8 GRAPH

8.1 Purpose

The GRAPH program generates simple diagnostics and plots for some standard meteorological variables. The GRAPH code will process multiple times and vertical levels, computing the same diagnostics for each time and level. The GRAPH code will provide simple vertical interpolation capability, cross-section figures, and skew-T plots. The GRAPH code is written to be used as a batch processor, so that all graphical choices are made from tables. The GRAPH code can accept the version 1 and 2 format data from TERRAIN, REGRID, RAWINS and little_r, INTERP (front and back) and MM5. Boundary condition data are not formatted for use with GRAPH. The GRAPH code does not produce any standard output for use by a subsequent program. The GRAPH program can overlay two plots.

The GRAPH code in MM5 V2 system is built on NCAR Graphics library (which is a proprietary software: http://ngwww.ucar.edu). It can be run on both Crays and workstations where NCAR Graphics is installed. When working on a Cray, a user can run GRAPH in batch or interactive mode. Examples of the interactive GRAPH use are shown in the section 8.7.

8.2 Typical GRAPH Jobs

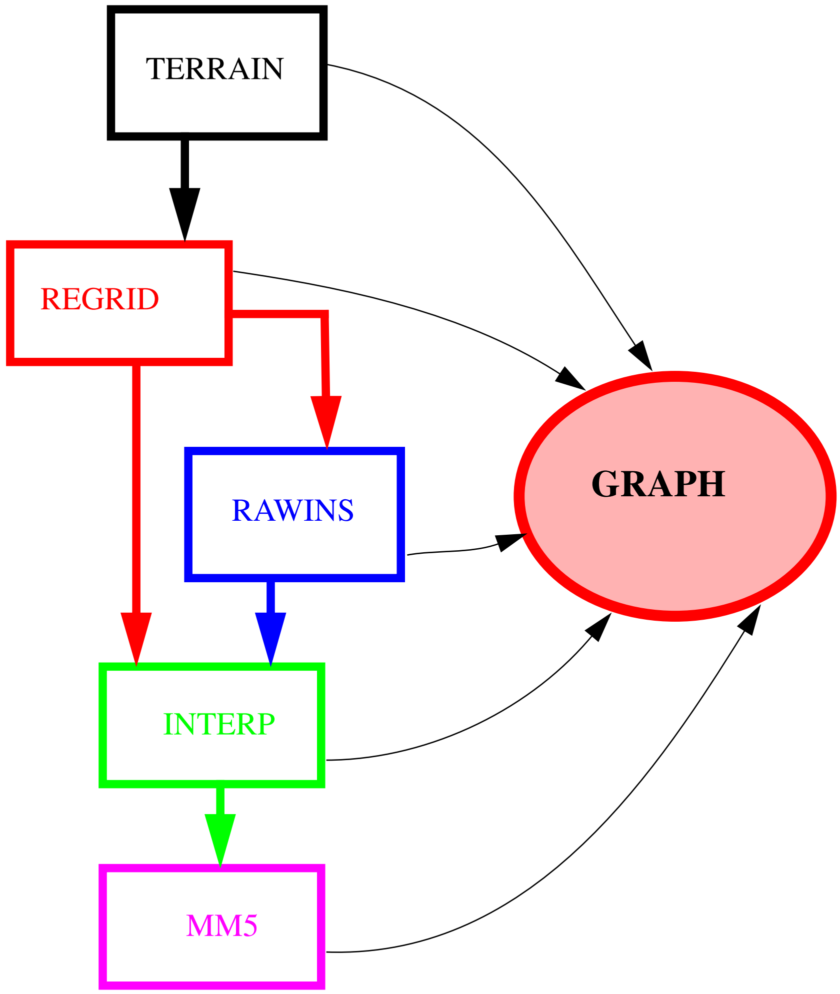


Fig. 8.1 Schematic diagram showing GRAPH accepting data from outputs of MM5 modeling system.

8.3 Plotting Table File: g_plots.tbl

This table is used to define times, levels and fields to be processed and plotted by Graph. An example is shown below:

```
TIME LEVELS (MDATE FORMAT YYMMDDHH): FROM 93031300 TO 93031400 BY 6 (A)
PRESSURE LEVEL MANDATORY: FROM SFC TO PTOP (B)
PRESSURE LEVEL NON STANDARD: FROM SFC TO PTOP BY 3 (C)
SIGMA LEVEL: FROM 23 TO KMAX BY 5 (D)
THETA LEVEL: NONE (E)
TITLE: MM5 Tutorial (F)
-----------------------------------------------------------------------------
PLOT | FIELD | UNITS  |  CONTOUR  | SMOOTH || OVERLAY  | UNITS  |CONTOUR |SMOOTH
T/F  |       |        |  INTERVAL | PASSES ||  FIELD   |        |INTERVAL|PASSES
-----|-------|--------|-----------|--------||----------|--------|--------|------
 T   |  TER  |  m     |   100     |   0    ||          |        |        |
 T   |  WIND |  m/s   |   5       |   0    ||   BARB   | m/s    |   2    |  0
TP500| HEIGHT|  m     |  30       |   0    ||   VOR    |10**5/s|   0    |  0
TI305| PV    |  PVU   |   1       |   0    ||   P      |  mb    |  20    |  0
 X   | 5     |  5     |  23       |   8    ||  PSLV    |  mb    |   2       |  0
 X   | THETA |  K     |   3       |   0    || CXW      | m/s    |  10    |  0
 X   | 5     | 18     |  23       |   5    ||  PSLV    |  mb    |   2    |  0
 X   |THETA  |  K     |   3       |   0    || CXW      | m/s    |  10    |  0
 T   |SKEWTLL|72469 DEN   DENVER, CO    |39.75  |-104.87  || | | |
 T   |SKEWTXY|STATION IN (X,Y)       | 19 | 30 || | | |
-----------------------------------------------------------------------------
```

Description of Table Header Rows:

(A)	*TIME*	with beginning and ending times given in YYMMDDHH format, and time increment given in hours defined by the number after *BY.* '*BY 0*' means to plot every output times.
(B)	*MANDATORY*	used for pressure level dataset (such as from DATAGRID and Rawins). Will plot every mandatory level from the maximum and minimum level requrested. ALL and NONE may also used to replace the entire string after the colon.
(C)	*NON STANDARD*	used for pressure level dataset. Will plot every mandatory level from the maximum and minimum level requrested. Optional use of *BY n* will make plots at every n levels. ALL and NONE may also be used to replace the entire string after the colon.
(D)	*SIGMA*	used for σ-level data. Will plot the levels specified by indexes (K=1 is at the top of the model). An increment is required, and defined by the number after *BY.* Can also use ALL or NONE.
(E)	*THETA*	Not used.
(F)	*TITLE*	except for a colon (:), any alpha-numeric character can be used to make a simple 1 line, 80 character title

Description of Table Columns:

PLOT T/F	True or False to plot this field. Removing the line from the table has the same effect as F. If the user requests a cross section plot, the letter is X. If the user requests a plot on pressure level, the first 2 characters are TP, followed by the pressure value (TP500 is 500 mb level); if the user requests a plot on isentropic surface, the first 2 characters are TI, followed by the potential temperature value (TI330 is 330 K level). The last two options only work with σ data.
FIELD	a field name to be plotted. See complete list in Tables 8.1, 8.2 and 8.3. If the field is a skew-T, the interpretation of the following columns is changed (see the explanation below).
UNITS	units used on a plot. For some fields, there are different standard units available. If you don't know the unit, use '?'.
CONTOUR INTERVAL	real or integer values used to contour a plot. If you don't know what contour interval to use, use '0'. For a vector field (e.g. BARB), this value specifies the grid interval. For streamline field (VESL), this value specifies how sparse or dense streamlines are drawn.
SMOOTH PASSES	number of passes of the smoother-desmoother used for each horizontal plot.
OVERLAY FIELD	a field name for the overlay plot. May be left blank.

* If the plot is a skew-T (SKEWTLL or SKEWTXY), the UNITS column is used to define location name, and lat/long or X/Y appear in the following two columns.

* For a cross-section plot, the location is defined by the 4 numbers in the columns following 'X', and they are in the order of X1, Y1, X2, and Y2.

8.4 Default Option Settings File: g_defaults.nml

This is a namelist file and it is optional. If this file exists in the current working directory when the Graph program starts executing, the file's contents replace the previously set defaults in the Fortran code. Since this is a namelist structured file, lines may be removed. Comments after ';' are not allowed on most platforms, but are shown here for easy reference only.

```
&JOEDEF ; defaults for graph
  ;   ISTART=1,             ; sub-domain plot beginning I location
  ;   JSTART=1,             ; sub-domain plot beginning J location
  ;   IEND=37,              ; sub-domain plot ending I location
  ;   JEND=49,              ; sub-domain plot ending J location
      LW1=2000,             ; line width, 1000 is thinnest
      LW2=2000,              ; line width for overlay plot
      DASH1=-682,            ; dash pattern, standard NCAR GKS method
      DASH2=-682,           ; 4092, 3640, 2730, -682
```

```
      COLOR1=12,  COLOR4=12,
      COLOR2=9,   COLOR5=9,
      COLOR3=8,   COLOR6=8,
      HDRINFO=F,            ; true=print header and stop
      LOGP=0,               ; cross section: 0=linear in p; 1=linear in ln p
      XPTOP=200.,           ; top of cross section plots (mb)
      LABLINE=1,            ; 0: no contour line labels
      LABMESG=0,            ; 1: no message below conrec plot
      NOZERO=0,             ; 0: allow zero line; 1:no min/max zero line;
                             ; 2: no zeor whatsoever
      IHIRES=0,      ; 1: use high resolution US county line/China coastline
&END ;
```

Description of variables in the namelist:

ISTART	integer	for a subdomain plot, this is the I-direction starting point
JSTART	integer	for a subdomain plot, this is the J-direction starting point
IEND	integer	for a subdomain plot, this is the I-direction ending point
JEND	integer	for a subdomain plot, this is the J-direction ending point
LW1	integer	line width for the first plot; 1000 is the thinest
LW2	integer	line width for the overlay plot
DASH1	integer	dash pattern for the first plot; standard NCAR GKS method. A '-' before a number means contour of positive values is solid, negative values is dashed 682: shorter-dashed line 2730: short-dashed line 3640: medium-dashed line 4092: long-dashed line
DASH2	integer	dash pattern for the overlay plot
COLOR1	integer	color index for the first contour plot, labeled lines
COLOR2	integer	color index for the overlay plot, labeled lines
COLOR3	integer	color index for a dot-point plot, labeled lines
COLOR4	integer	color index for the first contour plot, unlabeled lines
COLOR5	integer	color index for the overlay plot, unlabeled lines
COLOR6	integer	color index for a dot-point plot, unlabeled lines
HDRINFO	logical	T: will only print record header
LOGP	integer	for cross section plots: whether the vertical coordinate is plotted in linear p (LOGP=0), or log p (LOGP=1)
XPTOP	real	top of a cross section plot (in mb)
LABLINE	integer	=0: no contour line labels
LABMESG	integer	=1: no message below conrec plot
NOZERO	integer	=1: no min/max zero line; =2: no zero line whatsoever
IHIRES	integer	=1: use high resolution US county/China coastline

8.5 Map Options File: g_map.tbl

This table is used to modify map background specifics for a Graph plot.

```
-------------------------------------------------------------------------------
                                MAP DETAILS
LL | DASH | INT | LB | LSZ | LQL | P | TTL | TSZ | TQL | OUT | DOT | LW    | SP
-------------------------------------------------------------------------------
A  | PB   | D   | M  | 12  | 00  | Y | Y   |  8  | 00  | PS  | N   | D     |
-------------------------------------------------------------------------------
                                 MAP COLORS
LL LINES | LABELS | TITLE | STATES     | COUNTRIES    | CONTINENTS | PERIMETER
-------------------------------------------------------------------------------
1        | 1      |  1    | 1          | 1            | 1          |  1
-------------------------------------------------------------------------------
```

Description of variables in g_plots.tbl: (Text is provided by Dr. Mark Stoelinga of University of Washington.)

LL	lat/lon lines over land only (L), water only (W), none (N), or both land and water (D, A, or E)
DASH	lat/lon lines are dashed large (L), medium (M), small (SM), tiny (T), solid (SO), publ. style (P), or default (D) [LL.ne.N]
INT	lat/lon grid interval in degrees, or D for default [LL.ne.N]
LB	M for only MAPDRV labels (lat/lon on perimeter), N for none, or D or A for both
LSZ	lat/lon label size, 1 to 25 [LB.ne.N]
LQL	label quality: [LB.ne.N] 00 - Complex characters / High quality 01 - Complex characters / Medium quality 02 - Complex characters / Low quality 10 - Duplex characters / High quality 11 - Duplex characters / Medium quality 12 - Duplex characters / Low quality D - Default = 11
P	draw just a line perimeter (N) or a line perimeter with ticks (Y) [DASH.ne.P.or.LL.eq.N]
TTL	title flag: read the next two title parameters (Y) or skip to outline parameter (N)
TSZ and TQL	the same as LSZ and LQL except they refer to the title [both TTL.eq.Y]
OUT	determines which geo-political outlines will be drawn: NO - no outlines CO - continental outlines only US - U.S. State outlines only PS - Continental + International + State outlines PO - Continental + International outlines
DOT	determines whether geo-political boundaries will be dotted (Y) or solid (N) [OUT.ne.NO]

LW	gives the line width, in multiples of default (which is 1000 “units”). D gives default line width. [OUT.ne.NO.and.DOT.eq.N] (LW=2 would double the line width for geographic boundaries)
SP	gives dot spacing. Default (D) is 12 [OUT.ne.NO.and.DOT.eq.Y]

With each parameter is given a conditional statement. If that conditional statement is not met, then that particular box should be made blank. The most common error that occurs when the routine attempts to read this table is “Too many entries on line”, which simply means that the routine expected a box to be blank, but it wasn’t.

One can also do color-filled maps. To do so, add the following in the g_map.tbl:

```
                            MAP FILL
WATER |            SIX COLOR INDICIES WITH WHICH TO COLOR IN THE MAP
-----------------------------------------------------------------------
1     |                  2    |  2      |  2     |  2     |  2      |  2
-------------------------------------------------------------------------------------
```

In this example, the water will be colored white, and land light grey according to the color tabel described below.

8.6 Plot Color Options File: g_color.tbl

This table is used to define the color codes referred in the Graph program.

```
--------------------------------------------------------------------------------
                                 COLOR TABLE
COLOR                     | RED  | GREEN | BLUE | NUMBER
--------------------------------------------------------------------------------
WHITE                     | 1.00 | 1.00  | 1.00 | 1
LIGHT GRAY                | 0.66 | 0.66  | 0.66 | 2
DARK GRAY                 | 0.40 | 0.40  | 0.40 | 3
BLACK                     | 0.00 | 0.00  | 0.00 | 4
SKY BLUE                  | 0.20 | 0.56  | 0.80 | 5
BLUE                      | 0.00 | 0.00  | 1.00 | 6
LIGHT YELLOW              | 0.80 | 0.80  | 0.00 | 7
MAGENTA                   | 1.00 | 0.00  | 1.00 | 8
YELLOW                    | 1.00 | 1.00  | 0.00 | 9
GREEN                     | 0.00 | 1.00  | 0.00 | 10
FOREST GREEN              | 0.14 | 0.25  | 0.14 | 11
CYAN                      | 0.00 | 1.00  | 1.00 | 12
TAN                       | 0.40 | 0.30  | 0.20 | 13
BROWN                     | 0.25 | 0.20  | 0.15 | 14
ORANGE                    | 1.00 | 0.50  | 0.00 | 15
RED                       | 1.00 | 0.00  | 0.00 | 16
MID-BLUE                  | 0.00 | 0.50  | 1.00 | 17
DULL MID-BLUE             | 0.00 | 0.15  | 0.30 | 18
BRIGHT FOREST GREEN       | 0.20 | 0.40  | 0.20 | 19
DULL ORANGE               | 0.60 | 0.30  | 0.00 | 20
```

To make a color contour plot, change the background color from black to white using the following g_color.tbl:

```
----------------------------------------------------------------------------
                                 COLOR TABLE
COLOR                   | RED  | GREEN | BLUE | NUMBER
----------------------------------------------------------------------------
WHITE                   | 1.00 | 1.00  | 1.00 | 0
BLACK                   | 0.00 | 0.00  | 0.00 | 1
LIGHT GRAY              | 0.66 | 0.66  | 0.66 | 2
DARK GRAY               | 0.40 | 0.40  | 0.40 | 3
BLACK                   | 0.00 | 0.00  | 0.00 | 4
```
....

and change color used for maps in the MAP COLORS section of the *g_map.tbl* from 1 to a color code other than white for borders, tick marks, and map background.

8.7 Running GRAPH Interactively

Obtaining Graph tar file

To run GRAPH interactively on a workstation, the first step is to obtain the GRAPH tar file. The GRAPH tar file, ***graph.tar.Z***, can be otained from **~mesouser/MM5V2/Graph** (or **/fs/othrorgs/home0/mesouser/MM5V2/Graph**) from NCAR's Cray, **/MESOUSER/MM5V2/GRAPH/GRAPH.TAR.Z** on MSS, or from the anonymous site (**ftp://ftp.ucar.edu:mesouser/MM5V2/Graph**). This tar file contains the GRAPH source code, make files, as well as the table files required to produce plots.

To get the tar file from the anonymous ftp site:

1) `ftp ftp.ucar.edu`
2) login as `anonymous`
3) use your full email address as the password
4) `cd mesouser/MM5V2/Graph`
5) set the transfer to binary (or image), usually this is just "`bin`"
6) `get graph.tar.Z`
7) `quit`

Or to get the tar file on NCAR's Cray:

```
cd $TMPDIR
msread graph.tar.Z /MESOUSER/MM5V2/GRAPH/GRAPH.TAR.Z
```

or

```
cp ~mesouser/MM5V2/Graph/graph.tar.Z .
```

Compiling Graph code

Once you have the *graph.tar.Z* on the Cray's working directory or on the local workstation, the building process is to uncompress the file, untar the file and make the executable.

1) `uncompress graph.tar.Z`
2) `tar -xvf graph.tar`

   After untarring the file, you should find the following in your working directory among others:

   `configure.make`
   `g_color.tbl`
   `g_defaults.nml`
   `g_map.tbl`
   `g_plots.tbl`
   `graph.csh`

3) edit the file *configure.make* to change the RUNTIME_SYSTEM according to your machine; and comment and uncomment the corresponding compiler options.
4) if working on a workstation, edit file *word_length.incl* in the src/ directory to change LENS value according to your machine's specification.

If your dataset dimensions are greater than 200x200x40, you need to edit two more files in src/ directory: *scratch.incl* and *data.incl*

5) type “`make`”, and this will create a graph executable in src/ directory, called *graph.exe.*

(if working on NCAR’s Cray, a user can simply copy an existing executable from *~mesouser/MM5V2/Graph/graph.exe.Z* to src directory, uncompress it.)

6) type “`ln -s src/graph.exe graph.exe`” so the *graph.exe* shows up in the top directory.

7) edit the *g_plots.tbl* and *g_defaults.nml* (if needed) files.

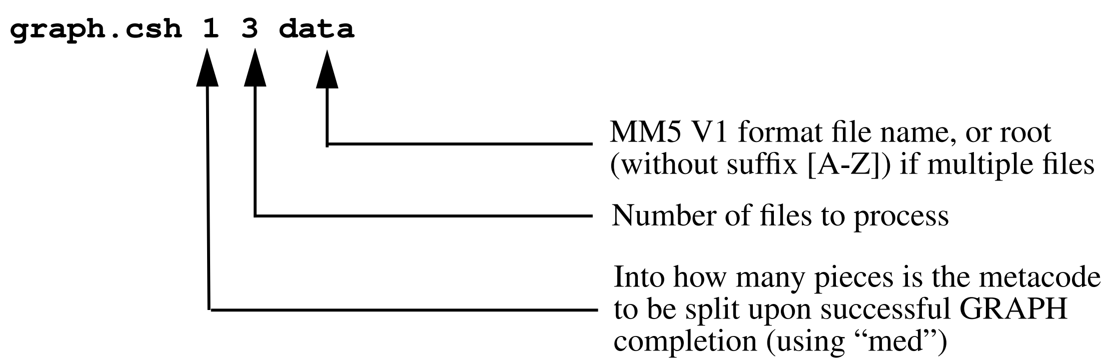
8) if a user is working on NCAR’s Cray, he/she needs to retrieve data from MSS by typing the following:

```
msread data[A-Z] MSSfilename &
```

The ‘&’ puts the `msread` command in the background. If the data was “bsplit”, `cat` it all together.

Running Graph

Program Graph can only process output from one domain at a time. To run Graph, type “**`graph.csh 1 1 dataA`**”, or



For example,

a) to run graph with one data file:

`graph.csh 1 1 dataA`

b) to run graph with 3 files named dataA, dataB and dataC:

`graph.csh 1 3 data`

Generated Files

The plot files generated by Graph are metacode files called ‘**gmeta**’ (and gmeta.split1, gmeta.split2, etc. if you choose to split the file), which can be viewed by NCAR Graphics utility **idt,** and/or transformed to postscript files using **ctrans** (also an NCAR Graphics utility). For example, to transfer a gmeta file to postscript file,

```
ctrans -d ps.mono (or ps.color) gmeta > gmeta.ps
```

Or to view the output interactively using an interface,

```
idt gmeta
```

8.8 Available 2-D Horizontal Fields

Note: Some variables (marked with [7]) are only available from MM5 output fields since Release-2-7 (Dec. 1997)

Table 8.1 List of 2-D horizontal fields available for plotting.

Field ID	Description	Default Units	Optional Units	Optional Units	Optional Units
CORIOLIS	Coriolis parameter	1/s			
ICLW	integrated cloud water	cm	mm	in	
IRNW	integrated rain water	cm	mm	in	
LATITDOT	latitude	degrees			
LI	lifted index	K	C		
LNDUS	land use categories	(no units)			
LHFLUX[7]	surface latent heat flux	W/m^2			
LONGIDOT	longitude	degrees			
LWDOWN[7]	longwave downward radiation	W/m^2			
MAPFACDT	map scale factor	(no units)			
PBL HGT[7]	PBL height	m			
PRECIPT	total accumulated precipitation	mm	cm	in	
PRECIPC	convective accumulated precip	mm	cm	in	
PRECIPN	stable accumulated precip	mm	cm	in	
PRECIPTT	total precip during time interval	mm	cm	in	
PRECIPTN	stable precip during time interval	mm	cm	in	
PRECIPTC	convective precip during interval	mm	cm	in	
PRH2O	precipitable water	cm	mm	in	
PSLV	sea level pressure	mb	hPa	Pa	inHg
PSFC	surface pressure	mb	hPa	Pa	inHg
PTEND	pressure change	mb	hPa	Pa	
RAINT	total accumulated precipitation	mm	cm	in	

Field ID	Description	Default Units	Optional Units	Optional Units	Optional Units
RAINC	convective accumulated precip	mm	cm	in	
RAINN	stable accumulated precip	mm	cm	in	
RTENDT	total precip during time interval	mm	cm	in	
RTENDC	convective precip during interval	mm	cm	in	
RTENDN	stable precip during time interval	mm	cm	in	
REGIME[7]	PBL regimes (values 1-4)	catagory			
SHFLUX[7]	surface sensible heat flux	W/m^2			
SOIL T 1[7]	soil temperature in 1 cm layer	K			
SOIL T 2[7]	soil temperature in 2 cm layer	K			
SOIL T 3[7]	soil temperature in 4 cm layer	K			
SOIL T 4[7]	soil temperature in 8 cm layer	K			
SOIL T 5[7]	soil temperature in 16 cm layer	K			
SOIL T 6[7]	soil temperature in deep layer	K			
SWDOWN[7]	shortwave downward radiation	W/m^2			
TER	terrain elevation	m	ft		
TGD	ground temperature	K	C		
THK	thickness	m			
TSEA	sea surface temperature	K	C		
UST[7]	frictional velocity	m/s			

8.9 Available Cross-Section Only Fields

Table 8.2 List of cross-section-only fields available for plotting.

Field ID	Description	Default Units	Optional Units	Optional Units	Optional Units
AM	absolute momentum	m/s			
AXW	wind speed tangential to the cross-section	m/s			
CUV	horizontal wind barb in plane	m/s			
CXW	circulation vectors in cross-section plane	m/s			
XXW	wind speed normal to the cross-section	m/s			

8.10 Available 3-D Fields (as 2-D Horizontal or Cross-Section)

Table 8.3 List of 3-D fields available for plotting.

Field ID	Description	Default Units	Optional Units	Optional Units	Optional Units
AGL	above ground level	m	cm	Dm	
BARB	wind barbs	m/s	kt	cm/s	
CLB	cloud boundary	g/kg	kg/kg	mg/kg	
CLW	cloud water	g/kg	kg/kg	mg/kg	
DIV	divergence of horizontal wind	10**5/s	1/s		
GRA	graupel	g/kg	kg/kg	mg/kg	
H	geopotential height	m			
HEIGHT	geopotential height	m			
ICE	cloud ice	g/kg	kg/kg	mg/kg	
MDIV	moisture divergence	10**7/s	1/s		
MSE	moist static energy	J/kg			
MSS	saturated moist static energy	J/kg			
NCI	number concentration of ice	number/m^3			

Field ID	Description	Default Units	Optional Units	Optional Units	Optional Units
OMG		ub/s	mb/s	hPa/s	
P	pressure	mb	Pa	hPa	
PP	pressure perturbation	mb	Pa	hPa	
PV	potential vorticity	PVU			
QDIV	q-vector divergence *(p data only)*				
QV	mixing ratio	g/kg	kg/kg		
QVEC	q-vectors (*p data only*)				
RDTEND[7]	atmospheric radiative tendency	K/day	K/h		
RH	relative humidity	%			
RNW	rain water	g/kg	kg/kg	mg/kg	
SLW	super-cooled liquid water	g/kg	kg/kg	mg/kg	
SNOW	snow	g/kg	kg/kg	mg/kg	
T	temperature	K	C	F	
TD	dew point temperature	K	C	F	
TDD	dew point depression	K	C	F	
THETA	potential temperature	K	C	F	
THETAE	equivalent potential temperature	K	C	F	
TKE	turbulent kinetic energy	J/kg			
U	u-component of wind	m/s	kt	cm/s	
V	v-component of wind	m/s	kt	cm/s	
VAB	absolute vorticity	10**5/s	1/s		
VECT	horizontal wind vectors	m/s	kt	cm/s	
VESL	horizontal wind streamlines	m/s			
VOR	relative vorticity	10**5/s	1/s		
W	w-component of wind	m/s	kt		
WIND	wind speed	m/s	kt	cm/s	

8.11 Some Hints for Running GRAPH

- make sure the following line is included in your .cshrc file on NCAR's Cray:
  setenv NCARG_ROOT /usr/local
- The GRAPH program uses the information in the record header to define the size and location of the data. This limits the "wrong" data that the user can provide to be related to the requested fields and levels to be plotted.
- GRAPH prints out information to allow you to track the program's status. It should inform the user that it is processing each of the requested time periods, and for each of the requested variables and levels.
- If the GRAPH program is not processing the time that you have requested, and it should be based upon the intervals that you have set, ask GRAPH to plot every time.
- GRAPH only vertically interpolates data that is on a σ coordinate.
- Contour intervals for precipitation are fixed for the RAIN and RTEND fields and user modifiable for the PRECIP fields.
- Do not request a subdomain and also process soundings. The GRAPH program will not place the sounding at the correct location for the large domain.
- Errors related to the NAMELIST or other temporary files are common when porting GRAPH to a different architecture. Use the NAMELIST format in the architecture's FORTRAN manual. Make sure and remove all temporary files and links prior to each initiation of the GRAPH C-shell.
- When GRAPH is compiled on a different architecture, the length of the records for the direct access files must be specified in bytes (4 or 8 per word) or words. This information is found in the include file word_length.incl.
- If you get the following, and Graph stops,
  `NEED MORE DATA SPACE`
  it means that you need to increase the dimensions in the data.incl and scratch.incl files.
- If you get error message related to 'Direct Access Files', it is usually an indication that the dimensions specified in data.incl and scratch.incl files are not large enough, or the running memory is not large enough.

8.12 Sample Graph Plot File

For some horizontal plots, please refer to Chapter 12, page 12-7 and 12-8.

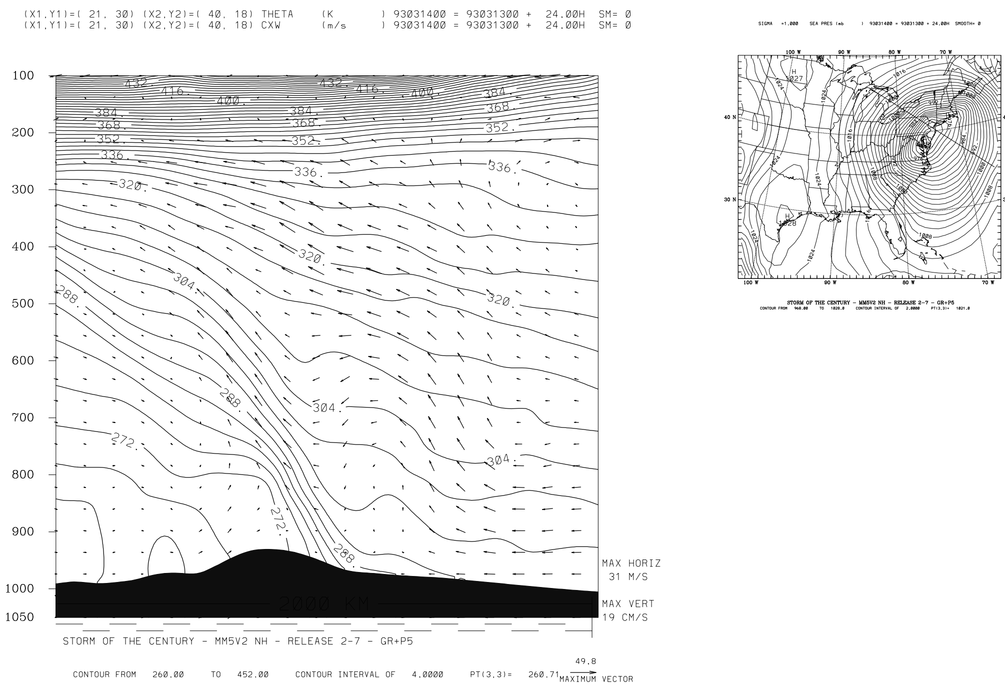


Figure 8.1 A NW-SE cross section of potential temperature (unit K and contour interval 4 K) and 2-D in-plane circulation vector through the warm front.

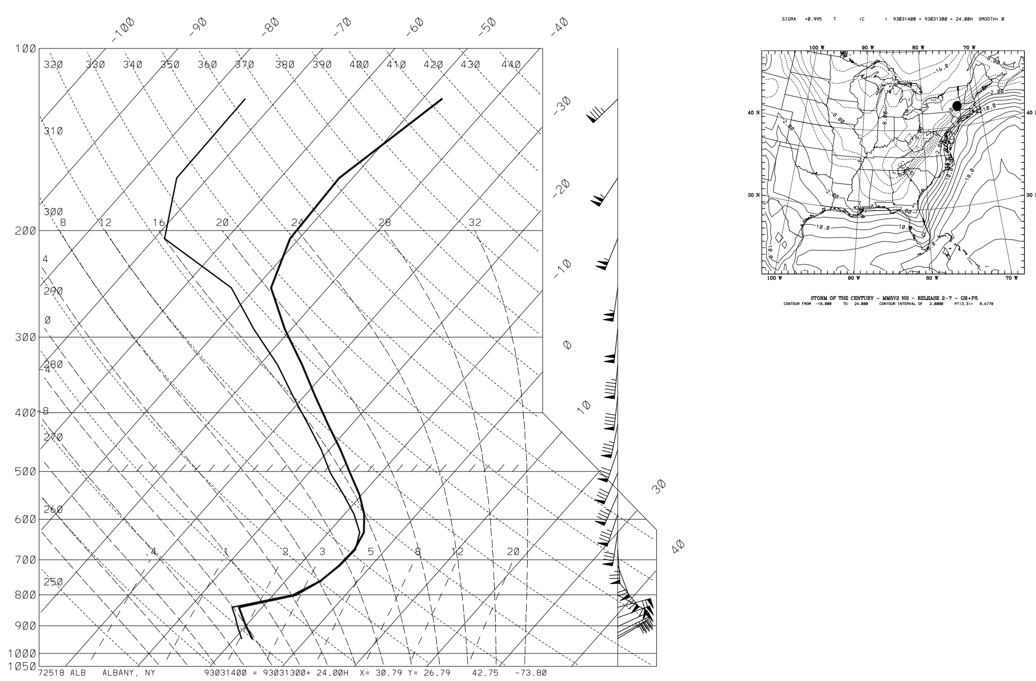


Figure 8.2 A skew-T plot from a 24-h simulation at Albany, New York.

8.13 Graph tar file

The graph.tar file contains the following files and directories:

CHANGES	Description of changes to the Graph program
Diff/	Will contain difference files between consecutive releases
Makefile	Makefile to create Graph executable
README	General information about the Graph directory and how to run Graph
Templates/	Job deck directory: batch deck for Cray only
configure.make	Rules for the Makefile
g_color.tbl	Color table for Graph job
g_defaults.nml	NAMELIST file for Graph job
g_map.tbl	Map table for Graph job
g_plots.tbl	Table for selecting plot variables
graph.csh	C-shell script to run Graph interactively
src/	Graph source code

8.14 Script file to run Graph job

```
#
#       this is INTERACTIVE or BATCH
#
if ( $?ENVIRONMENT ) then
   echo "environment variable defined as $ENVIRONMENT"
else
   setenv ENVIRONMENT INTERACTIVE
   echo "environment variable defined as $ENVIRONMENT"
endif
#
#       initializations, no user modification required
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
#
#       is it color
#
# set Color = BW
set Color = CO
#
if      (( $0 != GRAPH_NEW ) && ( $0 != GRAPH_TEST )) then
   set NewOne = NOPE
else if (( $0 == GRAPH_NEW ) || ( $0 == GRAPH_TEST )) then
   set NewOne = YES
endif
#
model >& /dev/null
set OK = $status
if ( $OK == 0 ) then
    set ForUnit = ftn
#       echo "This is an HP"
else
    set ForUnit = fort.
#       echo "This is not an HP"
endif
#
if ( -e numsplit.tbl ) rm numsplit.tbl
if ( -e med.input ) rm med.input
if ( ( -e ${ForUnit}20 ) || ( -l ${ForUnit}20 ) ) rm ${ForUnit}2*
#
#       simple error check on call
#
```

```
if (( $#argv == 0 ) && ( $ENVIRONMENT == INTERACTIVE )) then
   echo -n "into how many pieces is the metafile to be split (1) "
   set NumSplit = "$<"
   echo -n "how many input files are there                 (1) "
   set TotFiles = "$<"
   echo -n "what is the name of the file              (graphin) "
   set FileName = "$<"
else if ( $#argv < 3 ) then
   echo "graph.deck:  error in call"
   echo "usage:  graph.deck ns nf filename [filename2 filename3 ...]"
   echo "      where ns is the number of files into which metacode is split"
   echo "      where nf is the number of input files"
   echo "      where filename is either the root name, or several names"
   exit (1)
else if ( $#argv == 3 ) then
   set NumSplit = $1
   set TotFiles = $2
   set FileName = $3
else if ( $#argv >  3 ) then
   set NumSplit = $1
   set TotFiles = $2
endif
#
#       consistency checks on input
#
if (( $NumSplit < 1 ) || ( $NumSplit > 99 )) then
   set NumSplit = 1
endif
cat > numsplit.tbl << EOF
$NumSplit
EOF
if (( $TotFiles < 1 ) || ( $TotFiles > 99 )) then
   set TotFiles = 1
endif
if ( $#argv == 0 ) then
   if (( ! -e $FileName ) && ( ! -e ${FileName}A )) then
      echo "file $FileName does not exist"
      exit (1)
   endif
else if ( $#argv >=  3 ) then
   if (( ! -e $argv[3] ) && ( ! -e $argv[3]A ) && ( $0 != GRAPH_NEW )) then
      echo "file $argv[3] does not exist"
      exit (1)
   endif
endif
#
#   if NewOne == YES, recompile
#
if (( $NewOne == YES )) then
    if ( -e src/graph.exe  ) rm src/graph.exe
    make clean
    make
endif
#
if ( ! -e graph.exe ) ln -s src/graph.exe graph.exe
chmod +x graph.exe
#
#       make sure we have all the required tables
#
if ( -e g_map.tbl ) then
   echo "using local copy of g_map.tbl"
else
   echo "need a copy of g_map.tbl"
   exit (0)
endif
#
```

```
if ( -e g_color.tbl ) then
   echo "using local copy of g_color.tbl"
else
   echo "need a copy of g_color.tbl"
   exit (0)
endif
#
if ( -e g_plots.tbl ) then
   echo "using local copy of g_plots.tbl"
else
   echo "need a copy of g_plots.tbl"
endif
#
#       run graph program
#
if ( ( -e ${ForUnit}18 ) || ( -l ${ForUnit}18 ) ) rm ${ForUnit}18
ln -s     g_plots.tbl                  ${ForUnit}18
if ( -e .assign ) rm .assign
if ((( $TotFiles == 1 ) && ( $#argv == 3 )) || \
    (( $TotFiles >  1 ) && ( $#argv >  3 ))) then
   shift
   shift
   set NUMFIL = 1
   while ( $#argv )
      @ UNIT = 19 + $NUMFIL
      ln -s     $argv[1]               ${ForUnit}$UNIT
#     assign -a $argv[1] -Ff77 -Nieee  ${ForUnit}$UNIT
      shift
      @ NUMFIL ++
   end
else if (( $TotFiles > 1 ) || ( $#argv == 3 )) then
   set NUMFIL = 1
   while ( $NUMFIL <= $TotFiles )
      @ UNIT = 19 + $NUMFIL
      ln -s     $FileName$LETTERS[${NUMFIL}]              ${ForUnit}$UNIT
#     assign -a $FileName$LETTERS[${NUMFIL}] -Ff77 -Nieee ${ForUnit}$UNIT
      @ NUMFIL ++
   end
else if ( $#argv == 0 ) then
   set NUMFIL = 1
   while ( $NUMFIL <= $TotFiles )
      @ UNIT = 19 + $NUMFIL
      ln -s     $FileName              ${ForUnit}$UNIT
#     assign -a $FileName -Ff77 -Nieee ${ForUnit}$UNIT
      @ NUMFIL ++
   end
endif
#
#       run graph program
#
graph.exe
#
#       split metacode apart
#
if      ( $NumSplit != 1 ) then
   cat med.input
   med -f med.input
else if ( $NumSplit == 1 ) then
   cat med.input
   cp gmeta gmeta.split1
endif
#
rm med.input numsplit.tbl
rm tmp.*
echo "GRAPH run complete"
```