1994 November 15

The Difmap Cookbook

Greg Taylor
Department of Radio Astronomy, 105-24
California Institute of Technology, Pasadena, CA 91125
e-mail: gbt@astro.caltech.edu

Difmap was written by Martin Shepherd at Caltech, and is part of the Caltech VLBI Software Package.

Contents

1. INTRODUCTION

Difmap is a fast, flexible, editing and mapping program. While it is primarily intended for continuum VLBI data, it has been used to map VLA and other connected-element interferometer data in both continuum and spectral lines. It is designed to work on one source at a time and takes advantage of this and the large internal memory available in the current generation of workstations to minimize the time spent reloading data and regenerating models. Difmap also provides the user with a wide range of interactive plots available through simple commands. This frees the user from time-consuming specification of input parameters, and encourages a better familiarity with the data.

The purpose of this cookbook is to instruct and demonstrate how to drive Difmap effectively. A general mapping technique is presented and demonstrated for a snapshot VLBI observation of a core-jet source. Sample outputs are produced from all the displays generated by Difmap.

2. STARTING UP Difmap

Difmap is currently supported on Sparcstations running SunOS Release 4.1.*, and Solaris 2.*, by HP9000 series workstations running HPUX, and by IBM RS600 workstations running AIX. Start up Difmap by typing:

```
difmap
```

at which point you should see in response:

```
Caltech difference mapping program - version 2.0 (2 Feb 94)
Copyright (c) 1993 California Institute of Technology.  All Rights Reserved.
Type 'help difmap' to list difference mapping commands and help topics.
Started logfile:  difmap.log on Mon Aug 1 11:15:01 1994
0>
```

The `0>` is the prompt within Difmap. If you do not receive this response then Difmap has not been properly installed. Check your path environment variable to ensure that you can see the directory in which Difmap was installed, and consult with the person who claims to have installed Difmap at your site.

A log file called `difmap.log_*` is created upon the startup of Difmap. All commands are placed in this file as they are entered. Selected output lines are also written to the logfile, although they are prefixed with a `!` sign so that the log file may later be executed as a command file from within Difmap by typing

```
0>@difmap.log_5
```

Note that PGPLOT cursor events are not recorded in the log, so no interactive data editing or window setting will be reproduced when executing a log file. Typing `logfile` *filename* closes any existing log file and if a file name is provided opens a new log file with that name.

If you want to exit from Difmap type

```
0>exit
Exiting program
Log file difmap.log_5 closed on Wed Oct 20 13:55:53 1993
```

2.1 Getting Online Help

Simply typing `help` within Difmap will provide a list of the main modules. Help on all these modules can be obtained by typing `help` *module*, but the most useful commands and some general help topics can be found by typing `help difmap`. Help listings longer than a single page will prompt: `Press return to continue, Q [or command] to quit, or P to page`. If you have a favorite pager then this can be set to the default by setting your PAGER environment variable before starting Difmap (e.g. place the line `setenv PAGER less` into your .cshrc file), otherwise `more` will be used.

A list of Difmap commands is also provided in Appendix B of this cookbook. To get more detailed information about any command within Difmap type `help` *command*. The interpreter within Difmap performs minimum matching for all commands. To find out about new features in your release of Difmap type `help whatsnew`.

2.2. Reading Data

Your *u*-*v* data should be in the random groups UVFITS format (the same format as written by the AIPS task FITTP). The UV data must be sorted first in time (e.g. T*, TB). Multiple subarrays, IFs, spectral-line channels, and polarizations are all supported to a limited extent. For best results, the data must already have been fringe-fitted (in from VLBI) and amplitude-calibrated. The *merge* format is not supported in this version of Difmap. If you have a *merge* format file, this can be converted to FITS format using the MERGEFITS program of the Caltech VLBI software package. Read your data into Difmap by typing:

```
0> observe 0749+540.cal                a calibrated UVFITS format file
Reading UV FITS file:  0749+540.cal
AN table 1:  59 integrations on 103 of 105 possible baselines.
Apparent sampling:  0.714991 visibilities/baseline/integration-bin.
Found source:  0749+540

Number of spectral-line channels:  1

IF   Freq (Hz)   Chan BW (Hz)   Full BW (Hz)   Sideband
01   4.99199e+09       2e+06       2e+06           USB

Polarization(s):  RR
```

```
Read 37 lines of history.

Reading 4345 visibilities.
Selecting polarization:  RR, channels:  1..1
```

To get even more information about your observation type `header`.

In order for editing and self-calibration to work, visibilities from different baselines must be grouped with the same integration times. UV FITS files do not provide any means to map visibilities on different baselines into integrations. Each visibility has its own time-stamp, which need not agree with those on other baselines within the same logical integration. Difmap, on the other hand does require that visibilities be grouped into integrations. This is the reason for the 'binwid' argument of the OBSERVE command. If the visibilities do not lie on an integration grid then you must specify a suitable integration time into which visibilities should be binned into integrations. Depending on how the FITS file has been processed, it may already have visibilities grouped into integrations with identical time-stamps assigned to each grouped visibility, in which case no 'binwid' argument will be required. If you do not know what state your file is in, then try to read it with the observe command without specifying an integration time. Then if OBSERVE reports an apparent sampling of $\leq$0.5 then either run the UVAVER command to re-grid the data or equivalently re-run OBSERVE with a suitable integration time. Other symptoms of incompletely binned integrations are that selfcal flags all of your data due to the lack of closure quantities, and that station based editing in VPLOT behaves like baseline based editing (see §3).

2.3. Examining Data

If your data consists of only a single polarization and spectral line channel (as in the example above), then Difmap will automatically SELECT that polarization after it is read in. If you have more than one polarization then you'll need to to select a polarization and channel range for processing using the SELECT command (see also §9 and §10).

At this point it is often useful to take a look at a plot of amplitude versus u-v radius. This may be obtained by typing:

```
0> radplot
```

If you have not previously made any plots, you will be prompted for the output device for the plot. If you respond by typing `?` then you will be given a list of devices to choose from that will look something like:

```
Legal PGPLOT device types are:
/GIF   (Graphics Interchange Format file, landscape orientation)
/VGIF  (Graphics Interchange Format file, portrait orientation)
/HPGL  (Hewlett Packard HPGL plotter, landscape orientation)
/VHPGL (Hewlett Packard HPGL plotter, portrait orientation)
/NULL  (Null device, no output)
/PPM   (Portable Pixel Map file, landscape orientation)
/VPPM  (Portable Pixel Map file, portrait orientation)
/PS    (PostScript file, landscape orientation)
/VPS   (PostScript file, portrait orientation)
/CPS   (Colour PostScript file, landscape orientation)
/VCPS  (Colour PostScript file, portrait orientation)
/QMS   (QUIC/QMS file, landscape orientation)
/VQMS  (QUIC/QMS file, portrait orientation)
/TEK4010 (Tektronix 4010 terminal)
/GF    (GraphOn Tek terminal emulator)
/RETRO (Retrographics VT640 Tek emulator)
/XTERM (XTERM Tek terminal emulator)
/TK4100 (Tektronix 4100-series terminals)
/VT125 (DEC VT125 and other REGIS terminals)
/XDISP (pgdisp or figdisp server)
/XWINDOW (X window window@node:display.screen/xw)
/XSERVE  (A /XWINDOW window that persists for re-use)
```

The exact list will depend on your installation of PGPLOT. If running Difmap under OpenWindows or X11, then a suitable choice is `/xw`. At this point a window will be created on your console with a plot similar to Figure 1.

This display can be manipulated in a variety of ways. Move the cursor to the plot window. Now type `n`. This will highlight all visibilities involving the first antenna, and the name of that antenna will be displayed in the top right corner of the plot. Pressing `n` again will highlight visibilities of the next antenna, and so forth. If you're having trouble seeing your data points, try typing a period, ".", to make each point bigger. Type `h` to obtain the following list of options:

```
You requested help by pressing 'H'.
The following keys are defined when pressed inside the plot:
 X - Quit radplot
 L - Re-display whole plot
 . - Re-display plot with alternate marker symbol.
 n - Highlight next telescope
 p - Highlight previous telescope
 N - Step to the next sub-array to highlight.
 P - Step to the preceding sub-array to highlight.
 T - Specify highlighted telescope from keyboard
```

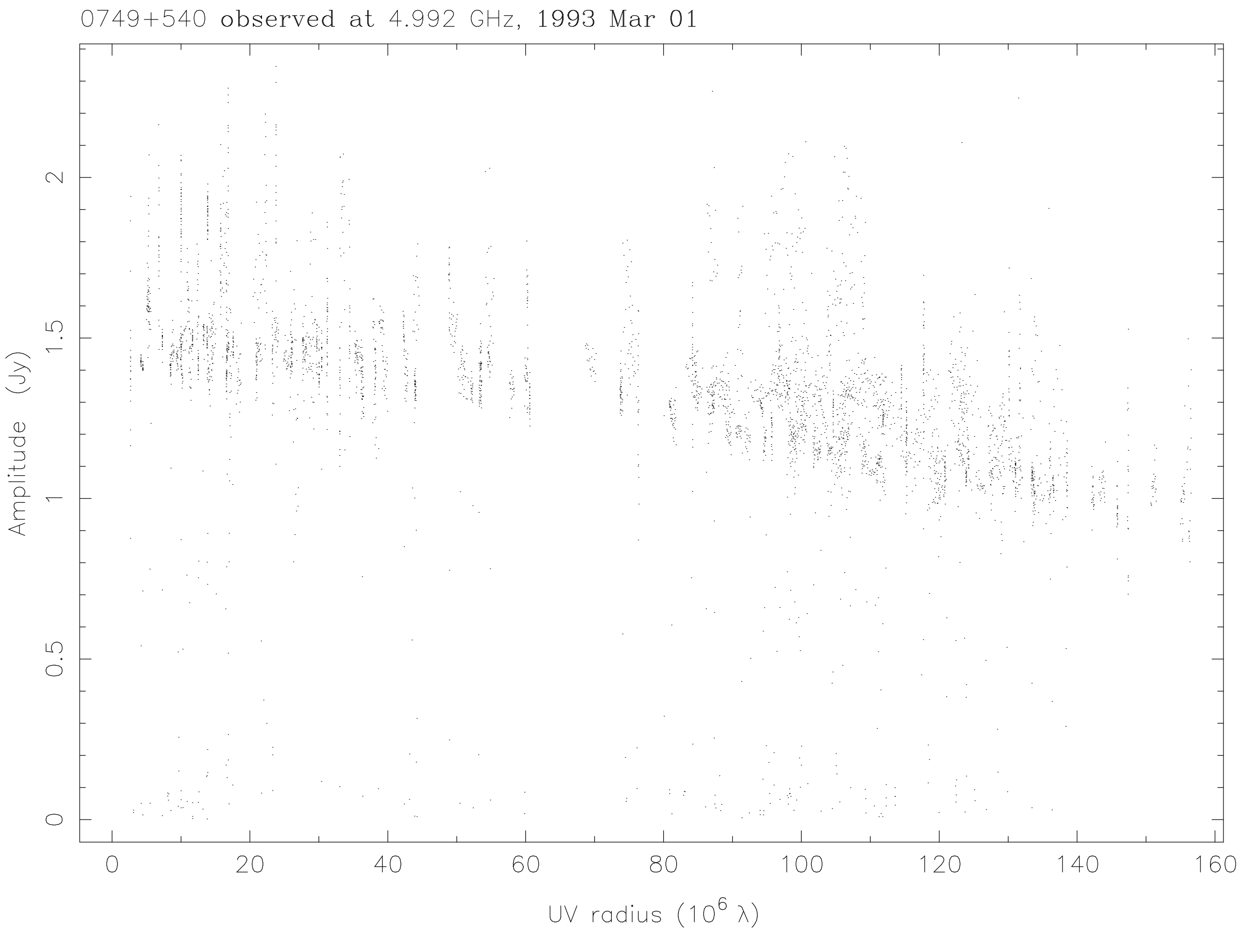

Figure 1. A plot of the amplitudes versus u-v radius for the source 0749+540 produced by RADPLOT. The data have been calibrated, but not yet edited.

```
 S - Show the baseline and time of the nearest point to the cursor
 A - (Left-mouse-button) Flag the point closest to the cursor
 C - Initiate selection of an area to flag.
 I - Toggle IF based editing.
 W - Toggle spectral-line channel based editing.
 Z - Select a new amplitude or phase display range.
 U - Select a new UV-radius display range.
Display mode options:
 M - Toggle model plotting.
 1 - Display amplitude only.
 2 - Display phase only.
 3 - Display amplitude and phase.
 + - Toggle whether to use a cross-hair cursor if available.
```

Although it it often tempting to edit away bad points in RADPLOT be aware that such edits are baseline-based edits, while systematic errors are more often antenna-based. See §3 on Editing data.

You may quit from within RADPLOT by pushing the third (far-right) mouse-button

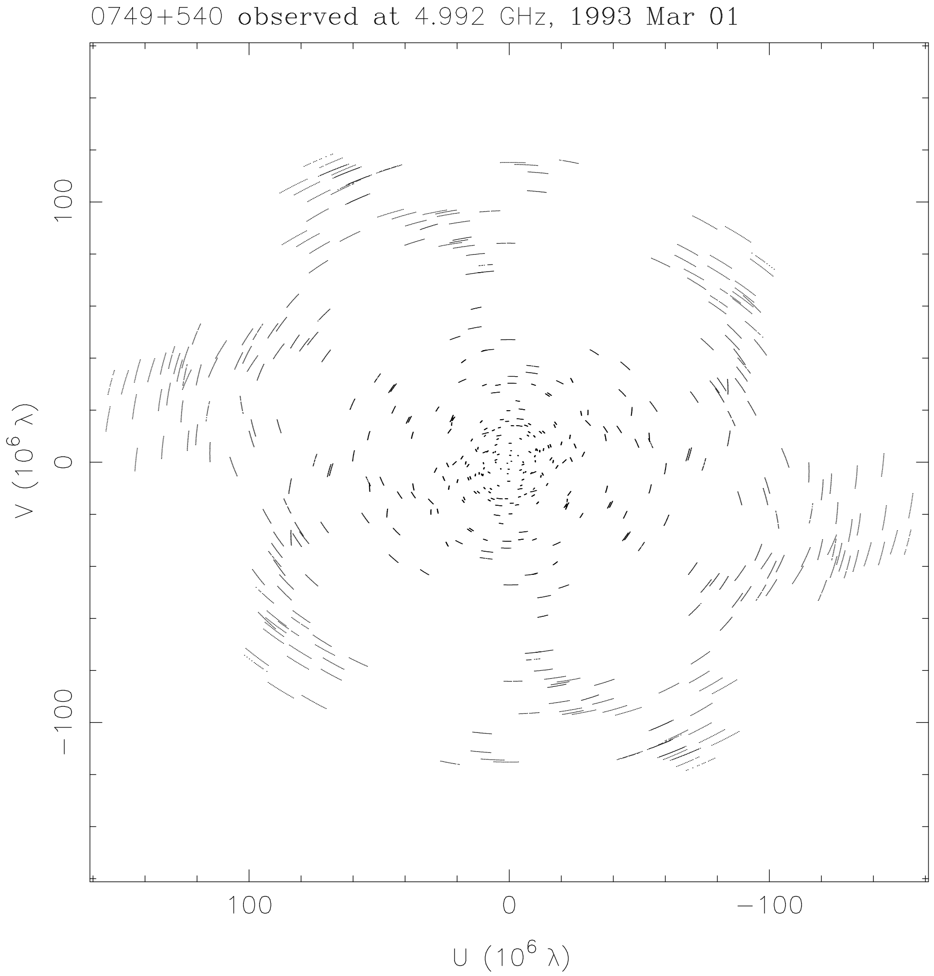

Figure 2. The *u*-*v* coverage for a snapshot (3 × 20 minutes) observation of 0749+540 produced by UVPLOT.

or typing `X`. Within Difmap the left, middle, and right mouse buttons are mapped to the A, D, and X keys.

Another useful display is a plot of the *u*-*v* coverage. This may be obtained by typing:

```
0> uvplot
```

to obtain a plot something like that shown in Figure 2. Visibilities common to each telescope can be highlighted by typing `n` with the cursor in the plot window. To zoom in on a region of the uvplot, type `z` followed by two opposite corners. Pressing the `z` key twice more will unzoom the plot. Other display options are similar to RADPLOT except that editing of individual points is not allowed. Regions may be selected for editing with the `c` key, but such edits are generally not recommended. To exit, type `X` or push the third mouse button.

To look at a cut of amplitude and/or phase along any radial line in the UV plane use the command PROJPLOT. This is especially useful for double sources, or sources with straight jets. For illustration purposes, I've selected a nearly equal double source, 0026+346 (also from Taylor *et al.* 1994), and done `projplot 56` to display the projected amplitude and phase with distance along the position angle (measured east of north) of the majority of source structure (Figure 3). The interactive options are the same as for RADPLOT with the addition of three keys ?,< and > for selecting the projection angle.

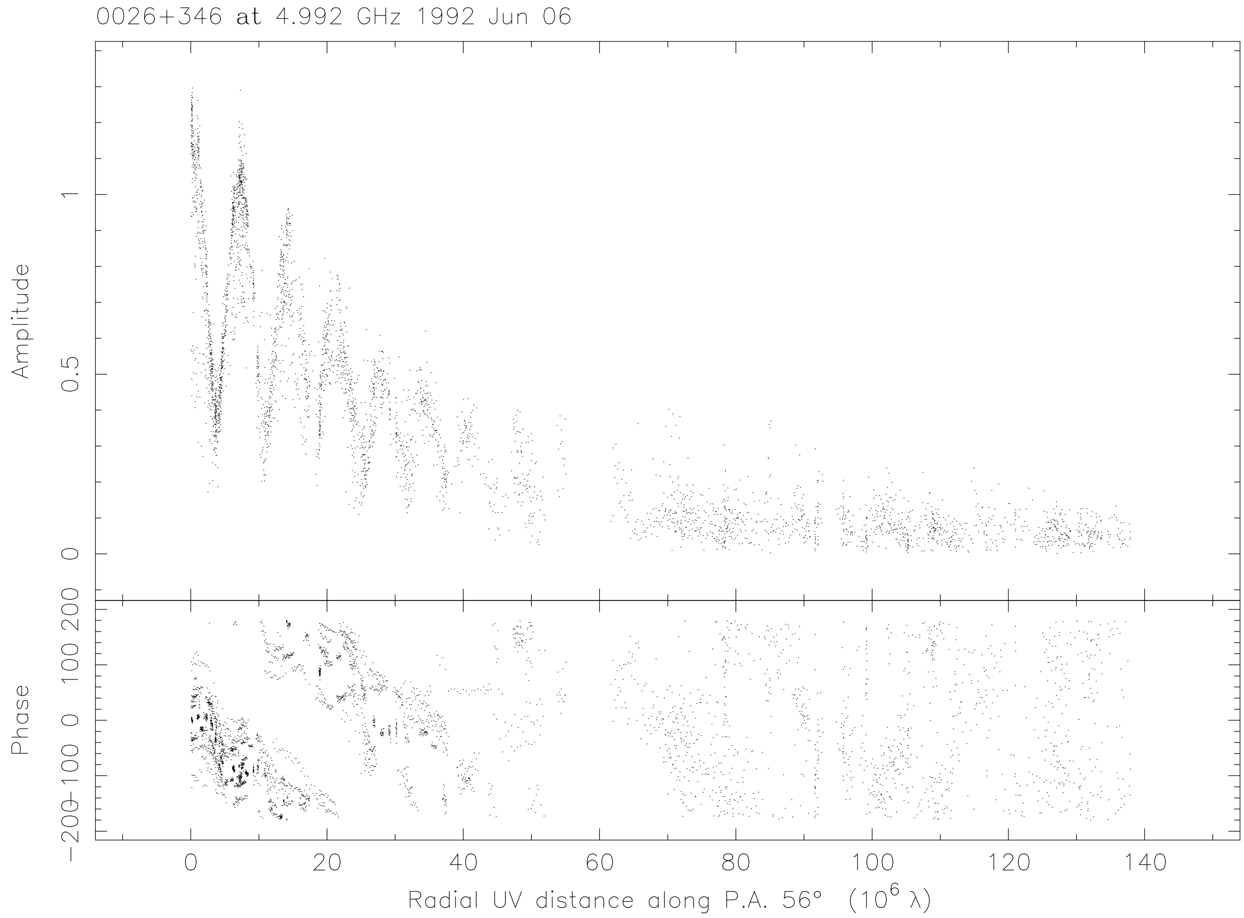

Figure 3. The result from PROJPLOT for an extended double source with components along a position angle of ∼56° . The beating of the two main components is clearly seen with a wavelength directly related to their spatial separation of 30 mas.

If you suspect that some of your data may be missing, or have gaps, you can verify this using TPLOT.

```
0>tplot
```

A dot is displayed for every time interval for which data exists for each telescope. See Figure 4 for a sample output. Data are colored green if there are no edits, yellow if any data to an antenna are flagged, blue if the antenna has been flagged in SELFCAL or CORPLOT, and red if all data to a given antenna are flagged. TPLOT is also useful if you should want to flag all baselines to a selected antenna, or all antennas for a given time-range.

3. EDITING DATA

It is a time honored tradition among scientists to get rid of bad data. With VLBI observations this should be done carefully yet thoroughly. While speaking about VLBI observations, a famous scientist once said, "bad data is worse than no data at all." Difmap provides an interactive data editor, VPLOT. Before invoking VPLOT you may find it useful to set a few "flags" which will effect how it runs. The setting of these flags for optimal performance depends on the dataset in question and your own personal preferences. A good choice for snapshot observations is:

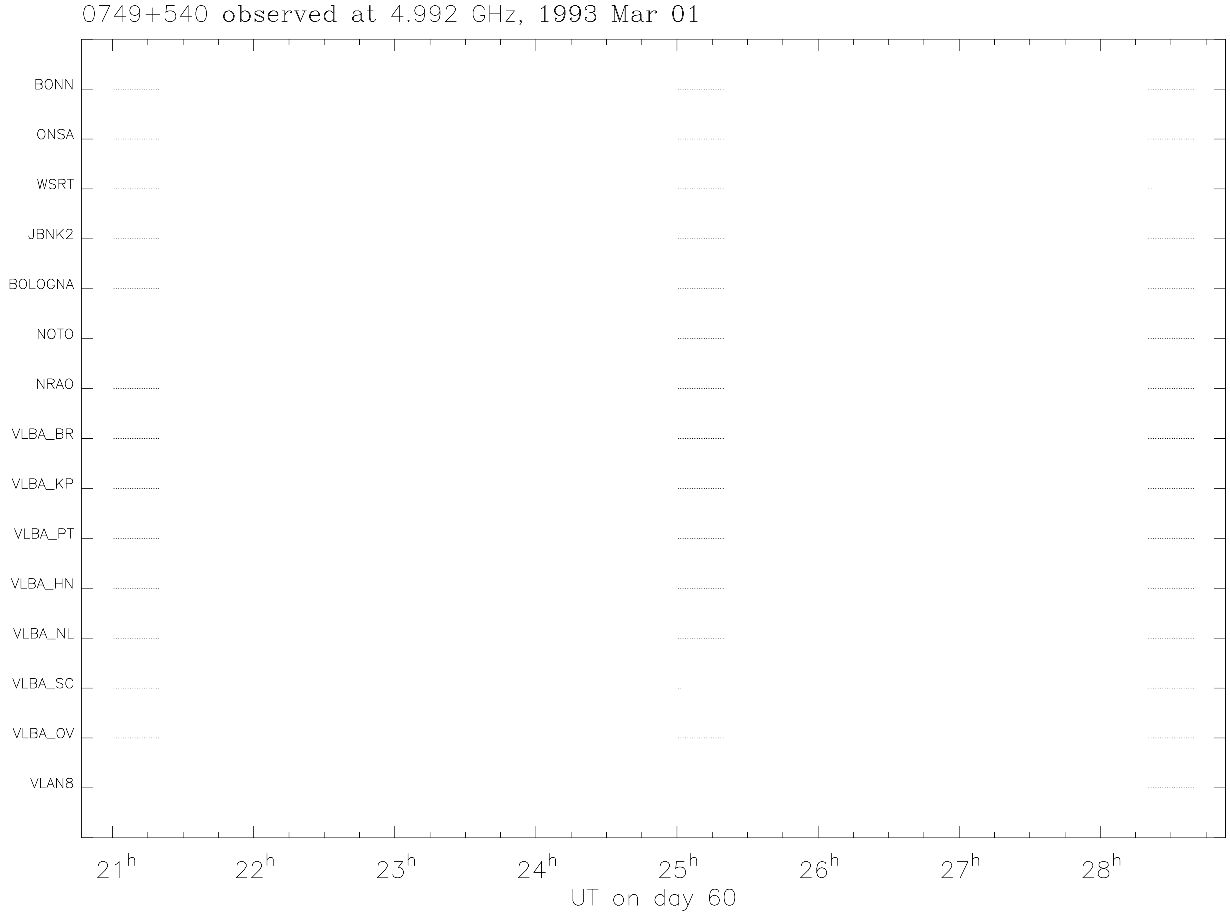

Figure 4. The plot produced by TPLOT for a snapshot (3 × 20 minutes) observation of 0749+540.

```
0>vflags="ebm"                        e: plot error bars, b: remove time gaps
0>vplot 5                             plot 5 baselines on each page
Over-riding default options with user defined vflags="ebm"
NB. Keyboard input will only be accepted when the cursor
   is in the plot window
For HELP, hit the 'H' key on your keyboard
```

By default, this will plot both amplitudes and phases versus time for the first 5 baselines (see Figure 5). The default interscan gap is one hour, but this can be changed with the SCANGAP command.

If the error bars on your data don't look right, you may have to adjust them globally using the WTSCALE command. See the help file for WTSCALE for the equation relating the weight scale factor to the bandwidth, integration time, correlator coefficient, effective collecting area, and system temperature.

At this point you may interactively edit the displayed data. Move the cursor (it should appear as a '+') to a bad datapoint (such as those evident at the beginning of each scan in Figure 5), and click the leftmost mouse button. If VPLOT is in "Station Editing" mode (look for these words in the upper left corner of the plot) then this action will flag

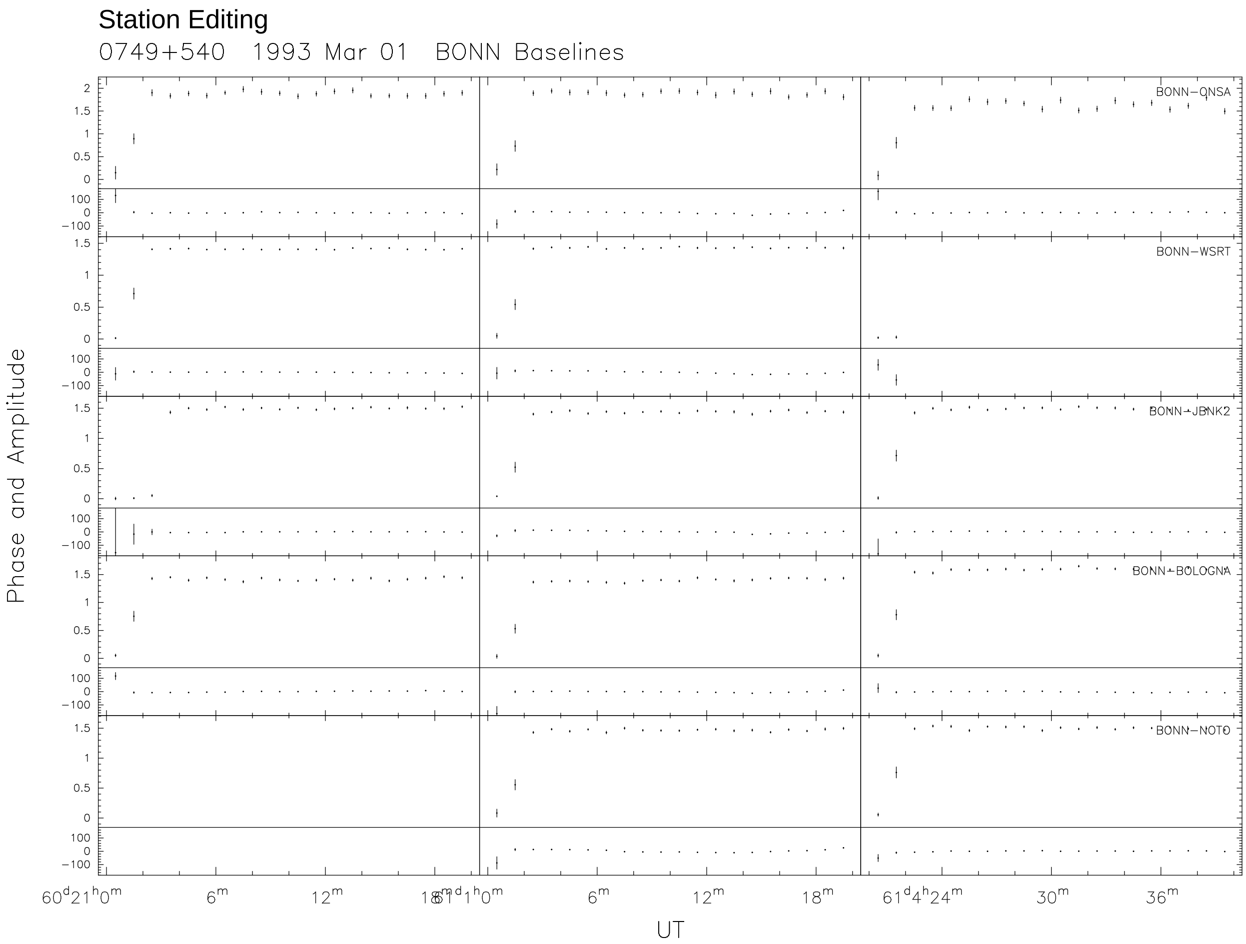

Figure 5. The first screen produced by VPLOT 5 with vflags set to "ebm".

all data points associated with antenna number 1 (Bonn in the example) for this one time. To perform baseline-based flagging operations, hit the spacebar. The words "Baseline Editing" should now appear in the upper left corner of the plot. If you have multiple IFs and want the edits on the currently selected IF to be applied to all IFs then toggle this on with the `I` key. The `W` key will toggle editing between the selected channel and all channels. To see the flagged data, type `f` and then press return. Flagged data will now be marked with a red cross. Clicking on a flagged data point will unflag it. The full range of options can be listed by typing an `h` in the plot window. The resulting output is:

```
Vplot key bindings:
H - List the following key bindings.
X - Exit vplot (right-mouse-button).
A - Flag or un-flag the visibility nearest the cursor (left-mouse-button).
U - Select a new time range (hit U again for the full range).
Z - Select a new amplitude or phase range (hit Z twice for full range).
C - Flag all data inside a specified rectangular box.
```

```
R - Restore data inside a specified rectangular box.
K - Flag all visibilities of a selected baseline and scan.
L - Redisplay the current plot.
n - Display the next set of baselines.
p - Display the preceding set of baselines.
N - Display the next sub-array.
P - Display the preceding sub-array.
M - Toggle whether to display model visibilities.
F - Toggle whether to display flagged visibilities.
E - Toggle whether to display error bars.
S - Select the number of sub-plots per page.
1 - Plot only amplitudes.
2 - Plot only phases.
3 - Plot both amplitudes and phases.
B - Toggle whether to break the plot into scans (where present).
V - Toggle whether to use flagged data in autoscaling.
T - Request a new reference telescope/baseline.
- (SPACE BAR) Toggle station based vs.  baseline based editing.
I - Toggle IF editing scope.
W - Toggle spectral-line channel editing scope.
```

It is often desirable to flag all points within a given region. To do this first move the cursor into the plot window and type a `c`. Next, select a corner of the region of bad points by moving the cursor there and pushing the leftmost mouse button. Finally, move the cursor to the opposite corner of the region and push the leftmost mouse button again.

If you make a change in the way the plot is displayed, either by toggling one of the flags or selecting a number, you will need to follow that with a return key to see its effect. Move forward to the next screen of baselines by pressing `n`, and backwards with `p`. If you have multiple sub-arrays then a capital `N` will move to the next sub-array. Continue to edit and move through the baselines until you encounter the message:

```
No plottable baselines in forward direction
```

At this point you will actually have seen every baseline twice. While this is redundant, it allows the user to work through the data antenna by antenna with all baselines plotted for each station.

Once you have finished editing the data, it is a good idea to write out a copy of it. First exit from the VPLOT window (far-right mouse button), then type:

```
0>wobs 0749+540.edt
```

You can enter and exit VPLOT any number of times and Difmap will remember your edits. If you quit from Difmap without first writing your data, then it is gone like the wind.

It is sometimes possible for the amplitudes and phases of a number of visibilities to look ok, even while there are serious problems with those data. One such example is when there is significant leakage of one polarization into another at the feeds of two or more antennas. These problems can show up in the closure phases. Closure phases are defined as the sum of visibility phases around a closed triangle of three antennas (Cornwell & Fomalont 1988). For a point source the closure phase should be zero. Closure phases are free of antenna-based errors such as those induced by atmospheric fluctuations and position shifts, and are used in self–calibration. Under antenna-based self–calibration the closure phases remain constant which also means that unlike some phase and amplitude errors that can be fixed by self–calibration, errors in the closure phase cannot. Examining the closure phases with CPPLOT may also be useful to determine whether self–calibration is possible. To examine the closure phases type `cpplot bonn,5`. This example plots the closure phase for the first 5 triangles involving the bonn telescope. Alternatively the command `cpplot bonn-nrao,6` will display the first 6 triangles containing the bonn-nrao baseline. Lastly, `cpplot bonn-nrao-noto` will display this single triangle. If your data have multiple sub-arrays the number of the sub-array should go before the telescopes. So typing `cpplot 3:bonn-nrao,5` will display the first 5 triangles in subarray 3 containing the bonn-nrao baseline. The scope of the editing is displayed on the top line of the plot and is dependent upon if CPPLOT was invoked with a station, baseline, or triangle. Triangle editing indicates that all baselines for the displayed closure triangle will be edited. As in VPLOT, the scope of the editing can be toggled with the spacebar, I, and W keys.

4. DIFFERENCE MAPPING

And now for the part which lends Difmap its name. In difference mapping the observer converges on a model by progressively cleaning the residual map, and self–calibrating using the latest model. At each SELFCAL–MAPPLOT–CLEAN iteration the model is subtracted from the data in the u-v plane, thus avoiding aliasing of side-lobes within the field of view. The model is retained in memory thereby saving considerable CPU time over programs that must recalculate it every time (such as in AIPS), and allows for easy comparisons between the model and the data. Upon convergence, the user obtains the self–calibrated data, a model, and a map of the source.

Before starting difference mapping the data should be amplitude calibrated (using the program CAL from the Caltech Package, or using the task ANCAL within AIPS) and edited. For sources with a very extended, diffuse component, the user may also want to use modelfit (see §8) to generate a starting model, as the CLEAN algorithm has some difficulty with such structures. For most VLBI sources a point source model is a good starting place (Cornwell & Fomalont 1988). To start with the default 1 Jy point source model at the map center type:

```
0>startmod
Applying default point source starting model.
Performing phase self-cal
Adding 1 model components to the UV plane model.
The established model now contains 1 components and 1 Jy

Correcting IF 1.
```

```
Fit before self-cal, rms=1.644838Jy sigma=35.631953
Fit after self-cal, rms=0.371699Jy sigma=7.936355

Reinstating the default IF.
clrmod:  Cleared the established, tentative and continuum models.
Redundant starting model cleared.
```

SELFCAL reports the rms difference between the model and the data and also sigma, which is the rms divided by the variances implied by the visibility weights (effectively sigma is the square root of the reduced χ^2), Don't worry about the (potentially huge) value of sigma at this point since the starting model probably doesn't match the visibility amplitudes well. The effect of STARTMOD is to fix up the phases so that you may begin mapping your source. If you want to begin with a more complicated model than a point source, supply the name of a file containing that model to STARTMOD.

The default units for Difmap are milli-arcseconds in the map plane and mega-wavelengths in the UV plane. If it is more convenient for you to work in units of arcseconds and kilo-wavelengths then type `mapunits arcsec`.

Now, define the image size and cell size you wish to map. The image size (square or rectangular) must be an integer power-of-two, and since Difmap can only properly deconvolve the inner quarter of the image, it should be at least twice the maximum source dimension. The cell size should be small enough to allow for 3 or more pixels across the synthesized beam. For the 5 GHz snapshot data on 0749+540, a reasonable choice is

```
0>mapsize 256,0.2
New map grid = 256x256 pixels with 0.2 milli-arcsec cellsize
```

Difmap will complain if the cell size chosen is too coarse to allow at least 2.5 pixels across the synthesized beam. If a certain cell size is required, the UV range can be restricted by typing `uvrange 0,51.6/`*cell* where *cell* is the cell size.

The default weighting scheme is uniform, with a binning size of 2 uv-grid pixels. With uniform weighting, the data are weighted inversely to the number of visibilities occurring within the bins. Equal weighting of all data points is called natural weighting, and provides higher sensitivity than uniform weighting, although uniform weighting provides higher resolution and in some cases may deconvolve more readily. Natural weighting is selected within Difmap by choosing a binning size of `0`. At the same time the u-v weighting is set, the user should also select the appropriate weighting of the visibilities by the errors. If the errors have been meaningfully calculated then -1 is a good choice for the error power, otherwise use 0. Type `help uvweight` for more information and see Sramek & Schwab (1988). For our sample data we'll start with uniform weighting:

```
0>uvweight 2,-1                       uniform weighting, error power of −1
New uniform weighting binwidth=2 pixels
Weights scaled by the amplitude errors raised to the power -1
Radial weighting cancelled
```

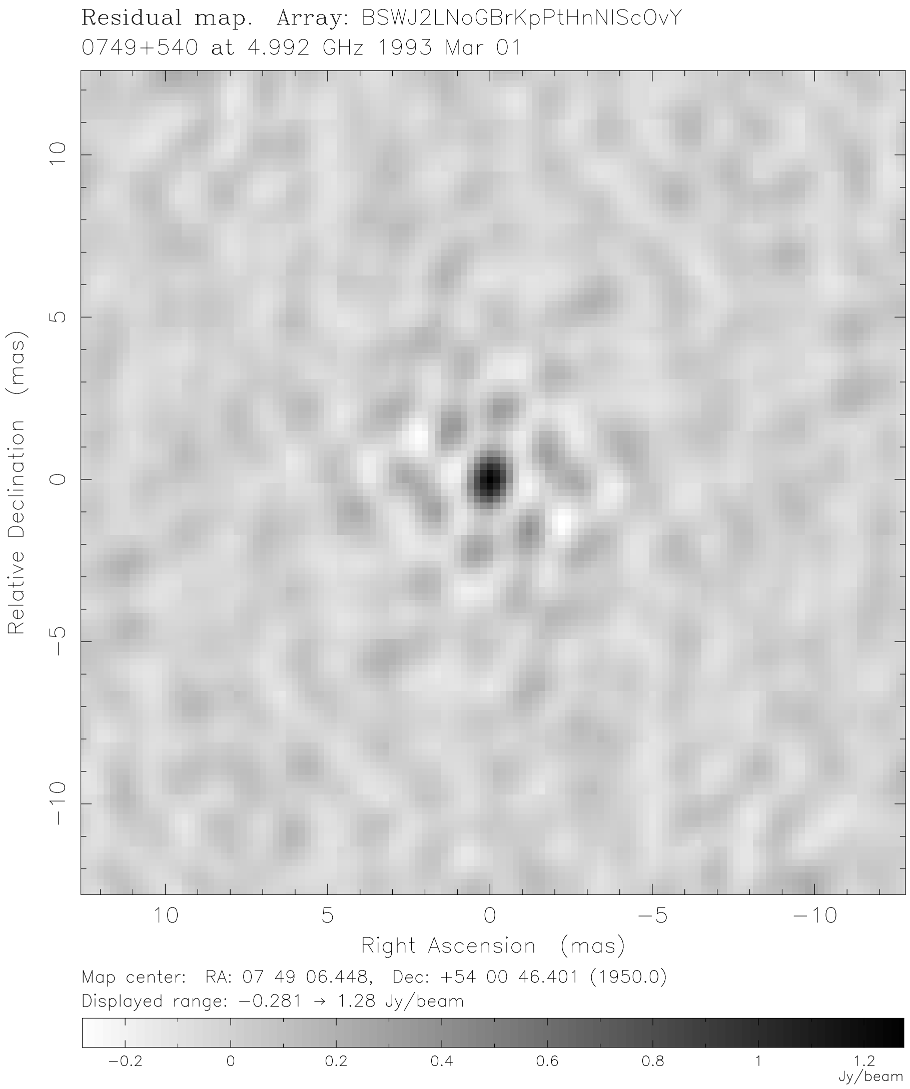

Figure 6. The uncleaned uniformly-weighted map of 0749+540, with a map size of 256 × 256, and cell size 0.2 mas/pixel.

Now you're ready to look at a first map of the source via

```
0>mapplot
Inverting map and beam
Estimated beam:  bmin=0.8978 mas, bmaj=1.237 mas, bpa=-21.65 degrees
Estimated noise=0.975165 mJy/beam.
Move the cursor into the plot window and press 'H' for help
```

This will generate a dirty map of the source (see Figure 6). The hexagonal sidelobes are due to the gaps in u-v coverage resulting from the three scans at widely spaced hour angles. With the cursor in the active window type `c` to change to a false-color transfer function. Hold down the `f` key and move the cursor within the graphics display window to interactively modify the transfer function. A capital `F` will reset the transfer function, and a `c` will switch to color. Other options can be explored by typing `h`. For example:

```
The following keys may be selected when the cursor is in the plot
X - Quit this session
A - Select the two opposite corners of a new clean window.
D - Delete the window who's corner is closest to the cursor
S - Describe the area of the window who's corner is closest to the cursor.
V - Report the value of the pixel under the cursor.
f - Fiddle the colormap contrast and brightness.
F - Reset the colormap contrast and brightness to 1, 0.5.
L - Re-display the plot.
G - Install the default gray-scale color map.
c - Install the default pseudo-color color map.
C - Install a color map named at the keyboard.
T - Re-display with a different transfer function.
G - Display the map in gray-scale.
Z - Select a sub-image to be displayed.
K - Retain the current sub-image limits for subsequent mapplot's
m - Toggle display of the model.
M - Toggle display of just the variable part of the model.
N - Initiate the description of a new model component.
R - Remove the model component closest to the cursor.
+ - Toggle whether to use a cross-hair cursor if available.
H - List key bindings.
```

Define a small window around the brightest component for cleaning. This is done by moving the cursor to a corner of the desired region, clicking the leftmost mouse button, and then moving to the opposite corner and clicking the leftmost mouse button again. Use the middle button, with the cursor at any corner, if you wish to delete a window. For a false-color representation type `c`, to get back to greyscale type `g`. When you are finished type `x` or click the rightmost mouse button. To look at the dirty beam, use `mapplot beam`. To look at the restored (clean) map, use `mapplot cln`. To zoom in on an interesting feature type `z` and then select the region of interest by selecting its opposite corners with the leftmost mouse button. To unzoom, type the `z` key twice.

4.1. Cleaning

Choose a number of iterations and a loop gain for cleaning. Once entered, these will become the default values. You may also set a cutoff if you wish – type `help clean` for more details. A slow, gradual cleaning works well for Difmap since you will see the residuals at each step. For our example source we'll choose:

```
0>clean 100,0.05
clean:  niter=100 gain=0.05 cutoff=0
Component:  050 - total flux cleaned = 1.208431 Jy
Component:  100 - total flux cleaned = 1.382937 Jy
```

```
Total flux subtracted in 100 components = 1.382937 Jy
Clean residual min=-0.019764 max=0.043627 Jy/beam
Clean residual mean=0.000333 rms=0.007438 Jy/beam
Combined flux in latest and established models = 1.38294 Jy
```

4.2 Self–Calibration

With the improved, but still basically point-like model just obtained, self–calibrate the phases by typing

```
0>selfcal
Performing phase self-cal
Adding 6 model components to the UV plane model.
The established model now contains 8 components and 1.382937 Jy

Correcting IF 1.

Fit before self-cal, rms=0.205283Jy sigma=2.113758
Fit after self-cal, rms=0.205262Jy sigma=2.111924
A total of 2 un-correctable visibilities were flagged.
```

The last line provides a warning concerning visibilities that are not part of closed arrays with a sufficient number of telescopes. A small number of such visibilities may result from vigorous baseline-based editing. The minimum, and also default, for phase–only self–calibration is 3 telescopes, and for amplitude self–calibration it is 4 telescopes. Correction flags applied by SELFCAL or in CORPLOT are different from ordinary flags in that they can only be undone in CORPLOT or with UNCALIB. In plot commands, visibilities that are flagged due to self-calibration are displayed in blue to differentiate them from good visibilities (in green) and otherwise flagged visibilities (in red). Type `help selfflag` for more details.

To see the residual map type `mappl`. Be aware that some spurious symmetrization may have resulted from using a point source model. Find the brightest component in the residual map and place a window around it as described in §4. Use a small window, since the idea behind difference mapping is to progressively build a model for the source by cleaning from the residual map. Type `clean` to clean another 100 components from the windowed regions. Iterate with SELFCAL, MAPPLOT and CLEAN until sigma is no longer decreasing, the cleaned flux is no longer increasing, and the residual map is roughly smooth and featureless. You can view (and edit) the corrections to the antennas with CORPLOT. Examine the goodness of fit of the model to the data in a general sense with RADPLOT, and in more detail with VPLOT using the `m` flag to request a plot of the model as well as data. To see a map with the clean components restored type `mapplot cln`. To see where clean components are located type a `m` in MAPPLOT, or start it up with `mapplot cln,true`. Positive clean components will be shown as green crosses, and negative components as red ones. Gaussian components will be displayed as ellipses (and see §8 on model fitting). You can get rid of clean components within a window by deleting that window within MAPPLOT, and then typing `winmod`. Individual model components can be removed by typing `R` with the cursor placed on the offending component.

If your model fails to match up with the data on the short spacings, you may be missing some flux. If this is the case you may want to switch to natural weighting `uvweight 0,-1`, before you start in with amplitude self–calibration.

If you have sufficient signal-to-noise in your data and a good model you may now consider amplitude self–calibration. As a first step, compute a single amplitude correction factor for each antenna:

```
0>gscale
Performing overall amplitude self-cal
Correcting IF 1.
   Amplitude normalization factor in sub-array 1:  1.0249
   Telescope amplitude corrections in sub-array 1:
  BONN      1.03   ONSA      0.82   WSRT      1.05    JBNK2     1.01
  BOLOGNA   1.01   NOTO      1.01   NRAO      1.00*   VLBA_BR   1.05
  VLBA_KP   0.95   VLBA_PT   0.98   VLBA_HN   1.01    VLBA_NL   1.03
  VLBA_SC   1.04   VLBA_OV   1.04   VLAN8     0.98
  Fit before self-cal, rms=0.170649Jy  sigma=2.025436
  Fit after  self-cal, rms=0.121868Jy  sigma=1.262237
Reinstating the default IF.
```

In this example the initial calibration was fairly good for all the antennas except Onsala. Use MAPPLOT to see the effect of GSCALE. To allow the telescope amplitude factors to float freely (which may alter the total flux) use `gscale true`. For further time–dependent amplitude self–calibration, it is best to start with long solution intervals (to insure a high enough signal-to-noise) and gradually shorten the solution interval as SNR permits. For example:

```
0>selfcal true,true,30               doamp=true, dofloat=true, solint=30
Performing amp+phase self-cal over 30 minute time intervals

Correcting IF 1.

Fit before self-cal, rms=0.121868Jy sigma=1.262237
Fit after self-cal, rms=0.104460Jy sigma=0.982351

Reinstating the default IF.
```

Be warned that an amplitude & phase self–calibration has the power to “make” the data fit the model, and can thereby “freeze” false components into the data. Be sure to use CORPLOT to ensure that the solutions are reasonable. Figure 7 shows an example of the CORPLOT solutions for the VLBA antenna at Pie Town with a 3 minute solution interval. CORPLOT can also be used interactively to flag bad solutions or data. Or if you find that you don’t trust the amplitude corrections, type

```
0>uncal false,true
uncal:  All telescope amplitude corrections have been un-done
```

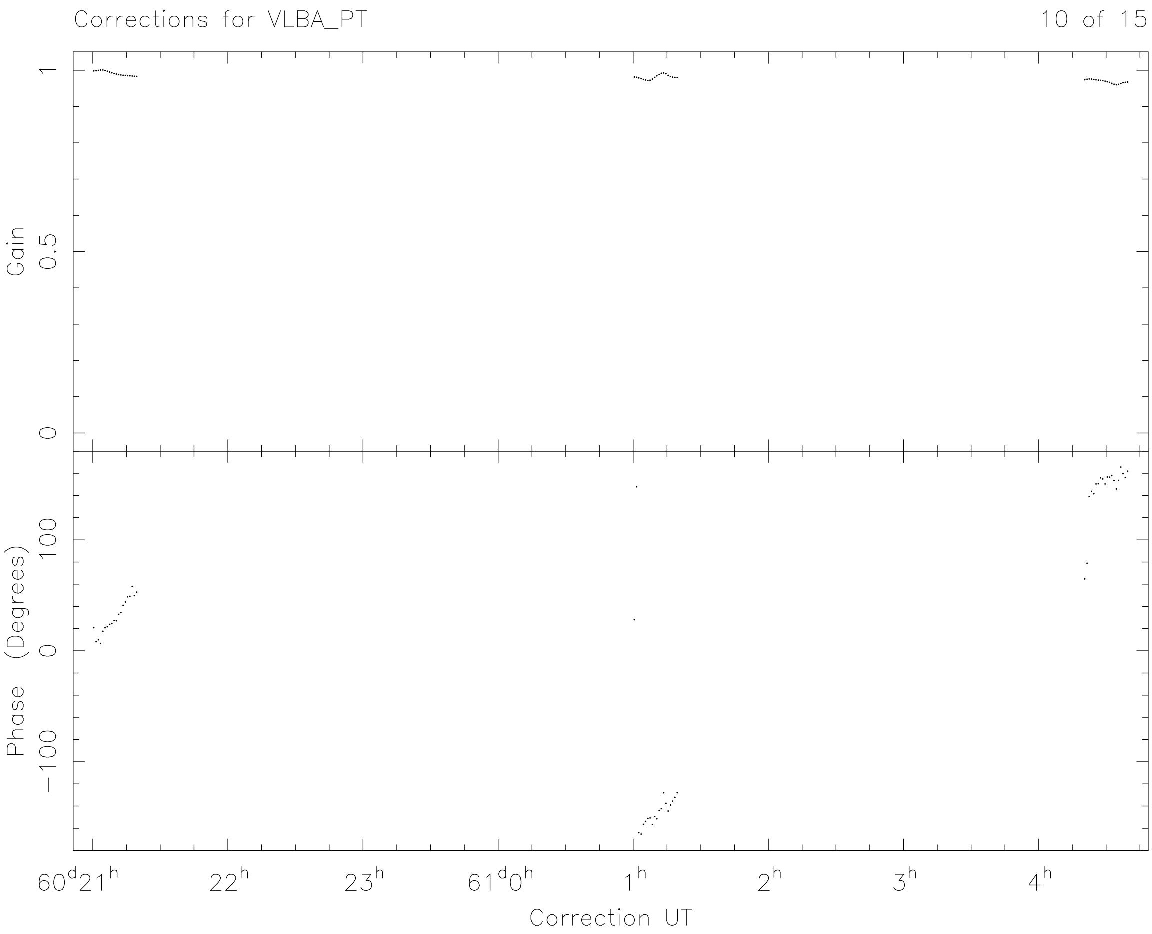

Figure 7. The phase and amplitude corrections for the VLBA antenna at Pie Town. The gain solution interval was 3 minutes. The `b` key was used to tell Difmap not to break the display into scans.

When you think you have obtained a fairly good model for your source, you may want to remove all antenna gain corrections and start over with this as your initial model, instead of a point source. This will often lead to some improvements. Also, in the course of cleaning and self-calibrating, the map may become too deeply cleaned in some areas. This will be obvious if the clean components clearly follow the edges of the clean windows, and/or if there are almost as many negative as there are positive clean components. If you think this has happened, you may want to try `clrmod true` to throw away your current model, and then iteratively issue `clean 200,0.03; keep; mapplot` until you've cleaned away all source structure and regained your previous cleaned flux. You may find it useful to change your windows at this time as well. The KEEP command is necessary to force subtraction of the clean components from the visibility data as opposed to subtraction in the image plane. Difmap stores both an "established" and a "tentative" model. The established model has been subtracted from the visibility data, while the tentative model has not been. The function of the KEEP command is to move all tentative components to the established model.

5. SAVING DATA, MODELS AND WINDOWS

At the final, or even at intermediate stages (depending on the time invested), you will want to write out your adjusted data, model, and windows. This can be done by typing

```
0>save 0749+540
Writing UV FITS file:  0749+540.uvf
Writing 60 model components to file:  0749+540.mod
wwins:  Wrote 4 windows to 0749+540.win
Writing clean map to FITS file:  0749+540.fits
Writing difmap environment to:  0749+540.par
```

The EXIT command asks the user for a prefix name to save to before exiting. A quick exit without saving can be accomplished using the command QUIT. The Difmap environment file (`name.par`) contains contour levels, mapsize, uvweighting, etc. The `.par` file created by SAVE is a command file that may be executed by typing: `@0749+540.par`. This will restore the data, model, windows and environment to what they were before the last save command with one essential difference – the antenna gains determined through self-calibration will have been applied to the data at the time of writing and these can no longer be removed with UNCAL.

Individual UVFITS, model, window, or map files may be written by typing:

```
0>wobs 0749+540.uvf
0>wmod 0749+540.mod
0>wwin 0749+540.win
0>wmap 0749+540.fits
```

There are also matching commands OBSERVE, RMOD, and RWIN to read in merge, model and window files. You can also write out the beam (`wbeam 0749+540.beam`) and a residual map (`wdmap 0749+540.map`). You may want to read §6 before writing out your final map.

6. SOME FINER POINTS ON MAPPING

Once you have satisfactorily edited and self-calibrated your data, you will want to make the best possible map for publication, or maybe just for posterity. What follows are a few suggestions to help you accomplish this:

Add a window as big as the entire field you have been mapping (or delete all clean windows using DELWIN) and clean a large number of iterations (perhaps 1000) with a very small loop gain (0.01). This will reduce the noise in the restored map as positive and negative residuals smaller than the restoring beam will be averaged out.

Use SHIFT to move your field of view around as necessary to encompass the source without a whole lot of blank sky. If you get lost, use UNSHIFT to put the data back to its original position. Then, choose an appropriate set of contour levels, such that a few positive and negative contours are made in the noise to show that you have contoured

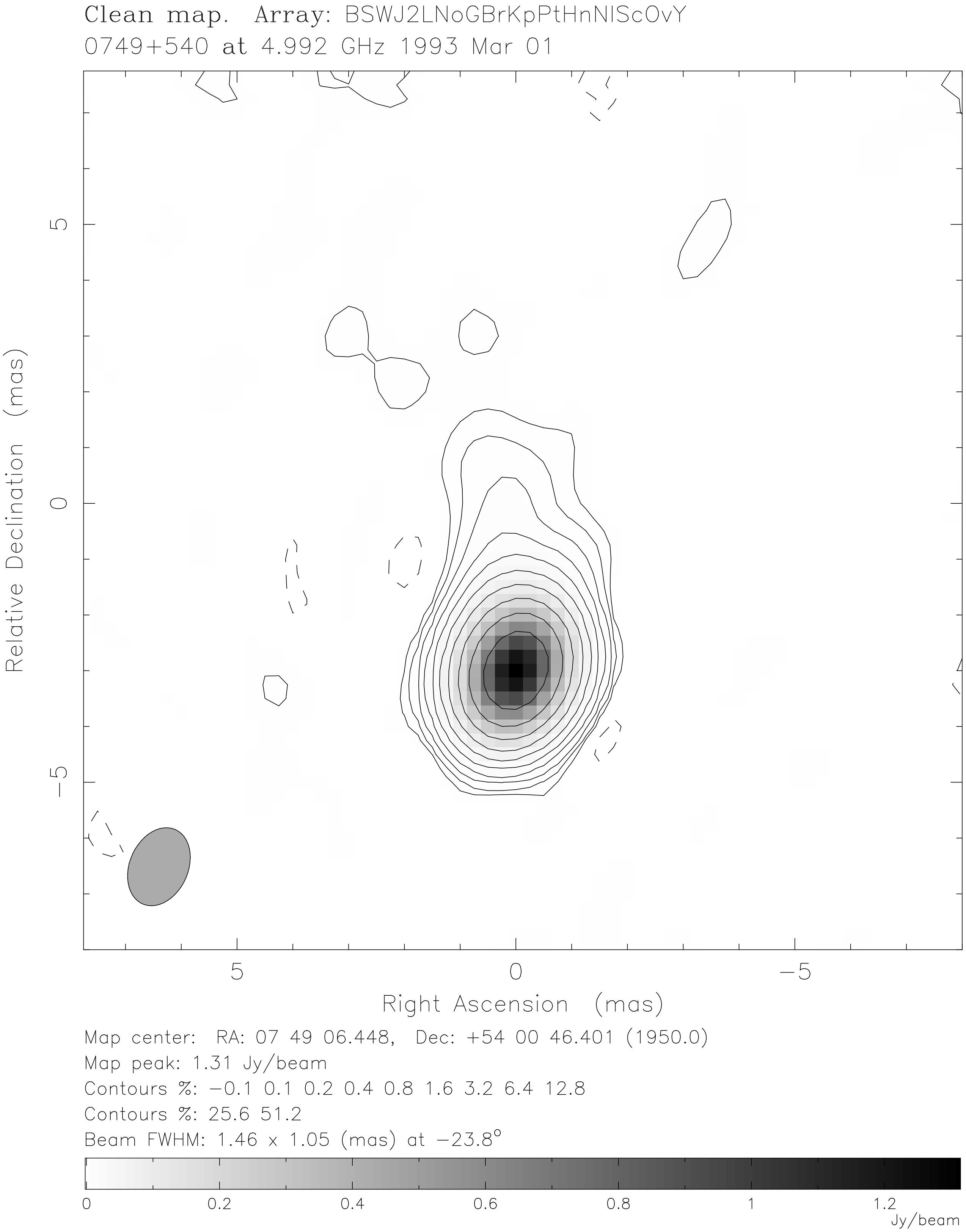

Figure 8. The final naturally weighted, 5 GHz image of 0749+540 at 1.05 × 1.46 mas resolution.

deeply enough, yet don't detract from the source. Figure 8 was made with the following commands:

```
0>uvweight 0,-1                natural weighting
0>mapsize 128,0.25
0>shift 0,-3                   shift field center north 3 mas
0>loglevs 0.1                  to set contour levels
0>mapplot cln
```

You can also try making some maps with a different restoring beam than the one estimated by Difmap. For example, to make a map with a 6 mas synthesised beam type:

```
0>restore 6
Inverting map
Restoring with beam:  6 x 6 at 0 degrees (North through East)
Clean map min=-0.006690 max=1.518844 Jy/beam
0>mapplot cln
```

For increased sensitivity to extended structure you may also want to apply a Gaussian taper to your visibility data. Type `help uvtaper` for more details on how this is done.

7. GENERATING OUTPUT FOR HARDCOPY

Any plot or map that you generate on your display screen may be redirected to a postscript file, and subsequently printed. For example, to generate Figure 8, I typed

```
0>device 0749+540.ps/vps
Attempting to open device:  '0749+540.ps/vps'
0>mapplot cln
0>device "/xw"
```

It is important to remember to reopen your display device (`/xw` in the example) before printing. Until that is done, the postscript file will remain open. Multiple pages, such as the output generated by VPLOT, can be placed in a single file. If you do not desire greyscales on your plot then type `mapcolor none` before invoking MAPPLOT. To turn off the contour plotting use `docont=false`.

8. MODEL FITTING

Model fitting is useful when it is difficult or inappropriate to make an image from the visibility data, or as a way of characterizing an image with a fairly small number of free parameters. For a good description of model fitting in general see Pearson (1994). The most commonly used model components are elliptical Gaussians, but other types are also recognized by Difmap (type `help modelfit` for the list of types).

In order to model-fit your data, you must first come up with an initial model. If you already have some experience at estimating model components you can use the command

ADDCMP or cobble up a file with your favorite editor by invoking EDMOD. To define components graphically, you may first want to clear away any existing clean component model using `clrmod`. Next start `mapplot` and then type `N` to initiate the description of a new component. To define an elliptical Gaussian will require three clicks of the `A` button: at the center of the component, at one edge to define the major axis, and at another edge to define the minor axis. Type `N` to complete the definition of the component. Remove components with `R`. The flux given to the component is simply the flux of the pixel under its center, so be aware that this may not be a very good guess, especially for a very extended component. Once you think you have a semi-decent model for your source, exit from mapplot and type:

```
0>modelfit 5
Partitioning the model into established and variable parts.
The fixed established model contains 0 components (0 Jy).
The variable part of the model contains 1 components (1.323 Jy).
There are 6 variables and 4345 usable visibilities.
This gives 2 x 4345 - 6 = 8684 degrees of freedom.
Reduced Chi-squared = Chi-squared / 8684.

Iteration 00:  Reduced Chi-squared=138.73986
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.32300v 0.0398043v 157.991v 1.17825v 0.628117v 0.00000v 1

Iteration 01:  Reduced Chi-squared=68.313182
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.44746v 0.0493185v -12.2193v 0.984119v 0.00000v 80.0371v 1

Iteration 02:  Reduced Chi-squared=9.2929050
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.46317v 0.00712541v 153.156v 0.638022v 8.97304e-09v 8.07632v 1

Iteration 03:  Reduced Chi-squared=1.2830196
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.47282v 0.000428626v 144.679v 0.506558v 0.465718v 45.0691v 1

Iteration 04:  Reduced Chi-squared=1.2362510
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.47499v 0.00127619v -169.754v 0.518428v 0.489876v 45.5630v 1

Iteration 05:  Reduced Chi-squared=1.2362465
!  Flux (Jy) Radius (mas) Theta (deg) Major (mas) Axial ratio Phi (deg) T
1.47503v 0.00125352v -169.282v 0.518532v 0.490083v 45.5749v 1
```

The above represents five iterations of model fitting, with the reduced χ^2 and the model reported at each iteration. A little `v` after each number in the model component indicates if this parameter is variable. Issuing a `mapplot map,true` will show you where your model component(s) wound up. Add additional components as necessary and `modelfit 5` again. Then use SELFCAL to correct for small phase differences between the model and the data and then try `modelfit 5` again. The command EDMOD can be used to invoke an editor with which to change the current model. Remember to write out your model file using `wmod` *filename*.

9. POLARIZATION MAPPING

All of the discussion so far has been concentrated on total intensity data and images. With the SELECT command you can choose to process any polarization available or derivable from your data (use the HEADER command to list the recorded polarizations). The names of recognized polarizations are: I Q U V RR LL RL LR XX YY XY YX. Editing is generally best performed on I, RR, or LL, but bad data can sometimes be found in the cross-hand data (RL and LR).

In the current release of Difmap polarizations must be imaged one at a time and saved separately. Stokes Q and U maps can be read into AIPS and combined there (using COMB) in order to generate plots of polarization angle and/or polarized intensity.

10. SPECTRAL-LINE DATA

Although Difmap can be used to read in, edit, and map spectral line u-v data many crucial tasks have not yet been added. Currently it is possible to process a single channel, or multiple discontinuous ranges. The desired channel range(s) are specified with the SELECT command. Edits can be applied to the selected range or to all channels. The self-calibration solutions are applied to all channels. Work is in progress and new commands related to the mapping and display of spectral-line data will be described in the next update of this cookbook. Type `help spectral` for a description of the current spectral line capability in Difmap.

11. LOOPS, COLORS, AND OTHER NIFTY TRICKS

In this chapter we explore a few handy tricks and some ways to customize Difmap to suit your tastes. For instance if you have trouble seeing the red representation of the model in RADPLOT, try typing `pgscr 2,1,1,1`. This will change color number 2 to white (the last three values specify the levels for red, green, and blue). If you always want this setting then you can put it into a special file with a name like `difmap.boot`. You'll also need to set an environment variable DIFMAP_LOGIN pointing to this file. Here is an example boot file with the above definition and a few shortcut macros defined:

```
device /xw
pgscr 2,1,1,1
#+phasecal selfcal false,false,%1
#+ampcal selfcal true,true,%1
#+natural uvw 0,-1
#+uniform uvw 2,-1
#+clr unshift; delwin; clrmod true; uncal true,true
```

The format for defining a macro is `#+name commands`. The `%1` in the above examples refers to the first input string on the command line.

Loops may also be useful. For example you could do three clean, self-cal iterations by typing:

```
0>do i=1,3
1>clean; selfcal
1>print "iteration",i,"done"; end do
```

Notice how the prompt changed to `1>` indicating that these commands were nested by one level.

Another useful command to know about is the DEVICE command. If you want to put two RADPLOTs side by side first type `device /xw,2,1`, then read in the visibility data you want to plot and issue a RADPLOT for each.

The functions IMSTAT and UVSTAT return several interesting statistics about the image and visibility data. These can be very useful when writing scripts in Difmap. See Appendix A on automatic mapping for an example script.

References

Cornwell, C., & Fomalont, E.B. 1988, in *A.S.P. Conference Series, Volume 6: Synthesis Imaging in Radio Astronomy*, ed. R.A. Perley, F.R. Schwab, and A.H. Bridle (Provo: Brigham Young Press) p. 185.

Pearson, T.J. 1994, Very Long Baseline Interferometry and the VLBA, eds. J.A. Zensus, P. Diamond, and P. Napier. ASP.

Pearson, T.J., Shepherd, M.C., Taylor, G.B., & Myers, S.T. 1995, BAAS, submitted.

Shepherd, M.C., Pearson, T.J., and Taylor, G.B. 1994, BAAS, 26, 987.

Sramek, R.A., & Schwab, F.R. 1988, in *A.S.P. Conference Series, Volume 6: Synthesis Imaging in Radio Astronomy*, ed. R.A. Perley, F.R. Schwab, and A.H. Bridle (Provo: Brigham Young Press) p. 117.

Taylor, G.B., Vermeulen, R.C., Pearson, T.J., Readhead, A.C.S., Henstock, D.R., Browne, I.W.A., & Wilkinson, P.N. 1994, ApJS, in press.

Appendix A. Automatic Mapping

While making maps is good fun, mapping hundreds (or thousands) of sources can become tiresome. Depending on the scientific goals, complexity of the sources and quality of the data, it may be possible to run Difmap in an automatic mode (Pearson *et al.* 1995). The following script is meant to be used as a starting point for an automatic mapping script that meets the requirements of your scientific program. It is NOT meant to be used as a "black box" from which maps come out ready for publication.

The following script produces fairly good maps for global VLBI observations at 6 cm using Mark 2 format recording. This script, and the following csh script are available via anonymous ftp to astro.caltech.edu (131.215.240.1) in the difmap directory.

```
!-----------------------------------------------------------------------
! Hands off mapping procedure for reasonably well calibrated and edited
! data.  Works only in versions 2.1 and later of Difmap.
!
! WARNING: The output of this script should be checked carefully!
! This script is not intended for use as a "black box" that spits
! out maps ready for publication.  Some parameters will require
! careful setting and adjustment before this script will produce
! reasonable results.  There are several common failure
! modes for this script.  A few of them are indicated below:
!
!   Failure Mode          Symptoms                 Key Parameter
! ------------------------ ----------------------   ----------------
! 1) Map size too small    high noise in map,bad fit field_size
! 2) dynam too low         too many clean boxes     dynam
! 3) dynam too high        real components missed   dynam
!
! Input:
!  uvfile  literal-string    The name of the UV FITS file.
!  field_size         int    The map dimension for a single field.
!                            This depends on the source size.
!  field_cell       float    The cell size to use with 'field_size'.
!                            This depends on the sampling of the data.
!  clean_niter        int    The number of CLEAN-iterations per cycle.
!  clean_gain       float    The CLEAN loop gain.
!  dynam            float    The minimum dynamic range required for a peak.
!                            This depends strongly on the SNR of the data.
!  soltime          float    Solution time for phase selfcal (minutes).
!                            This depends strongly on the SNR of the data.
!  thresh           float    The threshold peak clean flux above which
!                            unconstrained amplitude self-cal is viable.
!                            This depends strongly on the SNR of the data.
!  win_mult         float    Multiplier for the size of the clean windows
```

```
!                             in units of the restoring beam.
! Output:
!  One cleaned and self-calibrated map.
!
! Written by Martin Shepherd and Greg Taylor on 3/15/1994
!-------------------------------------------------------------------------
float field_size; field_size = 256
float field_cell; field_cell = 0.25
integer clean_niter; clean_niter = 50
float clean_gain; clean_gain = 0.03
float dynam;  dynam = 6.0
float soltime; soltime = 30
float thresh; thresh = 0.5
float win_mult; win_mult = 1.8
! Define the inner loop as a macro.
float old_peak
float new_peak
float flux_cutoff
#+map_residual \
flux_cutoff = imstat(rms) * dynam;\
repeat;\
 if (peak(flux) > flux_cutoff) peakwin win_mult;\
 clean clean_niter,clean_gain;\
 flux_cutoff = imstat(rms) * dynam;\
 selfcal;\
 new_peak = peak(flux);\
until(new_peak<=flux_cutoff)
! Assumes UV data has previously been read in, otherwise uncomment
! the following line:
! observe %1
! Create the map grid.
mapsize field_size, field_cell
! Self-cal to a point source.
startmod
! Start mapping the field, using uniform weighting.
uvw 2,-1
map_residual
print "************** FINISHED UNIFORM WEIGHTING CLEAN **************"
! See if natural weighting reveals any further flux in the current field.
uvw 0,-1
win_mult = win_mult * 1.6
clean_niter = clean_niter * 2
dynam = dynam - 0.5
map_residual
```

```
print "************** FINISHED NATURAL WEIGHTING CLEAN **************"
! check antenna gain calibration (computes 1 scale factor/antenna)
gscale true
dynam = dynam - 0.5
map_residual
! Amplitude self-cal with a long solution time.
selfcal true, true, soltime
dynam = dynam - 0.75
clean clean_niter,clean_gain
selfcal
map_residual
print "************** FINISHED AMPLITUDE SELF-CAL **************"
! Restore the map and if the peak flux in the clean map is over a certain
! threshold then run an unconstrained selfcal on amplitude as well as phase.
restore
if(peak(flux) > thresh)
  selfcal true, true
  clean clean_niter,clean_gain
  selfcal
  map_residual
  selfcal true, true
end if
! one last clean/selfcal loop for luck
clean
selfcal
! save data
save %1
! now clean the entire map
x = (field_size-8) * field_cell / 4
addwin -x,x,-x,x
clean (field_size*4),0.01
keep
! show the map (uncomment and put in a bottom contour level
! in cmul in units of Jy/beam if you want this displayed)
!device /xw
!cmul=0.001
!loglevs (imstat(rms)*3),10000,2
!mapl cln
```

If the previous script was located in the file muppet, the csh script below could be used to automatically run our mapping script on a UV data file. It also used MAPPLOT in the Caltech VLBI Software Package to make a postscript plot.

```
#!/bin/csh
# runs automatic mapping script muppet on file $1 to generate
# a fully self-calibrated and cleaned naturally weighted map. Then
# make postscript file with power-2 contours starting at X Jy/beam.
#
# usage: automap source.edt  ! start with edited data
# output: source.cmp_n    ! naturally weighted map
#         source.ps       ! contour postscript plot
#         source.uvf      ! self-calibrated uvdata
#         source.mod      ! set of clean components
#         source.win      ! clean windows
#         source.par      ! difmap parameters file
#         source.log      ! log file
#
# written 03/20/94 by Greg Taylor
#================================================================
# set the bottom contour level on the following line in Jy/beam
set x = "0.002"
#
# get root of source name
set n = $1:r
#
# remove previously generated postscript file, if any
test -f $n.ps; if ($status == 0) then
    \rm $n.ps
endif
#
# remove previous difmap log files
test -f $n.log; if ($status == 0) then
    \rm $n.log*
endif
#
# load uvdata and run automatic mapping script muppet
dif << eodif
    logfile $n.log
    obs $1
    @muppet $n
    wmap "$n.cmp_n"
    quit
eodif
#
\rm $n.fits
#
# get vector agreement factor between model and data and maximum in
map
set fit = `nawk '/Fit after self-cal/  fit = $6; END  print fit'
$n.log`
```

```
    set peak = `nawk '/Clean map min=/  last = $5 ; END print last'
$n.log`
    #
    # use mapplot to make a contour plot of the final map
    mapplot << eof
       opaque
       mapfil $n.cmp_n
       topleft="\fr   topright="\fr   charsize=1.4,1.0
       title="","$fit, $peak"
       annot=-1
       beam
       plev 0
       clev $x
       levs=-1,1,2,4,8,16,32,64,128,256,512,1024,2048
       plotfil "$n.ps/vps"
       /
    eof
    #
    # all done
    exit 0
```

Appendix B. List of Commands

```
 General help topics:
 -------------------
 editing
     General discussion of editing in difmap.
 models
     An overview of how models are used in difmap.
 multi_if
     Overview of multi-IF mapping.
 polarization
     Overview of Difmap polarization mapping.
 spectral_line
     An overview of the current spectral-line facilities.
 subarrays
     A discussion of sub-array handling in Difmap.
 whatsnew
     The details of recent difmap changes.
 Functions and commands:
 ----------------------
 addcmp
     Add a model component by hand to the tentative model.
 addhist
     Append a new line to the recorded history of an observation.
 addwin
     Append a new entry to the list of clean windows.
 basename
     Returns the name of baseline 'ibase' of sub-array 'isub'.
 clean
     Clean a residual dirty map with the dirty beam.
 clrhist
     Clear the history information recorded with the observation.
 clrmod
     Clear one or more of the established, tentative and continuum models.
 clroff
     Undo all resoff calibration-offsets applied during this difmap
session.
 corplot
     Display accumulated telescope based self-cal corrections versus UT.
 cpplot
     Interactive display and editing of observed and model closure phases.
 delwin
     Delete all clean windows.
 edmodel
     Modify the established and tentative models with an external editor.
 get
     Restore UV data, models and windows from files.
 getif
     Select the default IF.
 gscale
     Self calibrate to determine overall telescope amplitude corrections.
 header
     List useful parts of the observation header on one's terminal.
```

```
 imstat
     Return image plane map statistics.
 invert
     Grid selected UV data and Fourier invert it to a (residual) dirty map
 keep
     Establish the latest CLEAN model and compute its UV representation.
 loglevs
     Set logarithmic contour levels for mapplot.
 mapcolor
     Select the colormap type, brightness and contrast to use in mapplot.
 mapfunc
     Change the transfer function and data range displayed in mapplot.
 maplot
     Alternative name for the mapplot command - see help mapplot.
 mapplot
     Display the dirty beam, residual or restored map and modify CLEAN
windows.
 mapsize
     Request a new map grid of given width and cell size
 mapunits
     Change the units used to specify and label map and UV plane
distances.
 modelfit
     Fit image-plane model components to visibilities in the UV plane.
 nbase
     Return the number of baselines in a given sub-array.
 nchan
     Return the count of the number of spectral-line channels per IF.
 nif
     Return the number of IFs in the current observation.
 nsub
     Return the number of sub-arrays in the current observation.
 ntel
     Return the number of telescopes in a given sub-array.
 observe
     Read UV data from a random-groups UV FITS file.
 peak
     Return details of the min,max or absolute max flux in a map.
 peakwin
     Place a CLEAN window around the brightest pixel in the residual map.
 projplot
     Plot visibility amplitudes and/or phases versus projected UV
distance.
 radplot
     Plot visibility amplitudes and/or phases versus UV radius.
 resoff
     Determine and apply a baseline based amplitude and phase correction.
 restore
     Add the CLEAN model to the residual map to form a clean map.
 rmodel
     Read a new CLEAN model from a VLBI model file.
 rwins
     Read windows previously saved CLEAN windows from a text file.
```

```
 save
     Save UV data, models, windows, the restored map and a command file.
 scangap
     Change the time gap used to delimit neighboring scans.
 select
     Select the polarization and range of spectral-line channels to be
processed.
 selfant
     Set antenna based constraints for subsequent self-calibration.
 selfcal
     Self calibrate for telescope phase (and amplitude) corrections.
 selfflag
     Control the identification and fate of un-correctable visibilities.
 selflims
     Set limits on acceptable amplitude and phase corrections in selfcal.
 selftaper
     Sets a (1 - gaussian) taper to weight down SHORT baselines.
 setcont
     Add the current model to the continuum model.
 shift
     Shift the observation phase-center, CLEAN window and model positions.
 showhist
     Display the history that was recorded with the current observation.
 showpar
     Display a list of difmap configuration parameters.
 startmod
     Read a starting model from disk and phase self-calibrate against it.
 telname
     Returns the name of telescope 'itel' of sub-array 'isub'.
 tplot
     Plot time sampling for each telescope of an observation.
 uncalib
     Undo recorded telescope calibrations (eg. self-cal corrections).
 unshift
     Undo accumulated position shifts from the data, windows and models.
 uvaver
     Time average a UV data set.
 uvplot
     Plot observed UV points on the UV plane.
 uvrange
     Set the min and max UV radii of usable visibilities for gridding.
 uvstat
     Return UV plane visibility statistics.
 uvtaper
     Set a gaussian taper to weight down long baselines and reduce
sidelobes
 uvweight
     Set the type and degree of weighting for subsequent UV gridding.
 uvzero
     Set the estimated flux expected from a baseline of zero length.
 vplot
     Generalized visibility plotting and editing command.
```

```
wbeam
    Save the dirty beam to a FITS file.
wdmap
    Save the residual dirty map to a FITS file.
winmod
    Delete all components outside (or inside) the current clean windows.
wmap
    Save the clean (restored) map to a FITS file.
wmodel
    Write CLEAN models to a VLBI model file.
wobs
    Save UV data to a random-groups UV FITS file.
wtscale
    Change the scale factor that mutiplies the raw visibility weights.
wwins
    Save CLEAN windows to a text file.
xyrange
    Set the bounds of the sub-image to be displayed in mapplot.
```