

GMT
GRD2ASCII

**Programme de conversion de grilles
binaires (netCDF) en grilles ascii**

(Version 3.0)

Germinal GABALDA
Laboratoire de Géophysique - Orstom - Bondy

Février 1996

Fonds Documentaire ORSTOM



010017289

Fonds Documentaire ORSTOM
Cote: Ax 17289 Ex: 1

GMT

GRD2ASCII

**Programme de conversion de grilles
binaires (netCDF) en grilles ascii**

(Version 3.0)

**Germinal GABALDA
Laboratoire de Géophysique - Orstom - Bondy**

Février 1996

INTRODUCTION	5
---------------------	----------

UTILISATION	7
--------------------	----------

ANNEXES

ANNEXE 1: INSTALLATION	9
-------------------------------	----------

- 1. GMT - THE GENERIC MAPPING TOOL**
- 2. PREPARATION**
- 3. INSTALLATION DE GRD2ASCII**
- 4. MAKEFILE**

ANNEXE 2: MAN	11
----------------------	-----------

ANNEXE 3: GRD2ASCII.C	13
------------------------------	-----------

ANNEXE 4: BIBLIOGRAPHIE	17
--------------------------------	-----------

GRD2ASCII est un programme permettant de transformer une grille binaire « GMT » (format netCDF) en grille ASCII. Il est écrit en Langage C et il utilise les ressources du produit GMT (Generic Mapping Tool) et les bibliothèques netCDF. Se reporter à l'*annexe 1* pour l'installation de `grd2ascii` et de son manuel en ligne.

L'affichage du `man` (**`man grd2ascii`**) donne une description complète sur la fonction et l'utilisation de cette commande (*Annexe 2*), mais un rapide résumé peut être obtenu en exécutant le programme sans arguments (**`grd2ascii`**). Les différentes options possibles sont néanmoins développées plus longement dans la partie *UTILISATION*.

Pour obtenir de plus amples renseignements sur le format des fichiers « `grd` » il est conseillé de se reporter au document *Technical Reference and Cookbook - Appendix B* dont les références sont citées en annexe 4.

Pour toute suggestion ou renseignement complémentaire

Germinal GABALDA Orstom - Laboratoire de Géophysique 32, av. Henri Varagnat - 93143 - BONDY CEDEX FRANCE Email: gabalda@gravi.bondy.orstom.fr

La syntaxe complète de la commande `grd2ascii` est la suivante:

`grd2ascii grdfile [-Ccol] [-Nnodata] [-Rouest/est/sud/nord] [-r]`

Utilisé sans options `grd2ascii` lie un fichier binaire (**grdfile**) et écrit sur la sortie standard la représentation `ascii` correspondante. Un entête contenant toutes les informations nécessaires précède l'affichage de la grille. Par défaut, la grille est écrite par latitudes décroissantes (Nord en haut) et longitudes croissantes (Ouest -> Est). Les noeuds sans valeurs sont remplacés par la chaîne alphanumériques `NaN`. Il est cependant possible de modifier la grille en entrée à l'aide des options décrites ci-dessous:

-Ccol permet de limiter le nombre de champs par ligne. Cela peut être intéressant pour faire une sortie imprimante.

-Nnodata pour remplacer `NaN` par une valeur numérique.

-Rouest/est/sud/nord permet de définir une sous-région pour extraire une sous-grille. Les limites peuvent être indiquées en degrés et minutes [et secondes] avec le format **`dd:mm[:ss]`**.

-r pour inverser la grille (Ecriture des latitudes à partir du Sud).

Vous pouvez changer le format d'écriture des données en sortie en modifiant le paramètre **`D_FORMAT`** dans le fichier **`.gmtdefaults`**.

Ci-dessous, nous détaillons le fichier `ascii` résultat de la commande:

`grd2ascii Te.resu.grd -C11 -N9999 -R10/20/-5/5 > Te.congo.asc`

`Te.resu.grd` est une grille binaire contenant pour toute l'Afrique les valeurs des épaisseurs élastiques. La commande `gmt` **`grdinfo`** nous donne les renseignements suivant:

```
Te.resu.grd: Title: Epaisseur Elastique
Te.resu.grd: Command: grdmath Te.af.grd x afr_mask.grd = Te.resu.grd
Te.resu.grd: Remark:
Te.resu.grd: Normal node registration used
Te.resu.grd: grdfile format # 0
Te.resu.grd: x_min: -20 x_max: 55 x_inc: 0.5 units: degre nx: 151
Te.resu.grd: y_min: -40 y_max: 40 y_inc: 0.5 units: degre ny: 161
Te.resu.grd: zmin: 0 zmax: 103 units: km
Te.resu.grd: scale_factor: 1 add_offset: 0
```

L'en-tête ascii reprend les informations contenues dans le header du fichier binaire Te.resu.grd. Nous pouvons noter la mise à jour de *command*, des limites géographiques (*xmin*, *xmax*, *ymin*, *ymax*) et des dimensions de la grille (*nx*, *ny*). Les champs *no_data* (ligne 9) pour les valeurs inconnues (ici à 9999), *longitude range* (ligne 10) et *latitude range* (ligne 11) ont été rajoutés.

Les deux premières lignes (en gras) correspondent à la latitude +5°N

```

Title      : Epaisseur Elastique
Command    : grd2ascii Te.resu.grd -C11 -N9999 -R10/20/-5/5
Remark     :
Normal     : node registration
0          : grdfile format (not used)
10 20 0.5 degre 21      : x_min x_max x_inc units nx
-5 5 0.5 degre 21      : y_min y_max y_inc units ny
0 103 km      : z_min z_max units
1 0 9999      : scale_factor add_offset no_data
10 20        : longitude range
5 -5         : latitude range
23      23      21      41      39      67      59      57      37      39      41
53      31      103     81      69      49      85      103     103     103     103
25      19      31      17      23      25      29      73      29      101     45
59      23      75      65      59      59      103     37      39      57      41
21      23      27      21      39      45      47      77      53      51      41
57      75      73      61      31      25      103     75      37      9999
21      19      25      19      31      103     41      77      69      71      89
85      59      103     103     87      103     45      103     77      35      103
21      21      21      25      33      37      9999     53      49      79      103
57      103     103     103     79      103     103     103     103     103     103
21      25      9999     29      61      97      59      55      69      97      63
35      45      9999     103     61      103     103     103     103     103     103
25      21      23      27      41      59      87      63      55      83      103
41      9999     103     103     77      103     103     37      9999     9999
21      19      19      23      39      47      35      57      35      37      51
51      103     103     103     77      103     59      81      29      79      51
23      17      23      27      35      31      41      65      57      61      51
63      71      71      63      57      99      87      59      73      103     83
21      19      29      29      39      31      103     45      39      69
103     103     37      33      49      45      79      103     103     9999
9999     17      23      35      37      55      103     39      29      91      83
87      103     103     103     45      39      103     103     83      103     45
9999     17      25      43      43      103     103     103     37      103
47      103     57      47      103     103     71      103     9999     91
9999     9999     9999     67      47      69      45      95      39      23      43
103     89      103     103     49      103     79      103     71      77
9999     9999     23      51      47      35      35      103     103     103     91
73      103     81      103     103     87      103     89      103     103
9999     9999     9999     31      31      27      103     39      103     25      33
31      103     103     33      103     47      33      103     77      81
9999     21      29      27      29      25      45      65      33      59      79
37      27      103     103     51      103     69      81      77      73
9999     19      39      27      31      27      83      79      63      73
43      41      33      39      103     9999     77      37      59      47
9999     0       37      33      27      33      35      23      31      93      45
31      49      45      51      55      89      79      79      73      55
9999     0       0       57      29      27      37      87      103     77      41
103     47      59      63      61      75      73      9999     45      55
9999     0       0       0       51      35      39      61      103     71      35
103     103     97      31      31      29      29      87      103     47
9999     0       0       0       9999     9999     9999     103     103     39      29
47      103     103     39      29      9999     33      39      27      89
    
```

1. GMT - The Generic Mapping Tool

GMT est un ensemble d'outils Unix qui permettent de manipuler des données et de produire des dessins postscripts. GMT peut être obtenu gratuitement sur internet ainsi que les bibliothèques netCDF qui sont utilisées par plusieurs fonctions de GMT.

- ftp anonyme à partir de la machine **kiawe.soest.hawaii.edu** (128.171.151.16) sous le répertoire pub/gmt
- www à l'adresse **<http://www.soest.hawaii.edu/soest/gmt.html>**.

2. Préparation

- Créez un répertoire sous le répertoire contenant les sources GMT
- Copiez les fichiers **grd2ascii.c**, **grd2ascii.man** et **makefile** dans ce répertoire
- Editez le makefile et adaptez à votre site les variables **PRODUC**, **NETCDF**, **LIBDIR**, **BINDIR** et **MANDIR**
PRODUC = `grd2ascii`
NETCDF = Chemin pour atteindre netCDF
LIBDIR = Chemin pour atteindre les bibliothèques GMT
BINDIR = Répertoire d'installation du binaire `grd2ascii`
MANDIR = Répertoire d'installation du man

3. Installation de GRD2ASCII

- Pour générer et installer le programme, installer le manuel en ligne et nettoyer le répertoire, tapez **make all**
- Pour générer le programme, tapez **make bin**
- Pour installer le programme, tapez **make install**
- Pour installer le manuel en ligne, tapez **make man**
- Pour nettoyer le répertoire, tapez **make clean**

Si BINDIR et MANDIR ne sont pas des répertoires standards alors pensez à modifier les variables **PATH** et **MANPATH**.

4. Makefile

```

# Follow the instructions in this makefile to customize your setup.
# To compile/link/install the program, install man pages and clean, try "make all".
# To compile/link the program, try "make bin".
# To install the program, try "make install".
# To install man pages, try "make man".
# To clean out directory with "make clean".
#
# Author:          Germinal Gabalda
#                 ORSTOM - Geophysique
#                 Bondy - 93143 - FRANCE
# Date:           1-AUG-1995
#-----
# Set PRODUC,NETCDF, BINDIR, LIBDIR, and MANDIR before making anything!
#-----
# PRODUC          -> The name of the program
# NETCDF          -> Where to find netcdf sub-directories lib and include
# BINDIR          -> Where to install executable code
# LIBDIR          -> Where GMT support data files are installed. This path
#                 must be hard, i.e., starting from root /
# MANDIR          -> Where GMT manual pages are installed
#-----
PRODUC          = grd2ascii
NETCDF         = /usrx/NETCDF-2.3.2
BINDIR         = /usrx/GMT-3.0/bin
LIBDIR         = /usrx/GMT-3.0/lib
MANDIR         = /usrx/GMT-3.0/man/man1
#----- stop here -----
NETCDFLIB        = $(NETCDF)/lib
NETCDFINC        = $(NETCDF)/include
#-----
SHELL=/bin/sh
CC               = cc
SI               = -DSI
INSTALL          = install -s
CFLAGS           = -O -L. -I$(NETCDFINC) -DLIBDIR=$(LIBDIR)" $(SI)
CDF              = -L$(NETCDFLIB) -lnetcdf
PS              = -lpsl
GMT              = -lgmt
#-----
.SUFFIXES:
all: bin install man clean

# BINARY FILES -----
bin:   $(PRODUC)

$(PRODUC):   $(PRODUC).c
            $(CC) $(CFLAGS) $? -L. -lgmt $(CDF) -lm -o $@

# INSTALL -----
install: $(BINDIR)/ $(PRODUC)

$(BINDIR)/ $(PRODUC): $(PRODUC)
                  $(INSTALL) $? $(BINDIR)

# MANS FILES -----
man:   $(MANDIR)/ $(PRODUC).1

$(MANDIR)/ $(PRODUC).1: $(PRODUC).man
                        cp $? $@

# CLEAN -----
clean:
            rm -f *.o $(PRODUC)

```


GRD2ASCII(1) MISC. REFERENCE MANUAL PAGES GRD2ASCII(1)

NAME

grd2ascii - Converting a netCDF grdf file to an ASCII grid with header

SYNOPSIS

grd2ascii grdffile [-Ccol] [-Nnodata] [-Rwest/east/south/north] [-r]

DESCRIPTION

grd2ascii reads a binary 2-D grdf file and writes out a grid in ASCII format to standard output. The region option can be used to define a subregion to be cut out. You can modify the precision of the ASCII output format by editing the D_FORMAT parameter in your .gmtdefaults file.

grdffile. The 2-D binary file to be converted.

OPTIONS

- C col specifies the number of fields by line in the standard output.
- N nodata. Set value for nodes with NaN.
- R west, east, south, and north specify the Region of interest. To specify boundaries in degrees and minutes [and seconds], use the dd:mm[:ss] format
- r to print latitude from south [Default is north].

EXAMPLES

To convert, reverse and reduce the 2-D binary grid with 10 fields by line, try

```
grd2ascii nica_topo.grd -C10 -R-87/-86/12/13 -r
```

SEE ALSO

gmtdefaults, gmt, grdedit, grdinfo, xyz2grd, ascii2grd

REFERENCES

- Wessel, P., and W. H. F. Smith, 1995, The Generic Mapping Tools (GMT) version 3.0 Technical Reference & Cookbook, SOEST/NOAA.
- Wessel, P., and W. H. F. Smith, 1991, Free software helps map and display data, EOS Trans. AGU, 72, 441.

AUTHOR

Germinal Gabalda (gabalda@gravi.bondy.orstom.fr)
Geophysic Laboratory
ORSTOM - (F - 93143 Bondy Cedex)

Based on `grd2xyz.c` (version 2.0) of Paul Wessel

Sun Release 4.1 Last change: 15 February 1996

1

```

/*
 * grd2ascii.c reads a grd file and prints out an ascii grid file with header.
 *
 * Author:      Germinal Gabalda - Orstom (Geophysic Laboratory)
 *              Email: gabalda@gravi.bondy.orstom.fr
 * Date: 15-FEB-1996
 * History:     Based on grd2xyz.c (version 2.0) of Paul Wessel
 * Version:     3.0 (v1.0/1-AUG-1995)
 */

#include "gmt.h"

float *z;
char *type[2] = {"Normal", "Pixel"};

main (argc, argv)
int argc;
char **argv, {
    int i, j, ij, nm, nx, ny, dummy[4], one_or_zero, col;
    int error = FALSE, global = FALSE, reverse = FALSE, collim = FALSE;
    double w, e, s, n, d, no_data;
    char *grdfile = NULL;
    struct GRD_HEADER grd;

    argc = gmt_begin(argc, argv);

    no_data = gmt_NaN;

    w = e = s = n = 0.0;
    dummy[0] = dummy[1] = dummy[2] = dummy[3] = 0;

    for (i = 1; i < argc; i++) {
        if (argv[i][0] == '-') {
            switch (argv[i][1]) {

                case '\0':
                    gmt_quick = TRUE;
                    break;

                case 'C':
                    collim = TRUE;
                    col = atoi (&argv[i][2]);
                    if (col < 1) {
                        error++;
                        fprintf (stderr, "\t-C<col>, col is number of fields by line in
                            the output\n");
                    }
                    break;

                case 'N':
                    if (!argv[i][2]) {
                        fprintf (stderr, "grd2ascii: GMT SYNTAX ERROR -N
                            option: Must specify value\n");
                        error++;
                    }
                    else
                        no_data = atof (&argv[i][2]);
                    break;
            }
        }
    }
}

```

```

        case 'R':
            error += get_arg_R (argv[i], &w, &e, &s, &n);
            break;

        case 'r':
            reverse = TRUE;
            break;

        default:
            error = TRUE;
            fprintf (stderr, "grd2ascii: GMT SYNTAX ERROR: Unrecognized
                option -%c\n", (argv[i][1]));
            break;
    }
}
else
    grdfile = argv[i];
}

if (argc == 1 || gmt_quick) {
    fprintf (stderr, "grd2ascii - Converting a netCDF grdfile to an ASCII grid\n\n");
    fprintf (stderr, "usage: grd2ascii <grdfile> [-C<col>] [-N<nodata>]
        [-R<west/est/south/north>] [-r]\n\n");

    if (gmt_quick) exit (-1);

    fprintf (stderr, " <grdfile> is the grd file to convert\n");

    fprintf (stderr, "\n\tOPTIONS:\n");
    fprintf (stderr, " -C specifies the number of fields by line in the standard output\n");

    fprintf (stderr, " -N set value for nodes with NaN\n");

    fprintf (stderr, " -R west, est, south and north specify the region of interest. To specify\n");
    fprintf (stderr, "     boundaries in degrees and minutes [and seconds], use dd:mm[:ss]
        format.\n");

    fprintf (stderr, " -r to print latitude from SW coner [default is latitude from NW coner].\n");
    exit (-1);
}

if (!grdfile) {
    fprintf (stderr, "grd2ascii: GMT SYNTAX ERROR: Must specify input file\n");
    error++;
}
if (error) exit (-1);

if (read_grd_info (grdfile, &grd)) {
    fprintf (stderr, "grd2ascii: Error opening file %s\n", grdfile);
    exit (-1);
}

grd_init (&grd, argc, argv, TRUE);

nm = grd.nx * grd.ny;

```

```

if (e > w && n > s) {
    global = (fabs (grd.x_max - grd.x_min) == 360.0);
    if (!global && (w < grd.x_min || e > grd.x_max)) error = TRUE;
    if (s < grd.y_min || n > grd.y_max) error = TRUE;
    if (error) {
        fprintf (stderr, "grd2ascii: GMT ERROR: Subset exceeds data domain!\n");
        exit (-1);
    }
    one_or_zero = (grd.node_offset) ? 0 : 1;
    nx = rint ((e - w) / grd.x_inc) + one_or_zero;
    ny = rint ((n - s) / grd.y_inc) + one_or_zero;

    z = (float *) memory (NULL, nx * ny, sizeof (float), "grd2ascii");

    if (read_grd (grdfile, &grd, z, w, e, s, n, dummy, FALSE)) {
        fprintf (stderr, "grd2ascii: Error reading file %s\n", grdfile);
        exit (-1);
    }
}
else {
    z = (float *) memory (NULL, nm, sizeof (float), "grd2ascii");

    if (read_grd (grdfile, &grd, z, 0.0, 0.0, 0.0, 0.0, dummy, FALSE)) {
        fprintf (stderr, "grd2ascii: Error reading file %s\n", grdfile);
        exit (-1);
    }
}

fprintf (stdout, "Title : %s\n", grd.title);
fprintf (stdout, "Command : %s\n", grd.command);
fprintf (stdout, "Remark : %s\n", grd.remark);
fprintf (stdout, "%s\t: node registration\n", type[grd.node_offset]);
fprintf (stdout, "%d\t: grdfile format (not used)\n", grd_i_format);
fprintf (stdout, "%lg %lg %lg %s %d\t: x_min x_max x_inc units nx\n", grd.x_min, grd.x_max,
        grd.x_inc, grd.x_units, grd.nx);
fprintf (stdout, "%lg %lg %lg %s %d\t: y_min y_max y_inc units ny\n", grd.y_min, grd.y_max,
        grd.y_inc, grd.y_units, grd.ny);
fprintf (stdout, "%lg %lg %s\t: z_min z_max units\n", grd.z_min, grd.z_max, grd.z_units);
fprintf (stdout, "%lg %lg %lg\t: scale_factor add_offset no_data\n", grd.z_scale_factor,
        grd.z_add_offset, no_data);
fprintf (stdout, "%lg %lg\t: longitude range\n", grd.x_min, grd.x_max);
if (!reverse)
    fprintf (stdout, "%lg %lg\t: latitude range\n", grd.y_max, grd.y_min);
else
    fprintf (stdout, "%lg %lg\t: latitude range\n", grd.y_min, grd.y_max);

if (!reverse) {
    for (j = ij = 0; j < grd.ny, j++) {
        for (i = 0; i < grd.nx; i++, ij++) {
            if (bad_float (z[ij])) z[ij] = no_data;
            fprintf (stdout, gmtdefs.d_format, z[ij]), putc ('\t', stdout);
            if (collim && (i+1) % col == 0 && (i+1) != grd.nx) fprintf (stdout, "\n");
        }
        fprintf (stdout, "\n");
    }
}

```

```

else {
    for (j = grd.ny; j > 0; j--) {
        for (i = 0; i < grd.nx; i++) {
            ij = grd.nx * (j-1) + i;
            if (bad_float (z[ij])) z[ij] = no_data ;
            fprintf (stdout, gmtdefs.d_format, z[ij]), putc ('\t', stdout);
            if (collim && (i+1) % col == 0 && (i+1) != grd.nx) fprintf (stdout, "\n");
        }
        fprintf (stdout, "\n");
    }
}

free ((char *)z);

gmt_end (argc, argv);
}

int get_arg_R (item, w, e, s, n)
char *item;
double *w, *e, *s, *n; {
    char *text, string[100];

    /* get_arg_R interprete la ligne de commande pour l'option -R */

    int i, error = 0;
    double *p[4], ddmms_to_degree ();

    switch (item[1]) {
        case 'R':
            p[0] = w;    p[1] = e;    p[2] = s;    p[3] = n;

            i = 0;
            strcpy (string, &item[2]);
            text = strtok (string, "/");
            while (text) {
                *p[i] = ddmms_to_degree (text);
                i++;
                text = strtok (NULL, "/");
            }
            if ((i < 4) || (*p[0] >= *p[1]) || (*p[2] >= *p[3]) ) {
                error++;
                fprintf (stderr, "grd2ascii: GMT SYNTAX ERROR -R option. Correct
                    syntax:\n");
                fprintf (stderr, "\t-R<xmin>/<xmax>/<ymin>/<ymax>, dd:mm format
                    ok\n");
            }
            break;
    }

    return (error);
}

```

- Gabalda G., GMT, ASCII2GRD, Programme de conversion de grilles ascii en grilles binaires (netCDF), (Version 1.0 - Février 1996), Orstom, Laboratoire de Géophysique, Bondy, *Notice d'utilisation*, 27 p.
- Wessel, P., and W. H. F. Smith, New version of the Generic Mapping Tools released, *EOS Trans. Amer. Geophys. U.*, vol. 76, pp. 329, 1995.
- Wessel, P., and W. H. F. Smith, 1995, The Generic Mapping Tools (GMT) version 3.0 Technical Reference & Cookbook, SOEST/NOAA.
- Wessel, P., and W. H. F. Smith, New version of the Generic Mapping Tools released, *EOS Trans. Amer. Geophys. U. electronic supplement*, http://www.agu.org/eos_elec95154e.html, 1995.
- Wessel, P., and W. H. F. Smith, Free software helps map and display data, *EOS Trans. Amer. Geophys. U.*, vol. 72, pp. 441, 445-446, 1991.