattern # 20 in Appendix E.

```
/* Format_version=1, Width=64, Height=64, Depth=1, Valid_bits_per_item=16
 */
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0x0C00,0x0000,0x0C00,0x0000,
    0x1800,0x0000,0x1800,0x0000,0x1800,0x0000,0x1800,0x0000,
    ... <28 lines of the same format > ...
    0x0000,0xC000,0x0000,0xC000,0x0000,0xC000,0x0000,0xC000,
    0x0001,0x8000,0x0001,0x8000,0xFFFF,0xFFFF,0xFFFF,0xFFFF
```

Note that icons created with *Icon Editor* under *OpenWindows* 3.0 only write 4 hexadecimal numbers per line. This change is not reflected in the Format_version argument. However, GMT programs can decode these files as well.

Appendix C: How to make GMT Encapsulated PostScript Files

GMT actually produces Encapsulated PostScript files (EPS). However, there is much confusion over what an EPS file is and if other programs can read it. Much of this has to do with the claim by some software manufacturers that their programs can read and edit EPS files. We get much mail from people asking us to let GMT produce EPS files that can be read, e.g., by Adobe Illustrator. This is a problem with Adobe Illustrator and similar programs, not GMT! We hope to allow GMT to produce Adobe Illustrator-compatible PostScript in the future. Fortunately, there are now many ways to convert general PostScript files to the subset of the EPS format that is understood by Adobe Illustrator, Aldus Freehand, and others.

The public domain programs ps2ai and EpsConverter both read generic PostScript and produce Adobe Illustrator-compatible EPS files. They are both available over Internet. The commercial product Adobe Destiller will do the same. There are probably many other ways we are simply not familiar with. We have successfully converted GMT PostScript files with EpsConverter and modified these further in Aldus Freehand. We have also created Portable Document Format (PDF) files using Adobe Destiller and imported these into Adobe Illustrator 5.5.

If you do not want to modify the illustration but just include it in a text document: Many word processors (such as Word) will let you include a PostScript file that you may place but not edit. You wont be able to view the figure on-screen, but it will print correctly. All illustrations in the GMT documentation were GMT-produced PostScript files that were included as is into a Word document.

Users should be aware that with GMT plots composed of multiple overlays there is no way that theBoundingBox information created by the first GMT call can be modified by later calls. Hence, you may manually have to modify the BoundingBox dimensions for some EPS placement programs to work properly. Again, utilities exist both in the public domain and commercially that will alleviate these problems. For further information send email to gmthelp@soest.hawaii.edu; more experienced GMT users may be able to help you out.

These examples do not constitute endorsements of the products mentioned above; they only represent our limited experience with the problem.

Appendix D: Availability of GMT and related code

All the source code, support data, a *PostScript* file of this document, and the manual pages can be obtained by anonymous ftp. We also maintain a GMT page on the World Wide Web (URL: http://www.soest.hawaii.edu/soest/gmt.html). For instructions on how to obtain GMT, send mail to listserver@soest.hawaii.edu containing the single message

information gmtgroup

The listserver will then automatically send you information including the shellscript install_gmt which you may run to obtain all necessary files and complete the installation. gmtgroup is actually a mailing list to which users of GMT can subscribe. Users on this mailing-list will be notified about bug fixes and upgrades (See Chapter 7).

In addition to the main archive, two other archives may be of interest to you. GMT_scripts.tar.Z contains all the shellscripts and support data used in the Cookbook section, while GMT_supplements.tar.Z contains several programs written by us and GMT users elsewhere. This archive currently includes programs for extracting grdfiles from global databases (such as those on CD-ROMs from NGDC), plotting focal mechanisms (the seismologist's "beach balls") and error ellipses on maps, an X11 editor for grdfiles, programs for handling and plotting of MGD-77 files, and mex-files for users of Matlab® to read/write grdfiles (See Appendix A). Most of these programs were not developed by us and no support is offered. Direct bug reports etc. directly to the authors (see the appropriate README and man pages for each program).

The netCDF library that makes up the backbone of the grdfile I/O operations can be obtained from NCAR. A compressed tar file can be accessed (in binary mode) from the file netcdf.tar.Z in the anonymous FTP directory of unidata.ucar.edu (Internet 128.117.140.3). The software distribution includes a *PostScript* file of the netCDF User's Guide. [A mailing list, netcdfgroup@unidata.ucar.edu, is available for discussion of the netCDF interface and announcements about netCDF bugs, fixes, and enhancements. To subscribe, send a request to netcdfgroup-adm@unidata.ucar.edu].