RST

Global Convection Mapping Software

R.J.Barnes

Contents

Convection Map Tutorial

A Brief tutorial on how to generate and manipulate SuperDARN Global Convection Maps.

Tutorial

The APL Convection Mapping software consists of a number of programs that are used to process SuperDARN data to produce Global Ionospheric Convection Maps. Rather than perform the analysis as a single stage, monolithic task, the process is program down into simple stages that are calculated by individual programs. This has the advantage that the user can modify the analysis at any stage without having to re-process the data in its entirety. At first, you may find that the number of steps involved in the analysis is confusing and cumbersome, however once you have become familiar with the software you will be able to string multiple stages together feeding the output of one step into the input of another.

Generating Grid Files

Before you can start generating convection maps you must generate a grid file. These contain filtered line of sight velocity vectors that have been fitted to an equi-area global grid. A file usually contains data from one or more radar and multiple grid files are combined to generate the input required for the convection map analysis. If you have a handy source of pre-generated grid files you can obviously skip this section.
A grid file is generated from a fit file using the program `make_grid`:

```
make_grid 200112300k.fit > 2000112300k.grd
```

The program writes the grid file to the standard output, so the pipe "`>`" is used to redirect it to the file "`2000112300k.grd`". The "`inx`" file associated with the fit file is not required for this kind of processing.
To monitor what is going on you can use the verbose option:

```
make_grid -vb 2000112300k.fit > 2000112300k.grd
```

In the default mode `make_grid` produces records that contain two minutes of data, this can be over-ridden using "`-i`" option and specify the record length in seconds:

```
make_grid -vb -i 180 2000112300k.fit > 2000112300k.grd
```

The program uses the scan flag contained in the fit file to identify individual Radar scans, but you will often find that the scan flag is incorrectly set and `make_grid` will fail. To overcome this problem you can specify a fixed scan length using the "`-tl`" option:

```
make_grid -vb -tl 120 2000112300k.fit > 2000112300k.grd
```

In this mode, data is read from the file in two-minute chunks, synchronized with the start of the day. So all the data between 0:00UT and 0:02UT is assumed to come from the first scan, and so on.
It is important to distinguish the difference between the "`-i`" and "`-tl`" options as they can have a dramatic impact on the output produced. The processing consists of two stages, the first involves filtering whole radar scans to remove noise and ground scatter contamination. The "`-tl`" option sets the length of each input scan, and at the end of the filtering process the scan will consist of filtered, averaged data with no more than one velocity measurement in each range/beam cell. The second stage involves fitting the filtered data to an equi-area grid. Filtered scans are mapped to grid for the specified record length as defined by the "`-i`" option. If the scan interval is less than the record length, multiple scans will form an output record. If a grid cell contains multiple velocities these are averaged in the output record.
There are many more command line options that can be applied to `make_grid`, the most useful allow you to specify the start and end time of the period to process:

```
make_grid -vb -st 9:00 -et 13:00 -tl 120 -i 120 2000112300k.fit >
                                                  2000112300k.grd
```

You must run `make_grid` for the fit file for each Radar site that you are including in the analysis.

Combining Grid files

A grid file with data from only one Radar site is not particularly useful for generating global convection maps. The next step of the process is to combine the data from individual radars together into a single grid file. This is done using the program `combine_grid`:

```
combine_grid 2000112300e.grd 2000112300f.grd 2000112300g.grd >
                                              2000112300.grd
```

Again note that the program writes the file to standard output and so must be redirected into a file. One obvious short cut is to use wildcards :

```
combine_grid 2000112300?.grd > 2000112300.grd
```

One thing to be careful of is that your wildcarded name doesn't match the name of the output file, or you'll end up with a recursion and a really big output file. In the case above the "?" wildcard was used rather than "*" for this very reason.
The only option that `combine_grid` takes is "`-r`", which indicates that if two of the input files contain data from the same radar, only the data from the file last on the command line will be included in the output. This is a confusing way of saying that you can use this option to replace data from one file with another. This is particularly useful if you discover that you need to re-process the data from one radar and don't have access to the individual grid files:

```
combine_grid -r 2000112300.grd 2000112300e.grd.new >
                               2000112300.grd.new
```

Working with Grid Files

Once you've generated a grid file, you probably would like to see what it looks like. This can be done using the `grid_plot` program:

```
grid_plot -x 2000112300.grd
```

The "`-x`" option indicates that the data should be plotted in an X window. You can vary the delay between frames in fractions of a second use the "`-delay`" option:

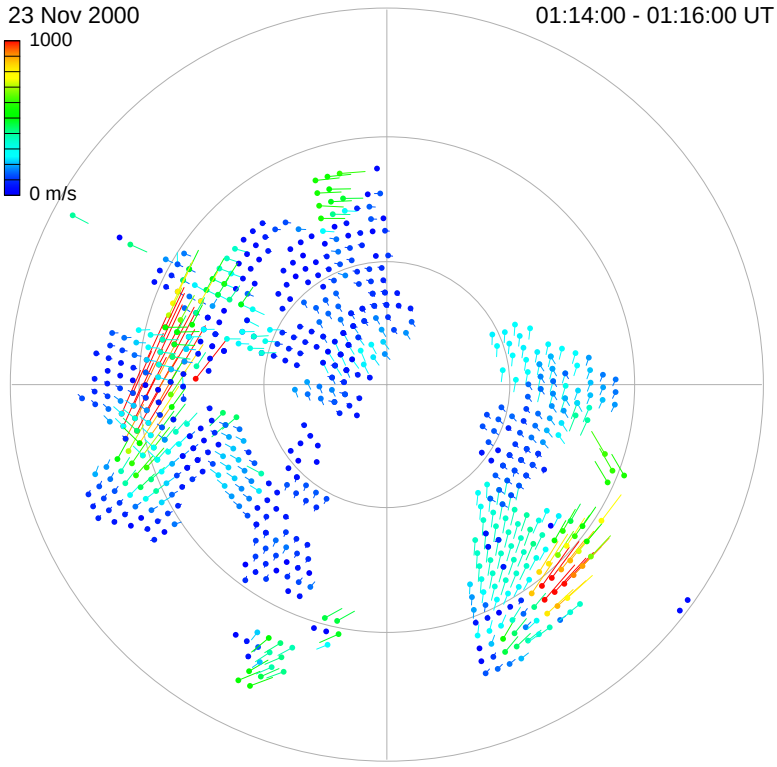
```
grid_plot -x -delay 1.5 2000112300.grd
```

Setting a delay time of zero will cause the program to wait between frames until the user clicks on the plot window.
You can also create either PostScript or PPM images using the "`-ps`" or "`-g`" options:

```
grid_plot -ps 2000112300.grd
```

The PostScript files created will be named "`0000.ps`", "`0001.ps`" and so on. If the "`-dn`" option is specified then filenames that correspond to the date and time of the record plotted are created, eg. "`20001123.1120.ps`".
You can also generate a single multi page PostScript file using the "`-mp`" option

```
grid_plot -ps -mp 2000112300.grd > output.ps
```



There are a lot of other command line options that can be applied to `grid_plot` and it can be irksome to have to type them out each time you want to plot some data. To avoid this you can place the options you want in a text file and use the " `-cf`" option to include it:

```
# My options for grid plot
-p -dn -key color.key -fcoast -coast -term
```

```
grid_plot -cf my.options 2000112300.grd
```

Often you only want to process a small section of a grid file and would like to be able to trim a section out of it. To do this you can use the `trim_grid` program:

```
trim_grid -st 11:00 -et 14:00 2000112300.grd > trim.grd
```

As usual, the program writes the file to standard output so it must be redirected to a suitable file.
If the grid file contains more than one day of data you might need to specify the start and end dates:

```
trim_grid -sd 20001123 -st 23:00 -ed 20001124 -et 1:00
              2000112300.grd > trim.grid
```

You can also use `trim_grid` to exclude data from a particular station from a file:

```
trim_grid -e 2000112300.grd > 20001123.not.e.grd
```

One final program that may be of use is `extract_grid`. This program will print out some statistics from a grid file:

```
extract_grid 2000112300.grd > 20001123.sct
```

These can be used to work out scatter statistics and show the contributions from the various radars.

Index Files

When dealing with large grid files it can take a long while to read them. If you are only interested in a small section of a file, this can be a pain, as the whole file must still be read. To solve this problem, you can create special index files from the grid files:

```
index_file 2000112300.grd > 2000112300.inx
```

The index file lets you quickly jump to the required section of the code. Most of the software that lets you add an index file as a command line argument:

```
trim_grid -st 11:00 -ex 1:00 2000112300.grd 2000112300.inx >
                                              trim.grd
```

Generating Convection Maps

Generating Convection maps from grid files is a multi-stage process. The first step is to check the solar wind data to determine the suitable delay time to apply to the IMF data. This can be done by either using the cdaweb web site, or by using the `istp_plot` tool:

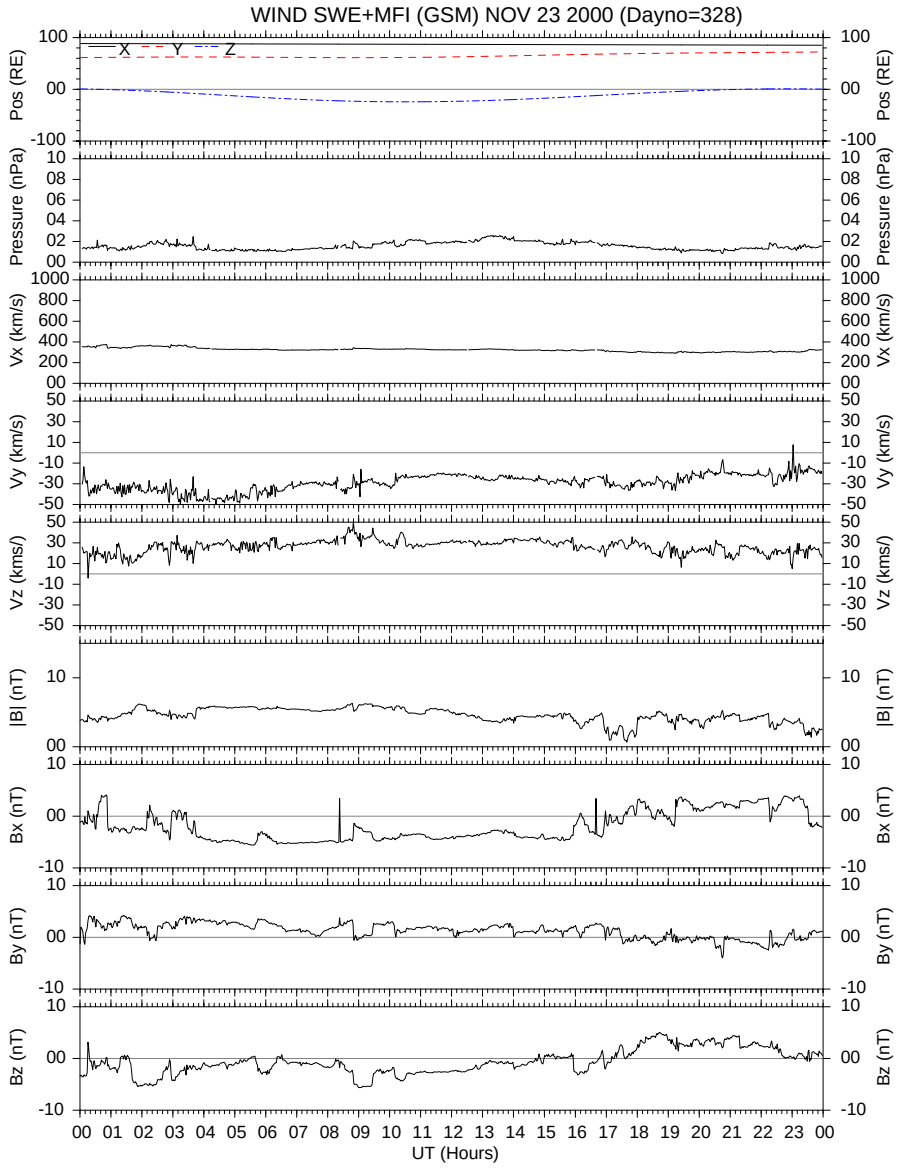
```
istp_plot -x -ace -swe -mfi -sd 20001123
```

The "`-x`" option indicates that the data should be plotted in an X window. The plot is produced in PostScript by default, but can be written to a PPM file the plot by specifying the "`-g`" option:

```
istp_plot –g -wind -swe -mfi -sd 20001123 > istp.ppm
```

The output is written to standard output so it must be redirected into an appropriate output file. The "`-sd`" option selects the start date, and you can also add an "`-ed`" option to set the end date:

```
istp_plot -wind -swe -mfi -sd 20001122 -ed 20001124 > istp.ps
```



Before you can create convection maps you must reformat the grid file into the map file format. This file contains a number of empty records that will be populated by the later stages of processing. The `map_grd` program is used to reformat the data:

```
map_grd 2000112300.grd  > 2000112300.empty.map
```

The next step of the analysis is to calculate the Hepner-Maynard Boundary for the data. This is done using the `map_addhmb` tool:

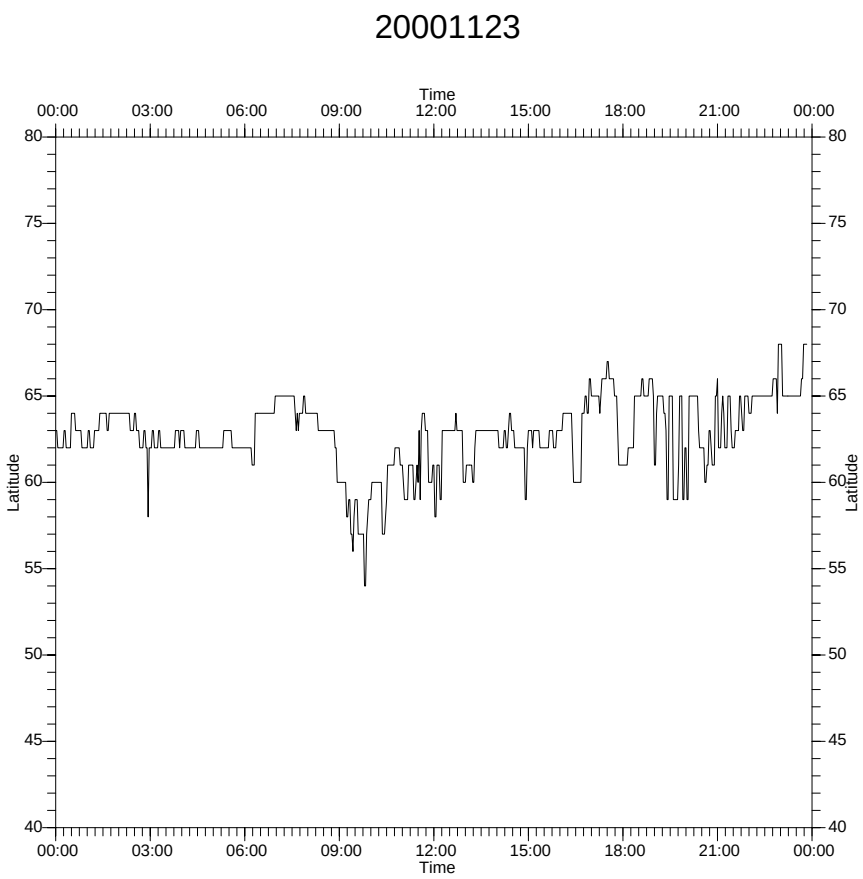
```
map_addhmb 2000112300.empty.map > 2000112300.hmb.map
```

The "`-vel`" and "`-cnt`" options control the velocity and point thresholds used by the algorithm that determines the location of the boundary:

```
map_addhmb -cnt 6 -vel 120 2000112300.empty.map >
                              2000112300.hmb.map
```

It is often helpful to check that the boundary generated is sensible, to do this use `hmb_plot` tool:

```
hmb_plot -x 2000112300.hmb.map
```



The next step of the process is to add the IMF data to the empty file using the `map_addimf` program:

```
map_addimf -ace -d 0:30 -vb 2000112300.hmb.map > 20001123.imf.map
```

In this example, a fixed delay of 30 minutes is used and the IMF data is taken from the ACE spacecraft. The program can also read IMF delay times from a text file, or read IMF data from a plain text file:

```
map_addimf -df delay.txt -if imf.txt 2000112300.empty.map >
                                     20001123.imf.map
```

Once the IMF data has been incorporated the model vectors can be calculated using the `map_addmodel` tool:

```
map_addmodel -o 8 2000112300.imf.map > 2000112300.model.map
```

The order of fit determines the number of model vectors that are added to the convection map, so this must be specified on the command line using the "`-o`" option. The level of doping can be varied using the "`-d`" option.

```
map_addmodel -o 8 -d l 2000112300.imf.map > 2000112300.model.map
```

The final, and most time consuming step of the process is to perform the actual fitting using the `map_fit` tool:

```
map_fit -vb 2000112300.model.map > 2000112300.map
```

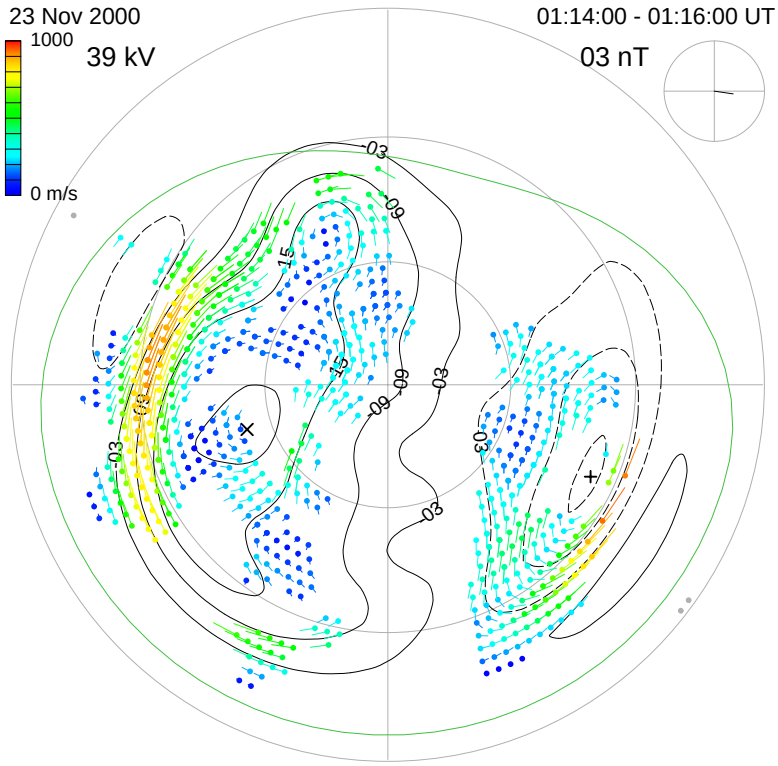
Working with map files

Having generated the convection map file, it would be nice to be able to plot it. The tool map_plot can be used to generate plots from the file:

```
map_plot –x 2000112300.map
```

The "`-x`" option indicates that the data should be plotted in an X window. You can vary the delay between frames in fractions of a second use the "`-delay`" option:

```
map_plot -x -delay 1.5 2000112300.map
```



As with `grid_plot`, setting a delay time of zero will cause the program to wait between frames until the user clicks on the plot window.
The program is a close cousin to `grid_plot` and shares many of the same command line options.

Often you will need to remove a small section from a map file for further processing. This is done using the `trim_map` tool, a close cousin to `trim_grid`.

```
trim_map –st 11:00 –et 13:00 2000112300.map > trim.map
```

The program extract_map is used to either re-create the original grid file from the map, or to extract some useful statistics from the file.

```
extract_map 2000112300.map > 2000112300.grd
extract_map –s 2000112300.map > 2000112300.sct
```

The final program, called `map_cnv` is used to generate some derived data products from the map file. The most useful one is a regularly spaced grid of potentials derived from the fit:

```
map_cnv –p 2000112300.map > 2000112300.pot
```

All of these programs are fully documented later in this guide.

Advanced Topics

The number of stages involved in processing the data might appear unnecessarily complicated. However, the different steps can be combined together:

```
map_grd 2000112300.grd | map_addhmb | map_addimf –ace –d 0:30 |
map_addmodel –o 8 | map_fit –vb > 2000112300.map
```

You could even generate a simple shell script to do this :

```
#!/bin/sh
# map_all
#
# script for one step convection maps.

delay=$1
order=$2
fname=$3

map_grd ${fname} | map_addhmb | map_addimf –ace –d ${delay} | \
map_addmodel –o ${order} | map_fit –vb
```

This script would be invoked like this:

```
map_all  0:30 8 2000112300.grd  > 2000112300.map
```

One advantage of breaking the analysis into a number of different steps is that changes can be made to the analysis without having to re-process all of the data. For instance to change the order of the fit, you need only re-run `map_addmodel` and `map_fit.`

If the IMF conditions are very variable and you need to specify multiple delay times for the period of data that you are working on, you can use the ability of `map_addimf` to read delays from a file:

```
2000 11 23 0 0 0  0 30
2000 11 23 11 30 0 1 0
```

```
map_addimf –df delay.txt 2000112300.hmb.map > 2000112300.imf.map
```

In a similar fashion, if the IMF data needs some post-processing to remove noise, you can generate a text file using the `istp_text` program, clean the output, and use this to supply the IMF data:

```
istp_text –sd 20001123 –ace –mfi > imf.txt
smooth_imf imf.txt > imf.new.txt
map_addimf –df delay.txt –if imf.new.txt 2000112300.hmb.map >
                                          2000112300.imf.map
```

Similarly if the Heppner-Maynard Boundary requires some cleaning you can generate a plain text version using the `map_addhmb` program:

```
map_addhmb –t 2000112300.empty.map > hmb.txt
smooth_hmb hmb.txt > hmb.new.txt
map_addhmb –lf hmb.new.txt 2000112300.empty.map >
                           2000112300.hmb.map
```

Tutorial

Quick Guide

The quick quide lists the most commonly used command line options of the programs.

```
make_grid [–vb] [–tl sec] [–i sec] [-st hr:mn]
          [-et hr:mn] fitfile > gridfile
```

```
combine_grid [-vb] gridfiles... > gridfile
```

```
trim_grid [-vb] [–st hr:mn] [-et hr:mn]
          [gridfile] > gridfile
```

```
grid_plot [-vb] [-x] [-ps] [-g] gridfile
```

```
map_grd [-vb] [gridfile] > [mapfile]
```

```
map_addhmb [-vb] [-v vel] [-c cnt] [mapfile] >
           [mapfile]
```

```
hmb_plot [-vb] [-x] [-ps] [-g] [mapfile]
```

```
map_addimf [-vb] [-ace] [-wind] [-d hr:mn]
           [mapfile] > mapfile
```

```
map_addmodel [-vb] [-o order] [mapfile] > mapfile
```

```
map_fit [-vb] [mapfile] > mapfile
```

```
map_plot [-vb] [-x] [-g] [-ps] mapfile
```

Environment Variables

Many of the programs use environment variables to locate data files and directories.

Environment Variables

Many of the tasks make use of environment variables to locate directories and configuration files. The following is a list of the environment variables that are used in the code. They can be divided into four groups; general purpose variables used by virtually all the code, analysis variables used by some of the analysis libraries, SupeDARN specific variables that are used by the SuperDARN data processing tasks, and Radar Operating System variables that are used to locate data directories and site configuration files.

General Purpose

FONTPATH

The pathname of the directory that contains the data files used to render fonts in graphics objects.

MAPDATA

The name of the file that contains the coastline data used for plotting maps.

MAPOVERLAY

The name of the file that contains the overlay data used for plotting maps.

Analysis

AACGM_DAT_PREFIX

The prefix used to construct the filenames of the AACGM co-efficient files.

IGRF_PATH

The pathname of the directory that contains the data files used by the IGRF library.

ISTP_PATH

The pathname of the directory containing the ISTP key-parameter CDF files.

SuperDARN

SD_LOGODATA

The filename of the data file used by the logo library to plot the SuperDARN logo.

SD_RADARNAME

The name of the file that contains the list of Radar names. The file contains the station number, identifier character, status and operator of each of the Radars.

SD_HARDWARE

The prefix used to construct the filenames of the hardware configuration files.

Software Reference

Reference Guide to the Software Tools used to Generate Convection Maps.

combine_grid

Usage

```
combine_grid [--help] [-vb] [-r] files...
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-r`	combine with replacement. As each input record is combined together to form the output, a check is made to see if any of the data is from a station already included. If a duplicate set of vectors is found they will replace the existing vectors in the output.
files	list of grid files to combine

Description

Combines together multiple grid files to produce a single file written to standard output. By default the output record is the simple combination of all of the input records. If two records contain data from the identical station, the two sets of vectors are both included in the output record.
The "-r" option combines with replacement so that as each input record is processed, a check is made to see if any of the vectors are from a station that has already been included. Any duplicate vectors replace the existing data in the output file.
This option is most useful when dealing with a grid file containing data from one station that is contaminated with noise. Rather than having to reprocess the entire file, the user can regenerate a new grid file for the affected station and then use the combine with replacement option to replace it in the existing grid file.
The resultant file is written to standard output.

Examples

```
combine_grid -vb 19981020?.grd > 19981020.grd
```

Combines together all files called "`19981020?.grd`" to produce a file called "`19981020.grd`". The status is logged to standard error.

```
combine_grid -vb -r 19991120.grd 19991120g.grd > 19991120.2.grd
```

Combines with replacement the file "`19991120.grd`" and "`19991120g.grd`" to produce the output file "`19991120.2.grd`".

extract_grid

Usage

```
extract_grid [--help] [-mid] [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-mid`	record the time at the middle of the record, rather than the start and end times.
file	grid file to process. If none is specified then standard input will be used.

Description

Extracts the scatter statistics from a grid file. The number of stations, their identifier codes and the number of vectors that they contribute to each record is extracted and written to standard output.
Each record in the grid file produces a single line of output containing the full start and end times of the record:

```
syear smonth sday shour sminute sseconds eyear emonth eday ehour eminute seconds….
```

If the "`-mid`" option is specified the line will contain the time at the middle of the record:

```
myear mmonth mday mhour mminute mseconds…
```

The remainder of the line lists the number of stations in the record followed by each stations identifier number, the total number of vectors in the record, and the number of vectors associated with each station:

```
nid idA idB idC…idn  nvec vecA vecB vecC… vecn
```

Examples

```
extract_grid 19981020.grd
```

Extracts the scatter statistics from the file "`19981020.grd`" and display them on the console.

```
extract_grid –mid 19991120.grd > 19991120.sct
```

Extract the scatter statistics from the grid file "`19991120.grd`" to produce the output file "`19991120.sct`". The middle time of each record is recorded in the file.

extract_map

Usage

```
extract_map [--help] [-mid] [file]
extract_map –p [--help] [-mid] [file]
extract_map –s [--help] [-mid] [file]
extract_map –l [--help] [-mid] [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-mid`	record the time at the middle of the record, rather than the start and end times.
file	map file to process. If none is specified then standard input will be used.
`-p`	extract cross polar cap potential and the statistics of the fit from the data file.
`-s`	extract the scatter statistics from the file.
`-l`	extract the lower latitude boundary from the file.

Description

Extracts information from a map file. By default, the program extracts the original grid file that was used to produce the map file. For the other options, each record in the map file produces a single line of output containing the full start and end times of the record:

syear smonth sday shour sminute sseconds eyear emonth eday ehour eminute seconds….

If the "`-mid`" option is specified the line will contain the time at the middle of the record:

myear mmonth mday mhour mminute mseconds…

If the "`-p`" option is specified, the remainder of the line lists the cross-polar cap potential in Kilovolts, the delayed IMF conditions, two numbers that identify the model used, the number of data points in the fit, the number of stations contributing to the fit, the X^2 error and the RMS error:

P Bx By Bz dir mag npnt nid X^2 *RMS*

For the "`-s`" option, the remainder of the line lists the number of stations in the record followed by each stations identifier number, the total number of vectors in the record, and the number of vectors associated with each station:

nid idA idB idC…idn nvec vecA vecB vecC… vecn

For the "`-l`" option, the remainder of the line contains the lower latitude boundary of the fit.

Examples

```
extract_map 19981020.map
```

Extracts the grid records from the file "`19981020.map`" and display them on the console.

```
extract_map –s –mid 19991120.map > 19991120.sct
```

Extract the scatter statistics from the map file "`19991120.map`" to produce the output file "`19991120.sct`". The middle time of each record is recorded in the file.

grid_plot

Usage

```
grid_plot [--help] [-g] [-ps] [-gp] [-x]
          [-display display] [-xoff xoff]
          [-yoff yoff] [-mp] [-tn] [-dn]
          [-pathg path] [-pathp path] [-logo]
          [-fcoast] [-coast] [-term]
          [-fov] [-mlt] [-mrg]
          [-st hr:mn] [-et hr:mn]
          [-sd yyyymmdd] [-ed yyyymmdd]
          [-ex hr:mn] [-l min]
          [-sf scale] [-flip]
          [-w wdt] [-s step] [-pwr] [-wdt]
          [-avg] [-max] [-min] [-nr] [-nvc]
          [-mxv max] [-mxp max] [-mxw max]
          [-bgcol rrggbb] [-txtcol rrggbb]
          [-grdcol rrggbb] [-trmcol rrggbb]
          [-fovcol rrggbb]
          [-cstcol rrggbb] [-lndcol rrggbb]
          [-seacol rrggbb] [-key kfile]
          [-cf cfgfile] [-delay sec] file [index]
```

Options

`--help`	displays the help message.
`-g`	produce portable PixMaP (PPM) output files.
`-ps`	produce PostScript output files. This is the default operation.
`-gp`	produce both PPM and PostScript output files.
`-x`	display output on an X terminal.
`-display` *display*	connect to the X terminal with the host name *display*.
`-xoff` *xoff*	open the X terminal window *xoff* pixels from the left edge of the screen.
`-yoff` *yoff*	open the X terminal window *yoff* pixels from the top edge of the screen.
`-mp`	produce a multi-paged PostScript plot, written to standard output.
`-tn`	create filenames of the form "*hrmn.sc.xxx*", using the record time. Where *hr* is the hour, *mn* is the minutes and *sc* is the seconds. The file suffix *xxx* is either "`ps`" or "`ppm`".

-dn	create filenames of the form "*yyyymmdd.hrmn.sc.xxx*", using the record time and date. Where *yyyy* is the year, *mm* is the month, *dd* is the day, *hr* is the hour, *mn* is the minutes and *sc* is the seconds. The file suffix *xxx* is either "ps" or "ppm".
-pathg *path*	store the PPM files in the directory pointed to by *path*.
-pathp *path*	store the PostScript files in the directory pointed to by *path*.
-logo	add the SuperDARN logo and credits to the plot.
-fcoast	plot filled coastlines.
-coast	plot coastlines.
-term	plot the terminator.
-fov	plot radar fields of view.
-mlt	plot the Magnetic Local Time labels.
-mrg	merge the line of sight velocity vectors to produce true vectors.
-st *hr:mn*	start time of the data period to plot. Expressed in the form "*hr:mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-et *hr:mn*	end time of the data period to plot. Expressed in the form "*hr:mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-sd *yyyymmdd*	start date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
-ed *yyyymmdd*	end date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
-ex *hr:mn*	extent or length of time of the data period to plot. Expressed in the form "*hr:mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-l *min*	set the lower latitude limit of the plot relative to the pole. to *min* degrees.
-sf *scale*	set the scale factor of the plot to *scale*.

`-flip`	plot the mirror image of the plot. This is used when dealing with Southern Hemisphere data, to plot the map as seen through the body of the Earth.
`-w` *wdt*	set the width in pixels or points of the plot to *wdt*.
`-s` *step*	skip *step* number of records in the file between each plot.
`-pwr`	plot the lambda power for each grid cell. This option is only applicable when using extended grid files that contain this information.
`-wdt`	plot the spectral width for each grid cell. This option is only applicable when using extended grid files that contain this information.
`-avg`	plot the average value of power or spectral width in each cell. Often two or more data points will share the same grid cell. This option will plot the average value of all the data points in the cell.
`-max`	plot the maximum value of power or spectral width in each cell. Often two or more data points will share the same grid cell. This option will plot the maximum value from the data points in the cell.
`-min`	plot the average value of power or spectral width in each cell. Often two or more data points will share the same grid cell. This option will plot the minimum value from the data points in the cell.
`-nr`	do not plot raw line of sight velocity vectors. This option can be used when plotting power or spectral width to stop velocity vectors being added to the plot.
`-nvc`	do not color the line of sight velocity vectors according to magnitude. The velocity vectors will be plotted in the foreground text color.
`-mxv` *max*	set the color bar limit for the magnitude of velocity to *max*.
`-mxp` *max*	set the color bar limit for lambda power to *max*.
`-mxw` *max*	set the color bar limit for spectral width to *max*.
`-bgcol` *rrggbb*	set the background color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.

`-txtcol` *rrggbb*	set the foreground text color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-grdcol` *rrggbb*	set the grid color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-trmcol` *rrggbb*	set the terminator color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-fovcol` *rrggbb*	set the radar field-of-view color to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-cstcol` *rrggbb*	set the coastline color to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-lndcol` *rrggbb*	set the color of the land to *rggbb*. here *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-seacol` *rrggbb*	set the color of the sea to *rggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-key` *kfile*	Reads the color key from the file *kfile*.
`-cf` *cfgfile*	read command line options from the configuration file *cfgfile*. The file should contain a space-separated list of options that will be read in an parsed as if they had been included on the command line. This provides a convenient method for repeating commonly used options when generating multiple plots.
`-delay` *sec*	set the delay in seconds between each plot when displaying on an X terminal to *sec* seconds. A value of zero will wait until a mouse button is pressed before displaying the next image.
file	name of the grid file to plot
index	index of the grid file. The index is a file that identifies the location of each record in the grid file and can be used to speed up searches for a specific interval of data.

grid_plot

Description

Plots the contents of a grid file. The output can be to an X terminal, Portable PixMap (PPM) files, or PostScript files. The default output is PostScript.
If the "`-x`" option is specified the program will display plots in an X terminal window. This option can be combined with the " `-g`", "`-ps`" and "`-mp`" options to produce PostScript or PPM output files in addition to the terminal display.
The output filenames are of the form "*nnnn.xxx*", where *nnnn* is the frame number starting at 0000 and *xxx* is the suffix "`ps`" or "`ppm`". The options "`-tn`" and "`-dn`" can be used to change this format and the directories that the files are written to can be set using the "`-pathp`" and "`-pathg`" options.
The option " `-mp`" will produce a multi-page PostScript document that is written to standard output.
The program usually plots the line of sight velocity vectors contained in the grid file. However the options "`-pwr`" and "`-wdt`" can be used to plot the power and spectral width information stored in extended grid files.
The option "`-key`" allows a user defined color key to be used.
The color key file is a plain text file that defines the red green and blue components for each index in the color bar. Any line in the file beginning with a "#" is treated as a comment and ignored. The first line that is not a comment defines the number of entries in the table. The remaining lines in the file contain color values for each index, one value per line. The values are hexadecimal numbers of the form *rrggbb*, where *rr* is the red component, *gg* is the blue component and *bb* is the blue component.
The following is an example of a color key file:

```
# Color key black – red – orange - yellow.
# Sixteen entries defined, starting at black.
16
000000
200000
400000
600000
800000
a00000
c00000
e00000
ff0000
ff2000
ff4000
ff6000
ff8000
ffa000
ffc000
ffe000
ffff00
ffff00
```

The number and complexity of the command line options makes typing them a laborious process, especially when producing multiple plots. To solve this problem, command line options can be placed in plain text file that can be parsed by the program using the " `-cf`" option. This allows the user to create a set of configuration files for producing different plots.

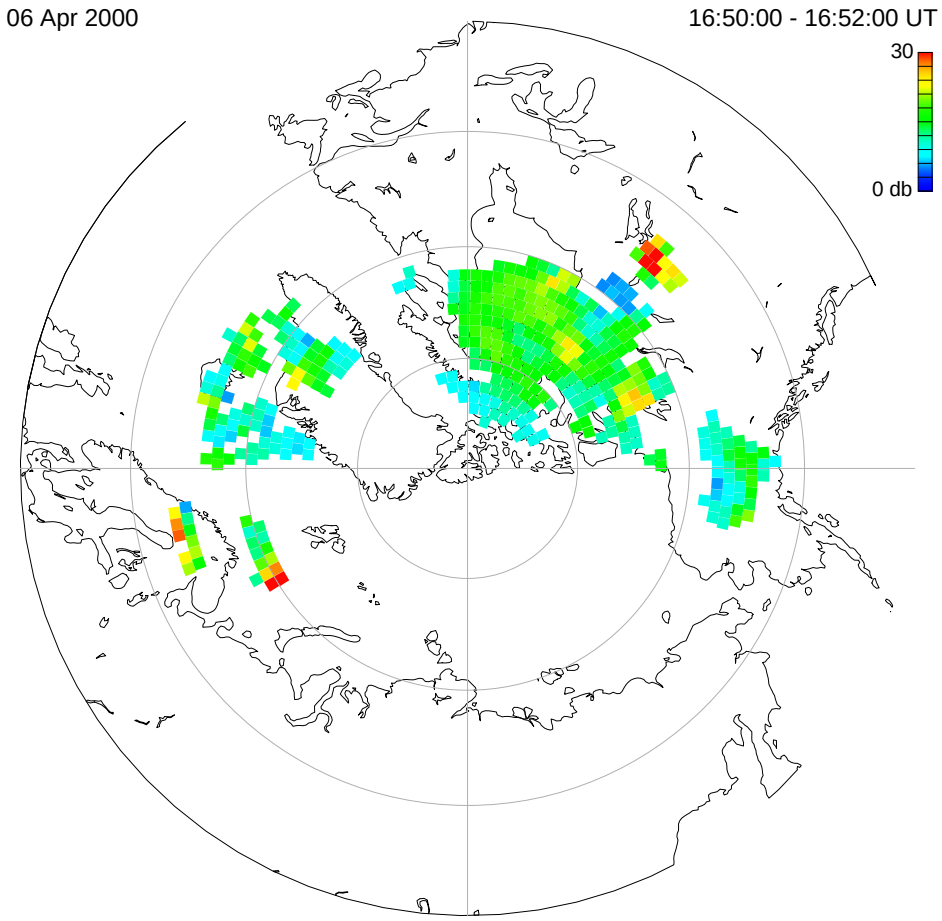
Examples

```
grid_plot -x –w 500 –fcoast –coast 19970410.grd
```

Plot the grid file "`19970410.grd`" on the X terminal using filled continents and oceans and coastlines marked in. The plot is 500 pixels wide.

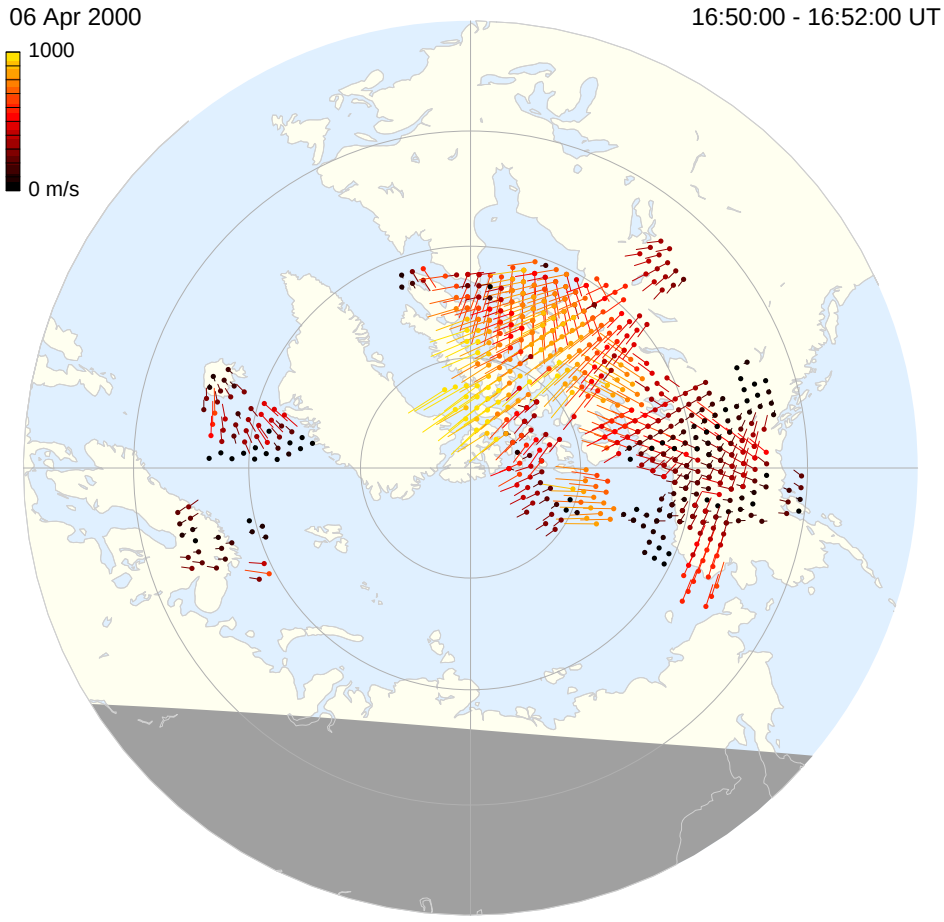
```
grid_plot –l 50 –dn -nr -pwr -avg -st 16:50 -coast -cstcol 00000
          20000406.grd
```

Generate PostScript files from the extended grid file "`20000406.grd`", without plotting velocity vectors but plotting the average power in each cell. The plots start at 16:50UT and the first file will be called "`20000406.2200.00.ps`". The coastlines are plotted in black and the lower latitude limit of the plot is 50°.



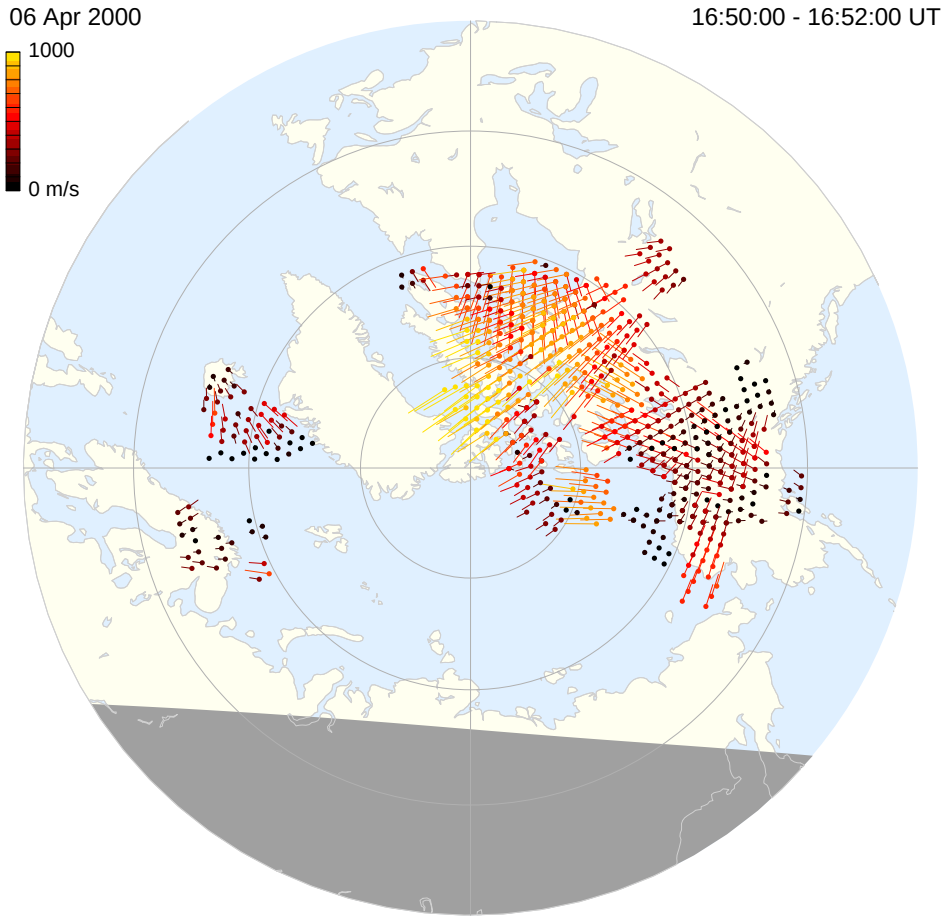
```
grid_plot –l 50 -st 16:50 –term -coast –fcoast –key broy.key
20000406.grd
```

Generate PostScript files from the grid file "`20000406.grd`", starting at 12:30UT. The plots start at 12:30UT and the first file will be called "`0000.ps`". Filled oceans and continents, together with the terminator and coastlines are plotted. The color key is taken from the file "`broy.key`".



```
grid_plot –l 50 -st 16:50 –term -coast –fcoast –key broy.key
20000406.grd
```

Generate PostScript files from the grid file "`20000406.grd`", starting at 12:30UT. The plots start at 12:30UT and the first file will be called "`0000.ps`". Filled oceans and continents, together with the terminator and coastlines are plotted. The color key is taken from the file "`broy.key`".



hmb_plot

Usage

```
hmb_plot [--help] [-vb] [-g] [-ps] [-x]
         [-display display] [-xoff xoff]
         [-yoff yoff] [-wdt wdt] [-hgt hgt]
         [-ex hr:mn] file
```

Options

`--help`	displays the help message.
`-vb`	verbose mode.
`-g`	produce portable PixMaP (PPM) output files.
`-ps`	produce PostScript output files. This is the default operation.
`-x`	display output on an X terminal.
`-display` *display*	connect to the X terminal with the host name *display*.
`-xoff` *xoff*	open the X terminal window *xoff* pixels from the left edge of the screen.
`-yoff` *yoff*	open the X terminal window *yoff* pixels from the top edge of the screen.
`-wdt` *wdt*	set the width of the plot in pixels or PostScript units to *wdt*.
`-hgt` *hgt*	set the height of the plot in pixels or PostScript units to *hgt*.
`-ex` *hr*`:`*mn*	extent or length of time of the data period to plot. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
file	name of the convection map file to plot

Description

Plots the Heppner-Maynard boundary from a convection map file. The output can be to an X terminal, Portable PixMap (PPM) files, or PostScript files. The default output is PostScript.
If the "`-x`" option is specified the program will display plots in an X terminal window. This option can be combined with the "`-g`" or "`-ps`" options to produce PostScript or PPM output files in addition to the terminal display.
The output is written to standard out.

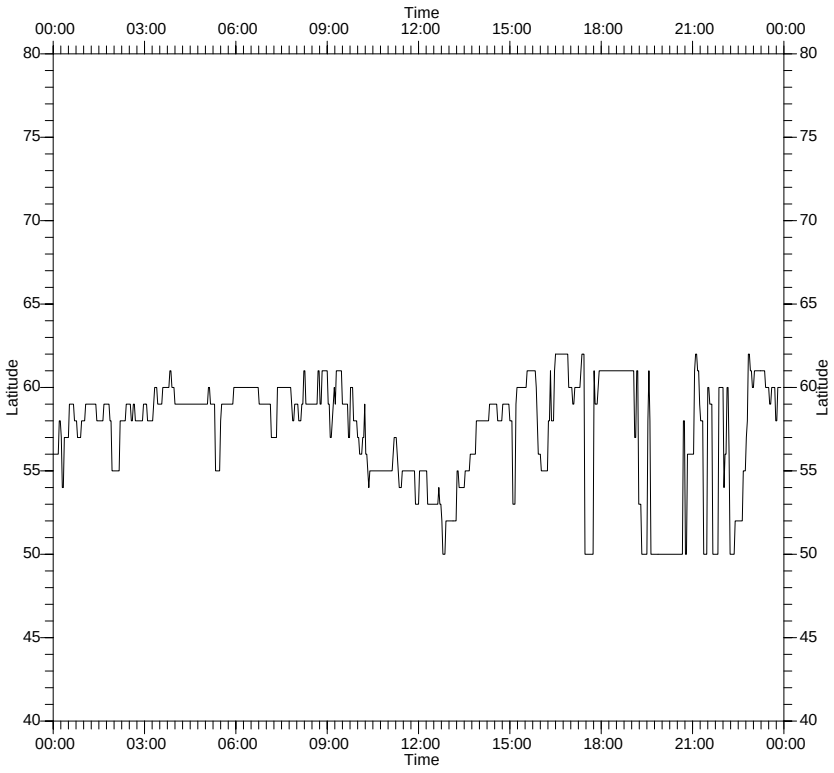
Examples

```
hmb_plot -x 19970410.map
```

Plot the HMB data from the file "`19970410.map`" on the X terminal.

```
hmb_plot 20001223.map > hmb.ps
```

Generate PostScript files from the convection map file "`20001223.map`".



index_file

Usage

```
index_file [--help] [file]
```

Options

`--help`	displays the help message.
file	name of the file to index, if none given then standard input is used.

Description

Generates an index of a data file in the universal text data format. The index is written to standard output.

The index contains the start and end times of each record, together with the file offset of the start of each record.

Examples

```
index_file test.dat > test.inx
```

Generates an index of the data file "`test.dat`" and stores it in the file "`test.inx`".

istp_plot

Usage

```
istp_plot –ace [--help] [-x]
          [-display display][-xoff xoff]
          [-yoff yoff] [-g] [-l] [-w wdt]
          [-h hgt] [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-mfi] [-swe]
          [-path path] [-cf cfgfile]
istp_plot –wind [--help] [-x]
          [-display display][-xoff xoff]
          [-yoff yoff] [-g] [-l] [-w wdt]
          [-h hgt] [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-mfi] [-swe]
          [-path path] [-cf cfgfile]
istp_plot –imp8 [--help] [-x]
          [-display display][-xoff xoff]
          [-yoff yoff] [-g] [-l] [-w wdt]
          [-h hgt] [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-mag] [-pla]
          [-path path] [-cf cfgfile]
istp_plot –geotail [--help] [-x]
          [-display display][-xoff xoff]
          [-yoff yoff] [-g] [-l] [-w wdt]
          [-h hgt] [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-mgf] [-cpi]
          [-lep] [-path path] [-cf cfgfile]
```

Options

`-ace`	plot data from the ACE spacecraft.
`--help`	displays the help message.
`-x`	display output on an X terminal. The default is to produce a PostScript file.
`-display` *display*	connect to the X terminal with the host name *display*.
`-xoff` *xoff*	open the X terminal window *xoff* pixels from the left edge of the screen.
`-yoff` *yoff*	open the X terminal window *yoff* pixels from the top edge of the screen.
`-g`	produce portable PixMaP (PPM) output files. The default

	is to produce a PostScript file.
`-l`	plot with landscape orientation.
`-w` *wdt*	set the width in pixels or points of the plot to *wdt*.
`-h` *hgt*	set the height in pixels or points of the plot to *hgt*.
`-sd` *yyyymmdd*	start date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-st` *hr*`:`*mn*	start time of the data period to plot. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes. The default time is 00:00UT.
`-ed` *yyyymmdd*	end date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-et` *hr*`:`*mn*	end time of the data period to plot Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-ex` *hr*`:`*mn*	extent or length of time of the data period to plot. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes. This option will override the end date and time if they are specified. If neither set of options are specified, 24 hours of data will be plotted.
`-gse`	plot in GSE co-ordinates. The default is to plot in GSM co-ordinates.
`-mfi`	plot data from the MFI instrument on ACE or Wind.
`-swe`	plot data from the SWE instrument on ACE or Wind.
`-path` *path*	set the pathname of the directories containing the data to *path*. The individual satellite data files are stored in the sub-directories named "`ace`", "`wind`", "`imp8`" and "`geotail`".
`-cf` *cfgfile*	read command line options from the configuration file *cfgfile*. The file should contain a space-separated list of options that will be read in an parsed as if they had been included on the command line. This provides a convenient method for repeating commonly used options when generating multiple plots.

`-wind`	plot data from the Wind spacecraft.
`-imp8`	plot data from the IMP8 spacecraft.
`-mag`	plot data from the MAG instrument on IMP8.
`-pla`	plot data from the PLA instrument on IMP8.
`-geotail`	plot data from the Geotail spacecraft.
`-mag`	plot data from the MGF instrument on Geotail.
`-cpi`	plot data from the CPI instrument on Geotail.
`-lep`	plot data from the LEP instrument on Geotail.

Description

Plot ISTP magnetic field and solar wind data from a set of CDF files. The output can be to an X terminal, Portable PixMap (PPM) files, or PostScript files. The default output is PostScript. The PostScript and PPM files are written to standard output.
The program usually plots 24 hours of magnetic field data in GSM co-ordinates for a given start date and satellite.
Magnetic field and solar wind data can be plotted together by combining the appropriate options.
The data files are taken from the sub-directories "`ace`", "`wind`", "`imp8`" and "`geotail`", of the path given by the "`-path`" option or by the environment variable *ISTP_PATH*.

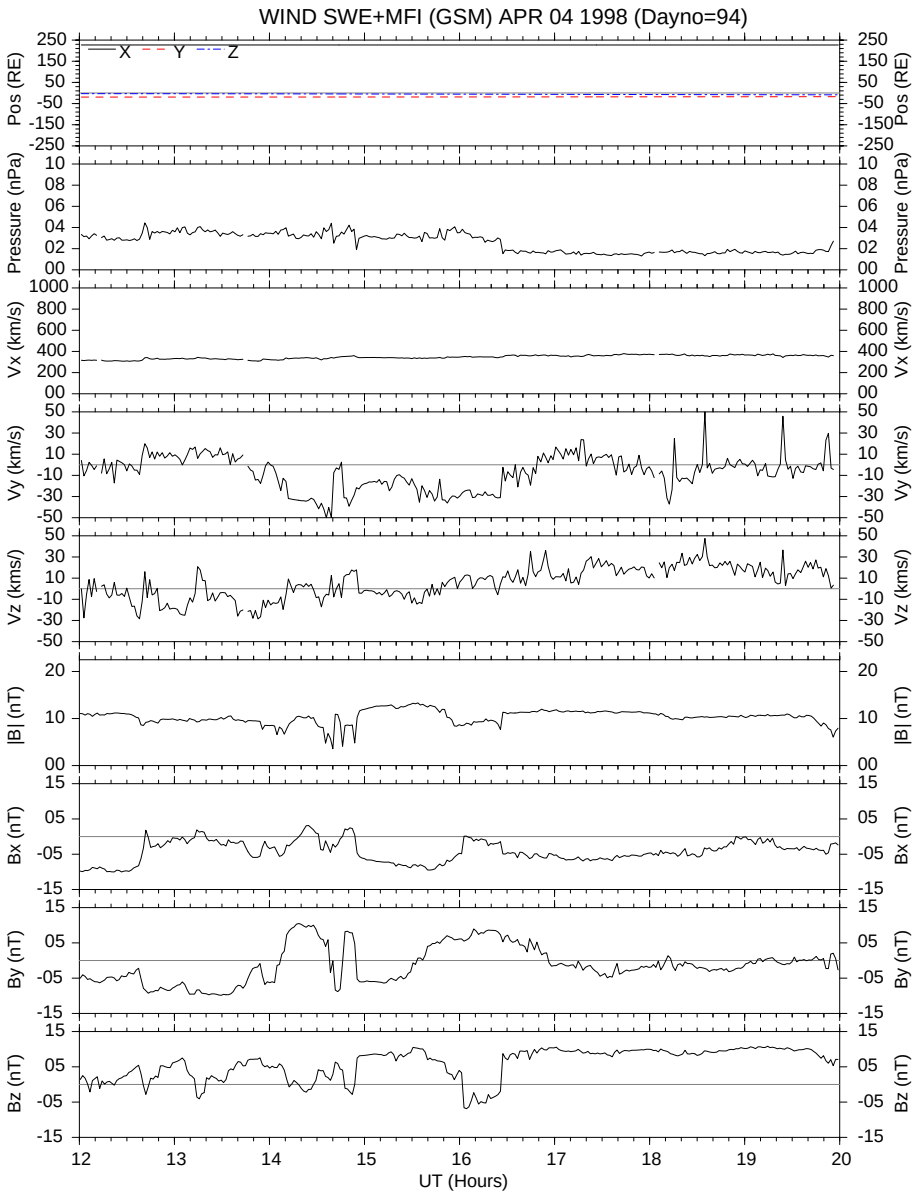
Examples

```
istp_plot -x  -ace -sd 19990406
```

Plot 24 hours of ACE data starting at 00:00UT on April, 6 1999 on the X terminal.

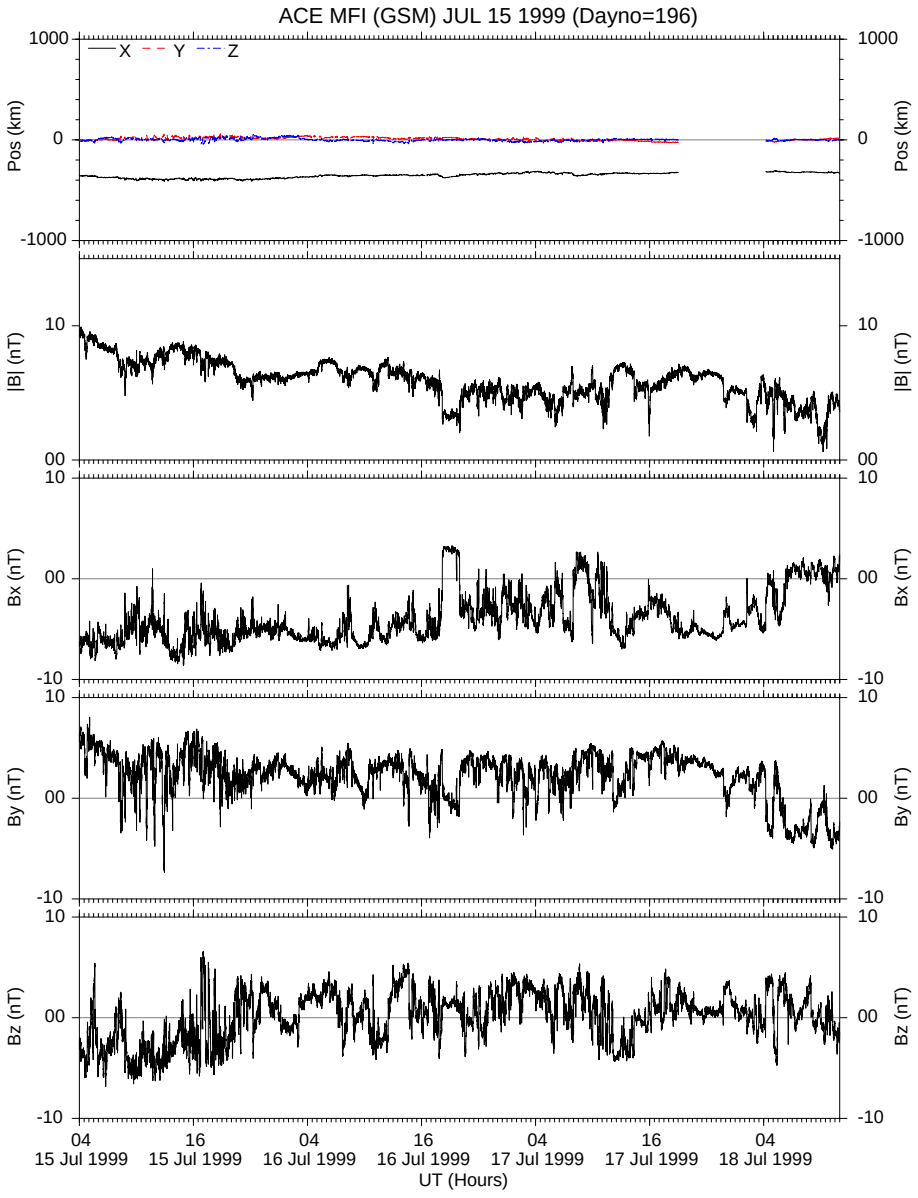
```
istp_plot -wind  -mfi -swe -sd 19980404  -st 12:00 -ex 8:00 >
          plot.ps
```

Generate a PostScript plot of Wind MFI and SWE data for the 8 hour period starting at 12:00UT on April 4, 1998. The plot is stored in the file "`plot.ps`".



```
istp_plot -ace  -mfi -sd 19990715 -st 4:00 -ed 19990718 -et 12:00 >
          plot.ps
```

Generate a PostScript plot of ACE MFI data for the period starting at 04:00UT July 15, 1999 and ending at 12:00UT July 18, 1999. The plot is stored in the file "`plot.ps`".



istp_text

Usage

```
istp_text –ace [--help] [-h]
          [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-pos] [-mfi]
          [-swe] [-path path] [-cf cfgfile]
istp_text –wind [--help] [-h]
          [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-pos] [-mfi]
          [-swe] [-path path] [-cf cfgfile]
istp_text –imp8 [--help] [-h]
          [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-pos] [-mag]
          [-pla] [-path path] [-cf cfgfile]
istp_text –geotail [--help] [-h]
          [-sd yyyymmdd] [-st hr:mn]
          [-ed yyyymmdd] [-et hr:mn]
          [-ex hr:mn] [-gse] [-pos] [-mgf]
          [-cpi] [-lep] [-path path]
          [-cf cfgfile]
```

Options

`-ace`	use data from the ACE spacecraft.
`--help`	displays the help message.
`-h`	include a header at the start of the file labeling the columns.
`-sd` *yyyymmdd*	start date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-st` *hr*`:`*mn*	start time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes. The default time is 00:00UT.
`-ed` *yyyymmdd*	end date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-et` *hr*`:`*mn*	end time of the data period to process Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.

`-ex` *hr*`:`*mn*	extent or length of time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes. This option will override the end date and time if they are specified. If neither set of options are specified, 24 hours of data will be processed.
`-gse`	use GSE co-ordinates. The default is to use GSM co-ordinates.
`-mfi`	use data from the MFI instrument on ACE or Wind.
`-swe`	use data from the SWE instrument on ACE or Wind.
`-path` *path*	set the pathname of the directories containing the data to *path*. The individual satellite data files are stored in the sub-directories named "`ace`", "`wind`", "`imp8`" and "`geotail`".
`-cf` *cfgfile*	read command line options from the configuration file *cfgfile*. The file should contain a space-separated list of options that will be read in an parsed as if they had been included on the command line. This provides a convenient method for repeating commonly used options when generating multiple files.
`-wind`	use data from the Wind spacecraft.
`-imp8`	use data from the IMP8 spacecraft.
`-mag`	use data from the MAG instrument on IMP8.
`-pla`	use data from the PLA instrument on IMP8.
`-geotail`	use data from the Geotail spacecraft.
`-mag`	use data from the MGF instrument on Geotail.
`-cpi`	use data from the CPI instrument on Geotail.
`-lep`	use data from the LEP instrument on Geotail.

Description

Generates a plain ASCII text file containing ISTP magnetic field and solar wind data from a set of CDF files. The file is written to standard output.
The program usually produces 24 hours of magnetic field data in GSM co-ordinates for a given start date and satellite.
The data files are taken from the sub-directories "`ace`", "`wind`", "`imp8`" and "`geotail`", of the path given by the "`-path`" option or by the environment variable *ISTP_PATH*.

Example

```
istp_text -ace -sd 19981112 > mfi.txt
```

Generate a text file containing 24 hours of ACE MFI data starting at 00:00UT on November 12, 1998. The output is stored in the file "`mfi.txt`"

```
istp_text -wind -mfi -swe -pos -sd 19971014 -st 4:00 -ex 8:00 >
          19981112.wnd.txt
```

Generate a text file containing 8 hours of Wind MFI, SWE and position data starting at 04:00UT October 14, 1997. The output is stored in the file "`19981112.wnd.txt`".

```
istp_text -ace -mfi -pos -sd 19990406 -st 6:00 -ed 19990408
          -et 12:00 > mfi+pos.txt
```

Generate a text file containing ACE MFI and Position data, starting at 06:00UT April 6, 1999 and ending at 12:00UT April 8,1999. The output is stored in the file "`mfi+pos.txt`".

make_grid

Usage

```
make_grid [--help] [-vb] [-st hr:mn]
          [-et hr:mn] [-sd yyyymmdd]
          [-ed yyyymmdd] [-ex hr:mn] [-i sec]
          [-tl sec] [-fwgt wgt] [-pmax max]
          [-pmin min] [-vmax max] [-vmin min]
          [-wmax max] [-wmin min]
          [-vemax max] [-vemin min] [-ion]
          [-gs] [-both] [-nav] [-nlm] [-nb]
          [-xtd] [-inertial] [-cn A|B]
          [-minrng rng]
          [-ebm bm,…] fitfile [inxfile]
make_grid –c [--help] [-vb] [-st hr:mn]
          [-et hr:mn] [-sd yyyymmdd]
          [-ed yyyymmdd] [-ex hr:mn] [-i sec]
          [-tl sec] [-fwgt wgt] [-pmax max]
          [-pmin min] [-vmax max] [-vmin min]
          [-wmax max] [-wmin min]
          [-vemax max] [-vemin min] [-ion]
          [-gs] [-both] [-nav] [-nlm] [-nb]
          [-xtd] [-inertial] [-cn A|B]
          [-minrng rng]
          [-ebm bm,…] fitfiles…
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-st` *hr*`:`*mn*	start time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-et` *hr*`:`*mn*	end time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-sd` *yyyymmdd*	start date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ed` *yyyymmdd*	end date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ex` *hr*`:`*mn*	extent or length of time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.

`-i` *sec*	sets the time interval to store in each record to *sec* seconds. The default is 120 seconds or 2 minutes.
`-tl` *sec*	causes the program to ignore the scan flag in the fit files and instead use a fixed scan length of *sec* seconds. The scan boundary is aligned with the start of the day.
`-fwgt` *wgt*	set the median filter weighting to *wgt*. A value of zero disables the filter.
`-pmax` *max*	set the upper limit for the lambda power to *max*. Data points in the fit file with lambda power that exceed this threshold will be ignored.
`-pmin` *min*	set the lower limit for the lambda power to *min*. Data points in the fit file with lambda power below this threshold will be ignored.
`-vmax` *max*	set the upper limit for the velocity magnitude to *max*. Data points in the fit file with velocity magnitude that exceed this threshold will be ignored.
`-vmin` *min*	set the lower limit for the velocity magnitude to *min*. Data points in the fit file with velocity magnitude below this threshold will be ignored.
`-wmax` *max*	set the upper limit for the spectral width to *max*. Data points in the fit file with spectral width that exceed this threshold will be ignored.
`-wmin` *min*	set the lower limit for the spectral width to *min*. Data points in the fit file with spectral width below this threshold will be ignored.
`-vemax` *max*	set the upper limit for the velocity error to *max*. Data points in the fit file with velocity error that exceed this threshold will be ignored.
`-vemin` *min*	set the lower limit for the velocity error to *min*. Data points in the fit file with velocity error below this threshold will be ignored.
`-cn A|B`	filter based on the Stereo channel, either A or B.
`-minrng` *rng*	exclude scatter below this range gate.
`-ion`	process only those vectors that are classed as ionospheric scatter. This is the default operation.
`-gs`	process only those vectors that are classed as ground scatter.

`-both`	process both ionospheric and ground scatter vectors.
`-nav`	do not perform temporal filtering. Usually three consecutive scans are used in the median filter. However this can obscure rapid transitions in the data. This option forces the median filter to operate on only a single scan.
`-nlm`	do not apply limits to changes in radar parameters between scans. When the radar parameters such as range separation or frequency change, the location of vectors in adjacent scans will change. These scans are normally ignored, as the median filter should only be applied to scans with similar operating parameters. This option disables this behavior and includes all scans in the analysis.
`-nb`	do not apply the bounding threshold to lambda power, velocity, spectral width or velocity error.
`-xtd`	generate extended files that contain lambda power and spectral width information.
`-intertial`	generate grid using an inertial reference frame. (The rotation of the earth is factored into the calculation of the velocities).
`-ebm` *bm,...*	exclude the comma separated list of beams starting with *bm* from the analysis.
fitfile	the name of the fit file to process.
inxfile	the name of the optional index file associated with the fit file. The index file speeds up the location of records in the fit file. It is only useful to include this file when an interval in the middle of a fit file is being processed.
`-c`	concatenate multiple fit files together for processing. Fit files usually contain only two hours of data and this option avoids the need to separately concatenate the files together before they are processed.
fitfiles...	a list of fit files to concatenate together for processing.

Description

Generates a grid file from one or more fit files. A grid file is a highly processed data product consisting of geo-magnetically located line of sight velocity vectors.
The algorithm optionally applies a median filter to the scan data to remove noise. Each range-beam cell together with its immediate neighbors in the current, preceding and following scans is examined. A weighted sum of all the cells containing scatter is calculated and if this sum exceeds a certain threshold, the median data value of the cells is substituted for the central cell. Various command line options control how the filter is applied.
Once the data has been filtered, the geo-magnetic location of each line of sight velocity measurement is calculated. The vectors are then fixed to an equi-area grid to ensure that the data is not biased according to its location in the radar field of view.

The vectors in each cell are averaged together over a fixed period of time to generate a data record, which is then written to standard output.
The program operates in two modes. The first operates on a single fit file. The second, specified by the "`-c`" option will concatenate multiple fit files together for processing.
In addition to the regular grid file output, the program can also produce "extended" grid files that contain information about the spectral width, power and composition of each data point by specifying the "`-xtd`" command line option.

Examples

```
make_grid -vb 1999112012k.fit > 1999112012k.grd
```

Generate a grid file from the fit file "`1999112012k.fit`" and store it in the file "`1999112012k.grd`". Report the status on standard error.

```
make_grid -c -i 240 20000510*a.fit > 20000510a.grd
```

Concatenate all the fit files in the current directory to create a grid file with a 4-minute record length. Store the output in the file "`20000510a.grd`".

```
make_grid –tl 120 –vemax 500 -both  -xtd
          1998101200g.fit > 1998101200g.grd
```

Generate a grid file from the fit file "`1998101200g.fit`" using a fixed scan length of 120 seconds. Ignore date points with a velocity error exceeding 500 m/s and process both ionospheric and ground scatter vectors. Store an extended format grid file in "`1998101200g.grd`"

```
make_grid   –nb –nlm 1997081012k.fit > 1997081012k.grd
```

Generate a grid file from the fit file "`1997081012k.fit`" without applying any thresholds to the vectors and any changes in radar parameters between scans, are ignored. Store the grid file in "`1997081012k.grd`"

map_addhmb

Usage

```
map_addhmb [--help] [-vb] [-vel vel]
           [-cnt cnt] [-ex id,…] [file]
map_addhmb -l latmin [--help] [-vb] [file]
map_addhmb -lf latfile [--help] [-vb] [file]
map_addhmb –t [-vel vel] [-cnt cnt] [--help]
           [-vb] [-ex id,…] [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-vel` *vel*	set the lower limit for the velocity magnitude to *vel*. For a point to be considered in the analysis it must have a velocity magnitude in excess of *vel*. The default value is 100 m/s.
`-cnt` *cnt*	set the minimum number of points required to determine the boundary to *cnt*. There must be at least this number of points lying along a test boundary for it to be accepted. The default value is 3.
`-ex` *id,…*	when calculating the location of the boundary, exclude data from stations whose identifier numbers match the comma separated list starting with *id*.
file	map file to process. If none is specified then standard input will be used.
`-l` *latmin*	set the Heppner-Maynard Boundary so that at magnetic local midnight it has a latitude of *latmin*.
`-lf` *latfile*	read latitudes of the Heppner-Maynard Boundary at magnetic local midnight from the file *latfile*.
`-t`	generate a list of latitudes of the boundary at magnetic local midnight rather than adding the boundary to the convection map file.

Description

Adds a Heppner-Maynard boundary to a convection map file, or generates a data file containing the latitudes of the boundary at magnetic local midnight for each record in the file. The file is written to standard output.
The default operation is to calculate a possible position of the Heppner-Maynard boundary from the line-of-sight velocity data in the convection map file. The two options "`-vel`" and "`-cnt`" adjust the algorithm. A boundary determination is made for each record in the map file. This is median filtered using the previous and subsequent records to reduce rapid fluctuations in the boundary. The median filtered boundary determination is then used to generate zero velocity model vectors that are

added to the map file to constrain the convection pattern. The location of the boundary is also stored.
If the "`-l`" option is specified, the location of the boundary is fixed so that at magnetic local midnight it has a latitude of *latmin*.
The "`-lf`" option will read the latitude of the boundary at magnetic local midnight from the plain text file *latfile*. Any lines in the file beginning the character "#" are treated as comments and ignored. Any other lines are expected to contain a time followed by two latitudes of the boundary at magnetic local midnight:

```
year month day hour minut second median actual
```

The two values correspond to a filtered and actual value of the latitude. Only the filtered value, *median*, is used to select the boundary for the map file. The boundary will be fixed at this value starting at the time specified and will only change if a subsequent entry in the boundary files alters it.
The "`-t`" option will generate a text file containing the latitude of the boundary at magnetic local midnight for each record in the map file. Each line of the output file contains the date and time of the start of the record followed by the median filtered boundary determination and the actual boundary determination of the record:

```
year month day hour minut second median actual
```

Examples

```
map_addhmb -vb19980410.map > 19980410.hmb.map
```

Locate the Heppner-Maynard boundary for the map file "`19980410.map`". The output is written to the file "`19980410.hmb.map`" and status is logged to standard error.

```
map_addhmb -l 64 19970415.map > 19970415.hmb.map
```

Add vectors to the map file "`19970415.map`" for a Heppner-Maynard boundary that crosses 64° at magnetic local midnight. The output is written to the file "`19970415.hmb.map`".

```
map_addhmb -lf lat.dat 19990830.map > 19980830.hmb.map
```

Add vectors to the map file "`19990830.map`" taking latitudes for the Heppner-Maynard boundary from the file "`lat.dat`". The output is written to the file "`19990830.map`"

```
map_addhmb -t –vel 150 -cnt 4  20000410.map > lat.dat
```

Generate a list of latitudes from the map file "`20000410.map`". Set the minimum velocity magnitude to 150 m/s and the minimum number of points to 4. The output is written to "`lat.dat`".

map_addimf

Usage

```
map_addimf [--help] [-vb
           [-bx x] [-by y] [-bz z]
           [file]
map_addimf –ace [--help] [-vb] [-d hr:mn]
           [-df dfile] [-bx x] [-by y] [-bz z]
           [-ex hr:mn] [-path path] [file]
map_addimf –wind [--help] [-vb] [-d hr:mn]
           [-df dfile] [-bx x] [-by y] [-bz z]
           [-ex hr:mn] [-path path] [file]
map_addimf –if iname [--help] [-vb] [-d hr:mn]
           [-df dfile] [-bx x] [-by y] [-bz z]
           [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-bx` *x*	set the initial IMF Bx component to *x*.
`-bx` *x*	set the initial IMF By component to *y*.
`-bx` *z*	set the initial IMF Bz component to *z*.
file	map file to process. If none is specified then standard input will be used.
`-ace`	use IMF data from the ACE spacecraft.
`-d` *hr:mn*	delay time to apply to the IMF. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-df` *dfile*	read the IMF delays from the text file specified by *dfile*. The file contains the IMF delay to apply at various times.
`-ex` *hr:mn*	expected length of the map file. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes. This is used to determine how much IMF data should be loaded. The default is 24 hours.
`-path` *path*	set the pathname of the directories containing the data to *path*. The individual satellite data files are stored in the sub-directories named "`ace`" and "`wind`".
`-wind`	use IMF data from the Wind spacecraft.

`-if` *ifile* read IMF data from the text file *iname*. The file contains the three components of the IMF in GSM coordinates defined at various times.

Description

Adds IMF data to a convection map file. The IMF applied to the map file can be fixed, taken from the ACE or Wind spacecraft, or read from a plain text file. The processed file is written to standard output.
If the "`-ace`" or "`-wind`" options are specified, the IMF data is taken from the appropriate CDF files for each spacecraft. The files are read from the "`ace`" and "`wind`" sub-directories of the path given by given by the "`-path`" option or by the environment variable "`ISTP_PATH`". The argument "`-ex`" is used to specify how much IMF data should be loaded. By default only 24 hours of data is read.
The IMF delay can either be fixed using the "-d" option or read from a text file using the "`-df`" option. Any lines in the file beginning the character "#" are treated as comments and ignored. Any other lines are expected to contain a time followed by the delay in hours and minutes:

```
year month day hour minut second dhour dminute
```

The delay will be applied to the IMF starting at the time specified and will only change if a subsequent entry in the delay file alters it.
The "`-if`" option will read the IMF from the plain text file *ifile*. Any lines in the file beginning the character "#" are treated as comments and ignored. Any other lines are expected to contain a time followed by the three components of the IMF defined in GSM coordinates.

```
year month day hour minut second bx by bz
```

The IMF will be fixed at this value and will only change if a subsequent entry the IMF file alters it.

Examples

```
map_addimf –vb –bx 1.5 –by -1.2 –bx 0.4 19970406.map >
           19970406.imf.map
```

Adds a fixed IMF of Bx=-15, By=-1.2 and Bz=0.4 to the map file "`19970406.map`". The output is stored in the file "`19970406.imf.map`" and status is logged to standard error.

```
map_addimf –ace –d 0:30 –ex 48:00 19981104.map > 19981104.imf.map
```

Adds IMF data from the ACE spacecraft to the map file "`19981104.map`". A delay of 30 minutes is applied to the data and the map file is expected to be 48 hours in length. The output is stored in the file "`19981104.imf.map`"

```
map_addimf –ace –df delay.txt  19990712.map > 19990712..imf.map
```

Adds IMF data from the ACE spacecraft to the map file "`19990712.map`". The IMF delays are read from the file "`delay.txt`". The output is stored in the file "`19990712.imf.map`"

```
map_addimf –if imf.txt –df delay.txt 2000312.map > 20000312.map
```

Adds IMF data from the text file "`imf.txt`" to the map file "`2000312.map`". The IMF delays are taken from the file "`delay.txt`". The output is stored in the file "`19990712.imf.map`".

map_addmodel

Usage

```
map_addmodel [--help] [-vb] [-o order]
             [-d l|m|h|e] [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-o` *latmin*	set the order of the fit to *order*. The default is 4.
`-d l|m|h|e`	set the doping level to (l)ow, (m)edium, (h)eavy, or (e)extreme. The default is light.
file	map file to process. If none is specified then standard input will be used.

Description

Adds model vectors to a convection map file. The file created is written to standard output.
The input map file must contain valid IMF data.

Examples

```
map_addmodel –d l –o 8 -vb 19981020.map > 19981020.model.map
```

Adds model vectors to the map file called "`19981020.map`". The order of fit is set to 8 and the doping level to light. The file created is called "`19981020.model.map`" and status is logged to standard error.

map_cnv

Usage

```
map_cnv [--help] [-vb] [-st hr:mn]
        [-et hr:mn] [-sd yyyymmdd]
        [-ed yyyymmdd] [-ex hr:mn]
        [-mlt] [-p] [-ef] [-v] [file] [index]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-st` *hr*`:`*mn*	start time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-et` *hr*`:`*mn*	end time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-sd` *yyyymmdd*	start date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ed` *yyyymmdd*	end date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ex` *hr*`:`*mn*	extent or length of time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-mlt`	express position in terms of Magnetic Local Time, rather than geo-magnetic longitude.
`-l` *min*	set the lower latitude limit of the grid relative to the pole. to *min* degrees.
`-p`	generate electrostatic potential.
`-ef`	generate northern and eastern electric field components.
`-v`	generate velocity magnitude and azimuth.
file	map file to process. If none is specified then standard input will be used.
index	index of the map file. The index is a file that identifies the location of each record in the map file and can be used to speed up searches for a specific interval of data.

map_cnv

Description

Generates a grid of longitude and latitude containing derived data from each record in a convection map file. The grid spacing is 2 degrees of longitude and 1 degree in latitude. Depending on which options are specified, the program will calculate the potential, electric field and velocity for each grid cell. It is also determined if the cell contains a line-of sight velocity vector.
The grid is written to standard output as a plain text file.

Examples

```
map_cnv -vb –st 11:00 –et 14:00 –p 19981020.map > 1998102011.dat
```

Extracts a 3-hour period starting at 11:00UT from the file called "`19981020.map`" to produce a file called "`1998102011.dat`". The file contains the potential calculated from the convection map file. The status is logged to standard error.

```
map_cnv –sd 19991121 –st 22:00 –ex 4:00 –p –v –mlt 199911.map >
         19991121.dat
```

Extracts a 4-hour period starting at 22:00UT on November 21, 1999 from the file "`199911.map`" to produce the output file "`19991121.dat`". The file contains the potential and the velocity components calculated from the convection map file. The location of the grid cells is expressed in terms of Magnetic Local Time.

map_fit

Usage

```
map_fit [--help] [-vb] [-ew y|n] [-mw f|n]
        [-s source] [-major ver] [-minor ver]
        [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-ew y|n`	error weighting, (y)es or (n)o. the default is yes.
`-mw f|n`	model weighting, (f)ixed or (n)ormalized. The default is normalized.
`-s` *source*	Overrides the text string embedded in the map file that indicates the source of the file. (Use of this option is not advised).
`-major` *ver*	Overrides the major version number embedded in the map file. (Use of this option is not advised).
`-minor` *ver*	Overrides the minor version number embedded in the map file. (Use of this option is not advised).
file	map file to process. If none is specified then standard input will be used.

Description

Performs spherical harmonic fitting on a convection map file. The file created is written to standard output.
The input map file must contain valid model data to ensure that the fit converges.

Examples

```
map_fit –ew y –mw n 19981020.map > 19981020.shf.map
```

Performs spherical harmonic fitting on the map file called "`19981020.map`". Errors are weighted and model weighting is set to normalized. The file created is called "`19981020.shf.map`" and status is logged to standard error.

map_grd

Usage

```
map_grd [--help] [-vb] [-l latmin] [file]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-l` *latmin*	set the lower latitude limit to *latmin*.
file	grid file to process. If none is specified then standard input will be used.

Description

Creates an empty convection map file from a grid file. The file created is written to standard output.
The output is in the map file format but most of the data fields are empty. Subsequent processing is required to add such things as the IMF data, Model vectors and coefficients of the spherical harmonic fit.

Examples

```
map_grd –l 60 -vb 19981020.grd > 19981020.map
```

Creates an empty map file from the grid file called "`19981020.grd`". The lower latitude limit is set to 60°. The file created is called "`19981020.map`" and status is logged to standard error.

map_plot

Usage

```
map_plot [--help] [-g] [-ps] [-gp] [-x]
         [-display display] [-xoff xoff]
         [-yoff yoff] [-mp] [-tn] [-dn]
         [-pathg path] [-pathp path] [-source]
         [-logo] [-fcoast] [-coast] [-term]
         [-fov] [-mlt]
         [-st hr:mn] [-et hr:mn]
         [-sd yyyymmdd] [-ed yyyymmdd]
         [-ex hr:mn] [-l min]
         [-sf scale] [-flip]
         [-w wdt][-s step] [-raw] [-model]
         [-bgcolor rrggbb] [-txtcol rrggbb]
         [-grdcol rrggbb] [-trmcol rrggbb]
         [-fovcol rrggbb] [-bndcol rrggbb]
         [-cstcol rrggbb] [-lndcol rrggbb]
         [-seacol rrggbb] [-key kfile]
         [-cf cfgfile] [-delay sec] file index
```

Options

`--help`	displays the help message.
`-g`	produce portable PixMaP (PPM) output files.
`-ps`	produce PostScript output files. This is the default operation.
`-gp`	produce both PPM and PostScript output files.
`-x`	display output on an X terminal.
`-display` *display*	connect to the X terminal with the host name *display*.
`-xoff` *xoff*	open the X terminal window *xoff* pixels from the left edge of the screen.
`-yoff` *yoff*	open the X terminal window *yoff* pixels from the top edge of the screen.
`-mp`	produce a multi-paged PostScript plot, written to standard output.
`-tn`	create filenames of the form "*hrmn.sc.xxx*", using the record time. Where *hr* is the hour, *mn* is the minutes and *sc* is the seconds. The file suffix *xxx* is either "`ps`" or "`ppm`".

-dn	create filenames of the form "*yyyymmdd.hrmn.sc.xxx*", using the record time and date. Where *yyyy* is the year, *mm* is the month, *dd* is the day, *hr* is the hour, *mn* is the minutes and *sc* is the seconds. The file suffix *xxx* is either "ps" or "ppm".
-pathg *path*	store the PPM files in the directory pointed to by *path*.
-pathp *path*	store the PostScript files in the directory pointed to by *path*.
-source	add the source of the data as indicated in the file to the plot..
-logo	add the SuperDARN logo and credits to the plot.
-fcoast	plot filled coastlines.
-coast	plot coastlines.
-term	plot the terminator.
-fov	plot radar fields of view.
-mlt	plot the Magnetic Local Time labels.
-st *hr*:*mn*	start time of the data period to plot. Expressed in the form "*hr*:*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-et *hr*:*mn*	end time of the data period to plot. Expressed in the form "*hr*:*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-sd *yyyymmdd*	start date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
-ed *yyyymmdd*	end date of the data period to plot. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
-ex *hr*:*mn*	extent or length of time of the data period to plot. Expressed in the form "*hr*:*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
-l *min*	set the lower latitude limit of the plot relative to the pole. to *min* degrees.
-sf *scale*	set the scale factor of the plot to *scale*.

`-flip`	plot the mirror image of the plot. This is used when dealing with Southern Hemisphere data, to plot the map as seen through the body of the Earth.
`-w` *wdt*	set the width in pixels or points of the plot to *wdt*.
`-s` *step*	skip *step* number of records in the file between each plot.
`-raw`	plot raw line-of-sight velocity vectors rather than fitted vectors.
`-model`	plot the model vectors.
`-bgcolor` *rrggbb*	set the background color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-txtcol` *rrggbb*	set the foreground text color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-grdcol` *rrggbb*	set the grid color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-grdcol` *rrggbb*	set the terminator color of the plot to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-fovcol` *rrggbb*	set the radar field-of-view color to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-bndcol` *rrggbb*	set the Heppner-Maynard Boundary color to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-cstcol` *rrggbb*	set the coastline color to *rrggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-lndcol` *rrggbb*	set the color of the land to *rggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-seacol` *rrggbb*	set the color of the sea to *rggbb*. Where *rr*, *gg*, and *bb* define the red, green and blue components of the color in hexadecimal.
`-key` *kfile*	Reads the color key from the file *kfile*.

`-cf` *cfgfile*	read command line options from the configuration file *cfgfile*. The file should contain a space-separated list of options that will be read in an parsed as if they had been included on the command line. This provides a convenient method for repeating commonly used options when generating multiple plots.
`-delay` *sec*	set the delay in seconds between each plot when displaying on an X terminal to *sec* seconds. A value of zero will wait until a mouse button is pressed before displaying the next image.
file	name of the convection map file to plot
index	index of the grid file. The index is a file that identifies the location of each record in the grid file and can be used to speed up searches for a specific interval of data.

Description

Plots the contents of a convection map file. The output can be to an X terminal, Portable PixMap (PPM) files, or PostScript files. The default output is PostScript.
If the "`-x`" option is specified the program will display plots in an X terminal window. This option can be combined with the " `-g`", "`-ps`" and "`-mp`" options to produce PostScript or PPM output files in addition to the terminal display.
The output filenames are of the form "*nnnn.xxx*", where *nnnn* is the frame number starting at 0000 and *xxx* is the suffix "`ps`" or "`ppm`". The options "`-tn`" and "`-dn`" can be used to change this format and the directories that the files are written to can be set using the "`-pathp`" and "`-pathg`" options.
The option " `-mp`" will produce a multi-page PostScript document that is written to standard output.
The option "`-key`" allows a user defined color key to be used.
The color key file is a plain text file that defines the red green and blue components for each index in the color bar. Any line in the file beginning with a "#" is treated as a comment and ignored. The first line that is not a comment defines the number of entries in the table. The remaining lines in the file contain color values for each index, one value per line. The values are hexadecimal numbers of the form *rrggbb*, where *rr* is the red component, *gg* is the blue component and *bb* is the blue component.
The following is an example of a color key file:

```
# Color key black – red – orange - yellow.
# Sixteen entries defined, starting at black.
16
000000
200000
400000
600000
800000
a00000
c00000
e00000
ff0000
ff2000
ff4000
ff6000
ff8000
ffa000
ffc000
ffe000
ffff00
ffff00
```

The number and complexity of the command line options makes typing them a laborious process, especially when producing multiple plots. To solve this problem, command line options can be placed in plain text file that can be parsed by the program using the "`-cf`" option. This allows the user to create a set of configuration files for producing different plots.

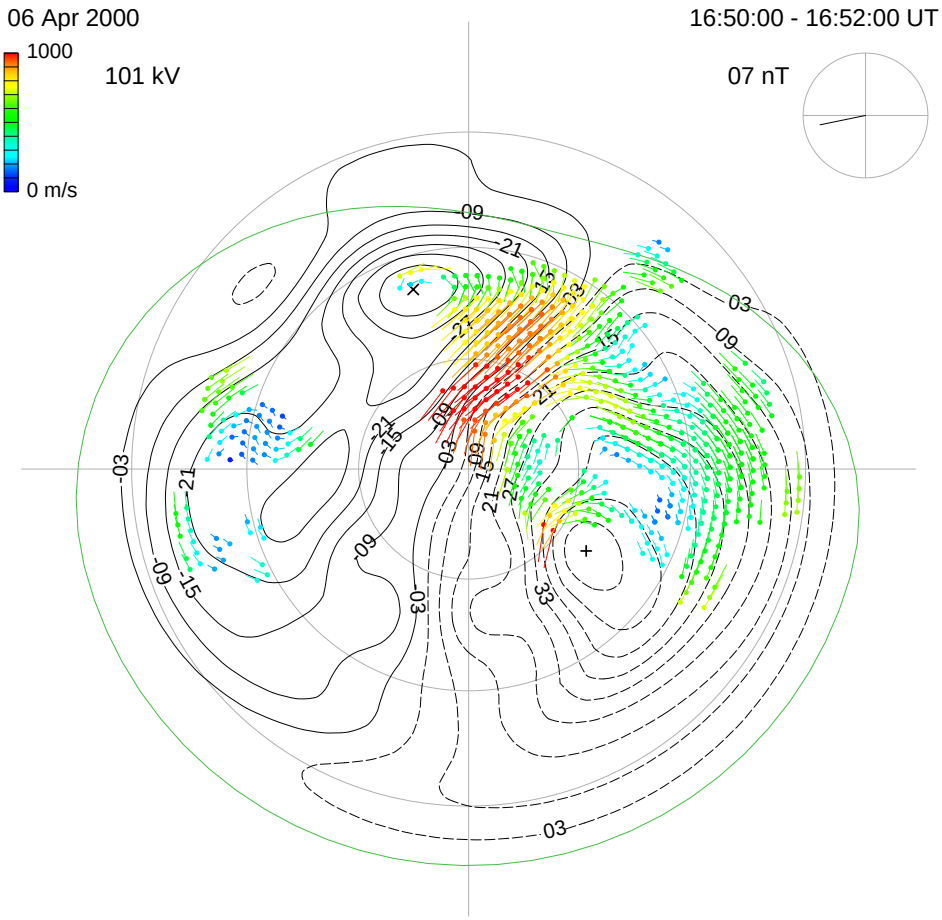
Examples

```
map_plot -x –w 500 –fcoast –coast 19970410.map
```

Plot the grid file "`19970410.grd`" on the X terminal using filled continents and oceans and coastlines marked in. The plot is 500 pixels wide.

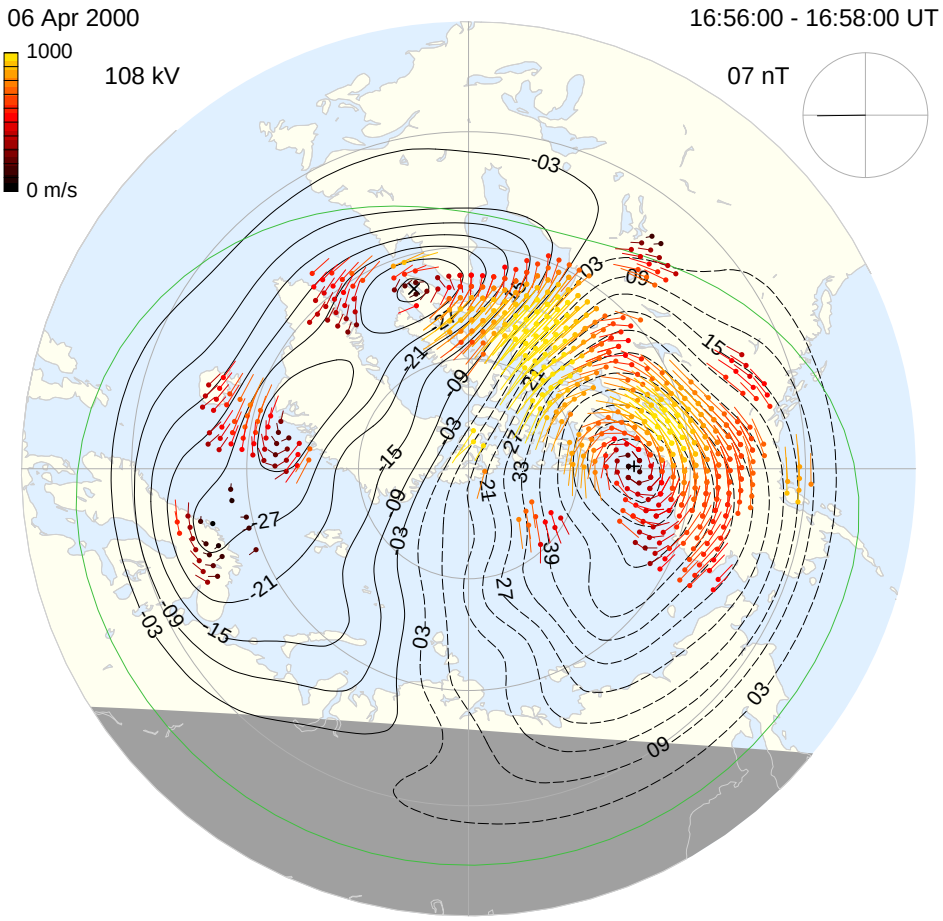
```
map_plot –l 50 –dn 20000406.map
```

Generate PostScript files from the convection map file "`20000406.map`".
The plots start at 16:50UT and the first file will be called "`20000406.1650.00.ps`". The lower latitude limit of the plot is 50°.



```
map_plot –l 50 -st 16:56 –term -coast –fcoast –key broy.key
20000406.map
```

Generate PostScript files from the map file "`20000406.map`", starting at 16:50UT. The plots start at 12:30UT and the first file will be called "`0000.ps`". Filled oceans and continents, together with the terminator and coastlines are plotted. The color key is taken from the file "`broy.key`".



trim_grid

Usage

```
trim_grid [--help] [-vb] [-st hr:mn]
          [-et hr:mn] [-sd yyyymmdd]
          [-ed yyyymmdd] [-ex hr:mn] [-i sec]
          [-{st}...] [-major ver] [-minor ver]
          [file] [index]
```

Options

`--help` displays the help message.

`-vb` verbose. Log status to standard error.

`-st` *hr*`:`*mn* start time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.

`-et` *hr*`:`*mn* end time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.

`-sd` *yyyymmdd* start date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.

`-ed` *yyyymmdd* end date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.

`-ex` *hr*`:`*mn* extent or length of time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.

`-i` *sec* re-integrate the grid file so that the time interval to stored in each record is *sec* seconds.

-{st} identifier code letter of a station to exclude from the grid file.

`-major` *ver* Overrides the major version number embedded in the map file. (Use of this option is not advised).

`-minor` *ver* Overrides the minor version number embedded in the map file. (Use of this option is not advised).

file grid file to process. If none is specified then standard input will be used.

index index of the grid file. The index is a file that identifies the location of each record in the grid file and can be used to speed up searches for a specific interval of data.

trim_grid

Description

Reprocesses a grid file. The program can extract a particular time period from a file, filter out contributions from specific radars or re-integrate a file to produce records of a different length.
The processed file is written to standard output.

Examples

```
trim_grid -vb –st 11:00 –et 14:00 19981020.grd > 1998102011.grd
```

Extracts a 3 hour period starting at 11:00UT from the file called "19981020.grd" to produce a file called "1998102011.grd". The status is logged to standard error.

```
trim_grid –sd 19991121 –st 22:00 –ex 4:00 –i 240 199911.grd >
          19991121.grd
```

Extracts a 4 hour period starting at 22:00UT on November 21, 1999 from the file "199911.grd" to produce the output file "19991121.grd". The records are re-integrated to produce records at 4-minute intervals.

```
trim_grid –g –k 20001120.grd > 20001120.2.grd
```

Removes vectors from the Goose Bay and Kapuskasing Radars from the file "20001120.grd" to produce the file "20001120.2.grd"

trim_map

Usage

```
trim_map [--help] [-vb] [-st hr:mn]
         [-et hr:mn] [-sd yyyymmdd]
         [-ed yyyymmdd] [-ex hr:mn]
         [-s source] [-major ver]
         [-minor min]
         [file] [index]
```

Options

`--help`	displays the help message.
`-vb`	verbose. Log status to standard error.
`-st` *hr*`:`*mn*	start time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-et` *hr*`:`*mn*	end time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-sd` *yyyymmdd*	start date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ed` *yyyymmdd*	end date of the data period to process. Expressed in the form "*yyyymmdd*", where *yyyy* is the year, *mm* the month, and *dd* is the day.
`-ex` *hr*`:`*mn*	extent or length of time of the data period to process. Expressed in the form "*hr*`:`*mn*", where *hr* is the number of hours and *mn* is the number of minutes.
`-s` *source*	Overrides the text string embedded in the map file that indicates the source of the file. (Use of this option is not advised).
`-major` *ver*	Overrides the major version number embedded in the map file. (Use of this option is not advised).
`-minor` *ver*	Overrides the minor version number embedded in the map file. (Use of this option is not advised).
file	map file to process. If none is specified then standard input will be used.
index	index of the map file. The index is a file that identifies the location of each record in the map file and can be used to speed up searches for a specific interval of data.

trim_map

Description

Extracts a section from a convection map file. The extracted section is written to standard output.

Examples

```
trim_map -vb –st 11:00 –et 14:00 19981020.map > 1998102011.map
```

Extracts a 3-hour period starting at 11:00UT from the file called "`19981020.map`" to produce a file called "`1998102011.map`". The status is logged to standard error.

```
trim_map –sd 19991121 –st 22:00 –ex 4:00 199911.map >
          19991121.map
```

Extracts a 4-hour period starting at 22:00UT on November 21, 1999 from the file "`199911.map`" to produce the output file "`19991121.map`".

Index