GRID2PS

A Program to Display Gridded Radar Data in PostScript

Susan Stringer, Wen-Chau Lee
Remote Sensing Facility
National Center for Atmospheric Research

October, 1997

GRID2PS TABLE OF CONTENTS

GRID2PS

1.0 INTRODUCTION

Grid2ps creates publication quality, native PostScript images from gridded radar data. Currently, grid2ps accepts data gridded in CEDRIC and a data format developed by Wen-Chau Lee. Additionally, routines to read or manipulate data can easily be added. Grid2ps can plot from one to ten images per page in both portrait and landscape orientation. The appearance of each image on the page is controlled by an input file. Each input file allows the user to specify the data file, coordinate system and a variety of other parameters that control the appearance of the plot. Grid2ps runs under the Solaris operating system.

Unpacking the software

The procedures for unpacking the software depends on whether you have accessed the software from a tape or via ftp.

Extracting the software from a tape

prompt> tar xvfp device name

For example, if the device name is /dev/rmt/0, then place the tape in the drive and type:

prompt> tar xvfp /dev/rmt/0

Extracting the software via ftp

Get the tar file:

ftp stout.atd.ucar.edu and login as anonymous
cd /pub/snorman
get grid2ps.tar.Z

Uncompress the tar file:

prompt> uncompress grid2ps.tar.Z

Untar the file:

prompt> tar xvfp grid2ps.tar

Building the software

There is Makefile included under the directory /grid2ps1.0/src along with all the source code. You must have NCAR Graphics installed on your system in order to run grid2ps. The CEDRIC read routine (CEDREAD.f) contains calls to NCAR Graphics. If you do not have NCAR Graphics installed, you may replace CEDREAD.f with your own routine. The call to CEDREAD.f must be removed from the main program. The program can then be compiled using the file ‘Makefile.alternate’.

Running the software

To run grid2ps, type:

prompt> grid2ps [template] [input1..]

The first argument (template) is the desired template and the rest of the arguments are the corresponding input files. For example, the following command creates a two panel image

(2 rows, 1 column):

prompt> grid2ps 2 inp1 inp2

In the example above, 2 is the template number and inp1, inp2 are the input files corresponding to each panel. If no command line arguments are given, then the user will be prompted for the template and the default input file name(s) are inp1, inp2...etc.

Templates

The user may select from 1 to 13 different templates to display the data. The choice of template reflects the number of panels per page and the orientation. The following table describes the templates.

TEMPLATE DEFINITIONS

TEMPLATE	# ROWS	# COLUMNS	# PANELS	ORIENTATION
1	1	1	1	Portrait
2	2	1	2	Portrait
3	3	1	3	Portrait
4	4	1	4	Portrait
5	5	1	5	Portrait
6	1	2	2	Portrait
7	2	2	4	Portrait
8	3	2	6	Portrait
9	4	2	8	Portrait
10	5	2	10	Portrait
11	1	1	1	Landscape
12	1	2	2	Landscape
13	2	2	4	Landscape

Input file

There are a variety of options that may be exercised to manipulate the appearance of the plot. These parameters are controlled through an input file. There is one input file for each panel on the page. The number of input files is dependent upon the template chosen by the user. The table above summarizes the configuration options. Please see Appendix B for a sample input file.

2.0 MANDATORY INPUT FILE PARAMETERS

The following parameters are mandatory. The program will not execute properly if these values are not filled in correctly. Grid2ps will generate a log file called ‘grid2ps.log’ that will list the parameters in the input file(s) and check for possible errors. If the program terminates prematurely, it is a good idea to check the log file. To avoid read errors, please enclose any string in quotes.

#OUTPUT_NAME
Name of the output postscript file.

#INPUT_FLAG
Flag that indicates the input format. For CEDRIC input (.mud file), enter C. For text input (Wen-Chau Lee’s format), enter W. If the use supplies their own read routine, then the flag must be some identifier (other that ‘C’ or ‘W’).

#DATA_FILE
Name of the input data file.

#COLOR
Enter ‘Y’ if the plot is color, or ‘N’ for a gray scale plot.

#CONTOUR_TYPE
This flag applies to color plots and indicates where a grey contour is filled.
0: The 0 contour line is filled with gray
1: The midpoint of range is filled with gray
2: No gray in contours

#FIX_AXIS
The fixed axis for the plot.

#FIELD1
Horizontal velocity field name. The name must match the name in the input data file.

#FIELD2
Vertical velocity field name. The name must match the name in the input data file.

#FIELD3
Primary contour field name. The name must match the name in the input data file.

#FIELD4
Secondary contour field name (used if plot_type = 4 or 5). The name must match the name in the input data file.

#PLOT_TYPE
Type of plot desired.
1. vector only
2. contour only
3. vector & primary contour
4. primary contour & secondary contour
5. vector, primary contour & secondary contour

#FILL_FLAG
Y = fill the primary contour field; N = do not fill the primary contour field.

#FILL_BEG
Beginning fill value (if fill_flag = ‘Y’)

#FILL_END
Ending fill value (if fill_flag = ‘Y’)

#FILL_INC

Fill increment (if fill_flag = ‘Y’)

#NOFILL_BEG

Beginning contour line value (if fill_flag = ‘N’)

#NOFILL_END

Ending contour line value (if fill_flag = ‘N’)

#NOFILL_INC

Increment for contour line (if fill_flag = ‘N’)

#PANEL_TITLE

Title of the panel (enclosed in single or double quotes).

#SCALE_TITLE

Title of the colorbar. Usually this is the units of the contour values plotted (enclosed in single or double quotes). For example, if the user is plotting reflectivity, then the title on the color might be ‘dBZ’.

#TITLE

Title of the entire plot (enclosed in single or double quotes). This title is located in the center top of the page.is produced

#LEVEL_FLAG

Flag to indicate whether the level you are choosing is by distance or level.
D = Distance
L = Level

#DISTANCE OR LEVEL

If level_flag = ‘D’, then enter the distance of the level to plot. If level_flag = ‘L’, then enter the level to plot.

#CATALOG_NAME

Name of the catalog file to generate a flight track. This file is produced using the translator. If ‘NONE’ is entered, then the flight track will not be plotted.

#PLANE_INC

Increment of the scans to plot a symbol on the flight track. For example, a value of 6 would plot a symbol every 6th scan.

3.0 OPTIONAL INPUT FILE PARAMETERS

The following fields of the input are optional. This part of the input file is divided into sections such as coordinate parameters, grid parameters..etc. To accept the default values for the entire section, then type a ‘Y’ under the line #DEFAULT [SECTION] PARAMETERS?. To accept only *some* of the default values in a section, then type a ‘N’ under the line #DEFAULT [SECTION] PARAMETERS? and then put a symbol (#, !, %..etc) in front of the parameter identifier to accept the default value. For example, to accept the default horizontal coordinates, but specify the vertical coordinates, the input file would look as follows:

```
#*************************************
# COORDINATE PARAMETERS
#*************************************
#DEFAULT COORDINATE PARAMETERS?
'N'
!#HORIZ_BEG      <-- The '!' in front of this line
                     means to accept the default value.
20.0
```

```
!#HORIZ_END
35.0
#VERT_BEG
10.0
#VERT_END
20.0
```

Because the parameter identifier #HORIZ_BEG and #HORIZ_END is preceded by a '!', then the default values are used, but the vertical values (#VERT_BEG & #VERT_END) are 10.0 and 20.0, respectively.

Descriptions of the parameter identifiers are listed below.

#HORIZ_BEG
Begin horizontal coordinate (km).
#HORIZ_END
End horizontal coordinate (km).
#VERT_BEG
Begin vertical coordinate (km).
#VERT_END
End horizontal coordinate (km).
#PRI_GRID1
Primary horizontal tick mark interval (km).
#SEC_GRID1
Secondary horizontal tick mark interval (km).
#PRI_GRID2
Primary vertical tick mark interval (km).
#SEC_GRID2
Secondary vertical tick mark interval (km).
#SCALE_FLAG
Flag to indicate whether or not to draw the color/gray scale.
ON = draw the scale
OFF = don't draw the scale
#SCALE_POS
Position of the color/gray scale.
HOR = Horizontal
VER
= Vertical
#SCALE_W
Width of the color/gray scale (cm).
#SCALE_H
Height of the color/gray scale (cm).
#ZERO_LINEWIDTH
Thickness of the zero contour line represented in points where 72 points = 1 inch.
#LINEWIDTH
Thickness of contour lines (except zero line) represented in points where 72 points = 1 inch.
#DASH_ON

Length of the solid portion of a dashed line (in points).
#DASH_OFF
Length of the space between dashes (in points).
#DASH_BEG
Distance into the dash pattern at which the dash is started (in points).
#PTSKIP
Number of grid spaces between vectors (-1 = Interleaved).
#VEC_SPEED
Reference speed for vectors.
#VEC_LENGTH
Reference distance for vectors.
#ARROW_H
Height of vector head represented as a percentage of the vector length. If this number is negative, then the absolute value is used and it represents the actual height of the vector head in points.
#ARROW_HH
Width of vector head represented as a percentage of the vector length. If this number is negative, then the absolute value is used and it represents the actual width of the vector head in points.
#ARROW_HT
Width of vector stem represented as a percentage of the vector length. If this number is negative, then the absolute value is used and it represents the actual width of the vector stem in points.
#FONT_TITLE
Size of the title font.
#FONT_NUM
Size of the number font.
#FONT_LABEL
Size of the label font.
#FONT_NAME
Name of the font.
#STRETCH_HORIZ
Horizontal stretch factor.
#STRETCH_VERT
Vertical stretch factor.
#SP_TEXT
Space between the boundary of the panel and the text (in cm).
#SP_SCALE
Space between the boundary of the panel and the color/gray scale (in cm).
#SP_ROW
Vertical space between panels (in cm).
#SP_COL
Horizontal space between panels (in cm).
#SP_TITLE
Space between the title and the upper panel(s) (in cm).
#TITLE_FLAG

Flag to indicate whether or not to display bottom label.
ON = display bottom label
OFF = do not display bottom label

#GRID_FLAG

Flag to indicate what type of grid to display.
BOTH = display both primary and secondary tick marks
PRIM = display only primary tick marks

#PANEL_W

Width of the grid (in cm).

#PANEL_H

Height of the grid (in cm)

#COLOR_TABLE

Name of the color table file. See Appendix D for the format of a color table file. The default color table is based on the carbone42 color table.

#PAGE_W

Width of the page (in cm).

#PAGE_H

Height of the page (in cm).

#L_MARG

Left margin (in cm).

#R_MARG

Right margin (in cm).

#T_MARG

Top margin (in cm).

#B_MARG

Bottom margin (in cm).

Please refer to Appendix C for a table of default values.

4.0 ADDING ROUTINES TO READ OR MANIPULATE THE DATA

The fortran code can be modified to read data formats not supported by grid2ps and to manipulate the data. In the file 'UserSub.f', there are two subroutines: ReadUserDat and MassageData. Within ReadUserDat, the header and data arrays must be filled. Please refer to Appendix A for the specifications for the header array ('id'). The data array is a global three dimensional real array with the dimensions fld_data(x, y, fld) where x is the horizontal grid point and y is the vertical grid point. The third dimension represents the field where:

1 = horizontal wind field (ie: U)
2 = vertical wind field (ie: V)
3 = first contour field (ie: DBZ)
4 = second contour field (ie: W)

For example:

fld_data(10, 5, 1) = 10.0 would mean at x = 10, y = 5, the horizontal wind component is 10.0 m/s.

fld_data(10, 5, 2) = 20.0 would mean at x = 10, y = 5, the horizontal wind component is 20.0 m/s fld_data(10,5, 3) = 35 would mean at x = 10, y = 5, the first contour field is 35 dBZ (if the first contour field was reflectivity). In a similar fashion, the data can be manipulated in the subroutine MassageData.

Both subroutines are called from the main program (main.f). ReadUserDat will only be called if the input_flag is ***not*** 'C' (for CEDRIC format) or 'W' for (Wen-Chau format).

5.0 EDITING THE POSTSCRIPT CODE

If the appearance of the plot is unsatisfactory, the user can either change the input file(s) and run grid2ps again or hand edit the PostScript code and view the results immediately. In addition, graphics such as boxes, circles, arrows and text may be overlaid on the figure by calling pre-defined subroutines in the PostScript code. There are three portions of the PostScript code that affect the appearance of the plot; global variables, local variables and user defined subroutines.

Structure of the PostScript code

The basic structure of the code is as follows:

```
%!PS
%********************
% SHORTEN COMMANDS
%********************
.....code.....
%********************
% CONSTANTS
%********************
.....code.....
%********************
% GLOBAL VARIABLES
%********************
...code..user can change these variables
%********************
% GLOBAL SUBROUTINES
%********************
....code.....
%********************
% DICTIONARIES
%********************
....code.....
%********************
% MAIN PROGRAM
%********************
....code.....
 %******************
 % PANEL 1
 %******************
 ...code..user can change these variables
     %CONTOURS
      .....code to draw contour lines.....
     %VECTORS
     .....code to draw vectors.....
     % FLIGHT TRACK
```

```
     .....code to draw flight track....
    % USER FUNCTIONS
     .....code to overlay graphics..user can change this code.....
 %******************
 % PANEL 2
 %******************
  ...code..user can change these variables
    %CONTOURS
     .....code to draw contour lines.....
    %VECTORS
     .....code to draw vectors.....
    % FLIGHT TRACK
     .....code to draw flight track....
    % USER FUNCTIONS
     .....code to overlay graphics..user can change this code.....
.
.
.
%******************
% PANEL N
%******************
...code..user can change these variables
    %CONTOURS
     .....code to draw contour lines.....
    %VECTORS
     .....code to draw vectors.....
    % FLIGHT TRACK
     .....code to draw flight track....
    % USER FUNCTIONS
     .....code to overlay graphics..user can change this code.....
```

Changing the global variables

The global variables are those variables which apply to the entire plot such as page size, margins, ..etc. Listed below is the code segment that defines the global variables:

```
%********************
% GLOBAL VARIABLES
%********************
   15/title_font ed
  10/label_font ed
   8/num_font ed
   1/template ed
   0.20/pct_hh ed
   0.18/pct_hw ed
   0.05/pct_vw ed
   0.25 cm/sp_text ed
   1.00 cm/sp_title ed
   0.50 cm/sp_scale ed
  10.00 cm/scale_w ed
   0.75 cm/scale_h ed
 -10/scale_beg ed
  50/scale_end ed
  10.00/scale_inc ed
```

```
  2.00 cm/l_marg ed
  2.00 cm/r_marg ed
  1.00 cm/t_marg ed
  1.00 cm/b_marg ed
 21.59 cm/page_w ed
 27.94 cm/page_h ed
  1.00 cm/sp_row ed
  1.00 cm/sp_col ed
(dBZ)/scale_title ed
(FASTEX TEST DATA)/plot_title ed
(VECTOR: U V CONTOUR: MAXDB)/field_label ed
(LATITUDE:   52 DEG   0 MIN   0 SEC
 LONGITUDE:  -16 DEG   0 MIN   0 SEC)/LatLon_label ed
(ORIGIN: ( 0.00, 0.00))/ori_label ed
(EVERY  3 POINTS PLOTTED)/points_label ed
(PLOT AIRCRAFT EVERY )/plane_label1 ed
( SCAN)/plane_label2 ed
HOR/scale_pos ed
ON/title_flag ed
ON/scale_flag ed
```

To change any of these variables, simply change the value in front of the '/' and redisplay the plot in ghostview. For example, to change the left margin to 3.0 cm, change:
'2.00 cm/l_marg ed' to '3.00 cm/l_marg ed'.

Changing the local variables

The local variables are those variables which apply to each individual panel such as vec_dist, vec_speed, panel_title..etc. There is a set of local variables for each panel displayed. Listed below is the code segment that defines the local variables:

```
%******************
 % PANEL 1
 %******************
  1/panel ed
  3/plot_type ed
 10/pri_tick ed
  1/sec_tick ed
 (X(KM))/h_label ed
 (Y(KM))/v_label ed
 (97/ 1/22)/date ed
 ( 9:52:41-10:24:59)/time ed
 (Z= 25.80)/level ed
 (test panel 1)/panel_title ed
    5.00/vec_dist ed
    5.00/vec_speed ed
    1.00/h_stretch ed
    1.00/v_stretch ed
  -70.00/h_beg ed
  170.00/h_end ed
 -100.00/v_beg ed
  100.00/v_end ed
 -999.00/box_w ed
```

```
-999.00/box_h ed
 1/plane_inc ed
CART/grid ed
PRIM/grid_flag ed
```

To change any of these variables, simply change the value in front of the '/ ' and redisplay the plot in ghostview. For example, to change the vertical stretch factor to 2.0, change: '1.00/h_stretch ed' to '2.00/h_stretch ed'.

Overlaying additional graphics

There are two ways to overlay graphics onto a figure generated in grid2ps; converting to an epsi file and importing into existing software or adding the graphics in the grid2ps output PostScript file. An epsi file allows you to view the graphic once it is imported into another application. There is a ghostscript utility called ps2epsi that will convert the file into an epsi file. The epsi file can be imported into another application and additional graphics can be overlaid.

There are also pre-defined subroutines in the grid2ps PostScript file that can easily be called. Each panel is structured as follows:

```
PanelDict begin
   %******************
   % PANEL 1
   %******************
    1/panel ed
    3/plot_type ed
   10/pri_tick ed
    1/sec_tick ed
   (X(KM))/h_label ed
   (Y(KM))/v_label ed
   (97/ 1/22)/date ed
                .
                .
                .
   (local variables)

    some miscelleneous code

   % CONTOURS
    [contour code]

   % VECTORS
   [vector code]

   % FLIGHT TRACK
   [flight track code]

   % USER FUNCTIONS
   [user function code...see description below]

end % PanelDict
```

The following code segment is an example where the code can be altered to include a box, circle, arrows or annotation.

```
% USER FUNCTIONS
gs
   Trans2Origin
   ClipIt
   MyFunction begin
   % User called functions
   % Relative to origin of grid
      %
      % Function to draw a square
      % Syntax: xbeg ybeg box_w box_h DrawSquare
      %   xbeg, ybeg: begin coordinates for box
      %   box_w, box_h:  width & height of box
      %   40 hkm -20 vkm 1 inch 2 inch DrawSquare

      % Function to draw a circle
      % Syntax: xori yori radius DrawCircle
      %   xori, yori: center of circle
      %   radius = radius of circle
      %    -40 hkm -60 vkm .50 inch DrawCircle

      % Function to draw an arrow
      % Syntax: tailx taily angle length DrawArrow
      %   tailx, taily: position of tail of arrow
      %   angle: direction of arrow
      %   length: length of arrow
      %   120 hkm 60 vkm 25 .25 inch DrawArrow

      % Function to draw text
      % Syntax: xbeg ybeg (text) justify fontsize Annotate
      %   xbeg, ybeg: position of text
      %   text: text [must be enclosed in ()]
      %   justify: 1 = left; 2 = right; 3 = center
      %   fontsize: size of font
      %   20 hkm 80 vkm (some text) 3 10 Annotate
   end % MyFunction
gr
```

Changing the color table

There are two methods to change the color/gray scale in the output from grid2ps: using a X Windows based utility called XACTE (X Answer to Color Table Editors) or hand editing the PostScript code.

Using XACTE

XACTE was developed by Forrest Cook (NCAR/RDP) and is a windows based utility that allows the user to visually adjust the color. The colors are adjusted using scroll bars and the output can be saved to a color table file and specified in the input file of grid2ps. The source code for XACTE is in the directory '/grid2ps1.0/xacte'.   Follow the directions listed in the README file. Please refer to Appendix E for a brief description of XACTE.

Changing the color table by hand

It is relatively easy to change the color/gray scale in the output from grid2ps. The following code segment is where the color table is defined:

```
/DefineColorTable {

   % COLOR TABLE..EACH ELEMENT IN CT IS THE RGB VALUE
   /ct 256 array def
   /o {ct exch get aload pop setrgbcolor} def
   ct  1 [ 1.00 1.00 1.00 ] put
   ct  2 [ .469 .020 .640 ] put
   ct  3 [ .445 .559 .996 ] put
   ct  4 [ .176 .633 .176 ] put
   ct  5 [ .938 .906 .703 ] put
   ct  6 [ .816 .578 .148 ] put
   ct  7 [ .629 .312 .375 ] put
   ct 44 [ 0.00 0.00 0.00 ] put

} def
```

The colors in PostScript reflect a combination of Red, Green & Blue values (RGB). The RGB values are stored in the array ct where the first value represents red, the second value represents green and the third value represents blue. The first and last element of the array ct represent white and black and should not be changed. The remaining elements can be changed. For example, the following code represents the color red:
'ct 2 [1.00 0.00 0.00] put'. If this were change to'ct 2 [0.00 1.00 0.00] put', then the color would change to green. This allows the user to change the color table and immediately view the results in ghostscipt or ghostview. For a gray scale, all three values are the same. For example, the code for a shade of gray might be 'ct 2 [.25 .25 .25] put'. Since the true colors are device dependent, there is a file called '/grid2ps1.0/examples/color.table.ps' that contains a grid of 1452 different color combinations that can be printed.

6.0 IMPORTING GRID2PS OUTPUT INTO EXISTING SOFTWARE

The output from grid2ps is a raw PostScript file. To view the output, the user may use ghostview or ghostscript or it can be sent to a printer. To convert the file to an encapsulated PostScript (eps) file, run the script '/grid2ps1.0/perl/ps2eps.pl'. The arguments to ps2eps.pl are as follows:

-i: Name of postscript file generated in grid2ps
-o: Name of output eps file
-b: Bounding box of graphic. [llx lly urx ury].
These are 4 integers representing the size of graphic in points (72 points = 1 inch).
llx = lower left horizontal coordinate
lly = lower left vertical coordinate
urx = upper right horizontal coordinate
ury = upper right vertical coordinate

Example: *ps2eps.pl -i test.ps -o test.eps -b '72 72 288 432'*

The lower left coordinates on the page are 72, 72 (1 inch, 1 inch) and the upper right coordinates are 288, 432 (4 inch, 6 inch). This means that the lower left corner of the graphic at 1 inch, 1 inch and the size of the graphic is 3 inches X 5 inches.

The bounding box can be obtained by viewing the PostScript file using ghostview and putting the cursor on the desired bounding box. In the upper left corner of the ghostview window, the coordinates will be displayed. The bounding box must be enclosed in quotes when running the script ps2eps.pl. Once the file is converted into an eps file, it can be imported into existing word processing applications such as FrameMaker or Word.

APPENDIX A

The header array (‘id’) is a 510 element array similar to the CEDRIC header array. Listed below are the array elements that must have values assigned in ReadUserDat:

(The array id is the header information from CEDRIC)

*id(16)=SCAN MODE
*id(17)=SCAN MODE
id(33)=COORDINATE ORIGIN LATITUDE DEGREES
id(34)=COORDINATE ORIGIN LATITUDE MINUTES
id(35)=COORDINATE ORIGIN LATITUDE SECONDS * 100
id(36)=COORDINATE ORIGIN LONGITUDE DEGREES
id(37)=COORDINATE ORIGIN LONGITUDE MINUTES
id(38)=COORDINATE ORIGIN LONGITUDE SECONDS * 100
*id(40)=DEGREES CLOCKWISE FROM NORTH TO X-AXIS * 64
id(41)=X COORDINATE OF HORIZONTAL AXIS ORIGIN * 100
id(42)=Y COORDINATE OF HORIZONTAL AXIS ORIGIN * 100
id(68) = GENERAL SCALING FACTOR (SF = 100)
id(69) = ANGLE SCALING FACTOR (CF = 64)
id(116)=YEAR BEGIN
id(117)=MONTH BEGIN
id(118)=DAY BEGIN
id(119)=HOUR BEGIN
id(120)=MINUTE BEGIN
id(121)=SECOND BEGIN
id(122)=YEAR END
id(123)=MONTH END
id(124)=DAY END
id(125)=HOUR END
id(126)=MINUTE END
id(127)=SECOND END
id(160)=MINIMUM X (KM) * 100
id(161)=MAXIMUM X (KM) * 100
id(162)=NUMBER OF X’S
id(163)=SPACING IN X (M)
id(165)=MINIMUM Y (KM) * 100
id(166)=MAXIMUM Y (KM) * 100
id(167)=NUMBER OF Y’S
id(168)=SPACING IN Y (M)
id(170)=MINIMUM Z (M)
id(171)=MAXIMUM Z (M)
id(172)=NUMBER OF Z’S
id(173)=SPACING IN Z (M)

* Only used if plotting a flight track

APPENDIX B

```
#***************************************************
# INPUT FILE FOR GRID2PS (MANDATORY PARAMETERS)
#***************************************************
#OUTPUT_NAME  !name of the output file
'970122.ps'
#INPUT_FLAG (C or T)  !C=Cedric; T=Text;
C
#DATA_FILE (if input_flag='T')  !name of the text file if input_flag=T
'/scr/port/snorman/mud/fastex/9701220950.mud'
#COLOR (Y or N)  !color plot? Y=yes;N=no
Y
#CONTOUR_TYPE (0=gray at 0;1=gray at midpoint of range;2=no gray)
2
#FIX_AXIS (X,Y,Z)  !fixed_axis
Z
#FIELD1 (VEC1)  !horizontal velocity field
U
#FIELD2 (VEC2)  !vertical velocity field
V
#FIELD3 (CONT1) !contour field 1
MAXDB
#FIELD4 (CONT2) !contour field 2
W
#PLOT_TYPE  !1=vec;2=cont;3=vec&cont;4=cont&cont;5=vec,cont&cont
3
#FILL_FLAG  !Y=contour field is filled;N=contour lines
Y
#FILL_BEG   !if fill_flag=Y, begin contour value
-10.0
#FILL_END   !if fill_flag=Y, end contour value
50.0
#FILL_INC   !if fill_flag=Y, contour increment
10.0
#NOFILL_BEG !if fill_flag=N, begin contour value
-10.0
#NOFILL_END !if fill_flag=N, end contour value
50.0
#NOFILL_INC !if fill_flag=N, contour increment
10.0
#PANEL_TITLE !title of the panel
'panel 1'
#SCALE_TITLE !title for the color bar (usually units)
dBZ
#TITLE !title of the plot (top of page)
'FASTEX DATA'
#LEVEL_FLAG  !D=distance;L=level
D
#DISTANCE OR LEVEL !distance (in km) or level, depending upon level_flag
4.0
#CATALOG_NAME  !name of catalog file,enter 'NONE' if no catalog file
'/scr/port/snorman/FASTEX/catalog/22Jan97/cat.970122095241.TA-ELDR.0'
```

```
#PLANE_INC  !plot the planes every plane_inc scans
10
#*************************************************************
# THE FOLLOWING PARAMETERS HAVE DEFAULT VALUES ASSIGNED TO THEM
# TO USE THE DEFAULT VALUE, COMMENT OUT THE VALUE WITH A #
#*************************
# COORDINATE PARAMETERS
#*************************
#DEFAULT COORDINATE PARAMETERS?
Y
#HORIZ_BEG  !begin horizontal coordinate
10.
#HORIZ_END  !end horizontal coordinate
80.
#VERT_BEG   !begin vertical coordinate
-30.
#VERT_END   !end vertical coordinate
40.00
#*************************
# GRID PARAMETERS
#*************************
#DEFAULT GRID PARAMETERS?
Y
#PRI_GRID  !horizontal primary tick mark plotted every pri_grid1 km
20
#SEC_GRID  !horizontal secondary tick mark plotted every pri_grid1 km
5
#*************************
# SCALE PARAMETERS
#*************************
#DEFAULT SCALE PARAMETERS?
Y
#SCALE_FLAG*  !ON=draw the scale or OFF=don't draw the scale
OFF
#SCALE_POS*  !HOR=horizontal;VER=vertical
VER
#SCALE_W*  !scale width
13.00
#SCALE_H*  !scale height
.50
#*************************
# LINE PARAMETERS
#*************************
#DEFAULT LINE PARAMETERS?
Y
#ZERO_LINEWIDTH*  !width (in points) of line representing zero
1.0
#LINEWIDTH*  !line width (in points)
.7
#DASH_ON*  !number of units to draw
5
#DASH_OFF* !number of units to not draw
3
#DASH_BEG* !dash offset
```

```
1
#************************
# VECTOR PARAMETERS
#************************
# NOTE: IF A NEGATIVE # IS ENTERED
# FOR ARROW_H, ARROW_HH, OR ARROW_HT,
# THEN THE VECTOR WILL *NOT* BE SCALED
# BY THE LENGTH (THE ACTUAL PARAMETERS
# WILL BE USED)
#************************
#DEFAULT VECTOR PARAMETERS?
N
#PTSKIP  !plot every ptskip vectors
3
#VEC_SPEED  !reference vector speed (in m/s)
5.0
#VEC_LENGTH !reference vector length (in km)
5.0
#ARROW_H  !height of head of vector (% of vector length)
.20
#ARROW_HH !width of head of vector (% of vector length)
.18
#ARROW_HT !width of vector (% of vector length)
.05
#************************
# FONT PARAMETERS
#************************
#DEFAULT FONT PARAMETERS?
Y
#FONT_TITLE* !font size of the title
20
#FONT_NUM*  !font size of the numbers
8
#FONT_LABEL* !font size of the labels
12
#FONT_NAME*
'Times-Roman'
#************************
# PLOT PARAMETERS
#************************
#DEFAULT PLOT PARAMETERS?
N
#STRETCH_HORIZ  !horizontal stretch factors
1.0
#STRETCH_VERT  !vertical stretch factors
5.0
#SP_TEXT (CM)*  !space between horizontal text & grid
.25
#SP_SCALE (CM)*  !space between color/gray bar & grid
1.25
#SP_ROW (CM)*  !space between rows
.8
#SP_COL (CM)*  !space between columns
.5
```

```
#SP_TITLE (CM)* !space between the plot & the title
.75
#TITLE_FLAG !ON or OFF
OFF
#GRID_FLAG  !PRIM=display primary tick marks only;BOTH=display prim & second-
ary
PRIM
#PANEL_W (CM)  !width of panel in cm
2.
#PANEL_H (CM)  !height of panel in cm
3.
#COLOR_TABLE  !name of color table file
test.ct
#************************
# PAGE PARAMETERS
#************************
#DEFAULT PAGE PARAMETERS?
N
#PAGE_W (CM)*  !width of page in cm
21.59
#PAGE_H (CM)*  !height of page in cm
27.94
#L_MARG (CM)*  !left margin in cm
2.00
#R_MARG (CM)*  !right margin in cm
2.00
#T_MARG (CM)*  !top margin in cm
1.50
#B_MARG (CM)*  !bottom margin in cm
1.50
************************************************
#EOF
```

APPENDIX C

.

TABLE OF DEFAULT VALUES

PARAMETER	VALUE	UNIT	PARAMETER	VALUE	UNIT
pri_grid1	10	km	**font_label**	10	
sec_grid1	1	km	**font_name**	Helvetica	
pri_grid2	10	km	**stretch_horiz**	1.0	
sec_grid2	1	km	**stretch_vert**	1.0	
scale_flag	‘ON’		**sp_text**	.5	cm
scale_pos	‘HOR’		**sp_scale**	.5	cm
scale_w	10.0	cm	**sp_row**	1.0	cm
scale_h	.75	cm	**sp_col**	1.0	cm
zero_linewidth	1.5	points	**sp_title**	1.0	cm
linewidth	.50	points	**title_flag**	‘ON’	
dash_on	6	points	**grid_flag**	‘BOTH’	
dash_off	3	points	**panel_w**	N/A	dependent upon template
dash_beg	0	points	**panel_h**	N/A	dependent upon template
pt_skip	-1		**color_table**	N/A	dependent upon # of contours
vec_speed	20.0	m/s	**page_w**	21.59	cm
vec_length	5.0	km	**page_h**	27.94	cm
arrow_h	.20		**l_marg**	1.0	cm
arrow_hh	.18		**r_marg**	1.0	cm
arrow_ht	.05		**t_marg**	1.0	cm
font_title	15		**b_marg**	1.0	cm
font_num	8				

APPENDIX D

FORMAT FOR COLOR TABLE

The following is an example of the format for the color table. The first line must begin with ‘colortable’ followed by any name. The last line must be ‘endtable’. The interior lines are the RGB (red, green, blue) values respectively.

```
colortable test
     0.469  0.020  0.640
     0.403  0.227  0.559
     0.164  0.055  0.582
     0.227  0.055  0.672
     0.289  0.055  0.766
     0.352  0.141  0.898
     0.414  0.375  0.996
endtable
```

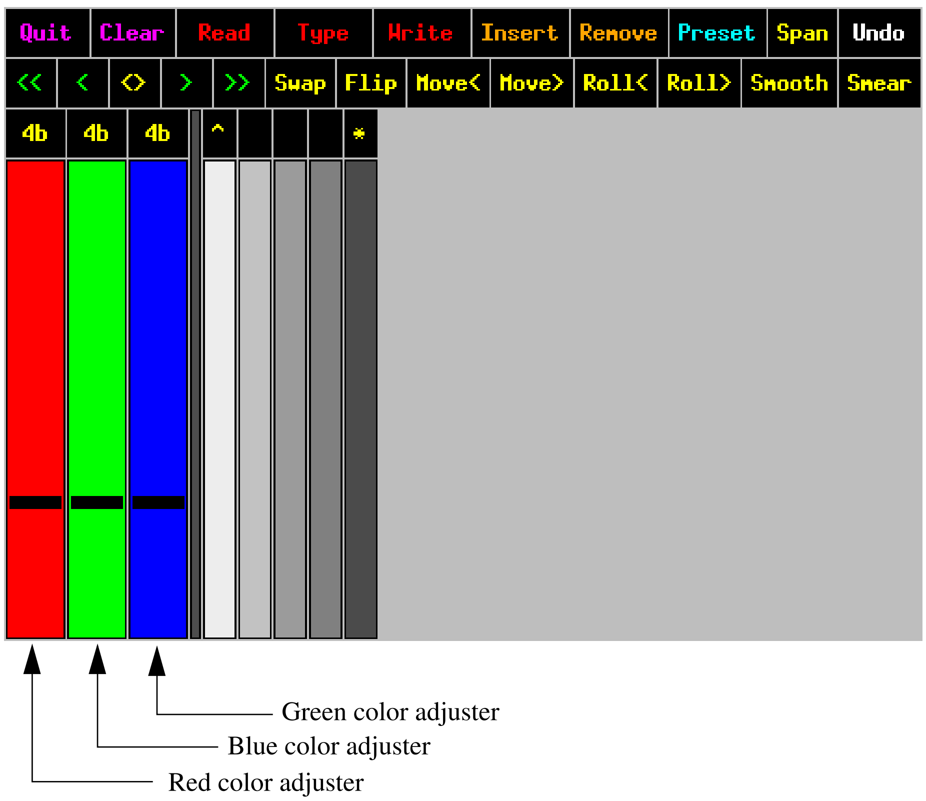
Note: The number of colors must match the number of contour increments listed in the input file.

APPENDIX E

To run XACTE, type
prompt> xacte #colors

For example, to create a color table with 6 colors, type:
prompt> xacte 5

The following window will appear:



Select the color bar by clicking on a color selector (an * will appear to indicate which color is being changed) and adjust the color by dragging the scroll bar for the color adjusters. You will notice the bar below the color selector changing color as you adjust the RGB values. When you are satisfied with the color table, continue to click the button labeled 'Type' until the work 'ZEB' appears. Next, click on the button labeled 'Write' to write the color table to a file in the proper format for grid2ps.