Mapping CookBook
IRAM Plateau de Bure Interferometer

S.Guilloteau[1], R.Lucas[1], K.Bouyoucef [1]

March 16, 1999

Version 1.3

(1) Institut de Radio Astronomie Millimétrique
300 Rue de la Piscine
F-38406 Saint Martin d'Hères

This document presents guidelines for producing images from IRAM Plateau de Bure Interferometer data. Only the UV processing part is specific to the interferometer, however, so that many of the image display facilities can also be used for data from the 30-m telescope.

Related information is available in:

- IRAM Plateau de Bure Interferometer: Introduction
- IRAM Plateau de Bure Interferometer: OBS Users Guide
- IRAM Plateau de Bure Interferometer: Atmospheric Calibration
- IRAM Plateau de Bure Interferometer: Calibration Cookbook
- CLIC: Continuum and Line Interferometric Calibration
- GILDAS: Grenoble Image and Line Data Analysis Software
- GreG: Grenoble Graphics

Contents

1 Introduction 4

2 Starting 4
2.1 Creating UV Tables . . . 4
2.2 Starting with mapping programs . . . 5

3 The Naked Tasks and Display Tools 5
3.1 Tasks . . . 5
3.1.1 Squeezing UV tables . . . 5
3.1.2 From UV data to Images . . . 6
3.1.3 Image Deconvolution . . . 6
3.1.4 UV analysis . . . 6
3.2 Display Tools . . . 8
3.3 Image Analysis . . . 8

4 The Tailored Mapping Environment 9
4.1 The INPUT and GO macros . . . 9
4.2 Mapping with the Tailored environnment . . . 9
4.3 UV Plots . . . 10

5 Mapping guidelines 10
5.1 Old correlator data . . . 10
5.2 Continuum maps . . . 11
5.3 Spectral line maps . . . 11
5.4 Map size . . . 11
5.5 Mapping and Cleaning . . . 12

6 The MAPPING program 12
6.1 Sample Map-making MAPPING session . . . 12
6.2 Sample Cleaning MAPPING session . . . 13
6.3 Sample WIPE MAPPING session . . . 14
6.4 Mosaics . . . 15
6.4.1 Mosaic Creation . . . 16
6.4.2 Cleaning . . . 16
6.5 WIPE . . . 17
6.5.1 WIPE Regularization . . . 17
6.5.2 WIPE Deconvolution . . . 17
6.5.3 Wipe Error Analysis . . . 22
6.5.4 Important Remarks . . . 23

7 MAPPING internal help 24
7.1 CLARK . . . 24
7.2 FIT . . . 24
7.3 HOGBOM . . . 24
7.4 LOAD . . . 25
7.5 MOSAIC . . . 25
7.6 MRC . . . 25

7.7 READ . . . 26
7.8 SDI . . . 26
7.9 SUPPORT . . . 26
7.10 UV_FLAG . . . 26
7.11 UV_MAP . . . 27
7.12 UV_SHOW . . . 27
7.13 UV_SORT . . . 27
7.14 UV_STAT . . . 27
7.15 WIPE . . . 28
7.16 WRITE . . . 28

1 Introduction

Mapping is the last stage in processing data from the IRAM Plateau de Bure Interferometer (PdBI). Since this is the last stage, we assume you are familiar with interferometry concepts, and you have good knowledge of the **CLIC** program. Here we introduce parts of the **GILDAS** (Grenoble Image and Line Data Analysis Software) software that have not been used in previous stages of data analysis.

Data calibration consisted of using a single (big) program from the **GILDAS** package, **CLIC**, and a single type of data, the .IPB files. In the mapping stage, the astronomer will use all the tools of **GILDAS**: a big program for data display, and many image processing programs called *Tasks*. He (or she) will also manipulate two or three kinds of data: *Images, Tables, and UV Tables.* An introduction to **GILDAS** is available in the **GILDAS** manual. One of the essential feature of **GILDAS** is its orientation towards spectral line mapping: most tasks handle 3-dimensional data cubes, processing all frequency planes in the same way.

The astronomer has three basic ways to proceed:

- the naked GRAPHIC and *Tasks*
  The GRAPHIC program is used to activate tasks (via command RUN) and to display the task results. All tasks are available in such a way.
- the tailored GRAPHIC and *Tasks*
  For the most frequently used tasks, procedures have been prepared to ease parameter specifications. Interaction with the task is performed from GRAPHIC via the procedure INPUT and GO. The normal RUN command is still available for those tasks wich cannot be controlled by INPUT and GO.
- **MAPPING**
  MAPPING is a superset of GRAPHIC with additional commands to make maps and deconvolve them. Because of its full interactivity, **MAPPING** offers more flexibility than the basic tasks. However, new developments are often available as tasks before being incorporated in **MAPPING** . Since **MAPPING** is a superset of GRAPHIC, the two other possibilities can be used directly from **MAPPING** .

2 Starting

2.1 Creating UV Tables

UV Tables are just special *Tables* suited for UV data handling that are created by **CLIC**. Mapping consists of transforming these tables into something more meaningful for the astronomer, either *Images* or numbers like positions, flux densities, sizes, etc.

UV Tables are created by **CLIC** using the `TABLE` command. The `TABLE` command offers several options in creating these tables. Four major options should be considered:

1. Weighting parameters. When the calibration has been carried out correctly, the weights should include both the `TSYS` correction and the amplitude `CALIBRATION` factor. Then proper weights are given to every data point, so all analysis programs can estimate errors consistently.

2. Frequency reference for spectral line observations. The `/FREQUENCY` option should be used to compute a velocity scale relative to a given spectral line rest frequencies.

3. `/RESAMPLE` option should be used when not all channels are required (or when the spectral resolution can be smoothed). `/RESAMPLE Nc` will select only the first `Nc` channels of a spectral line subband. `/RESAMPLE Nc Ref Val Res Type` offers a more general resampling.

To summarize, a **CLIC** procedure like the following is appropriate to create a table:

```
SET RECEIVER 1                      ! Select receiver number
SET RF ON FREQUENCY ANTENNA         ! Apply RF passband calibration
SET AMPLITUDE RELATIVE ANTENNA      ! Apply Flux calibration
SET PHASE RELATIVE ANTENNA INTERNAL ! Apply Phase calibration
SET SELECTION LINE USB LO1          ! Select the spectral line subband
SET WEIGHT TSYS ON                  ! Apply Tsys correction on noise
SET WEIGHT CAL ON                   ! Apply efficiency correction on noise
FIND/PROC CORR/TYPE O               ! Select Correlation data on Object.
! and create the table...
TABLE Name NEW/FREQUENCY "SiO" 86243.45/RESAMPLE 32
```

Of course, arguments of the option depend on your particular case. Please refer to the **CLIC** manual for more details. Note that the frequency is in MHz.

2.2 Starting with mapping programs

Creating *UV Tables* was the last stage requiring **CLIC**. Further on, only GRAPHIC, *Tasks* and/or **MAPPING** will be required. To get started, we recommend to activate the GRAPHIC display program, which can be run from any graphic terminal, since the *Tasks* can only be executed by command RUN. Use the `EXPLAIN` command whenever help on some task or kind of tasks is required.

3 The Naked Tasks and Display Tools

3.1 Tasks

Each task is documented on line by the `EXPLAIN` *Taskname* command. More thematic explanations are also available, using `EXPLAIN Summary`.

3.1.1 Squeezing UV tables

Because of flagged data and unused part of the IF bands, **CLIC** tends to produce bigger UV tables than necessary. Using the `/RESAMPLE` option of the `TABLE` command allows the selection of the minimum number of channels around a spectral line of interest, or the channel averaging required.

An alternative possibility is to use the *UV_EXTRACT* task to select channels, and the *UV_COMPRESS* task to produce channel averaged UV tables.

3.1.2 From UV data to Images

The following tasks can be used to produce images from UV data.

- *UV_MAP*
  This task sorts UV data and grid it to produce an image and a beam using an FFT. It is fast and efficient. Both the image and the beam can be 3-D cubes. *UV_MAP* can produce one beam per channel, or a common beam for all channels. It can also rotate and shift the field of view, though preserving the pointing information for final correction by the primary beam.
- *MX*

  MX is a combination of *UV_MAP* and *CLEAN* (see below) which makes a deconvolved map directly from a *UV* table.

3.1.3 Image Deconvolution

The CLEAN method is the major image deconvolution method available in **GILDAS** (several variants can be used). Recently, a new deconvolution method, WIPE, has been implemented in **MAPPING** .

- *CLEAN*
  The *CLEAN* task implements several variants: the basic algorithm as described by HOGBOM, the CLARK modification which cleans most of the image, and a Multi-Resolution-Clean (MRC). The HOGBOM and CLARK methods allow both the image and the beam to be 3-D cubes. For the MRC method the same beam must be used to clean all planes.
- *MX*
  The *MX* tasks implements a variant of the CLARK method, where clean components are substracted from the visibilities at each major cycles, rather than from the map. Aliased sidelobes are therefore suppressed.
- **MAPPING**
  is an interactive deconvolution program described below. **MAPPING** is more flexible and more powerful than CLEAN, and allows proper deconvolution of Mosaics. **MAPPING** includes new deconvolution methods like WIPE. **MAPPING** also includes interactive UV editing.

3.1.4 UV analysis

In addition to the basic *UV_MAP*, the following tasks can also be used to perform analysis or operation on UV tables.

- *UV_ASCAL*
  Self-Calibrate a UV data set. It references the Phases and Amplitudes to a point source measured (almost) simultaneously.
- *UV_CENTER*
  Find the position of peak emission (for a presumably centro-symmetric brightness distribution) from UV data.

- *UV_CLIP*
  Clip *UV* data to flag bad points that escaped previous notice.
- *UV_COMPRESS*
  Compress *UV* data by spectral channel averaging.
- *UV_EXTRACT*
  Extract some spectral channels from *UV* table.
- *UV_FIT*
  Fit radial brightness distributions to *UV* data, and determine the center of symmetry.
- *UV_FITC*
  A less intelligent, but faster radial brightness distributions fitting to *UV* data. The source is assumed to be centered at the phase tracking center. Combination of two simple source models is allowed, if both are centro-symmetric around the same position.
- *UV_FLAG*
  Flag data in a *UV* table, following conventions similar to those used in **CLIC**.
- *UV_MERGE*
  Can merge two different *UV* tables.
- *UV_SHIFT*
  Displace phase center of a UV data set, and optionally rotate the axis.
- *UV_SHORT*
  Make a "short spacings" *UV* table from a single-dish map (30-m or other telescope). Can be merged later with another *UV* table. Caution: there are some limitations about single-dish map properties for this method to work.
- *UV_SINGLE*
  Make a "short spacings" *UV* table from a single-dish table (30-m or other telescope, as produced by **CLASS**command `GRID`). No restriction of use. Can be merged later with another *UV* table.
- *UV_STAT*
  Make some "statistics" on a *UV* table. This task actually evaluates the beam size and noise level under various weighting and tapering combinations. It can be used prior to UV_MAP to get reasonable mapping parameters (specially UV_CELL and MAP_CELL).
- *UV_ZERO*
  Add zero spacing flux to a *UV* table (continuum or line).
- *UV_SORT*
  This task is a necessary step before some tasks can be used. It sorts *UV* data in an order appropriate for further gridding and mapping (negative and increasing V values) (*UV* order), or self-calibration and time averaging (*TB* order).

3.2 Display Tools

The `IMAGE` command and the **GreG** part of the GRAPHIC program can be used to display images (contour maps) and UV data. To display images, use the following basic commands:

```
IMAGE Filename    ! Connect to image file
LIMIT/RG          ! Compute limits of plots
LEVELS List       ! Set contour levels
RGMAP             ! Draw contours
BOX/UNIT SEC      ! Draw box, using arcseconds as angular units
```

See the **GreG** and **GILDAS** manuals for further details.

UV data also needs to be displayed. The UV tables are produced naturally in visibility order, which is not suited for display. Task *TRANSPOSE* or command `TRANSPOSE` in language VECTOR must be used first. We recommend these file extensions: .UVT for UV tables, and .TUV for their transposed counterparts. A .TUV table has the following format

```
Column 1   U coordinate in meters
Column 2   V coordinate in meters
Column 3   W coordinate in meters
Column 4   Day (in CLASS units, a Modified Julian Date)
Column 5   Time in seconds since 0:00 UT
Column 6   Start antenna
Column 7   End antenna
Column 8   Real part of channel 1 visibility
Column 9   Imaginary part of channel 1 visibility
Column 10  Weight of channel 1
Column 11, 12, 13,  etc. for other channels
```

.TUV tables can be plotted using the following commands

```
COLUMN X Ix Y Iy/TABLE Table.TUV ! Ix and Iy are the interesting
LIMITS                           ! column numbers
POINTS
BOX
```

or directly using Tables as **SIC** variables

```
DEFINE IMAGE UV Table.TUV READ
LIMITS/VARIABLE UV[Ix] UV[Iy]    ! Ix and Iy as above
POINTS UV[Ix] UV[Iy]
BOX
```

The GRAPHIC program can display data cubes in bit-map format.

3.3 Image Analysis

The user should remember that transpositions of the data cubes are often required in the final analysis stages to obtain velocity-position plots for example. The tasks *TRANSPOSE*, *REPROJECT* (change the coordinate system), *EXTRACT* (select a subset of a cube, or merges cubes), and *SLICE* (cut a slice through a cube) are useful tools to remember.

Standard fitting routines are available (1-D gaussian for spectral line profiles, 2-D gaussian for image features). Smoothing and image filtering tools also exist in **GILDAS** . However, for optimal sensitivity smoothing or high-pass filtering are better done by appropriate weighting functions in the UV plane, before mapping (i.e selecting the appropriate UV taper). This requires to go through the mapping and cleaning stages, unfortunately.

4 The Tailored Mapping Environment

4.1 The INPUT and GO macros

To help users and simplify routine operations, we have developped macro commands for mapping and display. These macros use **SIC** global variables that the user can change with simple `LET` assignments, and are accessible through the INPUT and GO symbol.

The INPUT symbol remembers the last operation described. INPUT INPUT will list the available macros, currently

```
INPUT & GO :  Commands for multichannel mapping:

- UV_MAP      compute an image from UV data
- MX          compute an image from UV data and clean it
- CLEAN       clean a dirty image
- LEVEL       compute best contour level list
- LIST        make a alphanumeric list file of UV data
- FLUX        measure the total flux in a user defined area
- FLAG        temporary flag some  UV data
- UVALL       make plots of UV data for all or selected channels
- MAP         plot a contour map of all or selected channels
- BIT         plot bitmap and contours of all or selected channels
- NICE        a nicer version of MAP
- SPECTRE     plot spectra from a data cube
- POS         plot a contour map for velocity-position cubes, or
              rotated images.
Type: INPUT command-name  for further information on a command
Type: GO command-name  to execute a command
```

`INPUT Macro` will give some help and the parameters of the specified macro. `GO Macro` will activate the specified macro, using its current parameters to run the **GILDAS** tasks which may be referred in the body of the macro. Variable names in the tailored environment are similar to the parameter names of the naked tasks; to get help you should thus use command `EXPLAIN Task Parameter`.

4.2 Mapping with the Tailored environnment

A standard session using the `INPUT` and `GO` macros would be the following

```
INPUT                ! Get started once.
LET NAME The_Object  ! Enter your basic UV table name.
```

```
INPUT UV_MAP          ! Look at the UV_MAP input parameters, and if
GO UV_MAP             ! all other INPUT parameters are OK, proceed.
LET TYPE LMV          ! Specify file extension for MAP.
LET FIRST 1           ! Look at spectral channels 1 to
LET LAST 15           ! 15 of your map
LET BEAM DIRTY        ! together with a dirty beam plot.
GO MAP                !
LET NITER 300         ! Estimate number of clean components from map.
GO CLEAN              ! Clean your data cube.
LET TYPE LMV-CLEAN    ! Now display the clean images,
GO MAP                ! if OK, your data reduction is finished.
```

If not OK, you may have to go back, selecting different UV tapers, map cell size, weighting function, clean parameters, etc., to produce the best images. You may even have to flag some discrepant data, using for example the *UV_CLIP* task, or going back to **CLIC**.

4.3 UV Plots

Although images are a wonderful way to display results, problems may be more obvious in UV plots of visibilities as a function of UV radius or time. `GO UVALL` will make these plots. This macro is controlled by 3 basic parameters: the *UV Table* name (`NAME`), the type of data to be plotted along the X and Y axis (`XTYPE` and `YTYPE`). These two variables can take the following values:

```
U and V       Fourier coordinates
ANGLE         UV position angle (Atan(V/U))
RADIUS        UV distance (Sqrt(U^2+V^2)
TIME          UT Time of visibility (in hours)
DATE          Date of visibility (in days)
NUMBER        Visibility number
REAL          Real part of the visibility
IMAG          Imaginary part of the visibility
AMP           Amplitude of the visibility
PHASE         Phase of the visibility
WEIGHT        Relative weight of the visibility
SIGMA         RMS noise on REAL and IMAG
```

First and last channels can also be specified.

5 Mapping guidelines

We mention here a number of tricks and potential traps in making maps from PdBI data.

5.1 Old correlator data

In the old correlator system, 10 wide band channels, C01 to C10, were always available in parallel with the spectral correlator. With this system, if continuum emission is your primary interest, we recommend you create the *UV* table using a `SET SELECTION CONTINUUM DSB C01 TO C10` command. This has the advantages of creating a small table, and automatically ignoring flagged subbands.

One possible problem is possible contamination by line emission. One workaround is to set a `Mark` on the corresponding correlator subband(s) when creating the table. However, this will mark both the USB and LSB receiver bands, thus loosing some sensitivity. An alternative is to create the *UV* table using `SET SELECTION CONTINUUM USB C01 TO C10` command and then add the other sideband by changing the selection to `CONTINUUM LSB`. The sensitivity can now be maximized, and since `USB` and `LSB` are stored in different visibilities, there is no difference in the data processing. Continuum observations at different frequencies can also be merged in such a way (but beware of spectral index problems).

Using `SET SELECTION LINE` may also be used (and can help checking upon line contamination), but a drawback is that synthesized beams may differ from one channel to another because of data flagging. Beam cubes must be produced by *UV_MAP* and used by *CLEAN*

5.2 Continuum maps

With the new correlator system, it is better to create continuum tables from the "line" backend, using command
`SET SELECTION CONTINUUM DSB L01 TO L06`
for example. When line contamination is possible, the best way is to build the UV table in pieces, as for example in the following command procedure:

```
FIND
SET SELECTION CONTINUUM USB L01 TO L04
TABLE CONT NEW
SET SELECTION CONTINUUM LSB L01 TO L02
TABLE
SET SELECTION CONTINUUM LSB L04 TO L06
TABLE
```

which creates a continuum table for data contaminated in subbands L05 L06 in upper sideband, and subband L03 in lower sideband.

5.3 Spectral line maps

If spectral line emission is the prime interest, the *UV* table must be created using a `SET SELECTION LINE USB L01` command for example. Flags are not channel based for the spectral correlator, but only subband based. Accordingly, the *UV_MAP* and *CLEAN* tasks can work on 3-D images with a common synthesized beam for all channels.

5.4 Map size

The map cell must be small enough to properly sample the synthesized main beam: a minimum of 3 cells per beam is required for CLEAN to work properly. Task *UV_STAT* can be used to determine the appropriate cell size.

The number of pixel along each dimension must be a power of two for efficient use of most image processing algorithm. Since CLEAN works properly only in the inner quarter of a map, the map size should be twice the region of interest. Remember the primary beam size of the 15-m antennas is 55" at 90 GHz, 22" at 230 GHz.

Map size can be reduced by adequate centering, using the UV_SHIFT variable and associated parameters in *UV_MAP* or *MX*, or the *UV_SHIFT* task before mapping. These parameters can also be used to align the map axis with a feature of interest, by rotation of the UV coordinates. This is especially interesting for bipolar flows or edge on galaxies, for example.

5.5 Mapping and Cleaning

Maximum sensitivity is obtained with natural weighting, and no taper. Because the phase noise may increase with baseline length, however better image quality may be obtained with a smaller taper, at the expense of spatial resolution. Maximum resolution is obtained using uniform weighting, and no taper. The uniform cell size can control to some extent the beam shape. Bigger cells will tend to increase noise, however. Uniform weighting enhances the contribution of (and the noise from) the long baselines. Task *UV_MAP* allows a compromise between uniform and natural weighting, by arguments UV_CELL. Best compromise can be determined using task *UV_STAT*.

In using CLEAN, avoid the following traps: cleaning to deep, undersampling the beam, cleaning more than the inner quarter of a map, or restoring with too small a beam.

Many astronomers believe that *MX* produces better results than *UV_MAP* and *CLEAN*. This is not true: *UV_MAP* and *CLEAN* will produce results very similar to *MX* provided they are used on images twice bigger. Despite this greater storage use, using *UV_MAP + CLEAN* is usually **much** faster than *MX*, and provides the advantage of checking the aspect of the dirty image.

In practice, never use MX at the beginning. MX is only worth (for Plateau de Bure data) for "final" deconvolution, if at all.

6 The MAPPING program

MAPPING is a program which allows interactive construction and/or cleaning of images. Rather than producing intermediate files for each attempt, this program reads in only once the UV data set (or dirty beam and image), and lets the user decide when saving an image is useful

6.1 Sample Map-making MAPPING session

A basic session to make maps with **MAPPING** is the following

```
CLEAN\READ UV  file.uvt          ! 1
CLEAN\UV_STAT WEIGHT             ! 2
SIC\LET MAP_CELL 0.5             ! 3
SIC\LET MAP_SIZE 128             ! 4
CLEAN\UV_MAP                     ! 5
CLEAN\FIT                        ! 6
GTVL\DEVICE IMAGE                ! 7
SHOW DIRTY                       ! 8
...                              ! 9
CLEAN\UV_FLAG                    ! 10
CLEAN\UV_MAP                     ! 11
...                              ! 12
CLEAN\WRITE DIRTY file.lmv       ! 13
CLEAN\WRITE BEAM  file.beam      ! 14
```

1. Read the UV table from file "file.uvt"
2. Analyse the UV data set to select appropriate weighting parameter and pixel size.
3. Specify the pixel size in arcseconds
4. Specify the map size in pixels
5. Make the map and beam. All other UV_MAP parameters can be listed by command INPUT UV_MAP
6. Determine the size of the dirty beam
7. Connect to an X-Window terminal for colour display
8. Show a bitmap of the dirty image. SHOW is a symbol defined to execute a command procedure which plots the image.
9. Iterate points 4, 5 and 8 until you are satisfied with the result.
10. Possibly flag some UV data points.
11. And recompute the map and beam.
12. If needed Iterate points 10 and 11
13. And once you are happy, write the dirty image
14. and beam on files.

Instead of point 13 and 14, you may directly proceed within MAPPING to the Cleaning stage.

6.2 Sample Cleaning MAPPING session

A basic Cleaning session with MAPPING is the following

```
CLEAN\READ DIRTY  file.lmv       ! 1
CLEAN\READ BEAM   file.beam      ! 2
SIC\LET NITER 200                ! 3
SIC\LET ARES  0.10               ! 4
SIC\LET FRES  0                  ! 5
SIC\LET GAIN 0.2                 ! 6
GTVL\DEVICE IMAGE                ! 7
FOR I 1 TO Nplane                ! 8
SHOW DIRTY I                     ! 9
CLEAN\SUPPORT/CURSOR             ! 10
CLEAN\HOGBOM                     ! 11
CLEAN\CLARK/PLOT                 ! 12
SHOW CLEAN                       ! 13
...                              ! 14
GREG2\WRITE POL file-'i'.pol     ! 15
NEXT                             ! 16
CLEAN\WRITE CLEAN file.lmv-sky   ! 17
```

1. Read the dirty image from file "file.lmv"
2. Read the associated dirty beam. Note that point 1 and 2 are not necessary if you have produced your dirty image within **MAPPING** .
3. Specify the number of clean components
4. Specify the absolute residual level
5. Specify the fractional residual (0 means use absolute level)
6. Specify the loop gain
7. Connect to an X-Window terminal for colour display
8. Start a loop over all channels
9. Show a bitmap of plane I of the dirty image. SHOW is a symbol defined to execute a command procedure which plots the image.
10. Define where clean components must be searched for for this plane, using the cursor to create a polygonal area.
11. Now make a simple HOGBOM clean.
12. or a faster CLARK-like one. Adjustment of the loop gain is possible after each major cycle. Note that HOGBOM and CLARK operate on the first plane defined in command SHOW
13. Show a bitmap of the corresponding plane of the clean image, to check if the result is satisfactory.
14. Iterate points 10, 11 or 12 and 13 until you are satisfied with the result.
15. optionally save the polygon which defined the support of the clean components in this plane.
16. Loop over all image planes
17. And once all planes have been cleaned, write the clean image on a file.

That's it. **MAPPING** should be used whenever the standard "CLEAN box" is not sufficient to specify where is located the true emission. The additional flexibility of modifying the loop gain in the CLARK method allows much better deconvolutions than using the standard techniques.

6.3 Sample WIPE MAPPING session

WIPE uses most of Clean inputs. A basic WIPE Cleaning session with MAPPING is the following

```
CLEAN\READ UV  file.uvt          ! 1
SIC\LET MAP_SIZE 128             ! 2
SIC\LET MAP_CELL 0.5             ! 3
SIC\LET WCOL 10                  ! 4
SIC\LET MCOL[1] 10               ! 5
SIC\LET MCOL[2] 10               ! 6
SIC\LET APERTURE[1] 1.2          ! 7
```

```
SIC\LET APERTURE[2] 1.2          ! 8
SIC\LET WEIGHT_MODE N            ! 9
CLEAN\UV_MAP                     ! 10
SHOW DIRTY                       ! 11
POLYGON                          ! 12
WIPE                             ! 13
CLEAN\WRITE NEAT file.lmv-neat   ! 14
CLEAN\WRITE BEAM  file.beam      ! 15
CLEAN\WRITE DIRTY file.lmv       ! 16
```

1. Read the uv table from file "file.uvt"
2. Specify the map size in pixels
3. Specify the pixel size in arcseconds
4. Select the channel to use for the weight
5. Specify the first channel to process
6. Specify the last channel to process
7. Select an amplification factor that will be used for the major axis of the ellipse defining the aperture to be synthesized
8. Select an amplification factor that will be used for the minor axis of the ellipse defining the aperture to be synthesized
9. Specify the weighting mode to be used, Natural or Uniform
10. Process the dirty map of selected channels
11. Show a bitmap of the dirty image. SHOW is a symbol defined to execute a command procedure which plots the image.
12. Define where clean or WIPE components must be searched for, using the cursor to create a polygonal area.
13. Now enter the main part of WIPE
14. And once all planes have been cleaned, write the neat image on a file.
15. write the beam in a file
16. write the dirty map in a file

Command lines 10 to 12 are optional and performed only if the user wants to define on the Dirty Map a "Clean Box" like support for the reconstruction process.

6.4 Mosaics

MAPPING is also able to clean mosaics. Mosaics are ensemble of overlapping fields, with typically half power primary beam overlap, and similar (but not identical) UV coverages for each field.

6.4.1 Mosaic Creation

Individual images for every field should be created with a common phase tracking center. The UV_SHIFT, MAP_RA and MAP_DEC variables in task *UV_MAP* must be used for this. The image names must derive from a generic name as follows (e.g. for a generic name MAP):
MAP-1.LMV, MAP-2.LMV, ...MAP-N.LMV
where N is the number of fields. For example

```
LET MAP_SIZE 256 512
LET MAP_CELL 0.5
LET UV_SHIFT YES
LET MAP_RA     20:39:10
LET MAP_DEC    68:01:20
!
LET WCOL 1
FOR I 1 TO 10
LET NAME CONT-'I'
GO UV_MAP
NEXT
```

Then the mosaic must be created using task *MAKE_MOSAIC*. This task produces a combined image (MAP.LMV), a data cube of primary beams (MAP.LOBE), a data cube of dirty beams (MAP.BEAM), and a weight image MAP.WEIGHT, which are required for cleaning. Two parameters are very important in the mosaic creation: the primary beam width (FWHM), and the truncation threshold of the primary beam. It is recommended to use a relatively high value for this threshold, 0.2 to 0.4, to avoid contaminating adjacent field in case of pointing errors.

6.4.2 Cleaning

Combined mosaic images and beam cube can be read in MAPPING as standard images. The **MAPPING** program will automatically switch to mosaic mode when a PRIMARY beam is read in using command
READ PRIMARY file.lobe
A logical variable, DO_MOSAIC, is set to indicate the mosaic mode is activated. To switch back to normal cleaning, reset this logical variable to NO.

The primary beam cube contains information about the threshold level for primary beams; this information is necessary to make a coherent deconvolution. Two additional variables are used for mosaic cleaning: the search level, SEARCH_W and the restoring level, RESTORE_W. This levels specify to which relative sensitivity level searching clean components and restoring the sky brightness should be performed. Since the noise increases at the mosaic edges, inversely proportionally to these levels, they should not be too low. However, the search level should be low enough to include all the emission; clean components searching is further limited by the support defined by command SUPPORT as for normal cleaning.

If necessary, confusing sources close to the mosaic edges should be removed in individual fields using *MX* before the final mosaic is built.

Note that, contrary to the standard clean which does not correct for primary beam attenuation, the mosaic clean produces *sky brightness* images.

6.5 WIPE

The reader interested in the theoretical foundations and developments of WIPE is refered to:

1. "Fourier Interpolation and Reconstruction via Shannon-type Techniques. Part I: Regularization principle", A. Lannes, E. Anterrieu and K. Bouyoucef, *Journal of Modern Optics*, 1994, vol. 41, no. 8, 1537-1574.

2. "Fourier Interpolation and Reconstruction via Shannon-type Techniques. Part II: Technical developments and applications", A. Lannes, E. Anterrieu and K. Bouyoucef, *Journal of Modern Optics*, 1994, vol. 41, no. 8, 1537-1574.

3. "CLEAN and WIPE", A. Lannes, E. Anterrieu and P. Maréchal, *A&A Supplement Series*, 1997, 123, 183-198.

We will focus here on the implementation of WIPE in **MAPPING** . WIPE deconvolution method can be divided in three main parts:

1. Regularization: includes the definition of the aperture to be synthesized from the experimental aperture, the definition of the Neat Beam and the computation of the Dusty Beam and the Dusty Map;

2. Deconvolution: is the heart of WIPE. WIPE uses clean as an initial solution to speed up the computations. WIPE defines an heuristic to build interactively the effective support of the observed object (the stability of the deconvolution is checked at each step);

3. Error analysis: allows one to compute the condition number, the eigenvalues and eigenvectors of the "mapping and gridding" operator.

6.5.1 WIPE Regularization

WIPE is a regularized deterministic Fourier synthesis method. We know from inverse problems theory that extrapolation is forbidden, and interpolation allowed to a certain extent. The first step is to define the amount of interpolation that we want to perform in the Fourier space from the gridded experimental data information.

The aperture to synthesize is defined as the ellipse fitting in the best manner the experimental data. According to the confidence we put in data, one can expand this ellipse using two multiplicative factors (variable APERTURE) applied on the major and minor axis. At the end of this step a window will appear with three areas showing the gridded experimental aperture, the synthesized aperture and the superposition of the two first ones (cf. figure 1).

Once the aperture to synthesize is defined, one can then compute the synthetic beam or "Neat Beam", using the power method to construct the corresponding spheroidal filter. The "Dusty Beam" is the one related to the experimental aperture. We also compute the "Dusty Map", as opposed to the clean "Dirty Map", because it is apodized by the Neat Beam. The window displaying these three images is displayed as the BEAM window (cf. figure 2).

6.5.2 WIPE Deconvolution

The second part of WIPE essentially concerns the final reconstructed image, the Neat Map. As WIPE is slow when reconstructing large images, it can be initialized with a few Clean iterations. The program will start by prompting for the number of clean components and the clean gain to use:

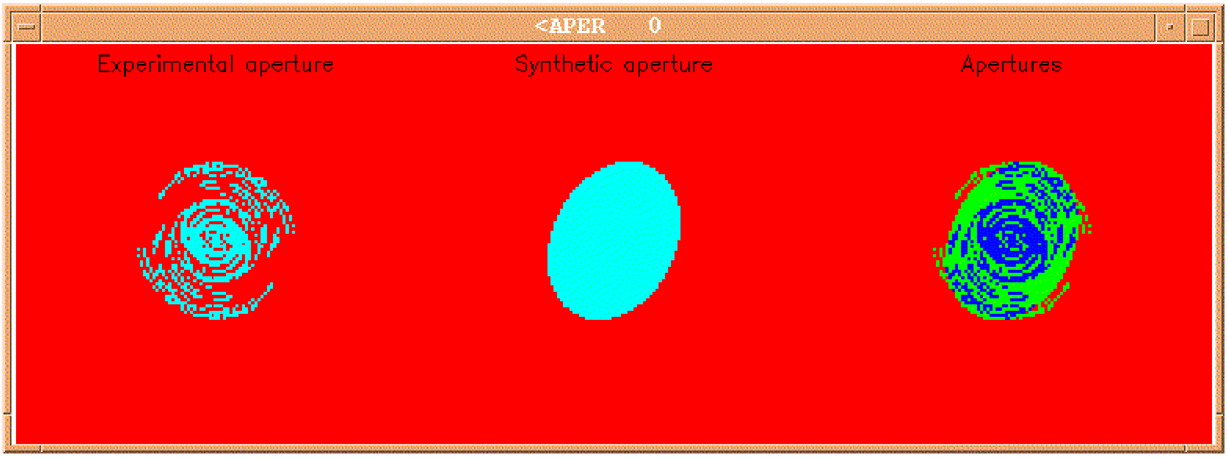

Figure 1: Example of aperture window.

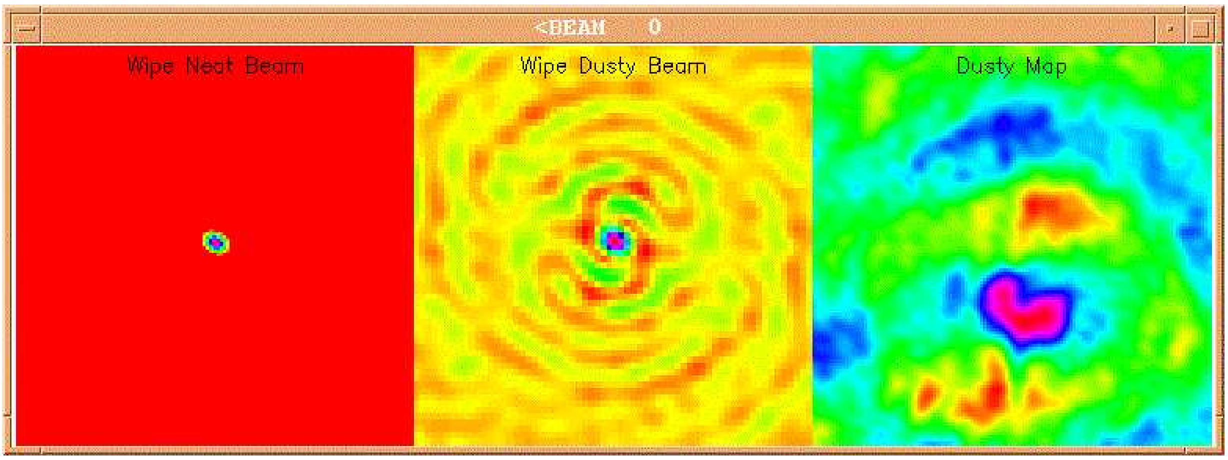

Figure 2: Example of a BEAM window.

```
clean_nbr_iter......: 10
clean_gain..........: .2
```

It will then select the `clean_nbr_iter` first components of clean and display the collected flux and the relative variation:

```
CLEAN (0) ....: component:[38,68], FLUX:1.500229E-01, RELATIVE_VARIATION:1.000000E+00
CLEAN (1) ....: component:[39,74], FLUX:2.957820E-01, RELATIVE_VARIATION:4.927923E-01
CLEAN (2) ....: component:[41,63], FLUX:4.298387E-01, RELATIVE_VARIATION:3.118767E-01
CLEAN (3) ....: component:[48,61], FLUX:5.520994E-01, RELATIVE_VARIATION:2.214468E-01
CLEAN (4) ....: component:[40,70], FLUX:6.701481E-01, RELATIVE_VARIATION:1.761532E-01
CLEAN (5) ....: component:[41,77], FLUX:7.874965E-01, RELATIVE_VARIATION:1.490145E-01
CLEAN (6) ....: component:[44,73], FLUX:8.867114E-01, RELATIVE_VARIATION:1.118909E-01
CLEAN (7) ....: component:[36,66], FLUX:9.877205E-01, RELATIVE_VARIATION:1.022648E-01
CLEAN (8) ....: component:[43,64], FLUX:1.088139E+00, RELATIVE_VARIATION:9.228493E-02
CLEAN (9) ....: component:[37,77], FLUX:1.181291E+00, RELATIVE_VARIATION:7.885618E-02
CLEAN (10) ....: component:[46,58], FLUX:1.271183E+00, RELATIVE_VARIATION:7.071504E-02
```

Selecting too many components for Clean can slow down WIPE later on, since WIPE will impose a positivity constraint and will have to process additional iterations of the conjugate gradients method to achieve it. By experience, we recommend a value between 5 to 20 components although it is problem dependant as one could imagine.

Another way to go, is to use Clean as usually, produce the expected Clean Map and then perform only one iteration of WIPE in order to check the value of the condition number κ.

At this point the program displays the current Clean Map and Clean Residu in the SOL window (cf. figure 3).

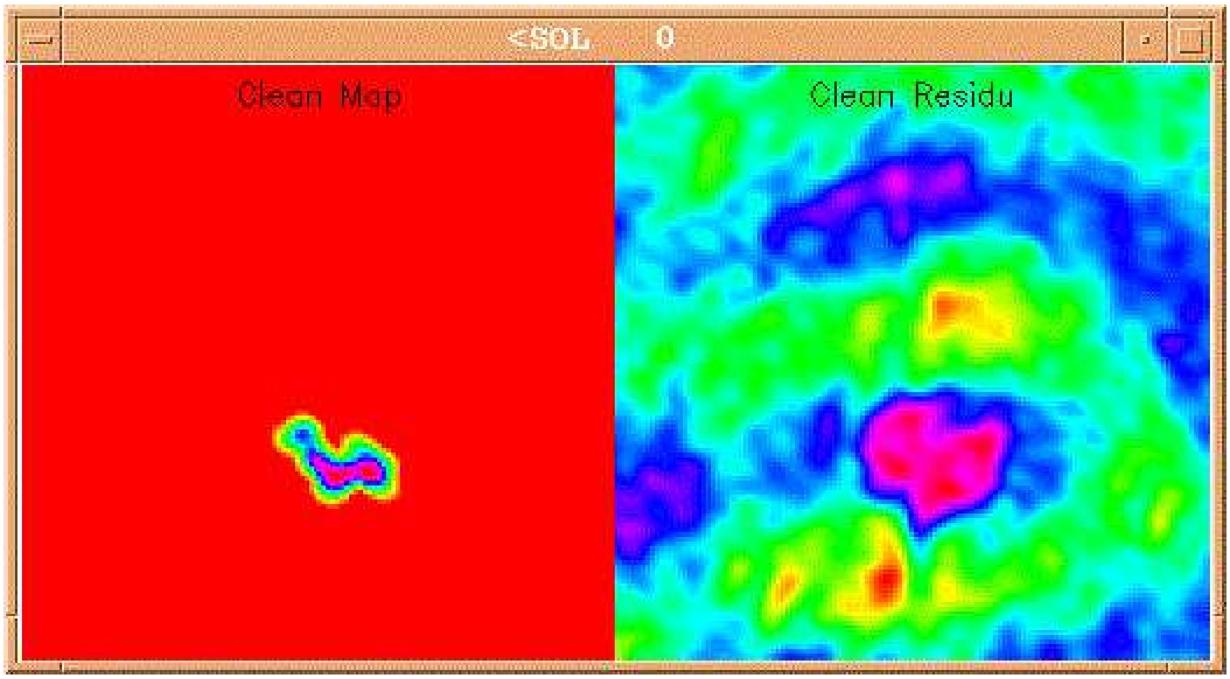

Figure 3: Example of a SOL window at the clean stage.

After this preliminary initialisation step, the user is prompted whether he (or she) wants to pursue the reconstruction. The user can come back to clean and add components or pass the hand over to WIPE:

```
Q-WIPE,  LEAVE CLEAN & WIPE ? [y,n] n
Q-WIPE,  CLEAN or WIPE ? [c,w] w
```

When switching to WIPE, the program starts by performing a first iteration of the WIPE process, optimizing the solution with the support corresponding to Clean, using the conjugate gradients methods. This last method, at the expense of a few additional iterations, allows one to exhibit an approximation of the condition number κ of the mapping and gridding operator. κ is a majorant of the amplification of the data error through the reconstruction process. Therefore it sets an upper bound for the reconstruction error and gives an idea of the stability and reliability of the reconstruction.

```
WIPE.......: wipe_iter=0
WIPE(0)....: iter_gc=0
WIPE(0)....: iter_gc=1 mu=1.346324E+00, mu'=2.088283E+00, kappa=1.245431E+00
WIPE(0)....: iter_gc=2 mu=1.046280E+00, mu'=2.561411E+00, kappa=1.564645E+00
WIPE(0)....: iter_gc=3 mu=7.935231E-01, mu'=2.902714E+00, kappa=1.912592E+00
WIPE(0)....: iter_gc=4 mu=6.996743E-01, mu'=3.116523E+00, kappa=2.110509E+00
WIPE(0)....: iter_gc=5 mu=6.120918E-01, mu'=3.296474E+00, kappa=2.320687E+00
WIPE(0)....: iter_gc=6 mu=5.726386E-01, mu'=3.376165E+00, kappa=2.428127E+00
WIPE(0)....: iter_gc=7 mu=5.535087E-01, mu'=3.404158E+00, kappa=2.479948E+00
WIPE(0)....: iter_gc=8 mu=5.400138E-01, mu'=3.416653E+00, kappa=2.515348E+00

I-WIPE,  FLUX:3.664394E+00; RELATIVE_VARIATION:6.530987E-01
```

After the first iteration of WIPE, the current solution is updated in the SOL window. The WIPE solution (Neat Map and Wipe residu) replaces the Clean one, as represented in the figure 4).

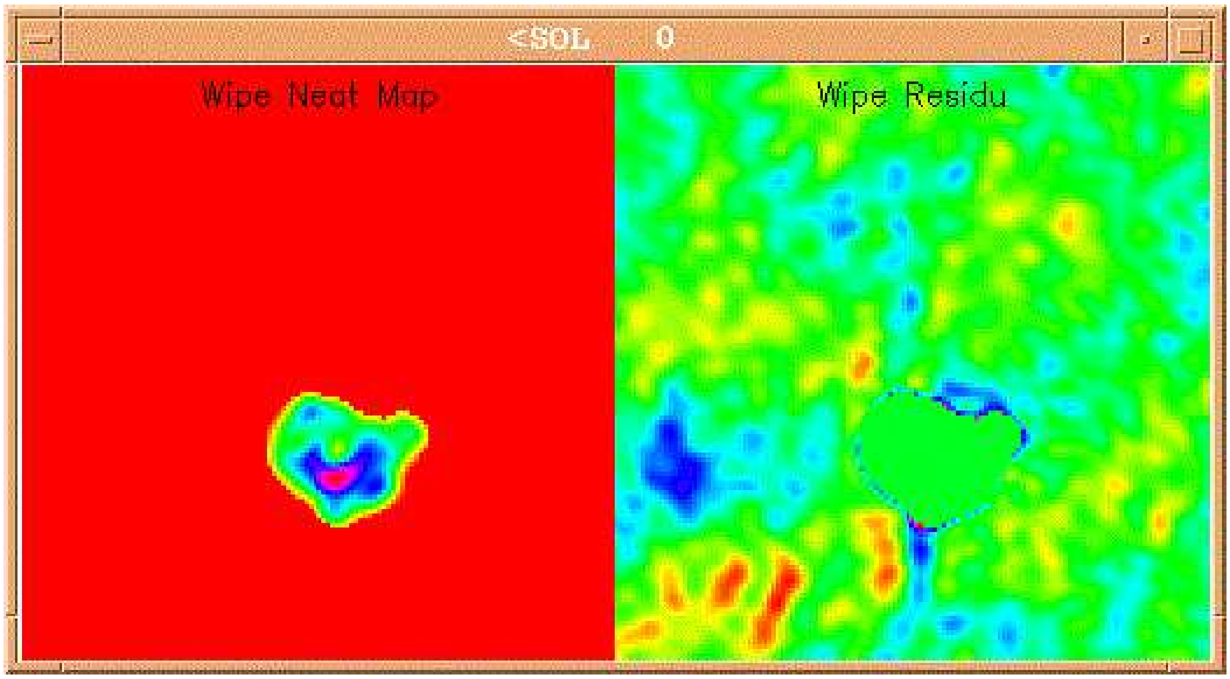

Figure 4: Example of a SOL window at the Wipe stage.

According to the value of κ and the information the user is interested in (if it is still in the residu and we are above the noise level) the program prompts for the number of additional Wipe steps to perform. Each Wipe step will increase the effective support following an embedded heuristic (the increase is of the order of the diameter of the Neat Beam).

```
WIPE.......: wipe_nbr_iter = 5
```

At each Wipe iteration, the Neat Map and the residu are updated. The outputs are similar to the ones of the very first WIPE iteration. After performing the requested number of iteration, the program asks whether to increase the support or not:

```
Q-WIPE,  EXTENSION OF THE IMAGE SUPPORT ? [y, n] n
```

And one could go on like that with one step increase and decide precisely when to stop the process.

The next request offers a way to go back to WIPE:

```
Q-WIPE   DO YOU WANT TO STOP WIPE ? [y, n] y
```

If we really want to leave WIPE, a final smoothing of the object support, removing small regions (basically a morphological operation), is performed:

```
I-WIPE,  SMOOTHING OF THE IMAGE SUPPORT
I-WIPE,  region 1 = 684 pixels
I-WIPE,  region 2 = 41 pixels
I-WIPE,  region 3 = 5 pixels
I-WIPE,  2 Discarded connected region(s)
I-WIPE,  MINIMIZATION ON THE SMOOTHED SUPPORT
WIPE.......: wipe_iter=7
WIPE(7)....: iter_gc=0
WIPE(7)....: iter_gc=1 mu=1.074615E+00, mu'=2.813862E+00, kappa=1.618173E+00
WIPE(7)....: iter_gc=2 mu=8.191533E-01, mu'=2.949595E+00, kappa=1.897573E+00
WIPE(7)....: iter_gc=3 mu=6.177616E-01, mu'=3.341397E+00, kappa=2.325699E+00
WIPE(7)....: iter_gc=4 mu=4.993783E-01, mu'=3.584762E+00, kappa=2.679263E+00
WIPE(7)....: iter_gc=5 mu=4.009354E-01, mu'=3.667690E+00, kappa=3.024538E+00
WIPE(7)....: iter_gc=6 mu=3.518280E-01, mu'=3.688257E+00, kappa=3.237765E+00
WIPE(7)....: iter_gc=7 mu=3.114740E-01, mu'=3.705962E+00, kappa=3.449369E+00
WIPE(7)....: iter_gc=8 mu=2.808061E-01, mu'=3.721767E+00, kappa=3.640586E+00
WIPE(7)....: iter_gc=9 mu=2.511003E-01, mu'=3.735208E+00, kappa=3.856859E+00
WIPE(7)....: iter_gc=10 mu=2.344449E-01, mu'=3.746117E+00, kappa=3.997333E+00
WIPE(7)....: iter_gc=11 mu=2.241068E-01, mu'=3.756186E+00, kappa=4.093983E+00
WIPE(7)....: iter_gc=12 mu=2.189550E-01, mu'=3.760925E+00, kappa=4.144478E+00
WIPE(7)....: iter_gc=13 mu=2.166695E-01, mu'=3.763188E+00, kappa=4.167533E+00
WIPE(7)....: iter_gc=14 mu=2.158919E-01, mu'=3.763985E+00, kappa=4.175474E+00
WIPE(7)....: iter_gc=15 mu=2.154566E-01, mu'=3.764431E+00, kappa=4.179937E+00

I-WIPE,  FLUX:5.370097E+00; RELATIVE_VARIATION:-5.083192E-02
```

At this step, we still have the chance to go back to Clean or WIPE:

```
Q-WIPE,  DO YOU WANT TO LEAVE CLEAN & WIPE ? [y,n] y
```

If we really want to exit Clean and WIPE, the solution reached at this point will be the final one. But it is not yet finished and one can move forward to error analysis in order to refine the computation of the condition number κ and the Neat Map solution, as we shall see in the next section.

6.5.3 Wipe Error Analysis

This last step of the program performs *a posteriori* control of the robustness of the reconstruction process.

When prompted by the program:

```
Q-WIPE   DO YOU WANT TO CONDUCT ERROR ANALYSIS ? [y,n] y
```

if the answer is yes, the program will compute the condition number κ, the eigenvalues λ and eigenmodes of the reconstruction operator.

```
I-WIPE   This may take some time. Please wait...

Krylov 2: mu = 3.995562e-01, mu' = 5.566639e-01, kappa = 1.180341e+00
Krylov 3: mu = 3.017594e-01, mu' = 8.600131e-01, kappa = 1.688193e+00
Krylov 4: mu = 2.584568e-01, mu' = 1.026722e+00, kappa = 1.993115e+00
Krylov 5: mu = 1.981331e-01, mu' = 1.194734e+00, kappa = 2.455597e+00
Krylov 6: mu = 1.479192e-01, mu' = 1.257864e+00, kappa = 2.916115e+00
...
Krylov 33: mu = 7.118590e-02, mu' = 1.295405e+00, kappa = 4.265852e+00
Krylov 34: mu = 7.118544e-02, mu' = 1.295405e+00, kappa = 4.265866e+00

lambda[0] = 0.071185    theta[0] = 88.789797
lambda[1] = 0.073799    theta[1] = 88.306748
lambda[2] = 0.121578    theta[2] = 82.808893
lambda[3] = 0.135521    theta[3] = 85.752680
...
lambda[32] = 1.287797   theta[32] = 89.913183
lambda[33] = 1.295405   theta[33] = 89.933163
```

μ and μ' are respectively the smallest and largest eigenvalues of the reconstruction operator. The condition number κ of the same operator is defined as

$$\kappa = \frac{\sqrt{\mu'}}{\sqrt{\mu}}$$

The angle θ_i between the solution, ϕ, and the eigenmode i, ϕ_i, is defined by

$$\cos(\theta_i) = \frac{|(\phi_i|\phi)|}{||\phi||^{1/2}}$$

In practice κ should be kept as low as possible (not exceding 4 or 5 in general). But this is not a strict rule and one should check whether the features present in the Neat Map can be connected to the ones present in the three first eigenmodes of the reconstruction operator. If so, one has to be careful because these features could be artefacts of the reconstruction. In such cases, we recommand to restart the process and to aim at a lower resolution and thus a more stable and reliable solution.

The window displaying the three first eigenmodes is displayed as the ERROR window (cf. figure 5).

Figure 5: Example of an error window.

Whether the error analysis is selected or not, a Neat Beam fitting is performed to allow comparison with the Clean method. Although the Neat Beam is a spheroidal filter and not a Gaussian function, it is fitted by a Gaussian to determine minor and major axis as well as the orientation angle:

```
I-MX_CLEAN,  Beam is     2.96" by     2.22" at PA    61.95 deg.
I-MX_CLEAN,  Errors (     .01)   (     .01)      (     .45)
S-WIPE,  Successfull completion
```

At this point, WIPE is over and pass back the hand to **MAPPING** . One can visualize the different results in the <GREG window and perform all kinds of operations, like flux measurements and what so ever.

But before leaving **MAPPING** , if one is satisfied with the results, he (or she) must save them on files:

```
WRITE NEAT file.lmv-neat  ! for the Neat map
WRITE BEAM file.beam      ! for the Neat Beam
WRITE DIRTY file.lmv      ! for the Dusty Map
```

6.5.4 Important Remarks

- The more *a priori* information we introduce, the more the problem is constrained. Although it is not presented in this section, the user can impose a support contraint on the solution *via* the POLYGON procedure. This could be particularly important for Clean, but not that much for WIPE since it has its own strategy to define the support.

- One point to stress here is the importance of the regularization step. Since it is the one which sets the objectives to reach in spatial frequency resolution. The size of the Neat Beam is only a consequence of the previous choice. The parameter that controls the spatial frequency resolution is APERTURE[2]: the multiplicative factor for the major and minor axis of the ellipse corresponding to the aperture to synthesize.

7 MAPPING internal help

7.1 CLARK

```
[CLEAN\]CLARK [Plane] [/PLOT Clean|Residu] [/FLUX Min Max]
```

```
Clean the specified plane (default: plane specified by  variable  FIRST)
with  the  B.Clark  method.  If  option /PLOT is given, a display of the
CLEAN or RESIDUAL map will be shown at each major  cycle,  depending  on
the  argument  (default  Residual),  and  the  user will be prompted for
continuation. Option /FLUX can be used to give the limit for the display
of the cumulated clean flux as a function of number of components.

In mosaic mode (See command MOSAIC), a mosaic clean is performed.
```

7.2 FIT

```
[CLEAN\]FIT [Field]
```

```
Fit the dirty beam to obtain the clean beam parameters.   This  is  done
automatically by CLARK, HOGBOM, MRC and SDI if needed.

Clean beam parameters can also be specified by the users, by setting the
variables  MAJOR_AXIS,  MINOR_AXIS and ANGLE. In this case, no automatic
fit will be perform by CLARK, HOGBOM, MRC and SDI.

In mosaic mode, you can specify on which field the fit has to  be  made,
using the first argument.
```

7.3 HOGBOM

```
[CLEAN\]HOGBOM [Plane] [/FLUX Fmin Fmax]
```

```
Clean the specified plane (default: plane specified by  variable  FIRST)
with  the  standard  Hogbom  method. The  cumulative  cleaned  flux is
displayed in real time while cleaning goes on in an  additional  window.
Option  /FLUX  can  be used to give the limits in flux for this display.
Cleaning can be interrupted by pressing C at any time.

In mosaic mode (See command MOSAIC), a mosaic clean is performed.
```

7.4 LOAD

```
    [CLEAN\]LOAD Variable

Define Variable (CLEAN, DIRTY, RESIDUAL and BEAM) to be the item  to  be
plotted  (for  RGMAP  and  PLOT  for example). Used in particular by the
procedure
        @ PR:P_PLOT Variable [First [[Last]]
 which will plot all channels of the specified variable.  This procedure
is available simply by typing
        SHOW Variable [First [[Last]]
First (default 0)  and  Last  (default  First)  are  optional  arguments
indicating  the first and last channel to be displayed. If not specified
the variables FIRST and LAST of the INPUT macro are used.

Several  variables  control  the  aspect  of  the  display.  DO_CONTOUR
indicates  whether contour levels should be drawn, DO_BIT whether bitmap
should be displayed.
```

7.5 MOSAIC

```
    [CLEAN\]MOSAIC On|Off

Turn on or off the  mosaic  mode.  Note  that  a  READ  PRIMARY  command
switches  automatically  in  mosaic  mode. The program prompt changes to
inform the user.
```

7.6 MRC

```
    [CLEAN\]MRC [Plane] [/PLOT Clean|Residu]  [/FLUX  Min  Max]  [/RATIO
Irat]

Perform a Multi-Resolution CLEAN on  the  current  dirty  image.   Seems
operational  since  version  2.0 of MAPPING. If option /PLOT is given, a
display of the CLEAN or RESIDUAL map will be shown at each major  cycle,
depending  on  the  argument  (default  Residual),  and the user will be
prompted for continuation. Option /FLUX can be used to  give  the  limit
for  the  display of the cumulated clean flux as a function of number of
components.

A summary plot with the Difference, Smooth, and total  CLEANed  maps  is
displayed at the end.

Option RATIO can be used to specify the smoothing factor, which must  be
a power of 2. The default is estimated from the image size.

MRC does not support mosaics yet.
```

7.7 READ

```
[CLEAN\]READ Variable File [/PLANE First Last]
```

Read the specified variable (UV, BEAM, DIRTY, CLEAN or PRIMARY) from input File. First and last planes read in can be specified using option /PLANE (valid also for UV tables).

7.8 SDI

```
[CLEAN\]SDI [Plane] [/PLOT Clean|Residu]
```

Perform a Steer-Dewdney-Ito CLEAN. This clean method selects an ensemble of clean components and remove them at once using FFTs. It works best for extended sources and UV coverages with short spacings. In such a case, it may avoid the "ringing" features which appear using the CLARK or HOGBOM techniques.

In mosaic mode (See command MOSAIC), a mosaic clean is performed.

7.9 SUPPORT

```
[CLEAN\]SUPPORT [Polygon] [/CURSOR] [/PLOT]
```

Define the support function for CLEAN components. Calls command POLYGON of GREG to do that. See HELP POLYGON for details.

With option /CURSOR, it calls the interactive cursor to define the polygon summits. Type any key to go to next summit, D to correct the last one and type E to end the polygon definition. The last polygon side will then appear. The polygon definition may be aborted by typing Q. For X11 displays, you may use the mouse buttons for the commands. The left mouse button draws a vertex, the middle mouse button deletes the last vertex, and the right mouse button ends the polygon definition.

7.10 UV_FLAG

```
[CLEAN\]UV_FLAG
```

Display UV data and calls the cursor to interactively select a region where UV data will be flagged. The parameters for the display are as in UV_SHOW.

Subsequent mapping with UV_MAP will ignore flagged data. Unflagging can only be done by re-loading the UV data set with command READ UV.

UV data flagged using command UV_FLAG can be saved on file with command WRITE UV File.

7.11 UV_MAP

```
[CLEAN\]UV_MAP
```

Compute a dirty map and beam from the UV data. UV data should have been loaded by command READ UV File. Parameters are specified through SIC variables as for the Task UV_MAP: use command INFO for the list.

7.12 UV_SHOW

```
[CLEAN\]UV_SHOW
```

Display UV data loaded by command READ UV File. Control parameters are available as SIC variables. Use command INPUT UV_SHOW to check them. Flagged UV data will appear in a different color.

7.13 UV_SORT

```
[CLEAN\]UV_SORT Order
```

Sort and transpose the UV data set, loaded by command READ UV File. Order is either TIME for Time-Baseline ordering, BASE for Baseline-Time ordering. The sorted UV data is then available in variable UVS for further plotting.

7.14 UV_STAT

```
[CLEAN\]UV_STAT Keyword [Step Start]
```

Perform analysis of the UV data set to predict the synthesized beam, expected noise level, and recommended pixel size for UV_MAP. Keyword can be TAPER, WEIGHT, or CELL.

For TAPER, beam sizes and noise level (in flux and brightness) will be computed for 9 different tapers (from Start to Start*Step^9). Default value for Step is sqrt(2), Default value for Start is 50 m. Weighting mode, UV cell size and "robust" parameter are taken from variable UV_CELL.

For WEIGHT, beam sizes and noise level (in flux and brightness) will be computed for 9 different "robust" weighting parameters (from Start to Start*Step^9). Default value for Step is sqrt(10), and default value for Start is derived to center the "robust" parameter values around 1. UV cell size is taken from variable UV_CELL[1], and Taper is taken from variable UV_TAPER.

For CELL, beam sizes and noise level are computed for 9 different UV cell sizes. That version is still under development.

UV_STAT allows the astronomer to select the best weighting parameters according to its personnal tradeoff between angular resolution and sensitivity.

7.15 WIPE

```
[CLEAN\]WIPE
```

Starts the WIPE deconvolution method. WIPE is a deterministic deconvolution method using CLEAN as a starting point. WIPE applies a regularization principle and provides error analysis on the deconvolution process. A strategy is defined in WIPE to build automatically the effective support of the object under observation. However a starting support (analog to a Clean Box) can be defined to restrict the search.

7.16 WRITE

```
[CLEAN\]WRITE Variable File
```

WRITE the specified variable (CLEAN, RESIDUAL, DIRTY, BEAM, SUPPORT, UV) onto a File.