NEWSTAR

Cookbook

J.E. Noordam (Editor)

September 10, 1992

A.G. de Bruyn

Chapter 1

GENERAL

NEWSTAR general section
GENERAL INTRODUCTION

Editor: J.E.Noordam

January 11, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]gen1_INTRO.tex

1 INTRODUCTION to the NEWSTAR Cookbook

NEWSTAR is an acronym for Netherlands East-West Synthesis Telescope Array Reduction. This Cookbook describes the use of the NEWSTAR uv-data reduction package for the Westerbork Synthesis Radio Telescope (WSRT). In particular it describes how to take maximum advantage of the strong points of the WSRT, which are:
- very small 'closure errors' (< 0.01 %)
- low instrumental polarisation, due to axi-symmetric telescopes
- good wide-field mapping, due to the East-West array
- redundant spacing information for model-independent calibration
The software can also be used for other East-West arrays, and there is an interface for data from the Australia Telescope (AT).

The programs were originally developed by J.E.Noordam in 1977-1983. They were then further enhanced (and professionally coded) by W.N.Brouw. This resulted in the 'R-series' of programs (REDUN, RLIGN, RMAP etc), which became also known as 'DWARF', after the local software environment developed by NFRA. In 1990-91, Brouw has written a complete revision, with many improvements and new options. This was first called the 'N-series' (with program names NSCN, NCALIB, NMAP etc), and then 'NEWSTAR '. During this entire development, A.G.de Bruyn played an indispensable role as an active and stimulating user.

The NEWSTAR 'package' consists of the NEWSTAR programs, the DWARF parameter interface, the Groningen Image Display System (GIDS), and PGPLOT routines for screen graphics. NEWSTAR runs on SUN and HP (and perhaps DEC) workstations, predominantly in Dwingeloo, Westerbork, Groningen and Sydney. However, if sufficient interest is expressed, the package can be exported to other institutes (like Bonn University) on a small scale. VAX and Alliant implementations still exist, and the package also runs on a Convex. NEWSTAR has completely superseded the old R-series, which has been frozen and will effectively disappear with the Alliants in Dwingeloo and Groningen.

1.1 Structure of this Cookbook

The Cookbook has a top-down structure, in which the amount of detail increases towards the end. It consists of the following main chapters:

1. **General aspects**, including a general introduction and oveview, background information on the hardware environment at NFRA in Dwingeloo, and the use of the DWARF parameter interface.
2. **Processing 'recipes'**, giving step-by-step guidance for the various data reduction methods. These recipes refer to options of specific programs described in a later chapter.
3. **File description**. Details of the structure and contents of the NEWSTAR data files and auxiliary files. Also given is an overview of the possible interactions with these 'objects'.
4. **Program description**. Details of the available options in the the various NEWSTAR programs. These include an explanation of the output on the screen and on the printer/plotter. For each program, there is a summary of its parameters ('keywords'), including their on-line HELP text.
5. **Appendices** with relevant information for NEWSTAR users.

Each *chapter* consists of about 5-10 *sections* of 2-10 pages each. The Cookbook will be updated continuously, as new processing methods are developed and old ones are improved (this includes the removal of bugs, inconceivable as they may be). For this reasons, full

realeases of the entire Cookbook will be rare. Instead, individual sections will be updated (or added) and distributed separately. As a consequence, the sections are as much as possible independent from each other. Cross-references to other sections are by name (not by page or section number), and page numbering is not consistent. Eventually, the Cookbook will be made available by anonymous FTP, as part of the WSRT User Documentation.

1.2 How to use this Cookbook

It is assumed that the user has some experience in radio aperture synthesis reduction. For the beginning NEWSTAR user, we recommend the following steps:

- Study the **table-of-contents** to get an idea of where to find what. Note that the page numbering will be meaningless in a Cookbook that consists of independent sections.
- Read this introduction, especially the **block diagram** that gives an overview of the (relation between) NEWSTAR programs and files. Note the basic functionality of each.
- Read the section 'Overview of NEWSTAR files'. Among other things, it contains a description of the three main NEWSTAR data files (SCN, MDL and WMP), and a **summary of possible interactions** with each of these 'objects'.
- Familiarise yourself with the structure of the uv-data file (SCN-file), and the various corrections that are stored therein.
- Read the section on 'Common features of NEWSTAR programs' to learn how to run NEWSTAR programs and to manipulate program keywords (parameters) under DWARF.
- Scan through the **processing recipes**, to see what is available. Find the recipe that is closest to your particular project, and go through the steps.

1.3 The NEWSTAR reduction concept

In many ways, NEWSTAR reduces uv-data in much the same way as other radio aperture synthesis packages like AIPS and MIRIAD. WSRT uv-data may be inspected, manipulated, calibrated etc. For the latter, one may use the data itself (SELFCAL), special calibrator observations, or externally obtained information about the ionosphere and clock.

The NEWSTAR reduction concept is *different* in the way it deals with calibration methods like SELFCAL:

1. The source model used in NEWSTAR is much more versatile than the usual collection of CLEAN components. It consists of multi-parameter source components, which are not limited to grid points, and can be used for many other purposes (see below). However, CLEAN components, generated in the normal way, may also be included in the model.
2. The uv-representation of the SELFCAL model is stored with the uv-data in the same SCN-file. It is always available for comparison with the observed uv-data, calculation of antenna-based gains, model 'updates', or any other manipulations that the user might choose to perform.
3. The redundant spacings of the very regular WSRT array are exploited as powerful extra constraints on the SELFCAL process. These constraints represent a 'higher truth' than constraints that are dependent upon the source model, which may be incorrect. Thus, it is more difficult to converge to the wrong model.

The NEWSTAR source model is usually not generated by CLEAN, but in the following way: A search programme is used to find sources in the map, which are then represented as multi-parameter components (position, flux, extent, polarisation, spectral index, etc) However, to avoid picking up the crossing points of grating rings as real sources, the process is done in a number of iterative steps. Initially, only very strong peaks in the map are located. These components are then Fourier transformed, and subtracted from the uv-data to form a 'residual map', which can again be searched to find weaker sources. In each iteration, the incomplete model is used to derive SELFCAL phase and gain corrections. There is an additional operation, called 'update' in which the source parameters (e.g. position and flux-densities) may be improved dynamically, using the uv-data and the uv-model.

The multi-parameter source components of the NEWSTAR source model may be used for many other purposes besides SELFCAL. An example is the accurate combination of different polarisations, different pointing centres (mosaicking), or different frequencies ('broad-band' mapping). They can also be used to deal with instrumental polarisation, Faraday rotation measures, and probably even with time-variable image-plane phenomena like non-isoplanaticity.

1.4 Overview of NEWSTAR programs and data files

The NEWSTAR functionality is distributed over a relatively small number of programs, each with a relatively large number of options. Thus, each NEWSTAR program represents a 'class' of operations. See also the block diagram in fig 2?. This approach differs from the structure of AIPS, which consists of a large number of small programs.

- **NSCAN:** Reading and manipulation of the uv-data file (SCN-file). Inspection and editing of uv-data and header information. Transfer to and from other formats (e.g. UVFITS).
- **NCALIB:** SELFCALibration, with or without redundancy constraint. Standard Calibration with the help of calibrator observations. Various polarisation corrections.
- **NMODEL:** SELFCAL model generation and manipulation.
- **NMAP:** Generation and manipulation of images (maps) of various types, with many options. Conversion of images to FITS format.
- **NPLOT:** Plotting of uv-data (including SELFCAL residuals and telescope errors) and maps of all types. Various graphics formats are generated, including X-screen format (using PGPLOT routines).
- **NGIDS:** Interface between the .WMP file and the GIPSY Image Dispay System (GIDS), for X-screen display of images.
- **NCLEAN:** CLEANing and restoring of map(s). CLEAN components are added to a SELFCAL model, and may be 'restored' in the residual map.
- **NGCALC:** Offers a variety of calculations on uv-data, to extract astronomical information from it.
- **NATNF:** Reading of uv-data from the Australia Telescope (E-W array only).

There are three main types of NEWSTAR data files, and some auxiliary files for logging, plotting etc. Just like the NEWSTAR programs, the data files are relatively large, in the sense that they may contain a collection of similar data that *is related in some way*:

- **.SCN files** contain uv-data and their corrections, and possibly the source components *and the uv-representation* of a SELFCAL model. The same file may contain many related uv-data sets, e.g. frequency channels, pointing centres (mosaicking), calibrator observations etc.

- **.WMP files** contain image data, including antenna patterns (for CLEAN) and residual images. Many related images can be held in the same file.
- **.MDL files** contain a single source model. This consists predominantly of multi-parameter source components, but ordinary CLEAN components may be mixed in.

NEWSTAR completely supersedes the earlier R-series of programs, which will disappear with the Alliant and the VAX. NSCAN replaces RSCAN and RWTAPE. NCALIB replaces REDUN and RLIGN. NMODEL replaces RMODEL. NMAP replaces RMAP and RMFID. NPLOT replaces RPLOT and RMPLOT. NCLEAN replaces RCLEAN. NGCALC replaces RGPLOT. The programs NSCAN, NMAP and NMODEL provide ways to translate existing R-series files.

1.5 Relation to the WSRT Standard Calibration

NFRA was the first to offer absentee observing and a calibration service to users of radio aperture synthesis telescopes. Upon request, the Dwingeloo Reduction Group will apply phase and gain corrections derived from calibrator observations before and after the user's observation. This is the socalled Standard Calibration. They can also apply externally obtained corrections for ionospheric Faraday rotation and refraction, and clock errors.

However, since the invention of SELFCAL and the exportable reduction package (AIPS), many users like to do their own reduction in their home institute. NEWSTAR caters to this desire by incorporating the functionality of the programs used by the Reduction Group in Dwingeloo. In addition it offers more advanced techniques like SELFCAL etc, and reversible data editing.

The tasks and the tools of the WSRT User Service are now being reviewed, with an eye on diversification. The service may be extended to include all the reduction techniques in NEWSTAR . Ideally, user and User Service should work with the same software package. This is now possible with NEWSTAR , and in the future with AIPS++.

1.6 Relation with GIPSY, AIPS and AIPS++

Since the development of SELFCAL, astronomers have wanted to reduce their uv-data themselves, preferably in their own institute. This has been recognised by NRAO, which has provided the very successful AIPS package for the reduction and analysis of VLA data. It has been difficult for NFRA to make the WSRT reduction programs available in a similar way, partly because of the large effort involved, and partly because users (and system managers) are resistant to "yet another package". The result is that the number of WSRT users is much smaller than it could be.

As an alternative, users can reduce their WSRT uv-data with AIPS, which is available and supported at many institites throughout the world. WSRT uv-data files can be transferred to AIPS via the 'UVFITS' format. However, since AIPS is rather VLA-specific (in terms of polarisation, coordinate-systems, redundant spacings etc), it will in general not be possible to do full justice to WSRT data in this way. The situation is better for WSRT image data (maps), which can be easily transferred to image analysis packages like AIPS (or GIPSY, or MIDAS) by means of the FITS format. However, astronomers do not like to move from one package to another, and would prefer to have WSRT uv-data reduction and image analysis in a single package.

The best solution would be to incorporate (and maintain) the specific WSRT programs in the "World Reduction Package". The possibility of using AIPS for this purpose was seriously discussed, even though it is difficult to program in AIPS, especially for non-VLA use. Fortunately, there will be a timely successor (AIPS++), which will be specifically designed for programmability and maintainability, and which will not be VLA-specific. AIPS++ will be available from 1994 (?) onwards, and will contain all the NEWSTAR functionality that is needed to get the most out of WSRT data.

1.7 Acknowledgements

The following people have made significant contributions to this Cookbook, by means of critisism (both kinds), suggestions and actual writing of sections:
- Ger de Bruyn
- Richard Strom
- Tapashi Ghosh
- Ernst Raimond

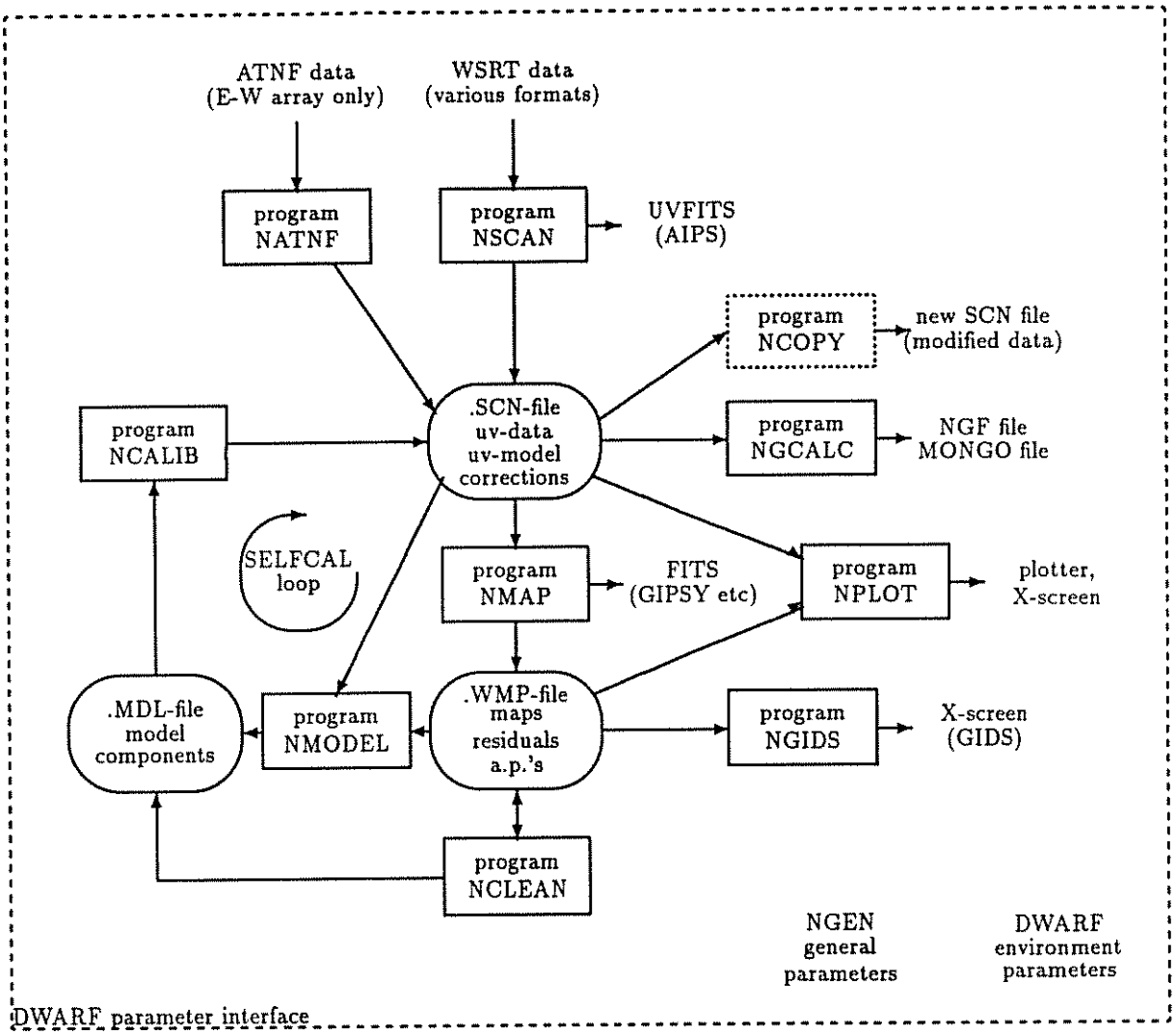


Figure 1: File: fig-gen1_INTRO_block.TEX

Block diagram of the NEWSTAR programs and data-files. The NEWSTAR package specialises in the processing of uv-data *from East-West arrays, in particular the WSRT. The heart of the package is the SELFCAL loop, although it has many other features. WSRT-style SELFCAL is different in two respects:*
- 1: It makes full use of the extra (model-independent!) constraint provided by the presence of redundant spacings in the WSRT array.
- 2: The source model contains multi-parameter source components, including extendedness, polarisation and spectral index. It can also contain CLEAN components.
The 3 types of data files (SCN, MDL and WMP) can be manipulated by means of the programs NSCAN, NMODEL and NMAP respectively. The program NGCALC can be used to extract a wide variety of (astrophysical) information from the data.

NEWSTAR general section
GETTING STARTED (in Dwingeloo)

Editor: J.E.Noordam

December 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]gen1_START.tex

1 GETTING STARTED (in Dwingeloo)

This section is meant for NEWSTAR users who want to work on the computers in Dwingeloo. Since the configuration is changing continuously, and especially in late 1992/early 1993, this section may soon be out of date in some details. Consult the Editor of the Cookbook or the author of this recipe if things do not work as expected.

1.1 Getting a machine and diskspace

There is a variety of computers, terminals and display devices available in Dwingeloo. Although you are free to work on any system that you fancy we strongly suggest that you work on either the SUN workstations or the HP workstations (which are due to arrive in January 1993). There are at least three good reasons to do so:
- these workstations are an order of magnitude faster than the VAX/Alliant
- these workstations have wonderful display (enough for 1024 x 1024 pixels through their X-terminal graphics
- they (will) have plenty of diskspace

Some details on the various systems are given below.

1.1.1 SUN workstations

At present there are two methods to get you into one of the SUN workstations. The direct way is through one of the X-terminals. There is one available for general users in the 'terminal kamer' which is located opposite the computer room entrance (ground floor). There are three more X-terminals in astronomer offices and several more will arrive in January 1993. All you need is an account (username and password) which can be gotten from Klaas Stuurwold (room 231), the system manager for the SUN and HP workstations. How to deal with X-windows is a separate story alltogether.

A second route to the SUN workstations is through 'telnet'. Within Dwingeloo you can connect to a different computer than the one you are working on by typing:
`telnet rzmws0` `(or rzmws1, rzmws2 or rzmws3)` to connect to one of the SUNs
`telnet rzmall` to connect to the Alliant
`telnet rzmvx4` to connect to the VAX3400

1.1.2 HP workstation

The procedure to get you into one of the HP workstations will probably be very similar to that to get to the SUN. Details like computer hostnames will become available soon.

1.1.3 Alliant

To log onto the Alliant (which will be disconnected sometime in 1993) there are two methods. The common route is through the LAT server which is available on all non-X-terminals (VT*** terminals) of which there are a few in the 'terminal kamer'. The second route is to transfer through the use of telnet, once you are already logged onto one of the computers. Contact Henny Lem (room 232) for an account. The procedure to log on via the server is as follows:

server: `c rzmall`	connect to Alliant
login: `username`	ask system manager
password: `password`	ask system manager

dir: `<CR>`	selects root directory
`> pwd`	show present working directory
`> df`	show disks

1.1.4 VAX3400 (discouraged)

Users are discouraged from reducing data on the two VAX 3400's (RZMVX4 and RZMVX5). These machines are being used by a large number of people for a great variety of tasks. Consequently, serious data reduction will be painfully slow (and slow down everybody else as well). Nevertheless, some people still prefer to use the VAX for the R-series of programs and the use of the DeAnza display.

To log onto the VAX you can either use the LAT server on one of the VT*** terminals (see below) or use telnet (see above). Contact Henny Lem (room 232) to get an account on the VAX.

server: `c rzmall`	connect to Alliant
login: `username`	ask system manager
password: `password`	ask system manager

1.1.5 DEC3100

The DEC3100 workstation is located in room 239 (second floor). At the moment NEWSTAR does not yet run on this workstation but it may be available some time in early 1993.

1.2 Getting your data

Some general comments on the organization of the data archive and the status and quality of the data in general is given here. How to actually read in your data is explained in Recipe nr 005.

1.2.1 The WSRT data archive

Soon all of the WSRT data taken since May 1979 will be available on optical disk. These disks can only be accessed via the VAX so you will need an account on the VAX to read your data. A general account for that purpose will be made available in the future. At present some of the data is still on magnetic tape (2400 ft reels). Such data can only be accessed trough the VAX or Alliant tapedrives. Consult a knowledgeable person to find out where the optical tapes and magnetic tapes are stored and how to operate the drives. Tapes and drives are all in the computerroom.

The archive contains data in a variety of formats and with different amounts of calibration applied. For example, data taken before July 2, 1985 will probably have been archived with calibration corrections applied, while data taken after that date will have been archived raw. Data taken before 1979 is not yet available. Data taken in the period from 1979 through 1984 will require special procedures to deal with them in NEWSTAR. These wil be outlined in the future.

When you come to Dwingeloo to process WSRT data in NEWSTAR you probably have been notified by the WSRT Users Service (formerly the Reduction Group) that the data is available and you will have information with you on their whereabouts (e.g. the optical disk and/or tape volume and label).

A convenient way to get a summary listing of all the data in the WSRT archive, including calibration observations observed before or after your object, is through the program ARCQUERY which runs on the VAX. An example of an ARCQUERY session is given in an Appendix.

1.2.2 The WSRT "green sheets"

Even before you are notified by Dwingeloo that your data is available in the archive you will have received some "green sheets" from Westerbork containing detailed information on the observations of your object. A detailed explanation on the information on those sheets will be made available by the WSRT Users Service in the near future. On those sheets there will be handwritten pencil notes by the WSRT Astronomer about the quality of the data. Usually you will also receive a preliminary map (profile, grey-scale and contour plot) as well as plots of the visibility Amplitudes on a selected set of baselines. Study them before you read in your data. On the basis of these plots you may e.g. decide that certain frequency channels or bands do not contain useful data because of interference or correlator problems and that they are not worth bothering about.

1.2.3 Transferring files between computers

The fastest and recommended way to transfer files (whether ascii or binary) is through the use of "ftp". An example dialogue of an ftp file transfer looks like:

```
>   ftp rzmvx4
... user:
... password:
ftp>cd user1:[dwl.ger]
ftp>binary
ftp>get NGC891.SCN
......
ftp>quit
>
```

When transferring files it is important to realize whether they are pure ascii files or mixed ascii/binary data. The default mode is ascii.

When transferring SCN-files (consisting of UV-data) or WMP-files (images) from/to the VAX to/from UNIX systems it is important to convert the internal presentation of numbers. You do this via the option CVX in the programs NSCAN (for SCN-files) or NMAP (for WMP-files). See the description of these programs for further details.

1.3 Backing up data

The recommended way to backup your data (raw data and any images you may have produced is through the "tar" command onto a DAT (Digital Audio Tape) unit. DAT drives exist on the VAX3400 (computer room), Alliant (computer room), SUN (rzmws0, in Klaas Stuurwold's room 231). A DAT will hold about 1 Gigabyte of data and can be written or read in about 2 hours time.

Currently the SUN DAT-unit can record two formats: GIGA-format and DDS-format. The latter is becoming the standard and is the format used on the VAX, Alliant and HP DAT-units.

Device names for the various units are:
- Alliant: sdt3
- SUN (ws0): rst1
- HP:
- VAX:

Examples of 'tar' commands (use 'man tar' to get on-line help):
`tar -tf /dev/nsdt3` lists the contents of a DAT on the Alliant.
`tar -cvf /dev/rst1 *.SCN` writes all SCN files onto a SUN DAT.

1.4 Some useful UNIX commands

The following UNIX commands should be sufficient to get you started on any machine:

Log in, using the machine-dependent dialogue (see below).

> `pwd`	show present working directory
> `df`	show disks
> `mkdir <subdir>`	create new sub-directory

It is recommended to create a new sub-directory for each project.

> `du`	show all sub-directories for this root directory
> `cd ~<subdir>`	select sub-directory
> `ls -l`	Show all files in the specified (sub-)directory

The default is the current sub-directory.

> `ls -l files`	Show file selection (*=wildcard)
> `rm filename`	Delete file safely (confirm)
> `dwe program`	Execute program
> `man command`	Give help on a specific command
> `^Z`	Stop (suspend) a program or process.
> `ps`	Show process statistics
> `kill <pid>`	Kill a suspended process (get pid from ps)
> `bg`	Continue the (suspended) process in the background.
> `^C`	Stop and Kill a program or process.
> `cp file1 file2`	Copy file1 to file2

If file2 is a dot (.), the files are copied to the current sub-directory, with their original name.

> `alias > filename`	put all known aliases in file
> `alias name command(s)`	Make your own alias (not saved)

Aliases are descriptive combinations of UNIX commands.

> `more filename`	display file on terminal screen
> `p750 filename`	print file on lineprinter (Alliant)
> `pvax filename`	print file on lineprinter (SUN)
> `cat logfile | grep "string" > file`	Find the specified string in an ASCII file (i.e. logfile), and copy its entire line into another file.
> `history 20`	Type the last 20 commands
> `!172`	Execute command line nr 172
> `!!`	Execute the last command line
> `!172:p`	Type command line nr 172
> `^oldstring^newstring`	Edit the last command and execute
> `..`	...

1.5 Koop X11 & NEWSTAR

NEWSTAR general section
RECENT MODIFICATIONS

Editor: J.E.Noordam

January 11, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]gen1_NNEWS.tex

1 NNEWS: Recent modifications to NEWSTAR

The user may keep abreast of recent modifications to NEWSTAR (bug fixes, new options etc) by typing 'nnews' at any time during the session. Note that the list extends backwards in time, with the most recent changes coming first.

```
> nnews
1 Nnews
921222 NPLOT:  Display option X11 now partly available (Do NOT use halftone)..
921218 Changed KEYWORD=LOG. All programs have now default YES iso. SPOOL
921106 Programs changed for J2000 and allow HA outside -90 - +90 degrees.
921022 NATNF, NSCAN and NMAP are now able to use mag-tapes on UNIX system.
       Do a setenv to see which mag-tapes are available.
       NPLOT and NGCALC can now plot on A3-plotter (B17).
920903 Many problems solved by WNB during his stay in Dwingeloo
       NGCALC: New program for data calculation and plotting (C2)
       NATNF:  New program to handle RPFITS files
       NSCAN:  Split mosaic data (B1)
               Error in old WSRT-tapes (1987) .. pointing set (A33)
               Wrong MJD when not correct ended WSRT observation (A34)
               Faster mosaic splitsing (2 times faster) (A35)
               Conversion LINOBS (I,Q,U,V) to XX,XY,YX,YY (B14)
       NMAP:   UV circular weighting function (B5)
               A option/loop (in FIDDLE/ADD) for line-data (B6)
               MOSCOM option to use noise as weight (B11)
               Extended NMAP fits header (B12)
               Change coordinates if shift (A23)
               Noise for extract/copy in NMAP (A28)
               Precission angle calculation (A30)
               Calculate offsets in map (A32)
               Logics MAP statistics (A36)
       NMODEL: Delete of non-clean components (B3)
               Delete of sources inside dl,dm box (B13)
               Merge source models: sometimes recalculate everything (A31)
               RSHOW problem solved (A22)
       NPLOT:  Option to plot hourangle against frequency for a baseline
               polarisation (C7)
               Halftone plotting problem solved (A19)
               Plot message by selfcal/align residuals with model (A24)
               On SUN: Problem plots bigger than A4 solved (A26)
       NCLEAN: Changed sign restore beam angle (A20)
920728 NGIDS:  New program to load maps into GIDS
               (GIDS = Groningen Image Display System)
920714 NSCAN updated for Online System nr. 63 (Change in extended FD)
920626 New delete option DCLOW, same as DNCLOW, but for cleaning components
       NMAP:   Problem with Beam-option correctred (B4)
               Problem with fsum corrected (A18)
               Problem with model subtraction for polarisation sources
               with RM corrected (A16)
920623 Name of software-pakket changed from N-series into NEWSTAR
       (Netherlands East West Synthesis Telescope Array Reduction-package)
920609 NSCAN:  Problem with flagging of two polarisations>
               Layout for MJD(start) corrected (A14)
       NMAP:   Problem with source subtraction corrected (A15)
               Problem with "LAYOUT" option for WMP-file corrected (A13)
920504 NMAP:   Problem with data in HA-baseline format corrected.
```

```
      NSCAN:  Problem with delete options (Rnoise Anoise) corrected (A11)
920407 Problem with option CVX to DEC-workstation corrected.
      NPLOT:  Use of double loops for plotting a MAP corrected (A5)
              Plotting more than one 'data', 'residuals' or
              'telescope corrections' problem corrected.
      NMODEL: Symmetirc extended sources problem corrected (A6)
      NCLEAN: Epoche problem corrected (A9)
      NMAP:   Writing REAL or AMP data from .WMP file to grey-scale
              problem corrected.
920131 NCLEAN with an unsaved residual map corrected
920131 NCLEAN major cycle statistics printout corrected
920130 NPLOT re-write to use full page in portrait mode
920129 NPLOT output to EPS and EPP can be viewed on the DECstation (if files
       produced there or ftp'ed) with the command: dxpsview &
       or with the PostScript preview application
       These files can also be incorporated in WordMarc (and other) documents
920128 Spool error PS plots
920128 Change prompt error in NMAP FIDDLE
920119 NMODEL UPDATE corrected for use of B1950 models
920119 NCALIB SELFCAL corrected for use of B1950 models
920117 NCLEN COMPON option
920117 NCLEAN restoring for extended sources
920117 NCLEAN restoring for non-clean point sources
920116 NCLEAN URESTore option for clean components
920116 NPLOT large plots now ok
920115 NCLEAN UVCOVER with restore option
920109 NSCAN UVFITS memory related crashes for large jobs solved
920109 NMODEL option to delete low-level non-clean changed in definition
920108 NCLEAN Clark type clean (UVCOVER) please test and comments
911230 NMODEL UPDATE corrected for logical error for clean components
911230 NMODEL new option DNCLOW to delete low-level non-clean components
911230 NPLOT better RA, DEC coordinates
911227 NPLOT renewed with ruled surface and polarisation vector possibilities
911227 Correct conversion to/from other coordinates in NMODEL
911219 A new set of Plot routines incorporated in NPLOT (maybe problems)
911209 Correct interferometer selection POL NCALIB
911118 Typo corrected Faraday correction
911115 All (de-)beams have a maximum limit of a factor 100
911105 MAPs made before today at 10:00 will not properly combine in MOSCOM
911105 NMAP FIDDLE MOSCOM will combine mosaic fields prop>
911104 Correct de-selection of clean components (X)UPDATE in NMODEL
911031 WERR option NSCAN will correct mosaic HA tape error
911025 All HA's on WSRT tape for mosaic wrong
911024 NMAP reference coordinate (mosaic) option checked
911023 NPLOT RES definition changed for Ampl and Phase
911014 NSCAN DELETE new CLIP option
911009 NCALIB: SET RENORM option added
911009 NMAP: correct multiple polarisations output
911008 Default: No complex for NCALIB pure redundancy
911007 Instrumental polarisation in first version
911004 NCALIB POLAR VZERO options CALC APPLY MANUAL ASK in first version
911003 The ncopy command will transfer and convert files from VAX
910930 NPLOT: repair logics
910930 NCALIB POLAR COPY has loops
910930 NCALIB REDUN check on extreme values
910927 Multiple input sets ok in NMAP; NPLOT option order reversed
910923 Add SET,COPY,EDIT option to POLAR option in NCALIB
910923 Correct sign of shift in NMAP
910918 Add option to NPLOT to only plot XY,YX
```

```
910918 quota directory|filename ... gives size (may have wildcards) (Alliant)
910917 Correct bug in all programs that mistreated APPLY and DE_APPLY
910917 Status of options as described in ITR198a
       Can be printed (on VAX) by: $ wm/print user5:[wnb.itr]itr2a.wnb
910917 The following programs exist:
       NSCAN, NMODEL, NCALIB, NMAP, NCLEAN, NPLOT (+NGEN for parameters only)
```

General section: PROBLEMS
Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 11, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]gen1_PROBLEMS.tex

1 PROBLEMS

names

What problems can be expected

Untested options in NEWSTAR

Checklist, Whom to call

Comments and wishes

Error report sheet

NNEWS

Help file, giving information about problems (same as NNEWS?). Local to Dwingeloo.

1.1 ..

Chapter 2

PROCESSING RECIPES

NEWSTAR processing recipe 001: Line 21cm

Editor: J.E.Noordam

October 8, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aseries.cookbook]rec1_001.tex

1 Processing recipe 001: 21 cm line

Author: A.G. de Bruyn

1.1 Scope of the recipe

This recipe deals with the reduction of a 21cm line dataset. It describes the calibration of the (complex) passbands using one or more calibrators, the flagging of bad data and the production of an image cube. It shows you how to display this image cube in a movie-like fashion using GIDS. The recipe also includes a description of how to visually identify groups of bad datapoints (e.g. bad interferometers or bad channels) and flag/clip those data. If the field is littered with bright continuum sources at the edge of the field, or when you are working with a relatively wide band, the subtraction of the continuum in the image plane will produce undesirable chromatic effects (the continuum emission itself will go away but not their frequency- scaled sidelobe and grating-lobe responses). A simple procedure to remove the continuum in such cases will be given.

Figure 1:

001 processing appetizer:...
This result was obtained in the following way:
Note the following features:

1.2 Introduction and background

1.2 Introduction and background

This recipe describes the series of steps to go through when you want to calibrate a WSRT 21cm line observation. It assumes that the S/N in the data is rather low and that selfcalibration is not required. It also assumes that you do not have to clean the final images.

1.3 Summary of the recipe

The following is a step-by-step summary of the processing recipe. For some of these steps, more detail is provided below.

1. **Load your data:** from tape or optical disk (NSCAN). See also Recipe 005.
2. **Inspect the data file layout:** (NSCAN)
3. **Determine complex passbands:** from a calibrator (NCALIB)
4. **Inspect the calibration quality:** (NPLOT)
5. **Transfer the calibration corrections:** (NCALIB)
6. **Make a preliminary cube of images:** (NMAP)
7. **Inspect the image cube:** (NGIDS)
8. **Subtract the continuum:** (NMAP)
9. **Make a cube of (calibrated) visibilities:** (NMAP)
10. **Inspect the UV-data cube:** (NGIDS)
11. **Plot your images:** (NPLOT)
12. **Transfer your images to GIPSY or AIPS:** for further analysis.
13. **Dealing with remaining imperfections:** If there are bad groups of datapoints or if the continuum has not been removed satisfactorily the following additional steps may be required:
    - **Flag bad groups of data:** (NSCAN)
    - **Make a new image cube:** (NMAP)
    - **Find and Update source parameters:** of discrete continuum sources(NMODEL)
    - **Make a new image cube:** with discrete sources subtracted (NMAP)

    Then proceed with step 7 and iterate if necessary.

1.4 More details for some of the steps

1.4.1 Load your data

Log in on the computer on which you want to read in the data. At present this is either the microVAX (RZMVX4) or the Alliant. You load data with the program NSCAN, option LOAD. Before you load your data you must know with what time resolution you want to read in the data, and which channels you wish to process. If your object is confined to a small region (say less than 10 arcmin) it may be sufficient to use 120 seconds averaging. If you wish to synthesize bigger fields I recommend using 60 seconds averaging, in order to avoid too much tangential smearing. If the data were Hanning tapered in Westerbork reading in every other channel may speed up the processing by a factor of two. Hence for a 32 channel observation you could answer at the channel question: "2 to 28 by 2". Reading in the continuum channel (channel 0) is generally not necessary; it contains continuum data with contaminating line emission. It is also advised to read in the calibrator data into the same scanfile. If you do this in the second jobstep the data will be assigned the next highest 'group index'.

1.4.2 Inspect the file layout and flag/clip bad points

To find out how the data in the file is organized, and how you can access certain subsets of it, it is suggested to move 'up and down' within the program NSCAN. There are five levels of actions to choose, all of which work on different parts of the dataset. The layout of the file, or the function of the five-digit 'index' that points to the various observations, channels etc. is particularly important to memorize. To avoid that obviously bad points (e.g. correlator spikes) ruin the calibration solution you may wish to flag ('delete') any points with a value higher than a certain limit. Usually the occurrence of correlator spikes is noted on the information you receive from Westerbork/Dwingeloo. Flagging is done on a point-by-point basis using the amplitude of the visibility. Be careful when you use this option if you have no information on the amplitude range which is normal. Preliminary clipping of calibrator data is generally not necessary. During program execution it will inform you at which hourangles interferometers have been clipped. The cutoff value for the object data can of course be much lower. When visibility samples have been clipped they will be ignored in all programs. You can also undo any clipping; you do this by first typing "undelete" at the question where the type of flagging criterion is selected (HA, AN, RN, IFR, CLIP ...).

1.4.3 Determine the instrumental gains/phases across the passband

Run the (self-)calibration program NCALIB (option redundancy) for the calibrator source(s). In order to do this you need a 'model' for the calibrator field containing the fluxes and positions of the calibrator source and any (strong) surrounding sources (see Recipe 013: Using external calibrators). Depending on observing frequency and source there could be anywhere between one and many hundreds of sources in the model file. Upon execution the complex gains are determined, and stored, per telescope/polarisation for each time interval and channel.

1.4.4 Plot the calibration results

In order to decide which calibrator(s) to use, or which part of the calibration observation, you must inspect the quality of the output from the calibartor NCALIB run. Both the printed LOG from the previous step and/or a plot of the telescope complex gains and the residuals from the (self-) calibration solution are helpful to make up your mind. The program NPLOT (options 'telescope' and 'residuals') does the plotting. In general it suffices to plot one channel, somehwere in the middle of the band, to judge the quality of the gains (called amplitude) and phases for the various telescopes. Phase and gain slopes across the passband are usually the same, to first order, for all telescopes. Phase drifts at 21cm are typically a few degrees of phase over a 12 hour period; gain drifts should generally be less than a few occasionally be larger. Consult the printout which contains summary information for each channel. If there are no instrumental problems the gain and phase errors should be equal to the thermal noise values (see Appendix ....).

1.4.5 Transfer the gains/phases from the calibrator(s)

The corrections from the calibrator(s) are averaged over all available (non- flagged) hourangle scans and stored in each scan of the object scanfile. If your calibrator observations have equal length, and no scans have been flagged, the resulting value is just the mean of the individual calibrator averages. There is no provision to weight for the S/N ratio or the amplitude of the calibrator. This means that the transferred gains/phases are representative for a point in time halfway the time of the calibrator observations. This would be a good assumption for gain/phase drifts that go linear with time. For galactic 21cm line observations you will have calibrators observed at frequencies lower and higher (usually by about 1 MHz) than the frequency of your object. Usually these are scheduled before and after the time of observation of your object but this is not essential. If the calibrator observations have equal length then the averaged gain/phase would be appropriate for a frequency halfway the calibrator frequencies as if there is a linear gain/phase drift with frequency. If your object

frequency is not halfway in frequency between the frequencies of the calibrators and you want to store gains/phases in your object file that refer to the frequency of your object, on the assumption that the gains/phase vary linearly with frequency, than you may play around with the length of the respective calibrator observations. This can be done using the NSCAN program (option DELETE, suboption HA (delete).

1.4.6 Make a cube of images

Assuming that you do not want to selfcalibrate the object data (which for line observations is usually the case) you can now proceed to make an image cube. When making images you want to delete any obviously bad baselines. Because you do not yet know whether such bad baselines exist you might as well proceed with making your first series of images. It is possible to exclude visibilities that have an amplitude higher than a certain value. This clipping is not permanent, but only for the execution of NMAP. If the images look good, then all participating baselines are probably good. In an observations with 32 frequency channels, channel 1,29,30 and 31 are usually of poor quality so one would make maps of channels 2 through 28 only. Note that the 3d digit in the index of the .WMP cube will always start counting at 0 and continues through 26. That is, channel 2 is really designated 0.0.0 and channel 28 is 0.0.26.

1.4.7 Visual inspection of images

Making final, good quality, images is generally an iterative procedure where you work in both the image plane and the UV-plane. This procedure is fastest if you can load the cubes directly into the memory of your display (e.g. X- terminal or workstation). GIDS, the Gipsy Image Display System, is a very useful program to do this. Currently it only runs on the Sun If you want to analyse the images in detail, using programs in GIPSY or AIPS) you will have to make FITS-images first. This can also be done using NMAP (option w16fits or w32fits) but is not shown here. Before you can run GIDS (via dwe NGIDS) you have to switch to an X-terminal (in Dwingeloo we have four HP X-terminals) or one of the two colour displays on the SUN (IPX's).

1.4.8 Subtract the continuum form the line channels

After inspecting the cube of line images you can decide which channels to use for the continuum. Use the option FIDDLE (suboption SUM) to create an averaged continuum image. Then use the option ADD to add the continuum channels with weight -1 to each line image to create continuum-free line channels.

1.4.9 Make a cube of the corrected visibilities

If the continuum-free line images look fine you are finished. However, there may be problems at some low level caused by interference, correlator errors or telescope/interferometer errors. Problems at certain position angles in the images correspond to certain hourangles and could be due to short-lived interference. An error pattern in the images, which is centered on the field centre (fringe-stopping centre) and not on strong continuum sources, shows that we are dealing with an additive, rather than multiplicative, error suggests that we are dealing with (DC-) offsets in the backend. To find out which baselines are causing the damage you can make a cube of the (calibrated) visibilities before they are gridded and transformed. You should use the option BASHA instead of UV in NMAP to generate such a cube. Answers to the questions on FFT and map size should be the defaults that are suggested. The best way to pick up low-level problems in the data (e.g. small offsets) is to create outputs for the real (COS) and imaginary (SIN) data. Working with the amplitudes and phases is generally only useful if you want to identify problems of a multiplicative nature and when the signal is well above the noise per visibility point.

1.4.10 Visual inspection of (calibrated) visibilities

The cube of UV-data has a different size than that of the images. With samples of 240 sec there are about 180 points in the horizontal direction and 152 points in the vertical direction (4 x 38). By choosing an 18m baseline increment when making the cube you will get the various baselines well separated and makes it easier to count on the screen which baselines are bad. The shortest baseline (9A=72m) is at the bottom of the 'image'. You can now proceed to make a new series of images where baseline 5B should be thrown out. Then you can define the 'continuum' channels and run NMAP to calculate an average continuum channel. This is not shown here because you may want to do this all in GIPSY or AIPS using FITS images. For the purpose of the demonstration this was done in Dwingeloo and the final 'continuum corrected' cube shows images of channels 7 though 23. These images were 512 x 512 in size and covered an area of 0.6 degrees.

1.4.11 Make contour plots of images

Use the program NPLOT, option MAP to make contour plots of images. There is a variety of output devices. In Dwingeloo you could first send a plot to a (graphics-) terminal (option REGIS) to check the contourvalues that you would like to plot. When you are satisfied you can, by using the 'loop' option, send a series of plots to the QMS plotter in either portrait (QMSP) or landscape (QMS) mode. The former gives you, for square plotareas, somewhat more space to plot in, if at least you want to restrict yourself to a single A4 sheet. Dealing with the size question will take some experience. The default size of 1,1 will give you a plot of at most 5 inch. This is the case if the number of grid points is a power of two (32, 64, 128 etc). When using a size greater than 1,1 on a power-of-two image area in landscape mode the plot will not fit on one sheet and you have to use scissors and tape. On the portrait mode you can actually increase the size to 1.4,1.4 before it needs more than a single A4 sheet.

1.4.12 Transfer images to AIPS or GIPSY

You can use the FITS write option in NMAP to generate 2 byte or 4 byte integer images in FITS format for further analysis in AIPS or GIPSY.

1.5 Dealing with remaining imperfections

If the inspection of the continuum subtracted image cube shows imperfections due to bad data or poor continuum subtraction you have to continue the reduction a bit further.

1.5.1 Further flagging of bad data.

If the inspection of the visibility cube using NGIDS has indicated that there are errors in certain hour-angles or baselines you can flag these using the delete option in NSCAN. If there are only bad baselines you can also decide to exclude these baselines when preparing the specifications for a new NMAP. That is, you do not have to actively delete them.

1.5.2 Make a new image cube.

If the new image cube shows, after subtracting some averaged continuum image, residual (chromatic) grating rings due to strong sources at the edge of the field you need to subtract these sources in the original UV-data.

1.5.3 Finding and updating discrete source parameters.

In order to determine the flux densities and positions of these sources you search the continuum image using the FIND option in NMODEL. To improve the estimates of flux and coordinates you can use the UPDATE option in NMODEL. A single update, using only the UV-data with coninuum emission, usually suffices. The update algorithm will only work for discrete (point) sources. If there are extended sources with residual grating lobes you need to use NCLEAN (option Beam) to decompose that source into a number of delta-functions. The final list of components (updated discrete sources and clean- components) can be merged into one model file.

1.5.4 Make a new image cube with sources subtracted.

Now you proceed with step 6, but subtract the list of sources found in the previous step.

NEWSTAR processing recipe 002: Linear polarisation

Editor: J.E.Noordam

January 11, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file users:[noordam.aseries.cookbook]rec1_002.tex

1 Processing recipe 002: Linear polarization

Author: R.G. Strom

1.1 Scope of the recipe

For obtaining polarization maps from standard continuum observations (all four dipole combinations must be present). See also the description of the program NCALIB, part 3.

1.2 Introduction and background

In a normal WSRT continuum observation, four dipole combinations (XX, XY, etc.) are measured for each fixed-movable (numbered-lettered telescope) baseline. Since mid-1983, the standard configuration is for the X-dipoles in all telescopes to be set parallel (hence the Y-dipoles are also). A polarization code is recorded for each observation, which gives the X-dipole position angle in the fixed and movable telescopes, in steps of 45°. The usual setting for the X-dipoles is 90° (i.e., directed toward increasing RA), which gives a polarization code of 22. This results in the following definition of the Stokes parameters:

$$I = \frac{1}{2}(XX + YY)$$

$$Q = \frac{1}{2}(YY - XX)$$

$$U = \frac{1}{2}(YX - XY)$$

$$V = -\frac{i}{2}(XY + YX)$$

(where $i = \sqrt{-1}$). In the remainder of this description it will be assumed that the observation being corrected has code 22, and of course that all four dipole combinations have good data.

Most continuum observations made before mid-1983 were done with the dipoles in the movable telescopes rotated by 45° to the fixed ones, usually with polarization code 21 [or in any event with the digits differing by an odd number]. For analyzing such observations, the user should consult with the WSRT User Service in Dwingeloo. From time to time, calibration sources are still observed in code 21 [generally indicated by '+×' appended to the source name]. In almost all line observations, only one or two of the dipole combinations (XX, YY) are measured; they are of course unsuitable for determining source polarization.

The determination of the (observed) Stokes Q depends, as shown by the first two equations above, upon how well the gains of the XX and YY channels can be ascertained (since usually, $I >> Q$). For the U and V determinations, the critical parameters are the dipole setting or orthogonality, and the ellipticity. Moreover, since V is usually small (i.e., $V << U$) its determination will also depend upon the quality of the XY and YX gains. The other parameter required to correct the XY and YX combinations is the $X - Y$ phase difference. The correction is usually small (since it should have been determined and applied online in Westerbork) and has to be determined from an observation of a polarized source, or a special 'crossed' (code 21) calibration source measurement. Finally, it may be necessary to correct for Faraday rotation in the ionosphere (this is usually only necessary at 49 and 92 cm), and for variations in the instrumental polarization if there is significant emission beyond the central part of the primary beam.

1.3 Summary of the recipe

The following is a step-by-step summary of the processing recipe. For some of these steps, more detail is provided below.

1. **Load your data:** from tape or optical disk (NSCAN, option LOAD) See also Recipe "Reading data in Dwingeloo".
2. **Inspect the data file layout:** (NSCAN, option SHOW)
3. **Determine the instrumental gain and phase corrections for a strong calibrator with known polarisation:** (NCALIB, option REDUN.)
4. **Determine or set the instrumental polarization corrections for an unpolarised calibrator:** (NCALIB, option POLAR.)
5. **Map and inspect Stokes Q, U, V for the calibrator (optional):** (NMAP, followed by NPLOT or NGIDS.)
6. **Copy (or set by hand) the instrumental polarization corrections for the observed field:** (NCALIB, option POLAR COPY.)
7. **Determine, and apply if necessary, the correction for ionospheric Faraday rotation (optional, usually only necessary at 49/92 cm):** (NCALIB, option SET FARADAY.)
8. **Map and inspect I, Q, U, V images of field:** (NMAP, followed by NPLOT or NGIDS)
9. **Primary beam correction for each Stokes parameter (may only be necessary if source extends over large fraction of primary beam):** (NCALIB, options BEAM_FACTORS (for Stokes I) and INPOL* (where * $= Q, U, oriV$ for the other three Stokes parameters).)
10. **Map the final corrected image in all Stokes parameters:** (NMAP)

1.4 More details for some of the steps

1.4.1 Load your data

Remember that in addition to your own observation, you will probably want to load one or more calibration source measurements. Choose an unpolarized calibrator (3C 147 is probably the best) observed within a day of your observation *and with the same instrumental settings* (load several calibrators if you want to check on the repeatability of the solution), and a polarized source (like 3C 286, except at 92 cm) to determine the $X-Y$ phase difference. To locate calibration measurements, use ARCQUERY (see Appendix B). General information on the use of external calibrators can be found in Recipe 13 (which refers to NCALIB option SET COPY), while loading data is described in Recipe 5 (which refers to NSCAN option LOAD).

1.4.2 Inspect the data file layout

Make certain, in particular, that all four polarization combinations (XX, XY, etc.) are present in all observations to be used, and that the calibrators are consistent with the observation to be corrected (frequencies and bandwidths should be the same, although a change in spacing shouldn't make any difference). If the time between calibrator and observation is more than a day or two, it is also advisable to check that no frontends have been changed (consult the logbook or reduction group). Inspect data for interference (which may be polarized – check the XY and YX combinations in particular) and other defects. For more information on inspecting data files, see NSCAN option SHOW.

1.4.3 Determine the instrumental polarization

Before determining the instrumental polarization using a calibrator, redundancy and selfcal solutions must be applied (use SELFCAL and ALIGN in the NCALIB option REDUN). This will also provide a useful check of the data quality. Note that at 92 cm (and in some cases 49 cm) there are background sources which may have to be included in the model used for ALIGN. Having run the ALIGN solution, NCALIB can be used to calculate the instrumental polarization (POLAR_OPTION: CALC) and examine the result (POLAR_OPTION: SHOW). Under normal conditions, the orthogonalities and positions of most dipoles should be under 1°, and the ellipticities generally under 1%. Large values (more than a few degrees or percent) probably indicate an instrumental problem (bad data) and require further investigation. Run a solution on a different calibrator as a check. Deviant points can be changed by hand (POLAR_OPTION: EDIT). Tables of instrumental polarization are also generated from time to time in Westerbork, and the values can be entered by hand (POLAR_OPTION: SET), or used to cross-check the solution from NCALIB.

As a check on the instrumental polarization thus generated, it may be useful to make a map of an unpolarized calibrator if one has been observed for a few hours or more during the same period (a shorter observation could also be used, but the map might prove difficult to interpret). Make sure that SELFCAL and ALIGN corrections have been successfully applied, and then look at the Q, U and V maps (made with NMAP). Ideally, they should be zero; the residual as a fraction of the flux density is an indication of the error which will be present in the polarization map of your observation. (If your source is very extended, however, the polarization error pattern generated by a point source may be misleading).

Finally, we have to determine (or at least check) the $X - Y$ phase difference. This is best done using a linearly polarized calibrator (strictly speaking, a source with strong U signal). The method assumes that V is much smaller than U (since $YX = U + iV$, a nonnegligible V affects the $X - Y$ phase), which is usually the case. The correction can be calculated (VZERO_OPTION: CALC, ASK, etc.) and applied to the data in several ways. Usually, the XY and YX phase zeroes have been determined and applied on-line to sufficient accuracy for most polarization maps, so the correction should be a few degrees or so. However, there have been instances where the difference was as much as 30°, so it is advisable to check. If various frequency channels are used, the phase difference should be calculated for each one separately, as the correction is usually frequency dependent.

1.4.4 Correct the observation for instrumental polarization

If you are happy with the instrumental parameters as applied to the calibrator(s), the values can be copied to your observation using NCALIB (under POLAR_OPTION select COPY). The corrections will then be applied to the data when making a map with NMAP. If you have run selfcal (ALIGN) on your XX and YY polarizations using a source model based on Stokes I ($XX + YY$), remember that solving for gain (keyword SOLVE) could remove most of the Q signal.

1.4.5 Correct for ionospheric Faraday rotation

At 49 and 92 cm, Faraday rotation in the ionosphere can be considerable. The effect is to change the position angle (p.a.) of the plane of linear polarization (for the WSRT the p.a. will always *increase*, so the shift is systematic). Since the ionosphere changes throughout the day, the amount of rotation may vary during an observation. This has two consequences: the average position angle will be increased, and more seriously, if the differential rotation exceeds about one radian, there will be decorrelation of the polarized signal and distortion of the Q and U maps.

To correct for ionospheric Faraday, the reduction group must be consulted to generate the predicted rotation using ionosonde data and information about the observation in question (the date, time and source position are needed, and the correction should be done for a frequency of 1 GHz). This will produce a table with hourly values of various parameters,

including the source HA and the amount of rotation. The HA and rotation at 1 GHz (both in degrees) have to be entered in an ASCII file (see the description in NCALIB, option FARADAY under SET_OPTION), and the correction is applied using FARADAY. Note that the HAs do not have to be hourly, or even regularly spaced: at night there may be little change in the Faraday rotation, and a single correction might suffice.

1.4.6 Correct for primary beam instrumental polarization

The instrumental polarization determined from a calibrator is, strictly speaking, only correct for a source at the beam center, though at most frequencies the variation within the half-power primary beam is small. For polarization mapping over large fields, there is a separate correction for off-axis instrumental polarization (NCALIB: NMODEL - BEAM).

1.4.7 How do I know that my data are correct? – additional hints.

Errors; some error patterns; effect of source extent.

Ionospheric effects and the I map.

What can be done if a polarization combination is missing?

Etc.

NEWSTAR processing recipe 003: MOSAICKING 21CM

J.E.Noordam (editor)

September 25, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.naeries.cookbook]rec1_RECIPENR.tex

1 Processing recipe 003: Mosaicing 21 cm.

Author: T. Ghosh

1.1 Scope of the recipe

This recipe suggests a way for making a 21-cm continuum mosaic map. In this example, a particular field was chosen where there were many previously-known radio sources within the region to be mapped. Hence, we knew that there was sufficient flux-density for almost all of the pointing centres to allow us to run *selfcal*. If the user is not sure of having enough SNR for this, a modified version can be used. Hence, consider this recipe as just a starting point, and *add salt to your taste.*

Figure 1:

003 processing appetizer:...
This result was obtained in the following way:
Note the following features:

1.2 Introduction and background

In the mosaic mode of observation, during one 12-hour period the telescopes, along with the fringe-stopping and the delay-tracking centres, cycle through a grid of pointings a number of times. In Fig 003.1, we used a 13 x 5 grid of these so called, *pointing centres.* There were about 21 cuts (or spokes), each containing two scans of 10-s duration for each pointing centre. The total 40-MHz bandwidth at 21 cm is usually divided into eight 5-MHz channels (although these are not contiguous). Hence, the data file had the following set indices (see also Ch 3.):

$$0.0.1 - 65.1 - 8.0 - 20$$

For a particular pointing centre, the reduction methodology is very much the same as that in Recipe 004. **However, all the maps are to be made using a common reference point (NMAP-option REFERENCE).** Usually, for a large number of pointing centres, book-keeping etc. could be quite difficult. Here, all the advantages of the Automatic Batch Processing facility (ABP, Appendix D) can be very easily exploited. In the following section we detail all the steps, indicating the pre-ABP, the ABP, and the post ABP stages, Stage I, II and III respectively, that were used to generate the map in Fig 003.1. These three stages also coincide with the *calibration using a different source*, the *Selfcal-Model formation loop for all the pointing centres*, and the *mosaicing* stages.

1.3 Summary of the recipe

STAGE I:

1. **Load your data:** Source data from tape or optical disk (NSCAN). See also Recipe "Reading data in Dwingeloo". Also read the data from observations of a calibration source, (e.g. 3C147 in the case of Fig 003.1) made just before or after the *source field.*
2. **Inspect the data file and flag bad points mentioned on the WSRT green sheet :** (NSCAN, option SHOW etc).
3. **Calculate antenna-based phase and gain correction factors for the Calibrator data:** (NCALIB, option REDUNDANCY, suboption SELFCAL, using a model file obtained from AGB)
4. **Copy the corrections to the source data:** (NCALIB, option SET, suboption COPY)

STAGE II:

This stage could be batch processed (APPENDIX D). In the following, we first write a name for each task, which briefly describes the function of the step too. Within the bracket, we mention the programme to be used, and comment on a few important input parameters. Depending upon the flux density of the strongest source in the map of a particular pointing centre, the Stage is stopped at various **Exit points**.

For each pointing centre:

1. **Raw:** (NMAP, make a $1024x1024$ pixel raw map of real size $1^\circ.2x1^\circ.2$)
2. **Find1:** (NMODEL, find model components down to a suitable limit e.g. 6 w.u. here)

Exit point 1: If the flux density of the strongest component, S_h is less than limit 1 (12 w.u., here) - STOP - Proceed to next pointing centre.

1. **Upd1:** (NMODEL, Update the model list, delete sources weaker than a certain limit after updating, e.g. 6 w.u)
2. **Upd2:** (NMODEL, Second iteration of the above)
3. **Self1:** (NCALIB, self-calibrate ONLY the PHASE of the data using model, Upd2)
4. **Sub1:** (NMAP, make $1^\circ.2x1^\circ.2$ map after subtracting model, Upd2)

Exit point 2: If $S_h <$ limit 2 (20 w.u. here) - STOP - Proceed to next pointing centre.

1. **Find2:** (NMODEL, find model components from map, Sub1 down to a suitable limit e.g. 5 w.u. here and add to the model, Upd2)
2. **Upd3:** (NMODEL, update the model list, Find2 and delete sources weaker than a certain limit e.g. 5 w.u here)

3. **Upd4:** (NMODEL, second iteration of the above)
4. **Self2:** (NCALIB, self-calibrate the data using model, Upd4)
5. **Del1:** (NSCAN, delete scans with selfcal-amplitude and/or phase noise $> 2\sigma$)
6. **Sub2:** (NMAP, make $1^\circ.2x1^\circ.2$ map after subtracting model, Upd4)

Exit point 3: If $S_h <$ limit 3 (40 w.u. here) - STOP - Proceed to next pointing centre.

1. **Find3:** (NMODEL, find model components from map, Sub2 down to a suitable limit e.g. 2.5 w.u. here and add to the model, Upd4)
2. **Upd5:** (NMODEL, update the model list, Find3 and delete soucres weaker than a certain limit e.g. 2.5 w.u here)
3. **Upd6:** (NMODEL, second iteration of the above)
4. **Self3:** (NCALIB, self-calibrate the data using model, Upd6)
5. **Del2:** (NSCAN, delete scans with selfcal-amplitude and/or phase noise $> 2\sigma$)
6. **Sub3:** (NMAP, make $1^\circ.2x1^\circ.2$ map after subtracting model, Upd6)

End of stage II

At this point, for all the pointing centres, one should have a good model list, and a residual map. The common model components in the overlapping region can be checked for positional agreement. These can be cross-checked against any source within the region whose accurate position is known from the literature. Usually, agreements within 1 arcsec was achieved at this stage, which was already 1/10 th of the synthesised beam.

Depending upon the quality of the maps, they can be grouped into four classes.

A. Good maps, where noise is already at the theoretical limit, though there may still be residual sources, and their grating rings and sidelobes.

B. Maps containing grating rings from sources beyond the mapped region.

C. Maps with bad scans.

D. Maps with amplitude or/and Phase calibration problems.

For Class B maps, make larger low-resolution maps and try to include the strong (offending) source in the model list, and then start again at the beginning of Stage I (retaining the 'outside sources', though).

For Class C maps, search out the bad data (no easy-solution, sorry), delete it, and start again.

For Class D maps, there may be spurious sources in the model-list. Plot them, (NPLOT) and try to locate suspects, delete these, and continue (refer to sec..... if this doesn't help).

STAGE III:

Once all the maps are of Class A-quality, do the following:

1. **Mkbeam :** (NMAP. Make beams for all the pointing centres)
2. **Clean :** (NCLEAN, option UVC. Clean the entire map with the number of components = 100, Gain = 0.1 and a Cycle-Depth value of 0.35)
3. **Restore :** (NCLEAN, option UREST. Restore both the model and the clean components)
4. **Extract :** (NMAP, option FIDDLE, suboption EXTRACT. Write out a smaller section of this map into one set index of a common map file that will be the input for the MOSAIC combination, here, inner 700 x 700 pixels of each pointing centre were used)

5. **Mosaic :** (NMAP, option FIDDLE, suboption MOSC. This part of the programme corrects the input maps for primary beam attenuation and then averages them with proper weightage)

6. **Ready to serve:** (NPLOT, NGIDS, write out FITS image and take it to AIPS ..... use the garnish of your choice)

Chapter 3

DATA FILE DESCRIPTION

3.1 Overview of NEWSTAR files

3.2 The SCN-file (uv-data)

3.3 The MDL-file (model components)

3.4 The WMP-file (image-data)

Overview of NEWSTAR files

Editor: J.E.Noordam

January 22, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aperlee.cookbook]file1_FILES.tex

1 Overview of NEWSTAR files

The NEWSTAR programs recognize three major types of data files, which are identified by their extension:
.SCN files: uv-data and uv-model
.MDL files: SELFCAL source model
.WMP files: maps of various kinds

NEWSTAR programs also produce a number of auxiliary files, with the following extensions:
.PLT-files: to be plotted
.LOG-files: to be printed
.NGF-files: produced by NGALC
.MNG-files: used to make plots with the MONGO package

1.1 Directories and nodes (files)

For historical reasons, the NEWSTAR programs refer to data files as **nodes** in a **database**. In fact, they are just files in a directory. Users are recommended to use a separate sub-directory for each data reduction project.

However, although they refer to the same thing, there is a slight difference between **node names** used by the user and the actual **file names**: Node names are a series of alpha-numeric character strings separated by dots, e.g.:

```
mynode
mynode.21cm.yesterday.s.x.c.d.file.dd
```

The maximum length of a node name is 80 characters (which means maximally 39 dots). This is converted to a *file name* by:
- appending the extension (.SCN, .MDL or .WMP),
- converting all dots (.) to underscores (_),
- converting all lowercase to uppercase,
- replacing the underscore nearest but less than 47 to a dot. (?)

This file name is then prefixed with the current database (default nothing).

Parts of a node name can be set aside for short-hand use. This can be done by the INFIX keyword (see COMMON keywords), or in a node specification by enclosing a part in parenthesis (). This enclosed part will from then on be available to all programs that are run in **the current stream**. Reference to 'the infix' is made by typing a #, e.g.:

```
mynode.21(cm.yesterday.s.x.c.d.file.d)d
```

Typing `mynode.92#e` will then produce `mynode.92cm.yesterday.s.x.c.d.file.de`

The database (directory) name can also be included in the node definition:

```
wnb/data/mynode
../other/mynode.21cm.yesterday.s.x.c.d.file.dd
```

If a database is specified in this way, it will be saved in such a way that all subsequent program runs **in the same stream** that have no explicit database specified, will use it, including the currently running program at all its further node questions.

1.2 The NEWSTAR data files (SCN, MDL, WMP)

These three NEWSTAR data files are each described in detail in their own 'File Description' section in this Cookbook. In this Overview section, it is explained how the basic units (called Sets) in such a file can be selected individually or in groups. For each of the three data files, an overview is given of the various ways in which the user may interact with these 'objects', with references to the relevant program options.

1.2.1 Data file logical organisation: Sets

The three main NEWSTAR data file (node) types are logically organised in the same way: they consist of a number of basic units called 'Sets', which can be selected by the user in two ways:

- By **direct reference** to the set index: `#ir` selects set nr `ir`, in which `ir` can be a range of contiguous sets (see below). Direct reference is only useful if the user knows exactly how the various Sets are ordered in the file.
- By **indexed reference** to a series of indices that represent parameters like map or channel nr. A 'selector' consists of a series on integer indices, separated by points. The indices may also be specified as ranges of indices or wildcards (*). For instance: `ir1.ir2.*.ir4` selects all Sets with index values in the indicated ranges.

Index ranges (`ir`) can take the following forms (**NB: Indices start at 0!**):

int	single index value at this level
*	wildcard: all possible index values at this level
int-	all index values at this level, starting at 'int'
int1-int2[:int3]	index values 'int1' through 'int2' by step 'int3' (dflt=1)
int1:int3	interpreted as int1-*:int3
omitted	in the middle: interpreted as .* (.. ≡ . * .)
omitted	at the end: interpreted as .* (..2 ≡ *. * .2. * . * .*)

1.2.2 Using loops

Sometimes a program must be run more than once for the same SCN-file, but for different Sets. In that case, the **LOOPS** keyword can used to specify subsequent ranges of Sets. A loop is specified by means of a pair of values: The first value indicates how often the loop should execute, the second specifies an increment to be given to the Set specification at each run.

For example: `loops=3,..2` indicates that the program has to run three times, each time incrementing the 3rd Set index by 2. So, if the first Set (selected with the keyword SETS) was `0.0-3.2.5.*`, the program will be run three times, for the Sets

```
0.0-3.2.5.*.
0.0-3.4.5.*.
0.0-3.6.5.*.
```

spatie belangrijk!

Nested loops can be specified by stringing loop definitions (pairs of values) together. For instance: `loops=3,..2,5,...3` adds an inner loop of 5 program runs in which the 4th Set index is incremented by 3 each time:

```
0.0-3.2.5.*.   0.0-3.2.8.*.   0.0-3.2.11.*.   0.0-3.2.14.*.   0.0-3.2.17.*.
0.0-3.4.5.*.   0.0-3.4.8.*.   0.0-3.4.11.*.   0.0-3.4.14.*.   0.0-3.4.17.*.
0.0-3.6.5.*.   0.0-3.6.8.*.   0.0-3.6.11.*.   0.0-3.6.14.*.   0.0-3.6.17.*.
```

1.3 Overview of interactions with the SCN file

The SCN-file contains uv-data, and possibly the uv-representation of a source model. A single SCN-file may contain uv-data of different (but related) observations.

For more information, see the dedicated SCN-file section in this chapter of the Cookbook, and also the section on the program NSCAN. It contains a description of the structure and the contents of the SCN-file.

1.3.1 Creating SCN files

- From WSRT circle files: NSCAN option LOAD.
- From ATCA (Australia Telescope Compact Array) files: NATNF.
- From old (R-series) SCN-files: NSCAN option FROM_OLD.
- Simulated uv-data: NSIMUL? (not implemented yet).

1.3.2 Inspecting the contents of a SCN file

- File Layout: NSCAN option SHOW
- File header: NSCAN option SHOW
- Set headers: NSCAN option SHOW
  - Telescope (dipole) angle/ellipt corr: NCALIB option POLAR SHOW
- Scan headers: NSCAN option SHOW
  - Telescope gain/phase corr: NCALIB option SHOW
- uv-data (corrected, converted): NSCAN option SHOW
- Display of (gridded) uv-data or uv-model: See WMP file below.
- Plot telescope gain/phase corr (REDC+ALGC+OTHC): NPLOT option TELESCOPE
- Plot Redundancy/Selfcal residuals: NPLOT option RESIDUAL
- Plot uv-data or uv-model: NPLOT option DATA or MODEL
- Print average telescope gain/phase corr (R+A+O): NCALIB option SHOW
- Extract various astrophysical info: NGCALC

1.3.3 Editing the header information of a SCN file

Almost every value (observation parameters, corrections, etc) in the SCN-file headers may be edited manually by means of NSCAN option SHOW EDIT. This means that, even if there is no specific NEWSTAR routine to change something, it can always be done by hand. This may be laborious in some cases, *but at least it is possible!*. This feature is particularly useful in those (rare) cases where erroneous information has somehow been put in the header during the observations in Westerbork.

1.3.4 Applying corrections to uv-data

In general, uv-data in a SCN-file is *never physically modified*. Corrections may be applied (or de-applied) to the uv-data whenever the data is read into memory to be processed. The user may specify which corrections are applied (or de-applied) by specifying the value of the general NEWSTAR keywords APPLY and DE_APPLY, which are used by all NEWSTAR programs that handle uv-data. Use `dws NGEN` (see Common features of NEWSTAR programs).

1.3.5 Modyfing the stored corrections

Zeroeing selected corrections: NCALIB option SET ZERO

Set Header (corrections that are 'constant' in time):
- Any value: NSCAN option SHOW EDIT

- POLC (dipole angle error and ellipticity corr):
  - Estimation: NCALIB option POLAR CALC
  - Manual: NCALIB option POLAR SET, EDIT, ZERO
  - Copying from calibrator: NCALIB option POLAR COPY

Scan Header (corrections that vary per HA):
- Any value: NSCAN option SHOW EDIT
- Telescope (dipole) gain/phase corr:
  - Estimation: NCALIB option REDUN
    - Redundancy (no model): REDC
    - Align: ALGC
    - Selfcal ALGC
  - Manual: NCALIB option SET MANUAL, ZERO, RENORM
  - Copying from calibrator: NCALIB option SET COPY, CCOPY, LINE
- Phase Zero Difference: OTHC
  - Estimation: NCALIB option POLAR VZERO CALC, APPLY, ASK, SCAN
  - Manual: NCALIB option POLAR VZERO MANUAL, ASK
  - Copying from calibrator: NCALIB option POLAR VZERO COPY
- Extinction corr (manual): NCALIB option SET EXTINCT
- Refraction corr (manual): NCALIB option SET REFRACT
- Faraday corr (input of ionosonde data): NCALIB option SET FARADAY

1.3.6 Modifying a uv-model in the SCN-file

See Overview of interactions with MDL file below.

1.3.7 Reorganisation of SCN files

- Create new 'Job' from Sets in the same SCN-file: NSCAN option REGROUP
- Create a 'secondary' SCN-file (data selection, correction): NCOPY
(not implemented yet)
- Delete SCN-file: Use UNIX command `rm <xxx>.SCN` (be careful!)

1.3.8 Export of uv-data from SCN files

- To UVFITS format (AIPS): NSCAN option UVFITS, PFITS
- To old (R-series) SCN-file format: NSCAN option TO_OLD
- To WMP file (as gridded uv-data or maps): NMAP option MAKE

1.4 Overview of interactions with an MDL file

A MDL file contains a collection of source model components. This may consist of a mixture of **multi-parameter components** and ordinary **CLEAN components**. The MDL file may also contain **reference coordinates** (obtained from a SCN-file) for the position of the field centre and the observing frequency.

For more information, see the dedicated MDL-file section in this chapter of the Cookbook, and also the section on the program NSCAN. It contains a description of the structure and the contents of the MDL-file.

NB: Note that the model in the MDL file may be manipulated by means of NMODEL **options** (i.e. options of the program NMODEL), or by MDL **handles**. The latter operations (e.g. READ, WRITE, EDIT, SHOW) are available in all the programs that deal with source models: NCALIB, NMAP, NCLEAN, NSCAN, NMODEL.

1.4.1 Adding source components to an MDL file

- Manual, by specifying source parameters: MDL handle ADD.
- Automatic search of a map (in a WMP file): NMODEL action FIND
- CLEANing a map (in a WMP file): NCLEAN option BEAM, UVCOVER, COMPON
- Save model from SCN file in an MDL file: NMODEL option SAVE
- Convert old (R-series) model to NEWSTAR MDL file format.

1.4.2 Modifying a model in an MDL file

Modifying **source components**:
- Improve source parameters by fitting to the uv-data: NMODEL option UPDATE, XUPDATE
- Manual editing: MDL handle EDIT, FEDIT
- Delete:
  - All components: MDL handle CLEAR, ZERO
  - Selected components: MDL handle DELETE, DNCLOW, DCLOW, DAREA
  - Selected components: MDL handle EDIT, FEDIT
- Calibrate (position, flux): MDL handle CALIB
- Combine components at the same position: MDL handle MERGE
- Correct for primary beam attenuation: NMODEL option BEAM, DEBEAM

Modifying **reference coordinates**:
- Change epoch or coordinate system: NMODEL option CONVERT

NB: Reference position and frequency are obtained from SCN-file.

1.4.3 Inspecting, displaying and sorting an MDL model

Inspecting:
- Show list of components: MDL handle SHOW (screen) or PRINT (log-file)
- Idem, in RA/DEC coordinates: MDL handle RSHOW, RLIST
- Show source list statistics: MDL handle TOT

Displaying:
- As gridded uv-model from SCN file: See WMP file below
- As a map of the uv-model in the SCN-file: See WMP file below
- As position markers in a map: NPLOT option MAP
- As 'restored' components in a CLEAN residual map: NCLEAN option UREST

Sorting:
- According to decreasing flux: MDL handle SORT, WRITE(!)
- According to the value of another source parameter: MDL handle FSORT

1.4.4 Relation with the uv-model in the .SCN file

The source components in an MDL file may be Fourier transformed to the uv-plane, to the uv-coordinates of the uv-data in a SCN-file. This **uv-model** is then 'saved' in the SCN-file, together with a copy of the MDL source components that produced it.

Whenever a uv-model is needed (e.g. for Selfcal in NCALIB, or source subtraction in NMAP) the user is always asked to specify a **input model** explicitly, even if there is already a **saved uv-model** in the SCN-file. This input model may be specified either by reading (and editing) components from an MDL-file, or by editing source components manually. In the following, the Fourier transform of the input model will be called the **input uv-model**. The user may choose (keyword MODEL_ACTION) one of the following possibilities:

- **Merge:** Replace the saved uv-model with the input uv-model, and use it.
- **Add:** Add the input uv-model to the saved uv-model in the SCN-file, and use it.
- **New:** Replace the saved uv-model in the SCN file by the input uv-model, and use it.
- **Temporary:** Use the input uv-model, but do not change the saved uv-model in the SCN-file.
- **Increment:** Use the sum of the saved uv-model and the input uv-model, but do not change the saved uv-model in the SCN-file.

NB: NEWSTAR regards the saved uv-model as *applied corrections*, i.e. corrections that were applied (added) to the cosmic noise before the uv-data were put into the SCN file. Hence, to subtract the saved uv-model from the data, one could specify 'MOD' to the (NGEN) keyword DE_APPLY! Specifying 'MOD' to the (NGEN) keyword APPLY will restore a model that was subtracted from the data before it was put in the SCN file (?).

1.4.5 The various uses of an MDL model

Present uses of an MDL model:
- To solve for telescope gain/phase errors (Selfcal, Align): NCALIB option REDUN
- To subtract sources from the uv-data: NMAP option MAKE
- To combine multiple observations:
  - With different frequencies (broad-band mapping)
  - With different pointing centres (mosaicking)
  - With different observing times
- To deal with instrumental polarisation
- To undo large Faraday 'rotation measures'
- To detect variability
- To simulate uv-data: clumsy at the moment

Potential uses of an MDL model:
- To deal with non-isoplanaticity
- To simulate uv-data: NSIMUL (not yet implemented)

1.5 Overview of interactions with a WMP file

A WMP contains a collection of 2-dimensional arrays, that are related in some way (but that can have different dimensions). These may be maps for various frequencies (line data) or pointing centres (mosaicking), antenna patterns and CLEAN residual maps. Even rectangular arrays of (gridded) uv-data may be put in a WMP file, for display purposes.

For more information, see the dedicated WMP-file section in this chapter of the Cookbook, and also the section on the program NSCAN. It contains a description of the structure and the contents of the WMP-file.

1.5.1 Creating maps in WMP files

- Making maps/ap's from uv-data (or uv-model) in SCN-file: NMAP option MAKE
  - Many types of maps: XX,YY,XY,YX,I,Q,U,V,cos,sin,ampl,phase,...
- Residual maps: NCLEAN option BEAM, UVCOVER
- Restored maps: NCLEAN option UREST
- Gridded uv-data from SCN-file: NMAP option MAKE REAL, IMAG, AMPL, PHASE
  - In various forms: real or imaginary parts,ampl,phase,...
- Gridded uv-coverage from SCN-file: NMAP option MAKE COVER
- Copy maps: NMAP option FIDDLE COPY
- Extract areas from maps: NMAP option FIDDLE EXTRACT
- Convert from old (R-series) map files: NMAP option FROM_OLD

Delete WMP files: Use operating system UNIX: `rm <....>.WMP` (be careful!)

1.5.2 Inspecting the contents of a WMP file

- Show header information: NMAP option SHOW.
- Display images on color screen (X): NGIDS
- Make plots on X-screen or plotter: NPLOT option MAP - Various types: contour, greyscale, polar, ruled surface

1.5.3 Editing the WMP header information

- Edit header information: NMAP option SHOW.

1.5.4 Operations on WMP images

Although NEWSTAR has primarily been designed for WSRT *uv-data* processing it offers some powerful image-plane features:

- Various map operations (very powerful): NMAP option FIDDLE
  - Add, subtract, average, extract, copy, mosaic, etc
- Cleaning and restoring: NCLEAN option BEAM, UREST
- Finding strong sources: NMODEL option FIND

For many astrophysical projects, this will be sufficient. However, for more advanced operations astrophysical image analysis, the user should transfer the images from the WMP file to other packages (e.g. GIPSY, AIPS) by means of the FITS format: NMAP option W16FITS, W32FITS.

1.6 Auxiliary NEWSTAR files

1.6.1 .LOG files

Each NEWSTAR program run produces a log-file, with the name `<progname>.LOG`. It contains the 'vital information' about the program run: keyword values (including the hidden ones) and essential results. In some cases (e.g. NCALIB REDUN), the user may specify how much information is printed in the log-file (keyword SHOW_LEVEL).

The user may specify (NGEN keyword LOG) what happens with the log-file upon completion: it may be spooled automatically to the line printer, or thrown away, or kept for later inspection. In the latter case, the log-file is automatically renamed with a unique name (e.g. `NCA<alphanumeric>.LOG`) when the program is run again.

1.6.2 .PLT files

All plot files produced by the programs NPLOT and NGCALC have the extension .PLT. The file names usually begin with the 3-4 letter code of the selected PLOTTER option (PSP, PAL, EMS etc), followed by a unique combination of alphanumeric characters derived from the date and time of creation.

1.6.3 .NGI files

NGI files are used to store the various information that the program NGCALC extracts from the SCN-file.

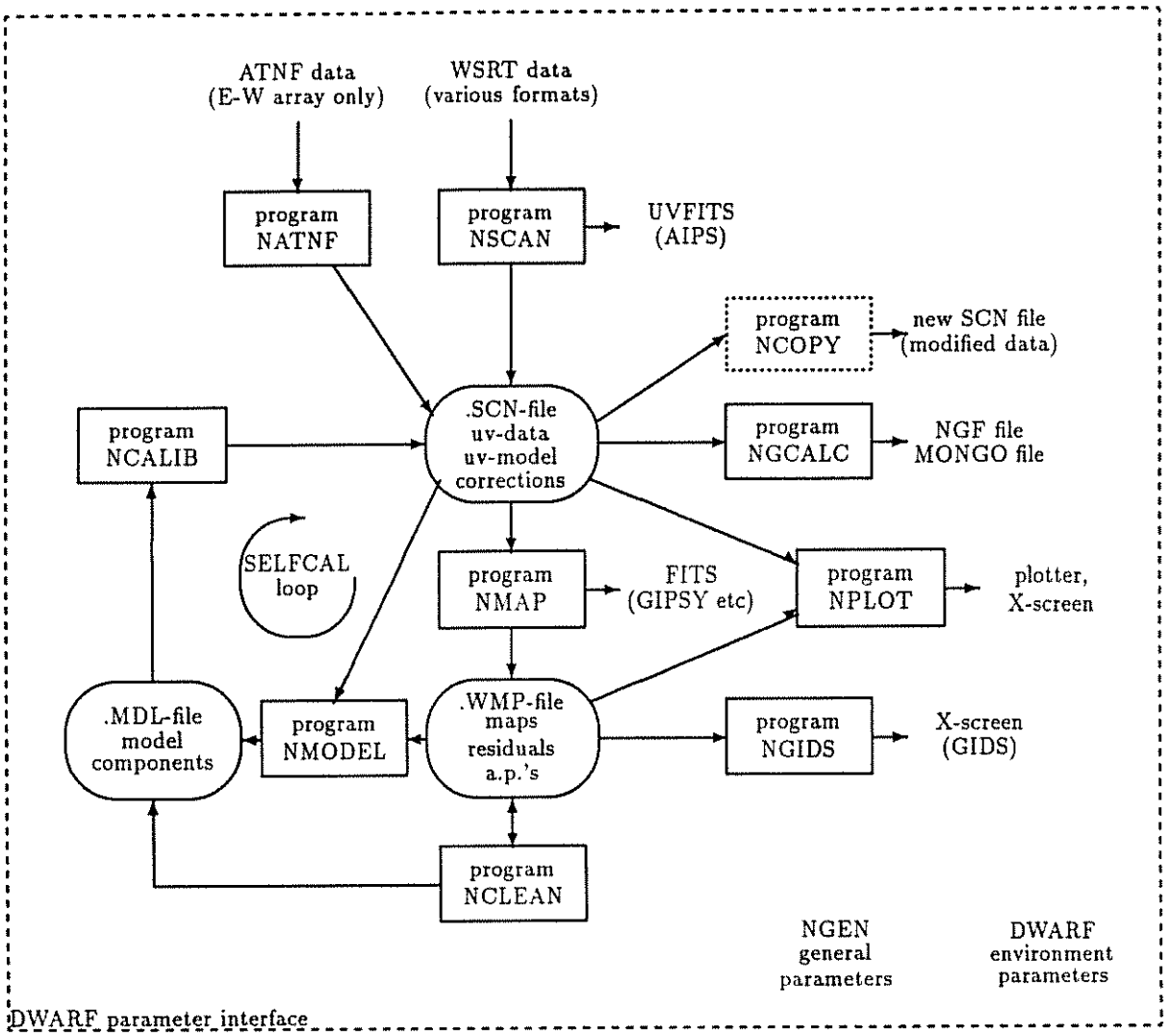


Figure 1: File: fig_gen1_INTRO_block.TEX

Block diagram of the NEWSTAR programs and data-files. The NEWSTAR package specialises in the processing of uv-data *from East-West arrays, in particular the WSRT. The heart of the package is the SELFCAL loop, although it has many other features. WSRT-style SELFCAL is different in two respects:*
- 1: It makes full use of the extra (model-independent!) constraint provided by the presence of redundant spacings in the WSRT array.
- 2: The source model contains multi-parameter source components, including extendedness, polarisation and spectral index. It can also contain CLEAN components.
The 3 types of data files (SCN, MDL and WMP) can be manipulated by means of the programs NSCAN, NMODEL and NMAP respectively. The program NGCALC can be used to extract a wide variety of (astrophysical) information from the data.

The NEWSTAR SCN-file
uv-data, corrections and uv-model

Editor: J.E.Noordam

February 10, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.newstar.cookbook]FILE1_SCN.tex

1 The SCN-file (uv-data)

The NEWSTAR SCN-file contains WSRT (or ATNF) uv-data and header information, including many types of corrections that may be applied (or de-applied) to the data. A single SCN file is a collection of uv-data samples, *that are related to each other in some way.* Examples are multi-channel data, mosaicking data (multiple pointing centres), data of the same object taken at different times, etc. Other examples are calibrator observations that were taken before and after the 'real' observation. This approach is convenient for processing, and for keeping related things together.

As shown in fig 1, the SCN-file may also contain a SELFCAL source model in two representations: as source components (from a MDL file), and Fourier transformed to the uv-coordinates of the data.

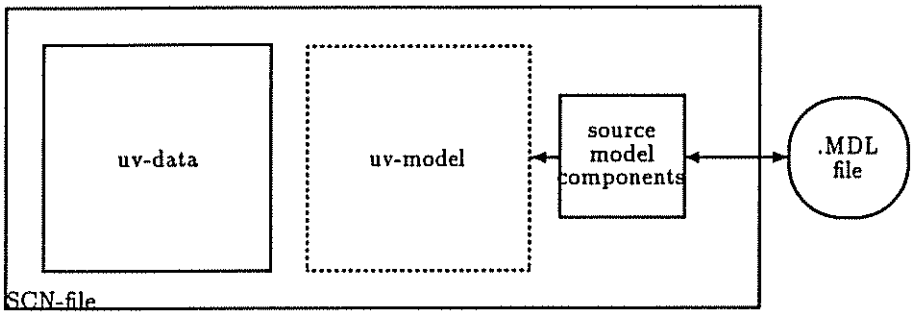


Figure 1: File: fig_file1_SCN_cont.TEX

The SCN-file contains uv-data and its corrections. It can also contain the uv-representation (i.e. mapped to the uv-coordinates of the uv-data) of a SELFCAL source model. The reason for this approach is that it is a very time-consuming process to transform a model of many many (> 100) source components to the uv-plane. If the uv-model is available, it may be used for various "casual" purposes (e.g. calculating SELFCAL residues, or subtraction from the uv-data) without a large time-penalty. It is also much quicker to recalculate the uv-model when only a few of the many source components have changed. Obviously, the component-representation (a copy of a .MDL file) of the stored uv-model must also be kept in the SCN-file.

1.1 Organisation of the SCN-file: Sets (segments)

Fig 2 gives a schematic overview of the organisation of the uv-data in the SCN file. The basic unit is the 'Set' (or segment). See also fig 3. A particular Set (segment) may be selected by means of 5 integer indices ('dimensions'):

1) job (j)	or group (e.g. observation or calibrator data)
2) label (l)	e.g. separate parts of the same observation
3) field (f)	pointing centre (mosaicking)
4) channel (c)	frequency point or DCB band
5) segment (s)	time-contiguous collection of HA-Scans

Example: `0.1.0.31.5` selects the 6th segment for frequency channel 31, for the 1st pointing centre, of the 2nd label of the 1st job.
The 'job', 'label' and 'segment' indices are running numbers **and start at 0!**.
The 'field' nr corresponds with the WSRT mosaicking field nr. For a single pointing centre, this index is 0.
The 'channel' index corresponds with the WSRT channel nr, with the 'continuum point' as channel 0.
The indices may also be specified as ranges or wildcards (see the section Overview of NEWSTAR files).

It is possible to quarrel about the 'best' order of the various indices. In particular, one might argue that it would be better to order the data in the way they have been acquired, i.e. in time-slices. This particular order one has been chosen to minimise the repetition of header information, and to optimise the sorting for particular projects.

A Set (segment) also has three 'dimensions' of *internal structure*, which may be specified by means of separate selectors, which are valid for all Sets:

6) HA-range	Scans
7) Polarisation	XX,XY,YX,YY
8) Interferometers	selected by name

A HA-Scan contains a header with information and corrections (see below), and the actual uv-data. The latter are stored as 'triplets' of 16-bit numbers: cosine, sine, weight. Scans may be reversibly 'deleted' (ignored) by setting delete-bits (in the Scan header). Similarly, individual data samples can be 'deleted' or by making the weight of a triplet negative (see program NSCAN, option DELETE).

NB: The user should be aware of an annoying ambiguity of terms here. Sets are also also sometimes called 'segment' or 'spoke'. Thus, the SCN-file layout refers to segments, but the header of such a segment is called a 'set header'.

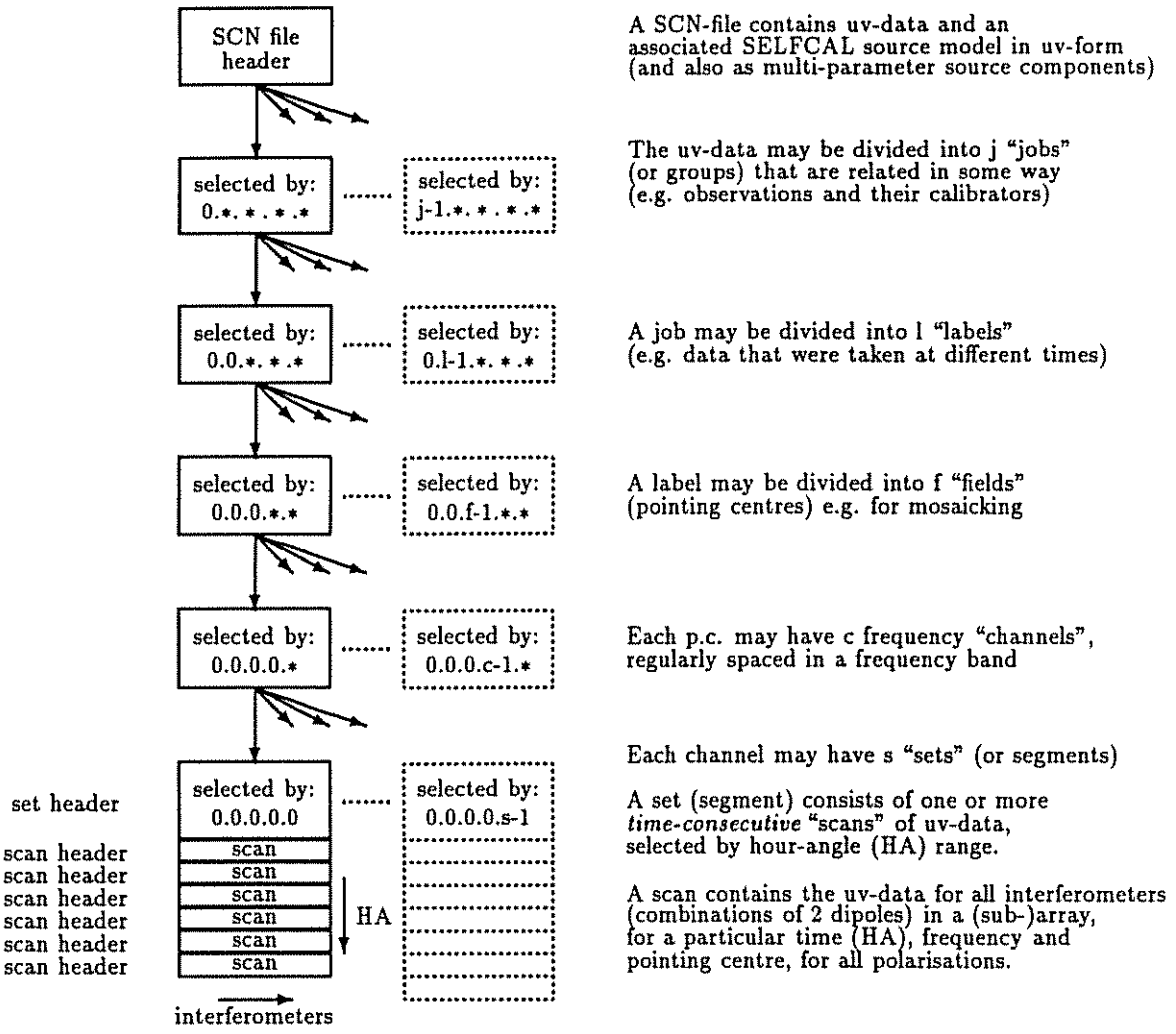


Figure 2: File: fig_file1_SCN_sets.TEX

Schematic overview of the organisation of the uv-data in the SCN-file. The basic unit is the Set (or segment), which can be selected by means of five integer parameters, divided by dots (j.l.f.c.s). Each starts at zero (!), and can be a wildcard (, meaning all) or a specific range with an increment (e.g. 3-15:3, or 4-). The uv-data within a Set can be selected by HA range, polarisation (XX,XY,YX,YY) and interferometers (9A, 8*, -FF etc).*
There are headers with information at three levels:
- The **file header** *describes controls access to the various parts, but contains no astronomical information.*
- Each **set header** *contains information about the source, and observational parameters like pointing centre, frequency etc. It also contains polarisation corrections.*
- Each **scan header** *contains information about HA etc. It also contains various phase and gain corrections per telescope.*

1.2 The file layout and file header

A summary of the SCN-file contents and layout may be obtained by using the program NSCAN:

```
     NSCAN$1 is started at 11-JUN-92 14:30:03
⊗ OPTION (LOAD,DUMP,FROM_OLD,TO_OLD,SHOW,DELET...) = QUIT:  show
⊗ INPUT_SCAN (input node name) = "":  u6446.240
File description of node U6446.240:
Created: 05-Jun-1992 17:39  Revision(3): 07-Jun-1992 17:37
File contains 96 datasets in 3 groups and has version 1
⊗ FILE_ACTION (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT:  layout
File layout:
0.0         contains    1 fields,   32 channels and    1 segments for 3C147
1.0         contains    1 fields,   32 channels and    1 segments for U6446
2.0         contains    1 fields,   32 channels and    1 segments for 3C286
```

This file contains three Jobs (0,1,2), for an observation of the source U6446 and calibration observations of 3C147 and 3C286. Each Job consists of a single Sub-job (.0), one pointing centre (field), 32 frequency channels, and one Segment (or Spoke, or Set) of contiguous Scans. The continuum channel (0) for 3C286 would be selected by `2.0.0.0.0` (note the start of the indices at zero!), or `2.*.*.0.*`.

The contents of the file header may be inspected also. It only contains information needed by the program to find its way in the file in an efficient manner (for a description of the file structure, see NFRA ITR197abcd by W.N.Brouw):

```
⊗ FILE_ACTION (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT:  show
File description of node U6446.240:
ID              .SCN      LEN             512     VER              1
CDAT      05-Jun-1992     CTIM           17:39     RDAT     11-Jun-1992
RTIM            15:40     RCNT               9     NAME       U6446.240
DATTP               3
LINK         00005DF0   00AD3740
NLINK              96
LINKG        00000200   009ABEE0
NLINKG              3    IDMDL               0     ID1              0
ID2                 0    USER                0
```

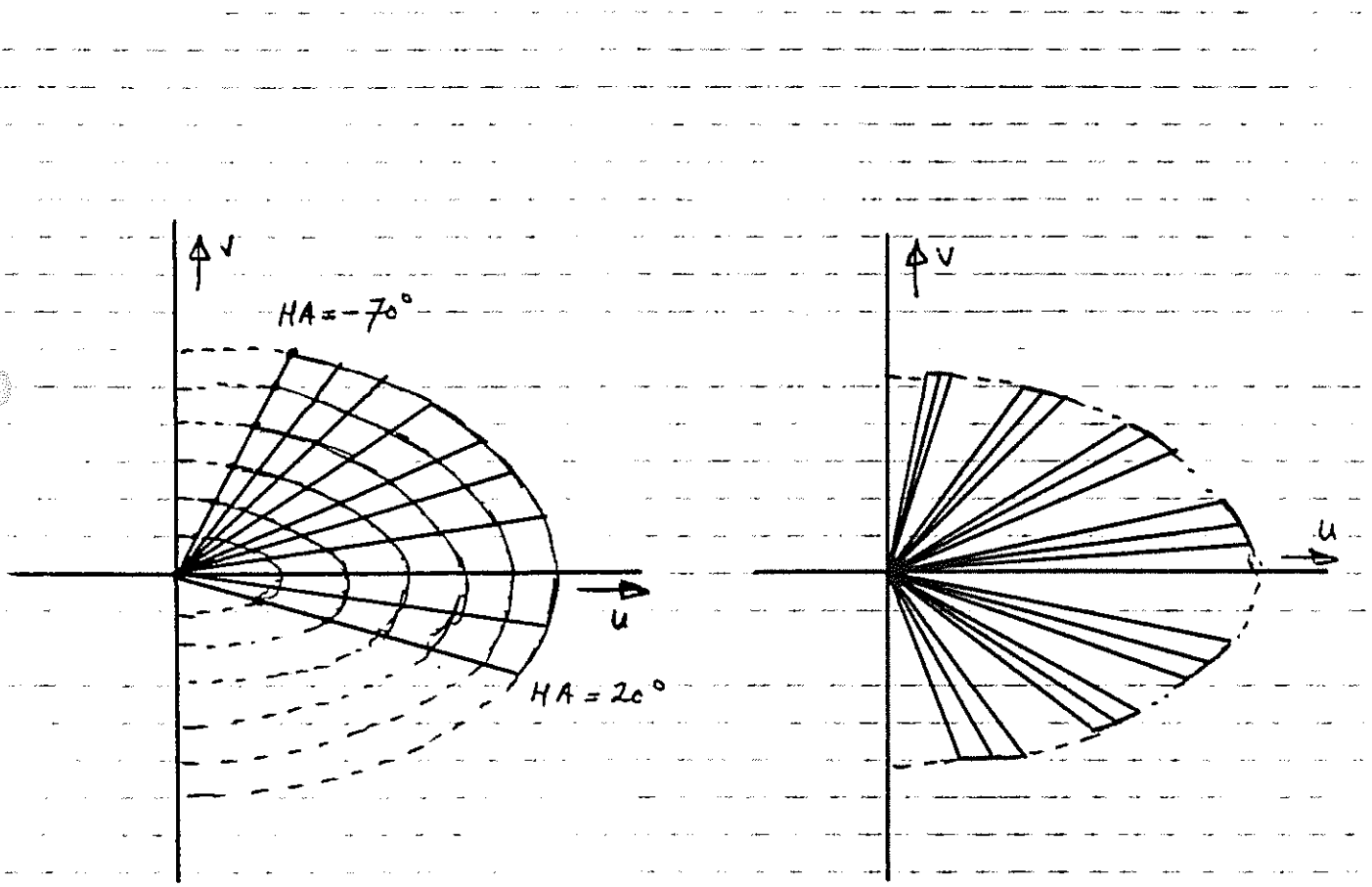


Figure 3: fig-file1-SCN-segment

A Set (segment) is a **time-contiguous** *collection of HA-scans, for one frequency channel and one pointing centre. Usually, the segment index will be 0, since there is only one segment. The single segment in the left figure contains 10 contiguous HA-scans, for one pointing centre and one frequency channel.*
The most common case of multiple segments is a mosaicking observation (right), where each pointing centre is observed for a few consecutive HA-intervals (scans) at a time, but revisited several times in the course of 12 hours. The figure on the right shows 7 segments of 3 contiguous HA-scans each, for one pointing centre and one frequency channel.

1.3 The Set (segment) header

When adressing a particular Set with the program NSCAN, a summary of the most important information in the Set header is dispayed:

⊗ **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT: <CR>
⊗ **SETS** (sets to do) = "": 0

```
 Set 0.0.0.0.0(#0) - 3C147 - Channel 0 - 23 Scans - 2 polarisations
RA (date)     85.4827 deg    HA(start)    -28.72 deg    Obs.day         104
DEC(date)     49.8517 deg    HA(end)       -6.66 deg    Obs.year         91
RA (1950)     84.6813 deg    HA(step)       1.00 deg    Epoch          1991.28
DEC(1950)     49.8286 deg    HA(average)    1.00 deg    Volgnummer 9101118
Frequency   1417.2484 MHz    # of ifrs     40           Backend          2
Bandwidth      2.3250 MHz    Prec. rot.     0.36 deg    Pointing Set     0
                                                        MJD(start)   48360.57755
Telescope positions 9, A, B, C, D = 1296, 1332, 1404, 2628, 2700
REDUN M.E.       0.0,      0.0,      0.0,      0.0
ALIGN M.E.       0.0,      0.0,      0.0,      0.0
```

The full information in the Set header may be inspected too:

⊗ **SET_ACTION** (NEXT,IFRS,SHOW,EDIT,CONT,QUIT) = CONT: show

```
Set header description 0.0.0.0.0:
LINK          0000F960      00000098
LEN            640          VER              2      SETN             0
BEC              2          PTS              0      VNR        9101118
CHAN             0          PLN              2      FIELD        3C147
RA         85.4826541        DEC       49.8517175    RAE       84.6812750
DECE       49.8285640        HAB      -28.7215206    HAI        1.0027379
SCN             23           OEP         1991.28     EPO         1950.0
FRQ      1417.248413          FRQE     1417.248413    BAND       2.325000
HAV         1.0027379
OBS            104            91
RTP            0.0000       143.9919     287.9837     431.9756    575.9674
               719.9592     863.9511    1007.9429    1151.9348   1295.9266
               1331.9297    1403.9216   2627.8513    2699.8472
NIFR             40          IFRP         00005D00     NFD            1920
FDP          000002A0        NOH             1536     OHP         00000A20
NSC            15360         SCP          00001020     NSH            2080
SHP          00005490        SCNP         000060C0     SCNL           1504
REDNS          0.0000        0.0000       0.0000       0.0000
ALGNS          0.0000        0.0000       0.0000       0.0000
OTHNS          0.0000        0.0000       0.0000       0.0000
MDL          00000000        00000000
MDD          00000000        00000000
PHI              0.4
POLC           0.0000        0.0000       0.0000       0.0000      0.0000
               0.0000        0.0000       0.0000       0.0000      0.0000
                                          Etc, Polarisation corrections: 56 4-byte numbers
                                          Position angle error and ellipticity for 28 dipoles
FRQO     1417.248413         FRQV     1417.248413      FRQC     1417.248413
VEL            0.0000        VELC              0       MJD       48360.57755
UTST          1.00274
```

1.3.1 Explanation of items in the Set header

- LINK: Link (pointer) to other sets
- LEN: Length of the header (bytes?)
- VER: Version nr of the header
- SETN: absolute Set nr, the one needed for direct reference (e.g. #setn)
- BEC: WSRT backend configuration nr
- PTS: Pointing centre nr (mosaicking)
- VNR: Obs nr (='volgnummer'+'cyclus'*66536) (WSRT code)
- CHAN: Frequency channel nr (0=continuum)
- PLN: Nr of polarisations (1 (XX,XY,YX or YY), 2 (XX or YY), 4)
- FIELD: Name of observed object/field
- RA: Obs RA (date) of field centre (degr)
- DEC: Obs DEC (date)of field centre (degr)
- RAE: Epoch RA(1950)
- DECE: Epoch DEC(1950)
- HAB: Start HA apparent (degr)
- HAI: HA increment between Scan centres (degr)
- SCN: nr of Scans in this set
- OEP: Obs epoch (e.g. 1990.12)
- EPO: Epoche (e.g. 1950.0)
- FRQ: Apparant central frequency (MHz)
- FRQE: LSR frequency
- BAND: Bandwidth (MHz)
- HAV: Averaging HA (degr)
- OBS: Observation day (e.g. 104) and year (e.g. 91)
- RTP: Telescope positions (m), starting at RT0
- NIFR: Nr of interferometers (per polarisation)
- IFRP: pointer to interferometer list
- NFD: FD block (WSRT tape format)
- FDP: Pointer to FD block
- NOH: Length of OH-block (WSRT tape format)
- OHP: Pointer to OH block
- NSC: SC block (WSRT tape format)
- SCP: Pointer to SC block
- NSH: SH block (WSRT tape format)
- SHP: Pointer to SH block
- SCNP: Pointer to Scan area
- SCNL: Length of Scan (bytes?)
- REDNS: Redundancy solution noise (gain/phase, X/Y)
- ALGNS: Align solution noise
- OTHNS: Noise of 'other' solution
- MDL: Pointer to model lists (source components)
- MDD: Pointer to model data (uv-representation)
- PHI: Precession rotation angle (degr)
- POLC: Polarisation corrections (posangle/ellipt, tels, X/Y)
- FRQ0: Rest frequency for line (MHz)
- FRQV: Real frequency for line (MHz)
- FRQC: Centre frequency for line (MHz)
- VEL: Velocity for line (m/s)
- VELC: Velocity code (0=cont, 1=heliocentric radio, 2=LSR radio, 3=heliocentric optical, 4=LSR optical)
- MJD: Start Julian Day
- UTST: Conversion UT/ST day length
- INST: Instrument (0=WSRT, 1=ATCA)
- VELR: Velocity at rest frequency (FRQC)

1.4 The Scan header

The Scan contains the actual data, and its header contains all the information that is relevant to that particular data. The Scan header can be inspected by NSCAN option SHOW:

```
      NSCAN$1 is started at 11-JUN-92 15:43:02
 ⊗ OPTION (LOAD,DUMP,FROM_OLD,TO_OLD,SHOW,DELET...) = QUIT:  show
 ⊗ INPUT_SCAN (input node name) = "":  u6446.240                    SCN-file
 ⊗ FILE_ACTION (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT:  <CR>           Next level
 ⊗ SETS (sets to do) = "":  2.0.0.16                                Job 2, channel 16
 ⊗ SET_ACTION (NEXT,IFRS,SHOW,EDIT,CONT,QUIT) = CONT:  <CR>         Next level
 ⊗ SCAN_ACTION (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,E,Q) = ">":  60.25  HA (degr)
HA          60.2972 deg    Maximum      3807.00 W.U.   Bits       00000000
Extinction   1.00000       Refraction      1.00000     Faraday       0.0 deg
Red. noise:    00.00,      00.00,      00.00,      00.00
Align noise:   15.93,      21.67,      16.43,      20.27
```

This produces a minimal selection of the information in the Scan header. The full Scan header may be inspected by:

```
 ⊗ SCAN_ACTION (XX,XY,YX,YY,ha,> [n],< [n],S,D,A,W...) = "> ":  s     show
Scan header description
HA          60.2971959      MAX          3807.000      SCAL          0.000000
REDNS           0.000        0.000         0.000         0.000
ALGNS          15.927       21.665        16.425        20.271
OTHNS           0.000        0.000         0.000         0.000
BITS         00000000
DOB              0000         0000
DEGEN               0            0             0             0
EXT           0.00000      REFR          0.00000       FARAD            0.00
REDC            0.0000       0.0000        0.0000        0.0000        0.0000
                0.0000       0.0000        0.0000        0.0000        0.0000
                                    Etc, Redundancy corrections: 56 4-byte numbers
                                                Gain and phase errors for 28 dipoles
ALGC            0.0443      -0.2488        0.0503       -0.1536        0.0235
                0.0854       0.0080       -0.1001        0.0373       -0.2338
                                    Etc, Align corrections: 56 4-byte numbers
                                                Gain and phase errors for 28 dipoles
OTHC            0.0000       0.0000        0.0000        0.0000        0.0000
                0.0000       0.0000        0.0000        0.0000        0.0000
                                    Etc, Other corrections: 56 4-byte numbers
                                                Gain and phase errors for 28 dipoles
IFRAC        00000000      IFRMC        00000000       AEXT          0.00000
AREFR         0.00000      AFARAD          0.00
AOTHC           0.0000       0.0000        0.0000        0.0000        0.0000
                0.0000       0.0000        0.0000        0.0000        0.0000
                                    Etc, Other applied corrections: 56 4-byte numbers
                                                Gain and phase errors for 28 dipoles
AIFRAC       00000000      AIFRMC       00000000
```

```
 ⊗ SCAN_ACTION (XX,XY,YX,YY,ha,> [n],< [n],S,D,A,W...) = "> ":  <CR>   Next Scan
```

Other Scan headers may be selected by HA, or by skipping n scans forward (e.g. > 44) or backward (e.g. < 72). The default is > 1. Each time, the summary of the Scan header is shown first.

1.4.1 Explanation of items in the Scan header

- HA: Apparant hour angle of this Scan (degr)
- MAX: Maximum cos or sin in this Scan (W.U.)
- SCAL: Cos/sin scale multiplier minus 1
- REDNS: Redundancy solution noise (W.U., gain/phase, X/Y)
- ALGNS: Align solution noise (W.U., gain/phase, X/Y)
- REDNS: 'Other' solution noise (W.U., gain/phase, X/Y)
- BITS: 32 general bits (BITS=00000001: Scan 'deleted')
- DOB: 32 do/undo bits
- DEGEN: Degeneracy count (gain/phase, X/Y)

The following are corrections which can be selectively **applied** (COMMON keyword APPLY) to the uv-data whenever they are read into memory:
- EXT: extinction factor minus 1 (so EXT=0 means: no extinction)
- REFR: refraction factor minus 1 (so REFR=0 means: no refraction)
- FARAD: Faraday rotation angle (radians)
- REDC: Redundancy dipole corr (log(gain)/phase, X/Y)
- ALGC: Align dipole corr (log(gain)/phase, X/Y)
- OTHC: 'Other' dipole corr (log(gain)/phase, X/Y)
- IFRAC: Pointer to additive interferometer corrections
- IFRMC: Pointer to multiplicative interferometer corrections

NB: Dipole corrections are also called 'telescope corrections', and are in units of log(gain) and radians.

The following are corrections that have already been applied to the data, and can be selectively de-applied (COMMON keyword DE_APPLY) if required:
- AEXT: Applied extinction factor minus 1
- AREFR: Applied refraction (MU-1)
- AFARAD: Applied Faraday rotation angle (radians)
- AOTHC: Applied total (REDC+ALGC+OTHC) dipole corr (log(gain)/phase, X/Y)
- IFRAC: Pointer to applied additive interferometer corrections
- IFRMC: Pointer to applied multiplicative interferometer corrections

NB: (gain/phase, X/Y) means: (Xgain, Xphase, Ygain, Yphase)

1.5 The Scan data

The actual uv-data in a particular Scan can be inspected with NSCAN option SHOW:

⊗ **SET_ACTION** (NEXT,IFRS,SHOW,EDIT,CONT,QUIT) = CONT: [<CR>] *Next level*
⊗ **SCAN_ACTION** (XX,XY,YX,YY,ba,> [n],< [n],S,D,A,W...) = "> ": [a] *ampl/phase*

Amplitude (1 W.U.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	.
	.	.	.	.	.	.	.	.	.	.	4854	5691	4461	4352	0
0	.	.	.	.	.	.	.	.	.	.	4805	5643	4423	4289	1
1	.	.	.	.	.	.	.	.	.	.	4786	5619	4407	4279	2
2	.	.	.	.	.	.	.	.	.	.	4477	5474	4317	4211	3
3	.	.	.	.	.	.	.	.	.	.	4765	5560	4363	4245	4
4	.	.	.	.	.	.	.	.	.	.	4719	5575	4370	4240	5
5	.	.	.	.	.	.	.	.	.	.	4823	5634	4421	4306	6
6	.	.	.	.	.	.	.	.	.	.	4956	5605	4393	4290	7
7	.	.	.	.	.	.	.	.	.	.	4684	5500	4340	4174	8
8	.	.	.	.	.	.	.	.	.	.	4676	5512	4319	4193	9
9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A
A	-12	-17	-4	-8	-20	-3	-3	-8	-7	-22	.	.	.	.	B
B	-9	-14	-1	-7	-16	0	0	-10	-4	-19	.	.	.	.	C
C	4	-1	12	6	-3	13	13	3	9	-7	.	.	.	.	D
D	1	-4	9	3	-6	10	11	0	6	-9	.	.	.	.	.
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	.

Phase (deg)

Each NEWSTAR visibility sample is stored as three 16-bit numbers: Cosine, Sine and Weight. The weights depend on the receiver noise temperature (cooled receivers have higher weights) and the integration time. A **negative** value means that the data sample has been reversibly 'deleted' (use NSCAN option DELETE). The weights can be inspected:

⊗ **SCAN_ACTION** (XX,XY,YX,YY,ba,> [n],< [n],S,D,A,W...) = "> ": [W] *weights*

Data weight (*1)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	.
.	.	.	.	.	.	.	.	.	.	-24	24	24	24	0
	.	.	.	.	.	.	.	.	.	24	24	24	24	1
		.	.	.	.	.	.	.	.	24	24	24	24	2
			.	.	.	.	.	.	.	24	24	24	24	3
				.	.	.	.	.	.	24	24	24	24	4
					.	.	.	.	.	24	24	24	24	5
						.	.	.	.	24	24	24	24	6
							.	.	.	24	24	24	24	7
								.	.	24	24	24	24	8
									.	24	24	24	24	9
										.	.	.	.	A
											.	.	.	B
												.	.	C
													.	D

The NEWSTAR MDL-file source model components

Editor: J.E.Noordam

January 11, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aseries.cookbook]file1_MDL.tex

1 The MDL-file (model components)

A NEWSTAR MDL file contains a collection of source model components. This may be a mixture of multi-parameter components and ordinary CLEAN components. The former are much more versatile, and represent one of the main differences between NEWSTAR and other uv-data reduction packages.

1.1 Example

The contents of a MDL file may be inspected in the following manner:

```
      NMODEL$1 is started at  8-DEC-92 22:20:59
 ⊗ ACTION (HANDLE,HELP,FIND,UPDATE,XUPDATE,PRO...) = HANDLE:  handle
 ⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST,...) A,QUIT)  read
 ⊗ MODEL_NODE (Model node) = "":  model
 ⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST,...) A,QUIT)  show
 ⊗ SOURCE_RANGE (Source number range) = *:  <CR>
```

```
  #       I          l          m        ID      Q     U     V      long  short   PA
         W.U.     arcsec     arcsec                 %     %     %    arcsec arcsec deg

Source list in local mode

    1  1000.000     0.00      0.00     1-00    0.0   0.0   0.00    0.00   0.00    0
    2   100.000    10.00    -10.00     2-00    0.0   0.0   0.00    0.00   0.00    0
    3    10.000  -100.00     100.00     3-00    0.0   0.0   0.00    0.00   0.00    0

3 sources (0 deleted) with 1110.000 W.U. (Max= 1000.000, Min= 10.000)
```

The above model is in 'local mode', which means that no reference position or reference frequency have been supplied (use NMODEL, option CONVERT, see also below). This information is stored with the model in the MDL file. If the position of the field centre is known, the model looks like this:

```
  #       I          RA           DEC          ID      Q     U     V      long  sho
         W.U.                                          %     %     %    arcsec arc

Sources at epoch 1950 at 05:38:43.51, 49.49.42.8, 1401.375 MHz

    1  1000.000  05:38:43.51  49.49.42.83     1-00    0.0   0.0   0.00    0.00   0.
    2   100.000  05:38:42.47  49.49.32.83     2-00    0.0   0.0   0.00    0.00   0.
    3    10.000  05:38:53.85  49.51.22.76     3-00    0.0   0.0   0.00    0.00   0.

3 sources (0 deleted) with 1110.000 W.U. (Max= 1000.000, Min= 10.000)
```

Some source parameters are also affected when the reference frequency is known.

1.2 Source component parameters

The source components in a NEWSTAR model have the following parameters:

- **I:** total flux in W.U. (even when extended or polarised)
- **l, m:** offset from the reference point (map centre) in arcsec
- **ID:** identification number
- **Q, U, V:** Stokes Q, U, V in percent (of I)
- **long, short, PA:** long and short axes (full halfwidth in arcsec) of an elliptic gaussian extended source, and the position angle of the long axis (degrees North thru East).
- **si:** spectral index ($flux \div frequ^{si}$)
- **rm:** Faraday rotation measure ($rot = rm \times (c/frequ^2)$ *in* rad/m^2)
- **flags:** (e.g. if a CLEAN component or a proper source (0))

1.3 Coordinate system

Model nodes come in three **flavours**, related to their sky coordinate system:
- **local:** unknown central position and frequency (these can be obtained from the relevant SCN file, using NMODEL 'action' CONVERT)
- **apparent:** the position of the field centre (and the parameters l, m) are in apparent coordinates, and the frequency is known (i.e. stored in the MDL file)
- **B1950:** position of the field centre (and l, m) are in B1950 coordinates, and the frequency is known

Models in MDL files can be converted between flavours (using NMODEL action CONVERT), in which case the l, m offsets, the spectral index and the rotation measure are changed.

1.4 Proper sources and CLEAN components

CLEAN components:

Are not corrected for beam smearing in SELFCAL.

Are confined to map grid postions, which may result in poor subtraction.

Their positions cannot be "updated" automatically after SELFCAL.

Use "proper sources" for strong point sources, and CLEAN components for weak extended sources (only if necessary).

1.5 Overview of model 'handling' options.

In various NEWSTAR programs (NMODEL, NCALIB, NMAP etc) the user is able to handle the list of model components in various ways. The keyword often used is MODEL_OPTION:

- **READ/WRITE:** Read/write from/to an external MDL file
- **CLEAR:** Clear the source component list, while resetting reference coordinates
- **ZERO:** Empty the source component list, but keep the coordinates of the field centre and frequency.
- **SHOW/LIST:** Show source list on terminal screen, or both terminal and LOG-file.
- **RSHOW/RLIST:** Show source list in RA/DEC coordinates
- **TOT:** Show source list statistics
- **ADD:** Add sources to the list by hand.
- **CALIB:** Convert the source list by scaling intensities and/or moving l,m positions.
- **EDIT:** Edit the sources in the list (an amplitude of zero will delete the source)
- **FEDIT:** Edit a 'field' (parameter) for a range of sources
- **MERGE:** Combine sources components that have the same position
- **SORT:** Sort the source list in decreasing amplitude (sorting will always precede a write)
- **FSORT:** Sort on a specified 'field' (parameter) in the source list
- **DEL:** Delete sources
- **DNCLOW:** Delete non-CLEAN components with low amplitudes
- **DCLOW:** Delete CLEAN components with low amplitudes
- **DAREA:** Delete sources in specified area

1.6 Conversion to a uv-model for the .SCN file

The calculation of uv-model data (for comparison with uv-data in NCALIB)i s time consuming. Therefore, a calculated uv-model is saved in the .SCN file for which it is calculated, together with the source list used. In all programs that use the model data it can be specified how the model calculation should be done, and if the calculation should be saved. The relevant question is MODEL_ACTION, which expects a list of three answers. The first one can be one of:

- **merge:** replace the model saved in the SCN node with the one specified by the user. However, first compare these two lists, and only add the difference to the saved model data. I.e. make the calculation as short as possible if the new list differs only slightly from the saved one.
- **add:** replace the saved model with the sum of the saved one and the one specified by the user. The model calculated on the basis of the user specified list is added to the saved data.
- **new:** replace the saved model by the model specified by the user, and calculate a completely new set of model data
- **temporary:** do not use any data in the SCN node, or write anything, but use the data based on the list specified by the user
- **increment:** use the saved model data, and add to it the model data based on the user specified list, but do not save anything

The second answer can be BAND or NOBAND, and specifies if in the model calculation source data should be corrected for bandsmearing to match the actual data better. The third answer can be TIME or NOTIME to indicate the use of integration time smearing.

The NEWSTAR WMP-file
image data (maps, antenna patterns, etc)

Editor: J.E.Noordam

February 4, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]file1_WMP.tex

1 The WMP-file (image-data)

The NEWSTAR WMP file contains 'image data', i.e. a collection of 2-dimensional arrays of data. The various 'images' in a WMP file *are related in some way*, but do not have to have the same dimensions. Examples are radio maps at various frequencies (line observations), polarisations, or pointing directions (mosaicking). There may also be antenna patterns and various kinds of residual maps, or even rectangular arrays ov uv-data.

1.1 Organisation of the WMP file: Maps

The basic unit in the WMP file is the Map (a 2-dimensional array of pixel values). It can be selected by the user by means of 6 integer indices:

1) group	spare index
2) field	pointing centre (mosaicing)
3) channel	frequency channel, or DCB band
4) polarisation	0=I or XX, 1=Q or XY, 2=U or YX, 3=V or YY
5) type	0=map, 1=ap, 2=cov, 3=real, 4=imag, 5=ampl, 6=phas
6) nseq	sequence nr within each type

Indices may also be ranges of indices, or wildcards (*), as explained in more detail in the section 'Overview of NEWSTAR files' in this Cookbook. Note that the Map in the WMP-file plays the same role as the Set in the SCN-file.

All indices are just running numbers (**starting at 0!**), except 'type' and 'polarisation', which have fixed codes (see above).

Neither the 'field' nor the 'channel' index nrs correspond with the 'field' or 'channel' nrs in the SCN-file. The reason for this is that a map may be made from a *combination* of fields or channels.

Usually, all Maps belong to the same 'group' (0). Therefore, the first index is called a 'spare index' here. However, any selection of Maps may be put into a new group in the same WMP file, using the NMAP option REGROUP.

The 6th index allows for a sequence of Maps of a certain type, usually derived from each other. Examples are residue Maps after CLEANing (see NCLEAN), or the Maps that result from combining other Maps (see NMAP option FIDDLE).

1.2 File layout and file header

A summary of the WMP-file contents and layout may be obtained by using the program NMAP, option SHOW:

```
>  exe nmap
    NMAP$1 is started at 03-FEB-93 13:38:36
⊗ OPTION (MAKE,SHOW,FIDDLE,W16FITS,W32FITS,WRLFITS,FROM_OLD, ...) = QUIT:  show
⊗ INPUT_MAP (node name) = "":  test                    short for TEST.WMP
File description of node TEST:

Created: 03-Feb-1993 13:31  Revision(0): 03-Feb-1993 13:31
File contains 10 datasets in 1 groups and has version 1
⊗ FILE_ACTION (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT:  layout
File layout:

0     contains   1 fields,   1 channels,   4 pol.s and    7 types for 3C147
```

This particular WMP-file actually contains the following (rather strange) collection of Maps:

```
g.f.c.p.t.n (#abs)                          group.field.chan.pol.type.nseq
0.0.0.0.2.0(#0) type COVE                     uv-coverage for XY-map
0.0.0.0.6.0(#1) type PHAS                          map of XY phases
0.0.0.0.0.0(#2) type MAP                                  XY-map
0.0.0.0.1.0(#3) type AP                  antenna pattern for XY-map
0.0.0.1.6.0(#4) type PHAS                          map of Q phases
0.0.0.1.0.0(#5) type MAP                                   Q-map
0.0.0.2.6.0(#6) type PHAS                          map of V phases
0.0.0.2.0.0(#7) type MAP                                   V-map
0.0.0.3.6.0(#8) type PHAS                         map of iV phases
0.0.0.3.0.0(#9) type MAP                                  iV-map
```

The 10 'datasets' (Maps) in this WMP-file belong to 1 'group' (1st index, =0). The number in parenthesis (#8) indicates the *absolute Map nr* within the file.

Note that the nr of polarisations is not really 4, and that the nr of types is not really 7. Indicated are the highest index values present, plus one.

The WMP file header only contains book-keeping information that allows the program to find its way around:

```
⊗ FILE_ACTION (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT:  show
File description of node TEST:

ID              .WMP      LEN              512      VER              1
CDAT      03-Feb-1993     CTIM            13:31      RDAT      03-Feb-1993
RTIM            13:31     RCNT                0     NAME            TEST
DATTP               7
LINK         00000340     001297c0
NLINK              10
LINKG        00000200     00000200
NLINKG              1      IDMDL              0      ID1              0
ID2                 0      USER               0
```

The Map header

Each Map in a WMP file contains a header with information, which can be inspected with the program NMAP, option SHOW:

⊗ **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT: <CR>
⊗ **MAPS** (maps to do) = "": 0.0.0.3.6.0

```
0.0.0.3.6.0(#8) type PHAS in node

Field: 3C147                          User comment: 3C147
RA: 84.68127 deg  Dec: 49.82856 deg   Epoch: 1950.0  Frequency: 1397 MHz

RA (1950)     85.36266 deg                          Obs.day          362
Dec(1950)     49.84731 deg                          Obs.year          84
Frequency   1396.92749 MHz                          Epoch         1984.99
Bandwidth     15.01141 MHz                          Map epoch     1950.00

Type: PHAS(VI)                        Size: 112*72      FFT size: 256*256
Fieldsize: 0.2988*0.2988 deg          Grid step: 4.22*4.22 arcsec
                                      Fieldshift: 0.00*0.00 arcsec
Maximum: 0.50 W.U. at 37,18           Minimum: -0.50 W.U.at -28,-19

Input baselines: 200                  Input sets: 5     Input points: 140629
Normalisation: 88028.8                Noise: 0.000 W.U.

Gaussian taper; Expsinc convolution( corrected); Not clipped; No subtractions; 0
```

⊗ **MAP_ACTION** (NEXT,SHOW,EDIT,CONT,QUIT) = CONT: show

```
Map header description 0.0.0.3.6.0:

LINK          001297c0      000e12a0
LEN                512      VER                1      SETN               8
FNM              3C147      EPO           1950.0      RA          84.6812750
DEC         49.8285640      FRQ       1396.927494      BDW         15.011412
RAO         85.3626582      DECO       49.8473088      FRQO      1396.927494
ODY                362      OYR                84      DCD                5
PCD                  0      SRA         0.0011719      SDEC        0.0011719
SFRQ          0.000000      NRA                112      NDEC              72
NFRQ                 1      ZRA                 56      ZDEC               0
ZFRQ                 0      MXR                 37      MXD               18
MXF                  0      MNR                -28      MND              -19
MNF                  0      MAX              0.500      MIN           -0.500
SHR          0.0000000      SHD          0.0000000      SHF         0.000000
SUM       8028.8133006      UNI                0.0      UCM             3C147
NPT             140629      TYP               PHAS      POL                VI
CD                   1            4              1            0            0
                     1            0              0
EPT                  1      OEP            1984.99      NOS             0.000
FRA             0.2988      FDEC            0.2988      FFRQ            0.000
TEL               WSRT      FSR                256      FSD               256
MDP            1185600      NBL                200      NST                 5
VEL                  0      VELC                 0      VELR           0.0000
INST                 0      FRQ0          0.000000      FRQV      1396.927494
FRQC       1401.390625
```

1.3.1 Explanation of items in the Map header

- LINK: Link (pointer) to other Maps
- LEN: Length of header block (bytes?)
- VER: Version nr of the header
- SETN: Abs Map (Set) nr, i.e. the one needed for direct reference (e.g. #setn)
- FNM: Field (pointing centre) name
- EPO: Epoch (e.g. 1950.0)
- RA: RA of field centre (degr)
- DEC: DEC of field centre (degr)
- FRQ: Central frequency (MHz)
- BDW: Bandwidth (MHz)
- RAO, DECO, FRQO: Observed RA (degr), DEC (degr), freq (MHz)
- ODY, OYR: Observed day (since January 0th) and year (since 1900)
- DCD: Data code (2=I, 4=J, 5=E, 8=D)
- PCD: Program code (0=NMAP)
- SRA, SDEC, SFRQ: Separation in RA (degr), DEC (degr) and freq (MHz)
- NRA, NDEC, NFRQ: Nr of points in RA, DEC and frequ
- ZRA, ZDEC, ZFRQ: Centre RA (1st point=0), DEC (1st line=0), frequ (1st map=0)
- MXR, MXD, MXF: Position max in RA, DEC, frequ
- MNR, MND, MNF: Position min in RA, DEC, frequ
- MAX, MIN: Max, min map value
- SHR, SHD, SHF: Shift in RA, DEC (add, degr?) or frequ (add, MHz)
- SUM: Normalising sum
- UNI: Multiplier to get Jy
- UCM: User comment
- NPT: Nr of input uv-data points
- TYP: Map type (MAP, AP, COV, PHAS etc)
- POL: Polarisation type (I,Q,U,V or XX,XY,YX,YY)
- CD: Codes (array of 8 integer switches, 0-7): taper type (0), convolution type (1), correct for convolution (2), clipping done (3), source subtraction (4), data type (5), uv coordinate type (6), de-beam count (7)
- EPT: Map epoch used (0=apparent, 1=as specified in EPO above)
- OEP: Observation epoch (e.g. 1985.78)
- NOS: Map rms noise (W.U.)
- FRA, FDEC, FFRQ: Field size in RA (degr), DEC (degr), frequ (MHz)
- TEL: Telescope name (e.g. WSRT)
- FSR, FSD: FFT size RA, DEC
- MDP: Map data pointer
- NBL: Nr of baselines that have contributed to the Map
- NST: Nr of uv-data sets that have contributed to the Map
- VEL: Velocity (m/s)
- VELC: Velocity code (0=continuum, 1=heliocentric radio, 2=LSR radio, 3=heliocentric optical, 4=LSR optical)
- VELR: Velocity at reference frequ (FRQC)
- INST: Instrument code (0=WSRT, 1=ATCA)
- FRQ0, FRQV, FRQC: Rest, Real and Centre frequency for line (MHz)

1.4 The Map data

The actual data in a Map can be displayed on the X-screen as a color map by using the program NGIDS, or as a contour or gray-scale plot with the program NPLOT. However, its is also possible to inspect small areas of a WMP Map, or its statistics, with the program NMAP, option SHOW:

⊗ **MAP_ACTION** (NEXT,SHOW,EDIT,CONT,QUIT) = CONT: [<CR>]
⊗ **DATA_ACTION** (S,N,O,Q) = Q: [s] *show*
⊗ **AREA** (Area l,m,dl,dm) = "": [?]

```
Specify an area of a map:
l       l of centre of area (0 is centre, <0 in direction of RA)
m       m of centre of area (0 is centre, >0 in dircetion of DEC)
dl      width of area in grid points
dm      height of an area in grid points
```

⊗ **AREA** (Area l,m,dl,dm) = "": [-16,-15,8,10]

```
Area(s) selected:
Total   : l= -16, m= -15, dl= 8, dm= 10

   l      m
 -20    -20     -0.3    -0.5    -0.5    -0.4    -0.4     0.5     0.5    -0.5
 -20    -19     -0.5    -0.4    -0.5    -0.5    -0.3    -0.4     0.5     0.5
 -20    -18      0.5    -0.2    -0.5     0.5    -0.4    -0.3    -0.5     0.5
 -20    -17      0.5     0.5    -0.2    -0.5    -0.5    -0.4    -0.4     0.5
 -20    -16      0.5     0.5     0.4    -0.4    -0.5    -0.5    -0.3    -0.4
 -20    -15     -0.5     0.5     0.5    -0.1    -0.5     0.5    -0.5    -0.3
 -20    -14      0.5     0.5     0.5     0.5    -0.1    -0.5     0.5    -0.5
 -20    -13      0.5    -0.5     0.5     0.5     0.5    -0.1    -0.5     0.5
 -20    -12      0.4     0.5    -0.5     0.5     0.5     0.5    -0.1    -0.5
 -20    -11      0.4     0.5     0.5    -0.5     0.5     0.4     0.4    -0.1
```

⊗ **DATA_ACTION** (S,N,O,Q) = Q: [n] *noise*
⊗ **AREA** (Area l,m,dl,dm) = 0,0,112,72: [<CR>] *Full map*
⊗ **AREA** (Area l,m,dl,dm) = "": [<CR>] *Multiple areas may be specified*

```
Area(s) selected:
Total   : l= 0, m= 0, dl= 112, dm= 72

Noise= 0.0504 W.U.
```

⊗ **DATA_ACTION** (S,N,O,Q) = Q: [o] *offset*
⊗ **AREA** (Area l,m,dl,dm) = 0,0,112,72: [<CR>] *Full map*
⊗ **AREA** (Area l,m,dl,dm) = "": [<CR>] *Multiple areas may be specified*

```
Area(s) selected:
Total   : l= 0, m= 0, dl= 112, dm= 72

Noise= 0.0475 W.U., Offset= 0.0604 (0.0026) W.U.
```

Note that the 'noise' option gives the rms of the pixel values in the selected area(s), while the 'offset' option gives the rms with respect to their average value (offset). A histogram of pixel values is printed in the log-file (NMAP.LOG).

Chapter 4

PROGRAM DESCRIPTIONS

Summary of the common keywords (parameters) that are used in several NEWSTAR programs

Editor: J.E.Noordam

July 23, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file: user3 [noordam.nseries.cookbook]prog3_COMMON.tex

1 COMMON parameters (keywords)

The following description of the program parameters is also available as on-line HELP.

1.1 NGEN (general) NEWSTAR keywords

The following keywords are called by many NEWSTAR programs, although they will usually be hidden. Their default values may be inspected and/or modified by typing `dws NGEN` or `dws NGEN`.

- **LOG** (SPOOL,YES,NO,CATEN) (log-file action)
  For each run of a NEWSTAR program, logging information may be put into a LOG-file, for subsequent inspection (and clogging up your room). The LOG-file has the same name as the program, and the extension .LOG (e.g. NCALIB.LOG).

  The LOG-file contains the values for all the parameter (keywords) that were used for this particualr program run, including the 'hidden' ones. It will also contain information about the program run itself, and any results.

  Specify what to do with the logging output:
  - NO: Make no LOG-file (not recommended)
  - YES: Make a LOG-file (preferred option, minimises line-printer output)
  - SPOOL: Spool the LOG-file to the printer at the end of the program run
  - CATEN: Add the new info to the existing LOG-file for this program

  Whenever a new LOG-file is made, any existing LOG-file of the same name is renamed to a unique name, which starts with the same first three letters:
  CALIB.LOG will become NCA34862063.LOG.
  In this way, LOG-files of earlier program runs can still be inspected.

  This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
  Its value may be modified (for a given $stream) in several ways:
  - Permanently: change the default value: dws[pecify] NGEN[$stream]
  - Interactively: remove the /NOASK switch from the default value
  - Incidentally: run a NEWSTAR program with the switch /ASK

  Default value(s): YES /NOASK

- **RUN** (run mode)
  Specify the run mode of the NEWSTAR program:
  - YES: The NEWSTAR program will run normally
  - NO: The NEWSTAR program will prompt the user for all parameters (keywords) but will then terminate. This mode should be used in conjunction with the '/save' option, in which all the specified keyword values will be saved in a .SAV file for later use (e.g. batch processing).

  The /save option is invoked by starting the program in the following way:
  dwe 'program'[$'stream']/save (e.g. dwe ncalib$3/save).
  The saved keyword values are used by running the program again:
  dwe 'program'[$'stream']/noask (e.g. dwe ncalib$3/noask)

  NB: An alternative approach is to use the '/norun' option:
  dwe 'program'[$'stream']/norun (e.g. dwe ncalib$3/norun)

This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
Its value may be modified (for a given $stream) in several ways:
- Permanently: change the default value: dws[pecify] NGEN[$stream]
- Interactively: remove the /NOASK switch from the default value
- Incidentally: run a NEWSTAR program with the switch /ASK

Default value(s): YES /NOASK

- **DATAB** (default directory name)
Directory names can be very long. In order to avoid having to type them fully each time, the user may store it in the keyword DATAB (the name has historic roots). From then on, node names (in the same $stream) are automatically preceded by the contents of DATAB.

NB: In general, however, it is recommended to keep all files of a reduction project in a single directory. In that case, no explicit directory name will be needed in any case.

Example: if node name is preceded by a directory name:
nb/wsrt/data/myproject/mynode
the value of INFIX (for the current $stream) will be:
nb/wsrt/data/myproject/
Subsequently, typing 'othernode' will produce the node name:
nb/wsrt/data/myproject/othernode
NB: IF THE NEWSTAR PROGRAM CANNOT FIND A FILE (NODE), WHILE IT IS CLEARLY
PRESENT IN THE CURRENT DIRECTORY, CHECK THE VALUE OF DATAB (AND INFIX)!!

This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
Its value may be modified (for a given $stream) in several ways:
- Permanently: change the default value: dws[pecify] NGEN[$stream]
- Permanently: include a directory specification in the node name
- Interactively: remove the /NOASK switch from the default value
- Incidentally: run a NEWSTAR program with the switch /ASK

Default value(s): "" /NOASK

- **INFIX** (node name shorthand)
NEWSTAR 'node' names can be very long. In order to avoid having to type them fully each time, the user may store (part of) the node name in the keyword INFIX. From then on, the user may specify node names in a short-hand notation which contains a hash (#) character. The latter is automatically replaced by the contents of INFIX, to produce the full node name.

Example: if part of the full node name is enclosed in brackets:
mynode.21(cm.yesterday.s.x.c.d.file.d)d
the value of INFIX (for the current $stream) will be:
cm.yesterday.s.x.c.d.file.d
Subsequently, typing '92#e' will produce the node name:
mynode.92cm.yesterday.s.x.c.d.file.de
NB: IF THE NEWSTAR PROGRAM CANNOT FIND A FILE (NODE), WHILE IT IS CLEARLY
PRESENT IN THE CURRENT DIRECTORY, CHECK THE VALUE OF INFIX (AND DATAB)!!

This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is

'hidden', i.e. the user is not prompted for it, but the default value is used.
Its value may be modified (for a given $stream) in several ways:
- Permanently: change the default value: dws[pecify] NGEN[$stream]
- Permanently: by enclosing part of the node name in brackets (...)
- Interactively: remove the /NOASK switch from the default value
- Incidentally: run a NEWSTAR program with the switch /ASK

Default value(s): "" /NOASK

- **APPLY**

The uv-data in a SCN-file are NEVER physically modified. Correction factors are stored in the Sector headers and Scan headers of the SCN-file, and may be inspected with NSCAN option SHOW. These corrections may be applied 'on-the-fly' to the uv-data whenever they are read into memory to be processed. The keyword APPLY allows the user to specify which corrections are to be applied (or not) in this way:
- ALL or *: apply all available corrections
- NONE or "": apply no corrections
- [NO]RED: redundancy corrections (telescope gain,phase)
- [NO]ALG: align corrections (telescope gain,phase)
- [NO]OTH: other telescope gain/phase corrections (i.e. from calibrator)
- [NO]EXT: extinction correction (elevation-dependent gain)
- [NO]REF: refraction correction (elevation-dependent phase)
- [NO]IREF: ionospheric refracion
- [NO]CLK: clock correction
- [NO]POL: polarisation corrections (dipole angle, ellipticity)
- [NO]FAR: Faraday rotation
- [NO]IFR: additve interferometer errors (e.g. DC offset)
- [NO]MIFR: multiplicative interferometer errors

Multiple corrections may be specified, separated by comma's:
Example: NONE,RED,POL,EXT
Example: ALL,NORED,NOPOL,NOEXT

NB: If a certain correction is to be dis-abled permanently, it is better to set it to zero in the SCN-file header(s), using NCALIB option SET ZERO.

This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
Its value may be modified (for a given $stream) in several ways:
- Permanently: change the default value: dws[pecify] NGEN[$stream]
- Interactively: remove the /NOASK switch from the default value
- Incidentally: run a NEWSTAR program with the switch /ASK

*Default value(s): * /NOASK*

- **DE_APPLY**

The uv-data in a SCN-file are NEVER physically modified. However, they may have been modified before they were stored in their present SCN-file, and some of the modification factors that were applied in that stage are stored in the Scan headers of the SCN-file. These 'A-corrections' have names that start with an 'A' (AOTHC, AEXT, etc), and may be inspected with NSCAN option SHOW. They may b
de-applied 'on-the-fly' to the uv-data whenever they are read into memory to be processed. The keyword DE_APPLY allows the user to specify which corrections are to be de-applied (or not) in this way:
- ALL or *: de-apply all available 'A-corrections'
- NONE or "": de-apply no 'A-corrections'
- [NO]OTH: other telescope gain/phase corrections (i.e. from calibrator)
- [NO]EXT: extinction correction (elevation-dependent gain)
- [NO]REF: refraction correction (elevation-dependent phase)

- [NO]IREF: ionospheric refracion
- [NO]CLK: clock correction
- [NO]FAR: Faraday rotation
- [NO]MOD: source model stored in SCN-file
- [NO]SHFT: shift data stored in scan file
- [NO]IFR: additve interferometer corrs (e.g. DC offset)
- [NO]MIFR: multiplicative interferometer corrs

Multiple 'A-corrections' may be specified, separated by comma's:
Example: NONE,OTH,EXT
Example: ALL,NOOTH,NOEXT

This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
Its value may be modified (for a given $stream) in several ways:
- Permanently: change the default value: dws[pecify] NGEN[$stream]
- Interactively: remove the /NOASK switch from the default value
- Incidentally: run a NEWSTAR program with the switch /ASK

Default value(s): NONE /NOASK

- **UFLAG** (NONE,ALL,MAN,OLD,CLIP,NOIS,ADD,SHAD,U1,U2,U3) (specify un-flag details)
The uv-data in a SCN-file can be flagged in a variety of ways, mostly through the DELETE(=FLAG) option in NSCAN, or by manual flagging in NGIDS. Different types of flagging will be given different flags. Flagged data can be used in any program by "unflagging" the corresponding type.
The keyword UFLAG allows the user to specify which flags are to be discarded by specifying one or more of the following keys:
  - NONE or "": do not use flagged data
  - ALL or *: use flagged data as if not flagged
  - OLD: use flagged data of the "OLD" class (i.e. flagged before 930609, and converted with NVS option)
  - MAN: use flagged data of MANUAL class
  - CLIP: use flagged data of CLIP class
  - NOIS: use flagged data of NOISE class
  - SHAD: use flagged data of SHADOW class
  - ADD: use flagged data of ADDITIVE class
  - U1: use flagged data of USER1 class
  - U2: use flagged data of USER2 class
  - U3: use flagged data of USER3 class

  Multiple flags may be specified, separated by comma's:
  Example: NONE,U1,NOIS

  NB: If a certain flag is to be dis-abled permanently, it is better to delete it in the SCN-file header(s), using NSCAN option DELETE(=FLAG) UNDELETE

  This keyword is part of NGEN, i.e. it is a NEWSTAR 'environment keyword', which is available to all NEWSTAR programs. The switch /NOASK indicates that it is 'hidden', i.e. the user is not prompted for it, but the default value is used.
  Its value may be modified (for a given $stream) in several ways:
  - Permanently: change the default value: dws[pecify] NGEN[$stream]
  - Interactively: remove the /NOASK switch from the default value
  - Incidentally: run a NEWSTAR program with the switch /ASK

  Default value(s): NONE /NOASK

- **DELETE_NODE** (yes/no)
Specify the deletion of a node (YES) or not (NO)
Default value(s): NO /ASK

- **DISPLAY** (Output Display)
  Specify the X-Display on which to produce output. E.g. if you are working from a workstation or Xterminal called e.g. rzmwx5 on another machine (maybe a fast processor, say rzmvfp); you can get the NGIDS display on your local machine by giving as answer: rzmwx5:0.0
  If you specify a wildcard (*) or an empty answer, the display will be taken from the environment variable DISPLAY; if this environment variable does not exist, the default :0.0 will be assumed.
  If your display machine is your working machine, give :0.0 (the default).
  Note: Before being able to use your local display you should enable it by typing while logged in on your local machine (rzmwx5) either:
  host +
  or:
  host +rzmfvp
  *Default value(s): * /NOASK*

1.2 Keywords for data-file (node) selection

- **SCN_NODE** (input/output 'node' name)
  Specify the node name.
- **WMP_NODE** (input/output 'node' name)
  Specify the node name.
- **MDL_NODE** (input/output 'node' name)
  Specify the node name.
- **NGF_NODE** (input/output 'node' name)
  Specify the node name.
- **FLF_NODE** (input/output 'node' name)
  Specify the node name.
- **INPUT_SCN_NODE** (input 'node' name)
  Specify the node name.
- **INPUT_WMP_NODE** (input 'node' name)
  Specify the node name.
- **INPUT_MDL_NODE** (input 'node' name)
  Specify the node name.
- **INPUT_NGF_NODE** (input 'node' name)
  Specify the node name.
- **INPUT_FLF_NODE** (input 'node' name)
  Specify the node name.
- **OUTPUT_SCN_NODE** (output 'node' name)
  Specify the node name.
- **OUTPUT_WMP_NODE** (output 'node' name)
  Specify the node name.
- **OUTPUT_MDL_NODE** (output 'node' name)
  Specify the node name.
- **OUTPUT_NGF_NODE** (output 'node' name)
  Specify the node name.
- **OUTPUT_FLF_NODE** (output 'node' name)
  Specify the node name.

1.3 Keywords for medium (unit) selection

- **UNIT** (0,1,2,3,4,5,6,7,8,9,D) ('tape' unit)
  Specify the input unit (in Dwingeloo):
  - 0,..9 Tape/optical disk/DAT unit
  - D Disk

  By convention (but changeable by setting logical/environment MAGn):
  - 0,1,.. 0.5 inch tape
  - 4,5,.. optical disk formatted as tape
  - 9,8,.. cassette tape (DAT)

  On Unix (?):
  - 0,[3] 1600 bpi
  - 1,[4] 6250 bpi
  - 2,[5] 800 bpi

- **INPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (input 'tape' unit)
  Specify the input unit (in Dwingeloo):
  - 0,..9 Tape/optical disk/DAT unit
  - D Disk

  By convention (but changeable by setting logical/environment MAGn):
  - 0,1,.. 0.5 inch tape
  - 4,5,.. optical disk formatted as tape
  - 9,8,.. cassette tape (DAT)

  On Unix (?):
  - 0,[3] 1600 bpi
  - 1,[4] 6250 bpi
  - 2,[5] 800 bpi

- **OUTPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (output 'tape' unit)
  Specify the output unit:
  - 0,..9 Tape/optical disk/DAT unit
  - D Disk

  By convention (but changeable by setting logical/environment MAGn):
  - 0,1,.. 0.5 inch tape
  - 4,5,.. optical disk formatted as tape
  - 9,8,.. cassette tape (DAT)

  On Unix (?):
  - 0,[3] 1600 bpi
  - 1,[4] 6250 bpi
  - 2,[5] 800 bpi

1.4 Keywords for uv-data selection

The following keywords are available in many NEWSTAR programs, to select uv-data inside a Sector.

- **POLARISATION** (XYX,XY,Y,X,YX) (polarisation(s))
  Select the polarisation(s) to be read:
  - XYX all four combinations (XX,YX,YX,YY)
  - XY XX and YY
  - X XX only
  - Y YY only
  - YX XY and YX

- **HA_RANGE** (DEG) (HA range)
  Specify the HA range in which the action(s) should occur
  minimum : -180.0000, -180.0000
  maximum : 180.0000, 180.0000
  non-descending order

- **SELECT_IFRS** (Select/de-select ifrs)
  Interferometers are specified by combinations of two telescopes,
  using the following codes:
  - 0,1,2,3,4,5,6,7,8,9,A,B,C,D are the WSRT telescope names
  - * means: all 14 telescopes (0-D)
  - F means: all 10 fixed telescopes (0-9)
  - M means: all 4 movable telescopes (A-D)
  - Y means: telescopes A and B
  - Z means: telescopes C and D
  - P means: show the interferometer selection table.
  - T means: AT telescopes (8-D), U the complement (0-7).

  Interferometer groups are SELECTed by preceding them with an optional (+),
  and DE-SELECTed by preceding them with a minus sign (-).
  The standard operation is on the cross-correlations. A telescope name
  followed with a # indicates the auto-correlations.
  Examples: -*A,3,+35,-89,*,-F#,-FF

- **SELECT_TEL** (Select/de-select tels)
  Telescopes are specified using the following codes:
  - 0,1,2,3,4,5,6,7,8,9,A,B,C,D are the WSRT telescope names
  - * means: all 14 telescopes (0-D)
  - F means: all 10 fixed telescopes (0-9)
  - M means: all 4 movable telescopes (A-D)
  - Y means: telescopes A and B
  - Z means: telescopes C and D
  - P means: show the interferometer selection table.
  - T means: AT telescopes (8-D), U the complement (0-7).

  Interferometer groups are SELECTed by preceding them with an optional (+),
  and DE-SELECTed by preceding them with a minus sign (-).
  The standard operation is on the cross-correlations. A telescope name
  followed with a # indicates the auto-correlations.
  Examples: -*A,3,+35,-89,*,-F#,-FF

1.5 Keywords for Set selection

- **SCN_SETS** (Set(s) to do: g.o.f.c.s)
  A NEWSTAR SCN-file contains uv-data of one or more objects. A GROUP typically
  contains all uv-data of a certain object (or calibrator), and will consist of
  one or more OBServations. Mosaicking observations contain multiple FIELDS.
  Line observations will have several frequency CHANNELs (0=continuum).
  A SECTOR is a time-contiguous collection of 'HA-scans'. Other keywords allow
  the user to select HA-range, polarisation and interferometers within a SECTOR.

  Thus, the basic unit of a SCN-file is the Sector. The user may select a
  'Set' of Sectors by means of 5 integer index-ranges, separated by dots (.)
  .o.f.c.s means: group.obs.field.channel.sector

  Index values start at zero (0). A WILDCARD value (*) means 'all'.
  Each index may also be specified as a RANGE: first-[last][:incr]
  Unspecified indices are assumed to be '*', i.e. ...1.0 means: *.*.*.1.0
  Wildcards at the end may be omitted, i.e. 1.0 means: 1.0.*.*.* (or 1.0...)

  Multiple Sets may be specified, separated by comma's: <Set1>,<Set2>,...
  Absolute Set nrs (#n) may be specified too (if you know what is what...)
  The associated LOOPS keyword allows even more looping over index values.

  NB: Type '@' or '>' to be prompted for each of the 5 indices separately,
  and for more specific explanation per index.
  You will get used to this powerful shorthand notation soon...

- **SCN_GROUPS** (1st index: group(s))
  Give the 'group' index-range (g) of a uv-data Set spec (g.o.f.c.s)
  (group.obs.field.channel.sector)
  Example(s) of multiple 'groups' in a SCN-file:
  n observation and its calibrator(s)
  Possible answers ([]=optional):
  0 means: take first (or only) group
  n1 means: take group nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over groups n1 through n2 [step n3]
  * means: loop over all available groups (wildcard)
  n1-[*] means: loop over all available groups, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **SCN_OBSS** (2nd index: observation(s))
  Give the 'obs' index-range (o) of a uv-data Set spec (g.o.f.c.s)
  (group.obs.field.channel.sector)
  Example(s) of multiple 'observations' in the same 'group':
  the parts of an interrupted 12h observation
  several 12h obs of the same object, with different array configs
  Possible answers ([]=optional):
  0 means: take first (or only) obs
  n1 means: take obs nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over obss n1 through n2 [step n3]
  * means: loop over all available obss (wildcard)
  n1-[*] means: loop over all available obss, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **SCN_FIELDS** (3rd index: field(s))
  Give the 'field' index-range (f) of a uv-data Set spec (g.o.f.c.s)
  (group.obs.field.channel.sector)
  Example(s) of multiple 'fields':
  different pointing centres (fields) in a mosaicing obs.
  Possible answers ([]=optional):
  0 means: take first (or only) field
  n1 means: take field nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over fields n1 through n2 [step n3]
  * means: loop over all available fields (wildcard)
  n1-[*] means: loop over all available fields, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **SCN_CHANNELS** (4th index: channel(s))
  Give the 'channel' index-range (c) of a uv-data Set spec (g.o.f.c.s)
  (group.obs.field.channel.sector)
  Example(s) of multiple 'channels':
  multiple DLB/DXB frequency channels or DCB bands
  Possible answers ([]=optional):
  0 means: take continuum (or only) channel
  n1 means: take channel nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over channels n1 through n2 [step n3]
  * means: loop over all available channels (wildcard)
  n1-[*] means: loop over all available channels, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **SCN_SECTORS** (5th index: sector(s))
  Give the 'sector' index-range (s) of a uv-data Set spec (g.o.f.c.s)
  (group.obs.field.channel.sector)
  Definition: A 'sector' is a TIME-CONTIGUOUS set of HA-scans, with the
  ame frequency channel (c) and the same pointing centre (f).
  Example(s) of multiple 'sectors' (with the same g.o.f.c):
  ultiple returns to the same pointing centre in mosaicking
  Possible answers ([]=optional):

0 means: take first (or only) sector
n1 means: take sector nr n1 (n1=0,1,2,3,....)
n1-n2[:n3] means: loop over sectors n1 through n2 [step n3]
* means: loop over all available sectors (wildcard)
n1-[*] means: loop over all available sectors, starting with n1
NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_SETS** (Set(s) to do: g.f.c.p.t.m)
All maps in a WMP-file usually belong to a single GROUP (0), unless the user has put a selection of them into a new GROUP (use NMAP option REGROUP). Mosaicking FIELD and frequency CHANNEL numbering may differ from the input SCN-file (uv-data), but the input UVDAT Sets can be found in the map header. The POLARisation codes are fixed: 0=I/XX, 1=Q/XY, 2=U/YX, 3=V/YY. The TYPE
codes are also fixed: 0=map, 1=ap, 2=cov, 3=real, 4=imag, 5=ampl, 6=phas.
For each (g.f.c.p.t) there may be more than one MAP, e.g. after doing Clean.

Thus, the basic unit in a WMP-file is the Map (2D image). The user may select a 'Set' of Maps by means of 6 integer index-ranges, separated by dots (.):
.f.c.p.t.m means: group.field.channel.polar.type.map

Index values start at zero (0). A wildcard value (*) means 'all'.
Each index may also be specified as a range: first-[last][:incr]
Unspecified indices are assumed to be '*': ...1.0 means: *.*.*.1.0
Wildcards at the end may be omitted: 1.0 means: 1.0.*.*.* (or 1.0...)

Multiple Sets of Maps may be specified: <Set1> , <Set2> , ...
Absolute Map nrs (#n) may be specified too (if you know what is what...)
NB: The associated LOOPS keyword allows even more looping over index values.

NB: Type '@' or '>' to be prompted for each of the 6 Map index ranges separately, and for more specific explanation (HELP) per index.
You will get used to this powerful shorthand notation soon...

- **WMP_GROUPS** (1st index: group(s))
Give the 'group' index-range (g) of a Map Set specification (g.f.c.p.t.m)
(group.field.channel.polar.type.map)
Example(s) of multiple 'groups' in a WMP-file:
A selection of maps may be put into a separate group (NMAP REGROUP).
Possible answers ([]=optional):
0 means: take first (or only) group
n1 means: take group nr n1 (n1=0,1,2,3,....)
n1-n2[:n3] means: loop over groups n1 through n2 [step n3]
* means: loop over all available groups (wildcard)
n1-[*] means: loop over all available groups, starting with n1
NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_FIELDS** (2nd index: field(s))
Give the 'field' index-range (f) of a Map Set specification (g.f.c.p.t.m)
(group.field.channel.polar.type.map)
Example(s) of multiple 'fields':
ifferent pointing centres (fields) in a mosaicing obs.
NB: The fields may be numbered differently from those in the input SCN-file,
nstance if only a sub-set of all UVDAT fields was transformed.
he input (UVDAT) field nr can be found in the map header.
Possible answers ([]=optional):
0 means: take first (or only) field
n1 means: take field nr n1 (n1=0,1,2,3,....)
n1-n2[:n3] means: loop over fields n1 through n2 [step n3]
* means: loop over all available fields (wildcard)

n1-[*] means: loop over all available fields, starting with n1
NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_CHANNELS** (3rd index: frequ channel(s))
  Give the 'channel' index-range (c) of a Map Set specification (g.f.c.p.t.m)
  (group.field.channel.polar.type.map)
  NB: The frequency channel numbering may differ from the input SCN-file,
  ecause a map may be the result of a combination of frequ channels.
  he input (UVDAT) channel nrs can be found in the map header.
  Possible answers ([]=optional):
  0 means: take continuum (or only) channel
  n1 means: take channel nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over channels n1 through n2 [step n3]
  * means: loop over all available channels (wildcard)
  n1-[*] means: loop over all available channels, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_POLARS** (4th index: polarisation(s))
  Give the 'polar' index-range (p) of a Map Set specification (g.f.c.p.t.m)
  (group.field.channel.polar.type.map)
  The polarisations have fixed codes:
  : I or XX, 1: Q or XY, 2: U or YX, 3: V or YY
  Possible answers ([]=optional):
  n1 means: take polar nr n1 (n1=0,1,2,3)
  n1-n2[:n3] means: loop over polars n1 through n2 [step n3]
  * means: loop over all available polars (wildcard)
  n1-[*] means: loop over all available polars, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_TYPES** (5th index: type(s))
  Give the 'type' index-range (t) of a Map Set specification (g.f.c.p.t.m)
  (group.field.channel.polar.type.map)
  The map types have fixed codes:
  mage data : 0=map, 1=antenna pattern,
  ridded uv-data: 2=uvcoverage, 3=real, 4=imag, 5=ampl, 6=phase
  Possible answers ([]=optional):
  n1 means: take type nr n1 (n1=0,1,2,3,4,5,6)
  n1-n2[:n3] means: loop over types n1 through n2 [step n3]
  * means: loop over all available types (wildcard)
  n1-[*] means: loop over all available types, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **WMP_MAPS** (6th index: map(s))
  Give the 'map' index-range (m) of a Map Set specification (g.f.c.p.t.m)
  (group.field.channel.polar.type.map)
  Examples of multiple 'maps' (with the same indices g.f.c.p.t):
  new map(s) after a FIDDLE operation (map arithmetic)
  new map(s) after CLEAN and/or RESTORE operations
  Possible answers ([]=optional):
  0 means: take first (or only) map
  n1 means: take map nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over maps n1 through n2 [step n3]
  * means: loop over all available maps (wildcard)
  n1-[*] means: loop over all available maps, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **MDL_SETS** (Set(s) to do)
  Specify the Set(s) of source component Models (MDL-file) to do:

  - k e.g. #32 do the specified sub-model
  - j... e.g. 0.1.2 do the specified model

All of the k,i,j,.. can be an *, indicating all at that level. Non-specified sub-levels are assumed to be *.
i,j.. can also be specified as a loop:
n1-n2 or n1- or n1-* means all models in range n1-n2 inclusive or n1-infinite
n1[-..]:n3 means the specified range (or -* if omitted) with an increment n3 (default increment is 1):
.5-.1-7:2 means all of 0.5.1, 0.5.3, 0.5.5, 0.5.7, 0.6.1, ...

- **NGF_SETS** (Set(s) to do)
A NGF-file consists of a number of dimensional cuts through the UV-data or associated corrections. These cuts are called 'plots' in some text connected with NGCALC, by lack of a better word, and to stress the fact that the normal output of NGCALC manipulation is a plot of data values. Each contains data as a function of Hour-angle (or frequency after an odd number of 'transpose' operations).
A group typically consists of all the output cuts in a single run of the program. The mosaicing FIELD and frequency CHANNEL match in general the FIELD and CHANNEL of the original UV-data input.
The POLARisation codes are fixed: 0=XX/X, 1=XY, 2=YX, 3=YY/Y. The Interferometer (IFR) codes specify the input interferometers (0,1,... in order of baseline length) or telescopes (0,1,...,13).
For each (g.f.c.p.i) there may be more than one CUT.

  Thus, the basic unit in a NGF-file is the Cut (1D vector). The user may select a 'Set' of Cuts by means of 6 integer index-ranges, separated by dots (.):
  .f.c.p.i.c means: group.field.channel.polar.ifr/tel.cut

  Index values start at zero (0). A wildcard value (*) means 'all'.
  Each index may also be specified as a range: first-[last][:incr]
  Unspecified indices are assumed to be '*': ...1.0 means: *.*.*.1.0
  Wildcards at the end may be omitted: 1.0 means: 1.0.*.*.* (or 1.0...)

  Multiple Sets of Cuts may be specified: <Set1> , <Set2> , ...
  Absolute Cut nrs (#n) may be specified too (if you know what is what...)
  NB: The associated LOOPS keyword allows even more looping over index values.

  NB: Type '@' or '>' to be prompted for each of the 6 NGF index ranges separately, and for more specific explanation (HELP) per index.
  You will get used to this powerful shorthand notation soon...

- **NGF_GROUPS** (1st index: group(s))
Give the 'group' index-range (g) of a Cut Set specification (g.f.c.p.i.c)
(group.field.channel.polar.ifr/tel.cut)
Example(s) of multiple 'groups' in a NGF-file:
Groups will be created at different 'runs' of NGCALC
Possible answers ([]=optional):
0 means: take first (or only) group
n1 means: take group nr n1 (n1=0,1,2,3,....)
n1-n2[:n3] means: loop over groups n1 through n2 [step n3]
* means: loop over all available groups (wildcard)
n1-[*] means: loop over all available groups, starting with n1
NB: The associated LOOPS keyword allows even more looping over index values.

- **NGF_FIELDS** (2nd index: field(s))
Give the 'field' index-range (f) of a Cut Set specification (g.f.c.p.i.c)
(group.field.channel.polar.ifr/tel.cut)
Example(s) of multiple 'fields':
  ifferent pointing centres (fields) in a mosaicing obs.
Possible answers ([]=optional):

0 means: take first (or only) field
n1 means: take field nr n1 (n1=0,1,2,3,....)
n1-n2[:n3] means: loop over fields n1 through n2 [step n3]
* means: loop over all available fields (wildcard)
n1-[*] means: loop over all available fields, starting with n1
NB: The associated LOOPS keyword allows even more looping over index values.

- **NGF_CHANNELS** (3rd index: frequ channel(s))
  Give the 'channel' index-range (c) of a Cut Set specification (g.f.c.p.i.c)
  (group.field.channel.polar.ifr/tel.cut)
  Possible answers ([]=optional):
  0 means: take continuum (or only) channel
  n1 means: take channel nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over channels n1 through n2 [step n3]
  * means: loop over all available channels (wildcard)
  n1-[*] means: loop over all available channels, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **NGF_POLARS** (4th index: polarisation(s))
  Give the 'polar' index-range (p) of a Cut Set specification (g.f.c.p.i.c)
  (group.field.channel.polar.ifr/tel.cut)
  The polarisations have fixed codes:
  : X or XX, 1: XY, 2: YX, 3: Y or YY
  Possible answers ([]=optional):
  n1 means: take polar nr n1 (n1=0,1,2,3)
  n1-n2[:n3] means: loop over polars n1 through n2 [step n3]
  * means: loop over all available polars (wildcard)
  n1-[*] means: loop over all available polars, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **NGF_IFRS** (5th index: ifr(s)/tel(s))
  Give the 'ifr' index-range (i) of a Cut Set specification (g.f.c.p.i.c)
  (group.field.channel.polar.ifr/tel.cut)
  The ifr have fixed codes:
  elescope data: telescope number (0,1,...,13)
  nterferometer data: 0,1,... starting at shortest baseline in obs.
  Possible answers ([]=optional):
  n1 means: take ifr nr n1 (n1=0,1,2,3,4,5,6)
  n1-n2[:n3] means: loop over ifrs n1 through n2 [step n3]
  * means: loop over all available ifrs (wildcard)
  n1-[*] means: loop over all available ifrs starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **NGF_CUTS** (6th index: cut(s))
  Give the 'cut' index-range (m) of a Cut Set specification (g.f.c.p.i.c)
  (group.field.channel.polar.ifr/tel.c)
  Possible answers ([]=optional):
  0 means: take first (or only) cut
  n1 means: take cut nr n1 (n1=0,1,2,3,....)
  n1-n2[:n3] means: loop over cuts n1 through n2 [step n3]
  * means: loop over all available cuts (wildcard)
  n1-[*] means: loop over all available cuts, starting with n1
  NB: The associated LOOPS keyword allows even more looping over index values.

- **LOOPS** (niter,Setincr ....)
  With the LOOPS keyword, the user may specify multiple program runs, with
  different Sets (of Maps, uv-data Sectors, Plots, etc).
  This is done by specifying pairs of values:
  niter1,Setincr1 niter2,Setincr2 etc. (separated by a SPACE).
  The 1st value (niter) indicates the number of times the loop has to execute.
  The 2nd value (Setincr) indicates how the Set specification is to be changed

for subsequent loops.

Example: the Set specification g.o.f.1-2.* selects the combination of all uv-data of frequency channels 1 and 2. (g,o,f, are constants here).
If one wishes to make 32 maps of such 2-by-2 combinations of frequency channels, one may either do it laboriously by hand, or one may define the LOOP 32,...2
This LOOP increments the 4th index (channel) of the original input Set specification by 2 for each time the program is run again:
g.o.f.1-2 g.o.f.3-4 g.o.f.5-6 ..... g.o.f.29-30 g.o.f.31-32

niter should be >= 0, and the increment can be any Set definition with only simple integers or * (interpreted as 0).

Nested loops are possible too. For example, to run the program for all 64 mosaicking fields (3rd index), for 10 odd channels (4th index) per field, starting at channel 7:
Specify the initial Set as SCN_SETS=*.*.0.7.0
and two nested loops as LOOPS=64,..1, 10,...2 (separated by SPACE!)

NB: A UVDAT Set spec has the format: group.obs.field.channel.sector
A MAP Set spec has the format: group.field.channel.polar.type.map
An NGF Set spec has the format: group.field.channel.polar.ifr/tel.cut
Default value(s): "" /ASK

Summary of the keywords (parameters) of the program NSCAN Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aseries.cookbook]prog3_NSCAN.tex

1 NSCAN parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **OPTION**
  Specify action to perform:
  - LOAD: load WSRT data into scan file
  - DUMP: dump WSRT data to disk file
  FROM_OLD convert old SCN data into new format
  - TO_OLD: convert new format into old SCN data
  - SHOW: show/edit data in file
  - DELETE: (un-)delete data in file
  - COPY: copy (un)corrected scan data
  - REGROUP: make new indices for specified sets
  - UVFITS: write UVFITS tape/disk for AIPS
  - PFITS: print AIPS like FITS tape info (but also other)
  - CVX: convert a SCN file from other machine's format to local machine's
  - NVS: convert a SCN file to newest version. Should be run if SCN file
  - : made before the dates:
  - : 910417 add MJD to set header
  - : 900907 add precession rotation angle
  - : 900220 add polarisation corrections
  - : 920828 recalculate MJD for observations aborted at Wbork
  - : any recalculate prec. rot. if equal zero
  - : any calculates XY from LINOBS Stokes output
  - WERR: correct mosaic WSRT tape errors
  - QUIT: finish

- **WERR_OPTION** (WE0,WE1,QUIT) (WERR action)
  Specify action to perform:
  - WE0: correct Hour angles for tape error in splitted mosaic tapes before
  - : online version 62
  - WE1: correct all Hour angles with a constant offset
  - QUIT: finish
  valid abbreviated options

- **INPUT_FILE** (input filename)
  Specify the input filename (without an extension for the LOAD from disk option, or the PFITS option: the extension will be made by the program on the base of the label number). In the case of other options, give the full file name, possibly as obtained from DATAB if on the VAX.

- **INTEGRATION_TIME** (integration time (sec))
  Specify the integration time per scan.

- **OUTPUT_FILE** (output filename)
  Specify the full output filename (without an extension for the DUMP option: the extension will be made by the program on the base of the label number). Specify a full file name for the other options, possibly followed by a DATAB creation if on the VAX.

- **INPUT_LABELS** (input labels)
  Specify the tape labels to be read. * specifies all labels on the tape

- **OUTPUT_LABEL** (output label)
  Specify the first output tape label. * or 0 indicates at the end of the tape.

The NEWSTAR program NSCAN
Interaction with SCN-file (uv-data)

Editor: J.E.Noordam

January 20, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aseries.cookbook]prog1_NSCAN.tex

1 The program NSCAN

1.1 Overview of NCSAN options

The program NSCAN allows the user to interact with the uv-data file (SCN-file). The structure of this file is explained in detail in a separate section of this cookbook. The program NSCAN offers the following main options:

- **LOAD:** Load WSRT uv-data (in WSRT circle format) from tape, disk, DAT or optical disk into a SCN-file. The input may be from multiple tapes and/or labels. The user may select data, and change the integration time.
- **DUMP:** Dump WSRT data (in WSRT circle format) from tape or DAT or optical disk to a disk file (in WSRT circle format).
- **FROM_OLD:** Convert an old (R-series format) SCN-file into a NEWSTAR SCN-file.
- **TO_OLD:** Convert a NEWSTAR SCN-file into an old (R-series format) SCN-file.
- **SHOW:** Show/edit the contents of a SCN-file: layout, header information (incl corrections), uv-data and uv-model. This is demonstrated in the section 'Description of the SCN-file' in this Cookbook.
- **DELETE:** Delete (or un-delete) uv-data in a SCN-file, according to certain selection criteria. Actually, the data is only disabled (flagged) reversibly by making the attached weight-factor negative.
- **COPY:** (not yet available) Copy selected Sets from a SCN file to a new (secondary) SCN-file. The uv-data may be physically modified (e.g. corrections, model subtraction, change of integration time) in this process. This option will probably be implemented in a separate program NCOPY.
- **REGROUP:** Select and reorganise the data in a SCN-file. Make a new group directory entry (job tree) with specified Sets in it.
- **UVFITS:** Convert a SCN-file into a UVFITS file (tape/disk) for further image analysis in AIPS. It is recommended to do all WSRT uv-data processing first in NEWSTAR , since AIPS uv-data processing is rather VLA-oriented, and does not do justice to WSRT data.
- **PFITS:** Print a summary of a UVFITS (AIPS) tape/disk file, showing all keywords and a limited set of data.
- **CVX:** Convert a SCN file from other machine's format to local machine's.
- **NVS:** Convert a SCN file to newest version. This should be run if SCN file made before the dates:
  910417: add MJD to set header.
  900907: add precession rotation angle.
  900220: add polarisation corrections.
  920828: recalculate MJD for observations aborted at Wbork.
- **WERR:** Correct mosaic tape errors. This only concerns mosaic data taken in 1991.
- **QUIT:** Exit the program NSCAN

- **POINTING_SETS** (1,...)
  Specify the pointing centres (mosaicking) to select as a number starting at 1 (*=all)

- **CHANNELS** (bands to select)
  Specify the bands to select (*=all)

- **UVFITS_POLAR** (IQUV,IQ,I,XYX,XY,X) (polarisation info)
  Specify the polarisation info to obtain:
  - I: I only ((XX+YY)/2 if both present, else either one)
  - IQ: I ((XX+YY)/2) and Q ((-XX+YY)/2)
  - IQUV: I, Q, U ((-XY+YX)/2) and V ((XY+YX)*I/2)
  - X: XX only
  - XY: XX and YY
  - XYX: all four combinations

- **IAT_UTC** (MJD, leap seconds,...)
  Specify the IAT-UTC values as pairs of MJD at which leap second occurs, and the total number of leap seconds as from that date. The first value is for 1 Jan 1991. Values before that are in program.
  Default value(s): 48257,26,900000,27,900000,27,900000,27,900000,2

- **WERR_RA** (DEG) (RA mosaic centre)
  Specify the Right Ascension of the mosaic area centre

- **WERR_HA** (DEG) (HA correction)
  Specify the Hour angle to be added to all hour angles in node

- **HAB_OFFSET** (start data offset in sec)
  Specify the offset from the start of an observation (each Dwelltime) at which integration should start, in seconds. I.e. an offset of n will throw away the first n seconds of data.

- **SET_PATTERN** (new index pattern)
  Specify the pattern into which to change the index for the input sets. Each field in the pattern containing an * is copied from the input set, other fields are used as is (true for the first four fields, i.e. job, label, field, channel). E.g to change the index of the sets 0.*.15283.* to 0.*.0.* give the first selection as the input sets, and the second as the pattern.

- **DELETE_TYPE** (ALL,HA,ANOISE,RNOISE,MAX,IFR,CLIP,>,<,DELETE,UNDEL,NEXT,QUIT)

  Specify the type of (un-)delete to perform:
  - DELETE: select deletion
  - UNDEL: select undeletion
  - Note:: one of DELETE and UNDEL must be specified first. At any other time
  - : you can switch between the two options
  - ALL: undelete (not allowed for delete) all scans selected by node, sets,
  - : HA range
  - HA: (un-)delete all scans at specified HA's in selected node, sets,
  - : HA range
  - ANOISE: delete scans in selected node, sets, HA range with a gain and/or
  - : phase alignment noise in any of the selected polarisations greater
  - : than the specified limit (and less than zero)
  - : undelete scans with a noise less than or equal limit (and <0)
  - RNOISE: delete scans in selected node, sets, HA range with a gain and/or
  - : phase redundancy noise in any of the selected polarisations greater
  - : than the specified limit (and less than zero)
  - : undelete scans with a noise less than or equal limit (and <0)
  - MAX: delete scans in selected node, sets, HA range with
  - : max(abs(cos),abs(sin)) less than lower limit and greater than
  - : high limit
  - IFR: (un-)delete data for scans in selected node, sets, HA range for
  - : the selected (+ in table) interferometers

- CLIP: (un-)delete data for scans in selected node, sets, HA range, for
- : the selected (+ in table) interferometers
- : for which the amplitude is greater than lower limit and less than
- : high limit
- >: (un-)delete the scan at the next HA in all selected sets
- <: (un-)delete the scan at the previous HA in all selected sets
- NEXT: goto next set selection
- QUIT: leave this node
Default value(s): NEXT

- **HA** (DEG) (HA(s) to do)
Specify the HA start and end of the range in which the action(s) should occur. If only the start value given, the end value will be the same.

- **LIMIT** ((un-)delete limit)
Specify the limit for the (un-)delete action

- **LIMITS** ((un-)delete limits)
Specify the limits for the (un-)delete action

- **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT)
Specify action to perform:
- LAYOUT: give layout of tape
- SHOW: display all fields in file header
- EDIT: edit fields in file header
- CONT: continue with set headers
- QUIT: finish this node

- **SET_ACTION** (NEXT,IFRS,SHOW,EDIT,CONT,QUIT)
Specify action to perform:
- NEXT: do next specified set
- IFRS: show interferometers present
- SHOW: display all fields in set header
- EDIT: edit fields in set header
- CONT: continue with scan headers
- QUIT: finish this set

- **SCAN_ACTION** (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,E,Q)
Specify action to perform:
- XX...: select polarisation (XX initial default)
- S[how]: show detailed scan header
- D[ata]: show cos/sin data at current (default: first) HA
- A[mpl]: show ampl/phase data at current (default: first) HA
W[eight] show data weights at current (default: first) HA
- E[dit]: edit data at current (default: first) HA
- Q[uit]: go back to SET mode
- ha: give a value (e.g. -12.36) to go to this scan
- >[n]: go to the n-th next (default=1) HA
- <[n]: go to the n-th preceding (default=1) HA

- **NGEN keywords, subset of COMMON keywords**
See COMMON keyword descriptions: NGEN subset
NGEN keywords are:
- LOG
- RUN
- DATAB
- INFIX
- APPLY
- DE_APPLY
- LOOPS
- DELETE_NODE

- MODEL_OPTION
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000
  - BEAM_FACTORS
  - DELETE_LEVEL

The NEWSTAR program NCALIB
Calibration/Correction of uv-data
Part 1: Overview, and options SET and SHOW

Editor: J.E.Noordam

January 22, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

1 The program NCALIB

The program NCALIB is the heart of the NEWSTAR package. It offers a wide range of options for the determination of all kinds of instrumental corrections (calibration) and their application to the data (correction). Corrections are stored in the Set headers and Scan headers of the SCN file (see SCN file description). They may be applied to the uv-data whenever they are read into memory, subject to the specifications given by means the APPLY keyword (see the section Common features of NEWSTAR programs). Corrections in the SCN-file may also be manipulated, or 'imported' from other sources.

1.1 Overview of NCALIB options

- **REDUNDANCY:** Calculate Redundancy, Align or Selfcal corrections.
  See section NCALIB, part 2.
- **POLAR:** determine/manipulate polarisation corrections:
  See section NCALIB, part 3.
  - **CALC:** Calculate polarisation corrections
  - **SHOW:** Show polarisation corrections
  - **SET:** Set corrections manually
  - **COPY:** copy polarisation corrections from one set to others
  - **EDIT:** Edit polarisation corrections
  - **ZERO:** Zero polarisation corrections
  - **VZERO:** Calculate X-Y phase zero difference, assuming V=0
- **SET:** Set some correction data
  See section NCALIB, part 1.
  - **ZERO:** Zero selected corrections
  - **MANUAL:** Copy corrections from manual input
  - **COPY:** Copy corrections from somewhere else
  - **LINE:** Copy all corrections from corresponding continuum channel
  - **EXTINCT:** Set extinction
  - **REFRACT:** Set refraction
  - **FARADAY:** Set Faraday rotation
  - **RENORM:** Renormalise telescope corrections
- **SHOW:** Show (on printer) the average telescope corrections (over all HA-Scans) in specified set(s).
  See section NCALIB, part 1.
- **QUIT:** Quit program NCALIB

1.2 Option SET: Set various corrections in the SCN-file

The NCALIB option SET is accessed in the following manner:

```
>  exe ncalib
    NCALIB$1 is started at 17-DEC-92 15:58:01
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:  set
⊗ SET_OPTION (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT:  ...
```

In the following, the various sub-options of the NCALIB option SET will be treated in some detail.

1.2.1 SET ZERO: Set selected corrections to zero

The uv-data stored in a SCN file are never physically modified. Corrections are stored separately in the Scan and Set header(s), and applied to the data whenever they are read into memory for processing. Thus, by setting some (or all) corrections to zero, a reduction process that has gone wrong can always be returned to a known initial state. The keyword ZERO allows the user to specify which of the various kinds of corrections are to be set to zero.

```
⊗ SET_OPTION (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT:  zero
⊗ ZERO (ALL,NONE,RED,ALG,OTH,EXT,REF,FAR,NOGAIN,NOPHASE) = NONE:  all
⊗ SCAN_NODE (node name) = "":  3c147
⊗ SETS (sets to do) = "":  *                                   All sets
⊗ POLARISATION (XYX,XY,Y,X) = XYX:  <CR>                       All 4 polarisations
⊗ HA_RANGE (DEG) = *:  <CR>                                    All HA-Scans
```

NB: Going trough a large SCN file (e.g. one with many line channels) will take some time.

1.2.2 SET MANUAL: Manual input of telescope corrections

The telescope gain and phase corrections may be specified manually by the user. The given values will be stored as 'other corrections' (OTHC) in the Scan headers of the specified range (Sets and HA-range).

Note of the editor: It is not yet clear to me what happens to the other telescope corrections (REDC and ALGC) in the Scan header. It seems reasonable that they are set to zero.

```
⊗ SET_OPTION (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT:  manual
⊗ SCAN_NODE (node name) = "":  3c147
⊗ SETS (sets to do) = "":  #0
⊗ POLARISATION (XYX,XY,Y,X) = XYX:  x                          X-dipoles only (this example)
⊗ HA_RANGE (DEG) = *:  -90,-80
⊗ GAIN_X (gain corr...) = 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1:  <CR>      factor
⊗ PHASE_X (phase corr...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:  <CR>    degrees
```

1.2.3 SET RENORM: Renormalise telescope corrections

In the Redundancy calibration process, the *average* telescope gain and phase corrections over all telescopes are arbitrarily set to zero. This is a 'reasonable' assumption, unless the correction for one or more telescopes happens to be anomalously large. In that case, the gain and/or phase corrections of all the other telescopes will be shifted by an 'unreasonable' amount (since the average must be zero). Therefore, it is sometimes desirable to RENORMalise by shifting the telescope corrections by a common amount, until the average is zero for a selection of 'good' telescopes.

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: `renorm`
⊗ **SCAN_NODE** (node name) = "": `3c147`
⊗ **SETS** (sets to do) = "": `#0`
⊗ **POLARISATION** (XYX,XY,Y,X) = XYX: `xy` *Average over X and Y dipoles!*
⊗ **HA_RANGE** (DEG) = *: `-90,-80`
⊗ **GAIN_NORM** (gain tel...) = 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1: `1,1,0,1,1,1,1,1,1,1,1,1,1,0,0`
De-select RT 2,C,D
⊗ **PHASE_NORM** (phase te...) = 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1: `1,1,1,0,0,0,0,0,0,0,0,0,0,0,0` *Select only RT 0,1,2*

```
....          NCALIB$1 is ended at 16:24:17                STATUS=SUCCESS
```

Note that the program terminates upon completion, i.e. it does not return to the level of the SET option.

1.2.4 SET LINE: Copy telescope corrections from continuum channel

Since Selfcal/Redundancy calibration requires a S/N of more than 2-5, it is often possible for the continuum channel (0), but not for the individual line channels. In those cases, the telescope gain and phase corrections that have been found for channel 0 may be transferred to the line channels. This is a 'reasonable' thing to do, since the total telescope errors will vary much more than the reletive errors between channels (i.e. the bandpass shape).

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: `line`
⊗ **SCAN_NODE** (node name) = "": `3c147`
⊗ **SETS** (sets to do) = "": `#0`
⊗ **POLARISATION** (XYX,XY,Y,X) = XYX: `x`
⊗ **HA_RANGE** (DEG) = *: `-90,-80`

```
     NCALIB$1 is ended at 16:19:30                STATUS=SUCCESS
```

1.2.5 SET COPY: Copy telescope corrections from somewhere else

In the Standard Calibration process, the telescope gain and phase errors g_i and p_i are calculated with the help of a strong calibrator source, which has been observed directly before (or after) the actual observation. In order to use these calibrator corrections to correct the latter, they must be transferred (copied) from the Scan header(s) of the calibrator to the Scan header(s) of the observed object.

There are two possibilities: The calibrator observation (and thus the desired corrections) may be stored in a separate SCN-file (node), or they may be stored as another 'job' of the same SCN-file as the observed object. In the following example, the observed object is stored as job nr 0, while the calibrator observation is stored in the same SCN-file, as job nr 1:

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: [copy]
⊗ **SCAN_NODE** (node name) = "": [3c147]
⊗ **LOOPS** (n,incr ....) = "": [<CR>]
⊗ **SETS** (sets to do) = "": [0.*.*.*.*] *All Sets in job nr 0*
⊗ **POLARISATION** (XYX,XY,Y,X) = XYX: [<CR>] *All polarisations*
⊗ **HA_RANGE** (DEG) = *: [<CR>] *All HA-Scans*
⊗ **INPUT_NODE** (input node name) = *: [?]

```
Specify the node name from which the corrections should be calculated.
* indicates the same as the output node name.
```

⊗ **INPUT_NODE** (input node name) = *: [<CR>] *Use same SCN-file*
⊗ **INPUT_SETS** (sets to use) = "": [1.*.*.*.0] *Copy from Set 0 of job nr 1*

```
NCALIB$1 is ended at 16:17:30                STATUS=ERROR
```

NB: Note that the program exits upon completion, and does not return to SET_OPTION.

1.2.6 SET CCOPY: Like COPY, but more intelligent

1.2.7 SET EXTINCT: Set extinction correction

The actual atmospheric extinction factor (as a function of telescope elevetion) may differ from the default value, which is based on a standard model of the atmosphere.

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: `extinct`
⊗ **EXTINCTION** (extinction-1...) = 0.00557, 0.00461,-0.544E-03: `<CR>`
⊗ **SCAN_NODE** (node name) = "": `<CR>`

1.2.8 SET REFRACT: Set refraction correction

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: `refract`
⊗ **REFRACTION** (refraction-1,...) = 0.31E-03, 0,0: `<CR>`
⊗ **SCAN_NODE** (node name) = "": `<CR>`

1.2.9 SET FARADAY: Set Faraday rotation

Information about the ionospheric Faraday rotation during the observation may be obtained externally, e.g. from ionosonde measurements. NFRA receives these values routinely from meteorological stations not too far from the WSRT. The information may be entered into the SCN-file as a function of HA. They are stored as corrections (FARAD) in the Scan header, and will be applied routinely to the data if specified by the keyword APPLY.

⊗ **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY, ...) = QUIT: `faraday`
⊗ **FARADAY_FILE** (Faraday rotation data file) = "": `?`

```
FARADAY_FILE (Faraday rotation data file)

Specify the name of a file with Faraday rotation data if you want to have
Faraday corrections. The file is a normal ASCII file. Each line should have
2 numbers separated by a comma. The first is an hour-angle in degrees, the
second the Faraday rotation in degrees at 1 GHz.
```

⊗ **FARADAY_FILE** (Faraday rotation data file) = "": `<CR>`

1.3 Option SHOW: Print average corrections (on line printer)

The average telescope gain and phase corrections that are stored in the SCN file can be printed, for a specified range of sets. The numbers printed are a *combination* of the several kinds of telescope corrections stored in the SCN file (see the SCN-file description section). The desired combination may be specified with the keywords APPLY and D_APPLY.

The output takes the form of 14 columns of 8 numbers:
- X gain: as gain factor and as percentage (%)
- X phase: in radians and in degrees
- Y gain: as gain factor and as percentage (%)
- Y phase: in radians and in degrees

```
>  exe ncalib
    NCALIB$1 is started at 17-DEC-92 16:25:48
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:   show
⊗ LOOPS (n,incr ....) = "":   <CR>
⊗ INPUT_NODE (input node name) = "":   3c147
⊗ INPUT_SETS (sets to use) = "":   #0
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:   <CR>
    NCALIB$1 is ended at 16:27:02                STATUS=SUCCESS
```

The output will now be printed on the line printer. Note that the program terminates upon completion, i.e. it does not return to the level of the SHOW option.

1.4 Option REDUN: See section NCALIB part 2

1.5 Option POLAR: See section NCALIB part 3

The NEWSTAR program NCALIB
Calibration/Correction of uv-data
Part 2: Option REDUN: Redundancy, Selfcal, Align

Editor: J.E.Noordam

January 18, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]prog2_NCALIB.tex

1 NCALIB part 2: Redundancy, Align and Selfcal

The NCALIB option **REDUN** actually covers three related methods of estimating telescope gain and phase errors from the uv-data itself: Redundancy, Selfcal and Align. All three methods require the socalled 'Selfcal assumption', which states that all phase and gain errors are *telescope-based*. This means that the errors can be fully decomposed into contributions from individual telescopes, and that interferometer-based errors can be ignored. This assumption implies a drastic reduction in the number of *independent* errors in the data taken with an N-telescope array: from $N(N-1)/2$ to $N-1$ per integration interval. This relatively small number of independent errors can be determined with a *least-squares fitting* technique.

Any *interferometer-based* errors (e.g. thermal noise or correlator errors) violate the basic Selfcal assumption, and cause propagating errors in the solution. Fortunately, the WSRT correlators only contribute very small interferometer errors (typically $< 0.01\%$), but the S/N per uv-sample should be at least 2-5 for a good solution.

Independent solutions can be obtained for telescope gain and phase errors, because they are mathematically 'orthogonal'.

- **Selfcal:** The telescope gain and phase errors are estimated by comparing the uv-data with a model of the observed source. In WSRT Selfcal, the information from redundant spacings can be added as *extra constraints* on the Selfcal solution. Since this extra information is model-independent, the Selfcal process is less likely to converge to the wrong result.
- **Redundancy:** Telescope gain and phase errors are estimated by comparing the uv-data of 'redundant' interferometers, i.e. interferometers that have the same baseline length and orientation. Since this is a comparative method, the absolute gain (flux) and the absolute phase gradient (position) cannot be determined. The result is a set of 'internally perfect' HA-scans, that still have to be 'aligned' (see below) to each other in flux and position.
- **Align:** The absolute gain and the absolute phase gradient for misaligned HA-scans can be determined with the help of a source model. This is similar to Selfcal, except that one only solves for one parameter per HA-scan, rather than for N telescope errors. This has the advantage that the source model may be less perfect, and the SNR of the uv-data may be lower. Therefore, this method may also be used to remove ionosferic phase gradients from data that have too little SNR to warrant Redundancy or Selfcal.

The figures of this section may help to illustrate the effects of these three methods. There is also a description of the relevant mathematical formalism. Finally, this section contains processing examples and an explanation of the output that is produced on the screen and in the log file.

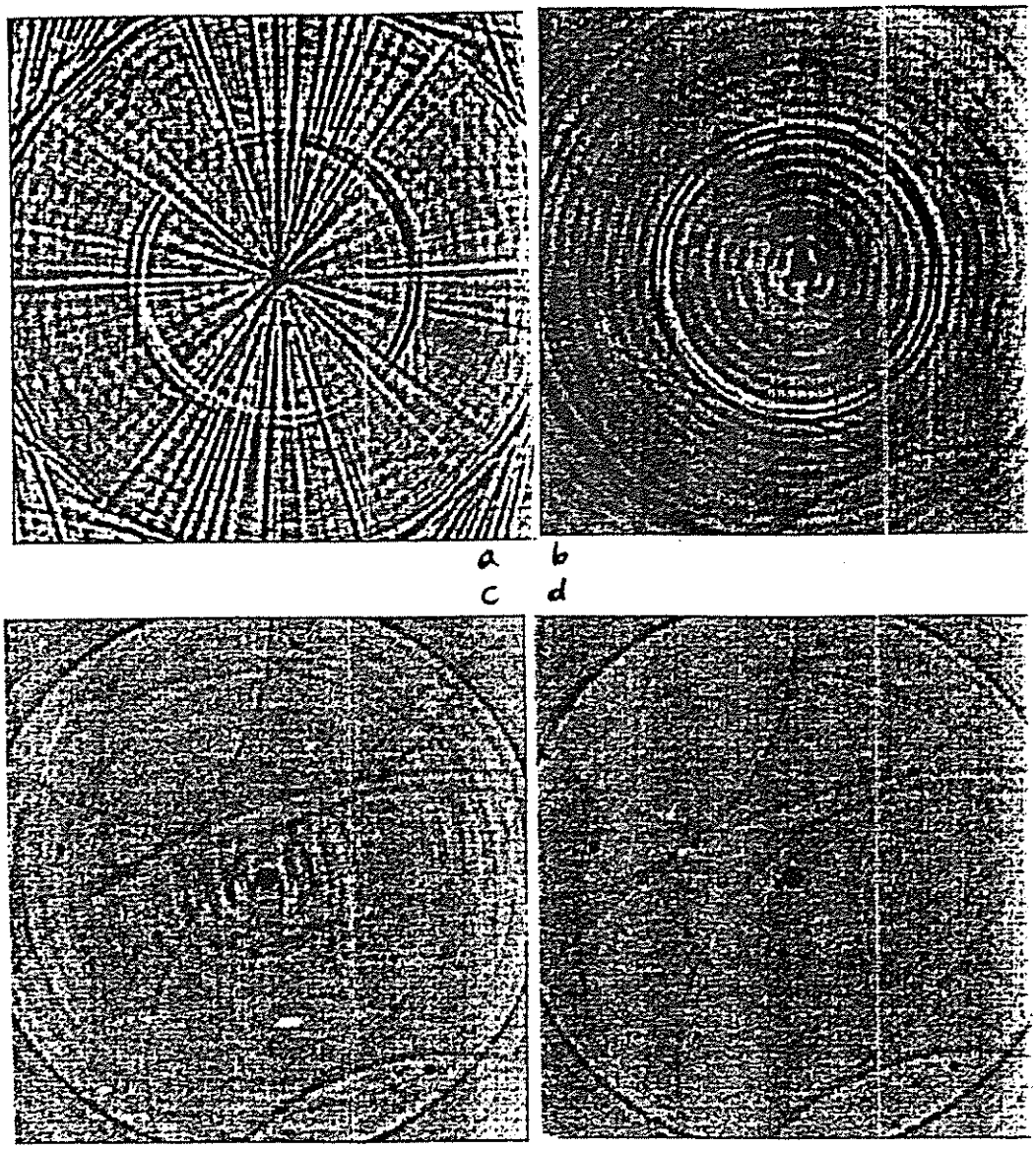


Figure 1: sg-NCALIB2-3C48

These four maps of 3C48 may serve to demonstrate the main features of the Selfcal, Redundancy and Align methods.

- The **radial stripes** *in (a) are caused by large-scale effects at specific Hour Angles. These limit the dynamic range of the 'Standard Reduction' to about 1:100 (20 dB). They are still present after a Redundancy solution, but can be be removed with an Align.*

- The prominent **rings** *in (b) are caused by a slowly varying gain or phase error between telescopes A and B. They could be removed by Redundancy, but only if there is a full solution for all 14 telescopes. Otherwise it can be removed with Align or Selfcal.*

- The **almost perfect map** *in (c) is the result of Selfcal, without Redundancy. The minor remaining problems are caused by the incompleteness of the source model, particularly the small source near 3C48.*

- The **finished product** *in (d) is essentially perfect, and has a dynamic range in excess of 1:10000 (40 dB), limited by the noise only. The remaining rings are the grating rings of sources which have not been completely subtracted. The result has been produced by a full Redundancy solution, followed by an Align with a model that contained about 10 point sources.*

NB: Any remaining problems in a WSRT map can always be easily recognised (and often diagnosed in detail), thanks to the very regular beamshape. Reduction artifacts will not easily be confused with real structure, even at very low levels.

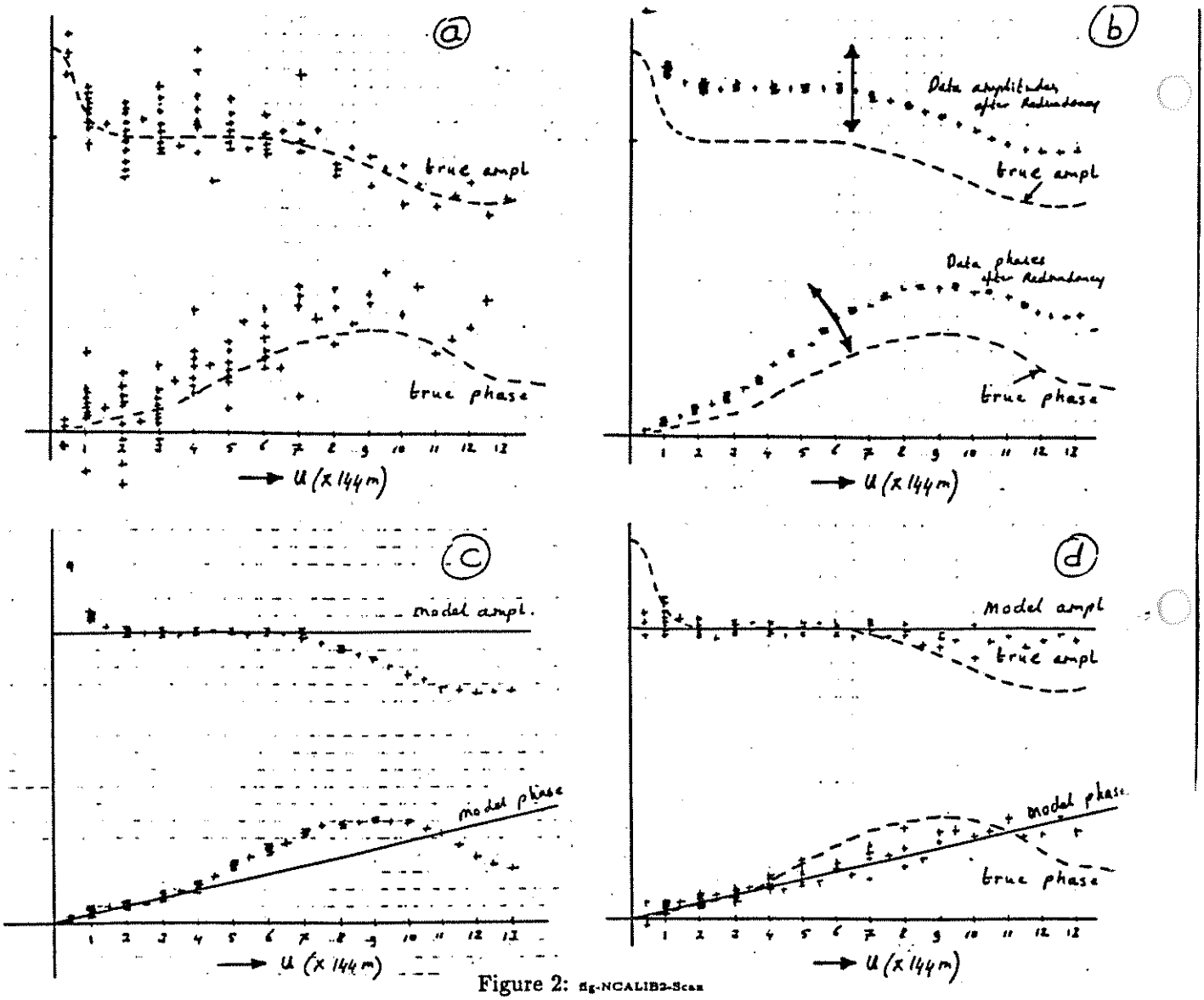


Figure 2: fig-NCALIB2-Scan

Illustration of the effects of Redundancy, Selfcal and Align on the visibility amplitudes and phases of a single HA-scan. In all four pictures, the broken line represents the 'true' visibility amplitudes and phases, as a function of baseline length (u). The actual uv-data are given by crosses. Note the multiple uv-data for redundant baselines. The source model in this example (full line) is an off-axis point source.

Redundancy: *In (a), he actually measured uv-samples are scattered, because of instrumental errors. After Redundancy calibration (b), the amplitudes and phases of redundant baselines are the same (except for a little residual scatter caused by noise). The result is an 'internally perfect HA-scan', i.e. its shape is as it should be, but two parameters are still missing: the absolute flux and the absolute position in the sky. A map made in this stage would show radial stripes around strong sources.*

Align: *In (c), the perfect HA-scans that are produced by Redundancy are 'Aligned' with the help of a model of the observed source. The 'rigid' Scans are moved as a whole to fit the model in a* weighted least-squares *sense. Arrows indicate the two parameters that are determined by Align: one to shift all amplitudes vertically, and one to rotate all phases around the origin. Note that the outcome can be influenced by giving more weight to certain baselines.*

Selfcal: *In 'normal' Selfcal (d), the source model is used to determine $2N$ telescope gain and phase errors. Since this number of independent parameters is larger than the two that were needed for Align, the Scan is less 'rigid'. This means that the data have greater freedom to adapt themselves to the* wrong *source model. To counter this effect, Redundancy constraints can be added to the Selfcal solution In the* NEWSTAR *implementation. If the Redundancy constraints would be given infinite weight, the outcome would be identical to that of Redundancy followed by Align. However, this has the disadvantage that a Redundancy solution may have 'frozen-in' errors caused by noise in certain critical baselines. Such errors may be 'thawed out' by giving more weight to Selfcal (model) constraints. In the present implementation, Selfcal and Redundancy constraints have equal weights, which may be close to optimum.*

file user5:[noordam.aseries.cookbook]FORM_REDUN.TEX

1.1 REDUN equations

1.1.1 Dipole gain and phase errors

The visibility observed by an interferometer between telescopes i and j can be described as: [1]

$$\mathcal{V}_{ij}{}^{data} = \mathcal{V}_b{}^{true} G_{ij} = \mathcal{V}_b{}^{true} G_i G_j{}^* N_{ij} \quad (1)$$

in which $\mathcal{V}_b{}^{true}$ is the (complex) true visibility, which only depends on the (projected) baseline length and orientation. $\mathcal{V}_{ij}{}^{data}$ is the observed visibility, which is 'corrupted' by the complex gain error G_{ij}. The latter can be decomposed into two *telescope-based* complex gain errors G_i and G_j (the $*$ denotes complex conjugation), and an *interferometer-based* complex gain error N_{ij} which includes the thermal noise.

NB: The two other types of telescope-based errors, dipole angle error and ellipticity, are ignored here, and assumed to be zero. They are treated in a later section on polarisation calibration (see NCALIB part 3). Their treatment is very similar to that of the gain and phase.

NB: Each WSRT telescope has two linearly polarised dipoles (X and Y), each with its own gain, phase, angle and ellipticity errors. Therefore, it would be much better to use the term *dipole errors* instead of telescope errors. Unfortunately, it is not easy to change.

The complex Visibility and Gain can be decomposed into: [2]

$$\mathcal{V} = |\mathcal{V}| \exp(i\Phi) = exp(\varrho + i\Phi) \quad (2)$$

$$G = g \exp(ip) = exp(q + ip) \quad (3)$$

Equation 1 can now be split into two independent *linear* equations for the ln(ampl) and the phase: [3]

$$\varrho_{ij}{}^{data} = \varrho_b{}^{true} + q_i + q_j + q_{ij} \quad (4)$$

$$\Phi_{ij}{}^{data} = \Phi_b{}^{true} + p_i - p_j + p_{ij} \quad (5)$$

For the calibration techniques discussed in this section (Selfcal, Redundancy and Align), we have to make the socalled **SELFCAL assumption**, i.e. that interferometer-based errors can be ignored: [4]

$$q_{ij} = p_{ij} = 0 \quad (6)$$

leaving only a small number $(N-1)$ unknown telescope-based errors, shared by a much larger number $(N(N-1)/2)$ uv-samples. Fortunately, this assumption turns out to be justified for the WSRT, provided that $S/N \gg 2$ per data sample (noise is interferometer-based).

[1] Equ 1 has LaTeX label equ-red-001
[2] Equs 2,3 have LaTeX labels equ-red-002,003
[3] Equs 4,5 have LaTeX labels equ-red-004,005
[4] Equ 6 has LaTeX label equ-red-008

1.1.2 Selfcal equations

Selfcal is a very successful technique, in which telescope gain and phase errors are estimated with the help of a tentative model of the observed source. It is een iterative process: the improved data are used to refine the model, which is used to determine the remaining errors, etc. This only works if the number of unknown parameters to be determined (i.e. telescope errors) is much smaller than the number of measured data.

Thus, if q_{ij} and p_{ij} are ignored in equs 4 and 5, and the 'true' visibilities are replaced with those of a tentative model of the observed brightness distribution, we get the **SELFCAL equations**: [5]

$$ln(ampl): \quad \varrho_{ij}{}^{data} - \varrho_b{}^{model} = q_i + q_j \tag{7}$$

$$phase: \quad \Phi_{ij}{}^{data} - \Phi_b{}^{model} = p_i - p_j \tag{8}$$

Since the left sides of the Selfcal equations are known, we can determine the unknown telescope errors on the tight side by solving the set of linear Selfcal equations for all interferometers in the array.

Since the gains and phases are mathematically *orthogonal* (i.e. they do not influence each other), their solutions can be obtained separately. The generic matrix equation to be solved in both cases has the form: [6]

$$[wdm] = W \times [e] \tag{9}$$

in which $[wdm]$ is a vector of known values ($\mathcal{W}(data - model)$), and $[e]$ is a vector of unkown telescope errors. Thus, each row in this matrix equation represents a *weighted* constraint on the ultimate solution. (Each equation is multiplied with a weight factor $\mathcal{W}$, which determines its relative influence on the solution).

The matrix W is rectangular because there are more equations (data) than unknowns. Inversion of this matrix equation is equivalent to a *weighted least-squares fit*: [7]

$$[e] = W^{-1} \times [wdm] = (W^T W)^{-1} W^T \times [wdm] \tag{10}$$

To save space and time, the NEWSTAR implementation uses the much smaller matrix $(W^T W)^{-1}$ and the vector $W^T[wdm]$, which is of course mathematically entirely equivalent.

NB: An interferometer only measures *phase-differences* $p_i - p_j$, but we are trying to solve for absolute telescope phases p_i. Therefore, the phase matrix cannot be inverted, unless an extra constraint equation is added to it, which has the effect of choosing an arbitrary phase-zero p_0. This will not affect the Selfcal solution in any way.

1.1.3 Redundancy equations

Two interferometers are 'Redundant' if:
- their baselines have the same length and orientation,
- they are observing the sky through the same 'filters' (primary beam shape, polarisation and frequency bandpass)

Since redundant baselines should yield identical values for the measured visibilities, they are a powerful tool to detect instrumental errors. This can be done in many ways, which have not all been explored yet. We will only treat the case of telescope-based gain and phase errors here. For this it is necessary to make the **Selfcal assumption**, i.e. that interferometer-based errors can be ignored.

[5] Equs 7,8 have LaTeX labels `equ-red-012,014`
[6] Equ 9 has LaTeX labels `equ-red-016`
[7] Equ 10 has LaTeX labels `equ-red-018`

By combining equations of the type (4) for redundant baselines, the ${\varrho_b}^{true}$ terms can be eliminated to yield the various types of **Redundancy gain equations:** [8]

$$\begin{aligned} {\varrho_{ij}}^{data} - {\varrho_{kl}}^{data} &= (g_i + g_j) - (g_k + g_l) \\ {\varrho_{ij}}^{data} - {\varrho_{kl}}^{data} + ({\varrho_{ik}}^{data} - {\varrho_{jl}}^{data}) &= 2\,(g_i - g_l) \\ {\varrho_{ij}}^{data} - {\varrho_{kl}}^{data} - ({\varrho_{ik}}^{data} - {\varrho_{jl}}^{data}) &= 2\,(g_j - g_k) \end{aligned} \qquad (11)$$

Similarly, by combining equations of the type (5) we get the **Redundancy phase equations:** [9]

$$\begin{aligned} {\Phi_{ij}}^{data} - {\Phi_{kl}}^{data} &= (p_i - p_j) - (p_k - p_l) \\ {\Phi_{ij}}^{data} - {\Phi_{jk}}^{data} &= p_i - 2\,p_j + p_k \end{aligned} \qquad (12)$$

The Redundancy equations look very much like the Selfcal equations: the left sides are known, and the right sides are linear functions of the telescope errors only. Again, separate solutions can be made for the ln(gain) and the phase, by matrix inversion (see section 1.1.2 above). Again, each equation is multiplied by a weight factor W to determine its relative influence on the solution.

Redundancy calibration cannot be used to determine absolute telescope errors. Since the outputs of redundant baselines are *compared* with each other, information about *absolute* position and flux is lost. The matrices can only be inverted if this missing information is supplied in the form of extra constraint equations. This can be done in two ways:

- **Implicitly**, by adding Selfcal equations. This is possible, since Selfcal and Redundancy equations have the same form. The absolute information is in the flux and position of the source model.
- **Explicitly**, by arbitrarily specifying the absolute error value(s) of one or more telescopes. In the latter case, the average of all telescope errors is usually set to zero. For the phase solution, it is also necessary to set the linear phase error gradient over the array to zero. These are assumed to be 'reasonable' assumptions.

1.1.4 Align equations

It often happens that the dominant gain and phase errors are large-scale, i.e. they are shared by all the telescopes in the array. Examples are:
- A common gain error, caused by atmospheric absorption.
- A linear phase gradient over the array, caused by ionospheric refraction.
Both of these effects are also present after a Redundancy solution, in which the absolute gain and the absolute phase gradient have been chosen arbitrarily, but the HA-scans are otherwise 'perfect'.

In these cases, one may use a Selfcal model of the observed source to properly 'Align' the various HA-scans, which would otherwise cause radial stripes around strong sources in the map. The difference with 'normal' Selfcal is, that one does not attempt to solve for individual telescope gain or phase errors, but only for the common gain factor or linear phase gradients over the array.

This leads to the **Align equations:** [10]

$$\begin{aligned} ln(gain): \quad {\varrho_{ij}}^{data} - {\varrho_b}^{model} &= q_0 \\ phase: \quad {\Phi_{ij}}^{data} - {\Phi_b}^{model} &= p^{grad}\, b + p_0 \end{aligned} \qquad (13)$$

[8] Equs 11 have LaTeX label equ-red-022
[9] Equ 12 has LaTeX label equ-red-024
[10] Equ 13 has LaTeX label equ-red-032

in which b is the baseline. The solution by matrix inversion is entirely analogous to Selfcal and Redundancy. However, since the number of unknowns is smaller in this case, the S/N ratio per visibility sample may be smaller, or the source model may be less perfect, while still producing a good result.

In a slightly more general formulation of Align, the telescopes are divided into k independent groups ($k = 1, 2, 3, ...$), for which k common gain errors must be determined, or k phase gradients: [11]

$$\begin{aligned} ln(gain): \quad & \varrho_{ij}{}^{data} - \varrho_b{}^{model} = \sum_{k=1}^{n} g_{0k} \\ phase: \quad & \Phi_{ij}{}^{data} - \Phi_b{}^{model} = \sum_{k=1}^{n} (p_k^{grad}\, b + p_{0k}) \end{aligned} \tag{14}$$

This is the case when it has not been possible to link all telescopes in a single Redundancy solution. The grouping of the telescopes, and thus the Align process, is controlled by the socalled *Freedom parameters.*

1.1.5 Complex solution

[11] Equ 14 has LaTeX label equ-red-034

1.2 Redundancy

Redundancy-only is selected by **not** specifying a source model. The resulting telescope gain and phase errors are stored in the Scan file headers, as REDC.

```
> exe ncalib
    NCALIB$1 is started at  5-OCT-92 09:16:12
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:  redun
⊗ SCAN_NODE (node name) = "":  3c147                          SCN-file
⊗ LOOPS (n,incr ....) = "":  <CR>
⊗ SETS (sets to do) = "":  *                                  Process all Sets
⊗ POLARISATION (XYX,XY,Y,X) = XYX:  X                         XX only (for example)
⊗ MODEL_OPTION (READ,WRITE, ...) ) QUIT:  <CR>                No model: Redundancy
0 sources in list

Pure redundancy selected
⊗ HA_RANGE (DEG) = *:  <CR>                                   Process all Scans
All cross interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  <CR>              Use all ifrs
⊗ SHOW_LEVEL (Level of type, print output) = 1,2:  2,2        screen, log-file
⊗ QDETAILS (more details?) = NO:  <CR>                        see section 1.6
```

```
  HA     Rk  A(%) P(deg) A(WU)  P(WU) Amax    Aavg   Arms    dAmax     dPmax  I
Set: 0.0.0.0
 11.53X New gain constraints:
          1   1   1   1   1   1   1   1   1   1   1   1   1   1
 11.53X New phase constraints:
        -37 -35 -33 -31 -29 -27 -25 -23 -21 -19 -18 -17   0   1
         38  36  34  32  30  28  26  24  22  20  19  18   1   0
 11.53X 12   0.1   0.0    3.2    2.7  4882  4396.2 226.4   -0.2 37   -0.1 13 2
 12.03X 12   0.1   0.0    3.2    2.4  4874  4387.9 226.6    0.3 78   -0.1 13 2
 12.53X 12   0.1   0.0    3.5    2.2  4888  4395.3 227.4    0.2 15   -0.1 49 2
 13.03X 12   0.1   0.0    2.8    2.3  4871  4388.6 228.4   -0.2 03    0.1 04 2
Set: 0.0.0.1
 11.53X 12   0.1   0.1    5.6    5.6  4904  4422.4 206.4   -0.3 36   -0.2 26 2
 12.03X 12   0.1   0.1    4.6    4.3  4905  4413.6 211.2    0.3 79   -0.1 07 2
 12.53X 12   0.1   0.1    4.7    5.3  4899  4414.5 210.3   -0.3 02    0.2 24 2
 13.03X 12   0.1   0.1    4.8    4.0  4910  4410.7 214.9   -0.4 03   -0.1 01 2
Set: 0.0.0.2
     ⋮
Set: 0.0.0.8

X average amplitude= 4392.835 (20.785)

X overall noise (gain, phase in W.U.):      6.6      6.1

     NCALIB$1 is ended at 09:20:24              STATUS=SUCCESS
```

1.3 Align

Align is selected by specifying a source model, and explicitly specifying the **ALIGN_OPTION**. It is then assumed by default that the Scan is 'perfect', i.e. that all 14 telescopes are grouped together. In that case, only two parameters have to be determined: the absolute gain, and the absolute phase gradient over the array. Experienced users can specify multi-parameter solutions for more than one independent groups of telescopes by manipulating the **FORCE_FREEDOM** keyword. The **MWEIGHT** keywords are used to give greater weight to those baselines (by length), for which the model is 'known' to be most accurate. The resulting telescope gain and phase corrections are stored in the Scan headers, as ALGC.

```
> dwe ncalib
    NCALIB$1 is started at  5-OCT-92 09:22:07
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:  redun
⊗ SCAN_NODE (node name) = "":  3c147                          SCN-file
⊗ LOOPS (n,incr ....) = "":  <CR>
⊗ SETS (sets to do) = "":  *                                  Process all Sets
⊗ POLARISATION (XYX,XY,Y,X) = XYX:  x                         XX only (for example)
⊗ MODEL_OPTION (READ,WRITE, ...) ) QUIT:  read                not shown here
1 sources in list
⊗ MODEL_ACTION (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL:  <CR>
⊗ MWEIGHT_TYPE (STEP,GAUSSIAN,TRIANGLE,ISTEP,IGAUSSIAN,ITRIANGLE) = STEP:  <CR>
⊗ MWEIGHT_DATA (centre, halfwidth in m) = 0,100000:  <CR>
                                          ≫ 3km: all baselines have equal weight
⊗ ALIGN_OPTION (SELFCAL,ALIGN) = SELFCAL:  align
⊗ FORCE_FREEDOM (gain,phase) = NO,NO:  <CR>                   telescope groups
Align selected
⊗ HA_RANGE (DEG) = *:  <CR>                                   Process all Scans
All cross interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  <CR>
⊗ SHOW_LEVEL (Level of type, print output) = 1,2:  2,2        screen, logfile
⊗ QDETAILS (more details?) = NO:  <CR>                        see section 1.6
```

```
  HA    Rk  A(%) P(deg) A(WU)  P(WU) Amax    Aavg   Arms    dAmax     dPmax  I
Set: 0.0.0.0
 11.53X New gain constraints:
          1   1   1   1   1   1   1   1   1   1   1   1   1   1
 11.53X New phase constraints:
        -37 -35 -33 -31 -29 -27 -25 -23 -21 -19 -18 -17   0   1
         38  36  34  32  30  28  26  24  22  20  19  18   1   0
 11.53X 12   0.1   0.0    5.2    2.8  4424  4396.9  11.4    0.3 67    0.1 58 5
     :
X average amplitude= 4392.959 (20.718)
X overall noise (gain, phase in W.U.):      5.8      4.6
      NCALIB$1 is ended at 09:25:44              STATUS=SUCCESS
```

1.4 Selfcal

Selfcal is selected by specifying a source model, and explicitly specifying 'Selfcal' to the **ALIGN_OPTION**. Redundancy constraints (equations) are included automatically if redundant spacings have been selected with **SELECT_IFRS**. The **MWEIGHT** keywords are used to give greater weight to those baselines (by length), for which the model is 'known' to be most accurate. The resulting telescope gain and phase corrections are stored in the Scan headers, as ALGC.

```
> dwe ncalib
    NCALIB$1 is started at  5-OCT-92 09:26:10
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:  [redundancy]
⊗ SCAN_NODE (node name) = "":  [3c147]                          SCN-file
⊗ LOOPS (n,incr ....) = "":  [<CR>]
⊗ SETS (sets to do) = "":  [*]                                  Process all Sets
⊗ POLARISATION (XYX,XY,Y,X) = XYX:  [X]                         XX only (for example)
⊗ MODEL_OPTION (READ,WRITE,CLEAR, ...) ) QUIT:  [read]          not shown here
1 sources in list
⊗ MODEL_ACTION (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL:  [<CR>]
⊗ MWEIGHT_TYPE (STEP,GAUSSIAN,TRIANGLE,ISTEP,IGAUSSIAN,ITRIANGLE) = STEP:  [<CR>]
⊗ MWEIGHT_DATA (centre, halfwidth in m) = 0,100000:  [<CR>]
                                          ≫ 3km: all baselines have equal weight
⊗ ALIGN_OPTION (SELFCAL,ALIGN) = SELFCAL:  [<CR>]
Selfcalibration selected
⊗ HA_RANGE (DEG) = *:  [<CR>]                                   Process all Scans
All cross interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  [<CR>]              include Redundancy constraints
⊗ SHOW_LEVEL (Level of type, print output) = 1,2:  [2,2]        screen, logfile
⊗ QDETAILS (more details?) = NO:  [<CR>]                        see section 1.6
```

```
  HA     Rk  A(%) P(deg) A(WU)  P(WU) Amax    Aavg   Arms    dAmax     dPmax  I
Set: 0.0.0.0
 11.53X New phase constraints:
          1   1   1   1   1   1   1   1   1   1   1   1   1   1
 11.53X 01   0.6   0.2   12.9    7.1  2012  2000.0   5.2    1.9 67    0.5 58 3
     ⋮

X average amplitude= 2000.005 (0.892)
X overall noise (gain, phase in W.U.):     14.2     11.1
      NCALIB$1 is ended at 09:29:02              STATUS=SUCCESS
```

1.5 Discussion of the screen/log output

The following information per HA-scan may be printed in the LOG-file and/or displayed on the terminal screen (keyword **SHOW_LEVEL**).

```
  HA     Rk  A(%) P(deg) A(WU)  P(WU) Amax    Aavg   Arms    dAmax      dPmax  I
Set: 0.0.0.0
 11.53X New phase constraints:
           1   1   1   1   1   1   1   1   1   1   1   1   1   1
 11.53X 01   0.6   0.2   12.9    7.1  2012  2000.0   5.2    1.9 67    0.5 58 3
     :
```

HA	(e.g. -83.86X)	Hour-angle (degr) and polarisation
Rk	(e.g. 12)	Rank of the gain and phase solution matrices
A(%)	(e.g. 3.3)	RMS gain residual (%)
P(deg)	(e.g. 2.2)	RMS phase residual (degr)
A(WU)	(e.g. 1.6)	RMS gain residual (WU=Westerbork Unit)
P(WU)	(e.g. 1.8)	RMS phase residual (WU)
Amax	(e.g. 63)	Maximum ampl (WU)
Aavg	(e.g. 50.1)	Average ampl (WU)
Arms	(e.g. 5.1)	RMS ampl (WU)
dAmax	(e.g. -12.4 23)	largest gain residual (WU), ifr=23
dPmax	(e.g. 6.3 CD)	largest phase residual (WU), ifr=CD

```
X average amplitude= 2000.005 (0.892)
X overall noise (gain, phase in W.U.):     14.2     11.1
```

- **Average amplitude:**
- **Average gain and phase errors per telescope:** Over the whole observation (or rather, the part that has just been processed). These number can also be calculated separately by means of NCALIB option SHOW.
- **Overall noise:** Useful for automatic deletion of 'bad' scans (see NSCAN). Redundancy-only is model-independent, so the overall noise should be equal to the thermal noise. If not, it is an indication of problems. A difference between SELFCAL noise and Redundancy-only noise is an indication of the completeness of the SELFCAL model (caution: there are different interferometers involved).
- **Graphs:** There are three kinds of line-printer graphs produced in the log-file. They give various overall gain and phase quantities per interferometer. The baseline length increases to the right. The gain axis (A) is on the left, and the phase axis (P) on the right.
  - **Graph: Average residual error X (W.U.):**
  - **Graph: Average residual error X (%, deg):**
  - **Graph: RMS X (W.U.):**

The SELFCAL and Redundancy residuals contain a wealth of information about the quality of the data and the completeness of the SELFCAL model. The user is urged to make residual plots by means of the program NPLOT.

1.6 QDETAILS: Hidden parameters

For all REDUN options, the user is prompted for 'more details?' by the NCALIB keyword QDETAILS. Hidden behind this are a number of keywords that may be manipulated by experienced users in special cases. Their default values are optimised for normal use, and wil be printed in the NCALIB LOG-file.

⊗ **BASEL_CHECK** (M) = 0.5 M: <CR>
Criterion for two baselines to be considered redundant, i.e. of identical length.

⊗ **WEIGHT_MIN** (Minimum weight accepted) = 0.01: <CR>
The weight of Selfcal and Redundancy equations is proportional to the amplitude of their uv-data. If the amplitude is very small, the information will be very noisy, and may do more harm than good to the solution. Therefore, it may be excluded by raising the value of WEIGHT_MIN, as a fraction of the maximum weight.

⊗ **SOLVE** (Solve for gain, phase (Y/N) = YES,YES: <CR>
Normally, both a gain and a phase solution will be required. But it is possible to ask for only one at a time.

⊗ **COMPLEX** (Complex solution (Y/N) = NO: <CR>
The non-linear conversion to gain and phase skews the gaussian distribution of the noise on the measured cos/sine values. Therefore, gain and phase solutions will produce a 'noise bias', which is more serious for low S/N data. This can be avoided by specifying a 'complex solution' (see also section ...

⊗ **FORCE_PHASE** (DEG) = 0 DEG,0 DEG,0 DEG,0 DEG,0 DEG,0 DEG,0 DEG, 0 DEG,0 DEG,0 DEG,0 DEG,0 DEG,0 DEG,0 DEG <CR>
If the visibility phases are close to ±180 degr, the conversion from cos/sine to phase may cause phase ambiguities (jumps) of 360 degr. This will cause problems in the phase solution, where the phases are assumed to be on a linear scale between ±∞. In order to avoid this, the user may specify initial phase corrections for all talescopes, which will be used to move the data away from ±180 degr before processing. This is of course taken into account for the total correction afterwards.

⊗ **CONTINUITY** (Continuity in solution (Y/N) = YES: <CR>
Normally, the HA-scans are processed in HA-order. The gain and phase errors determined for the last Scan may be used to correct the data of the next Scan before processing. (This is of course taken into account for the total correction afterwards). This approach is useful to keep phases away from ±180 degr, where they may cause unwanted phase-ambiguity problems.

⊗ **CHECKS** (Maximum deviations) = 5,5,3: <CR>
The user may set some threshold values for an automatic check on the quality of the solution of each HA-scan. A warning will be issued in the log whenever the tresholds are exceeded. However, no further action is taken by the program!

More information about these keywords can be found in the on-line help text, which is also printed in the Summary of NCALIB keywords in this Cookbook.

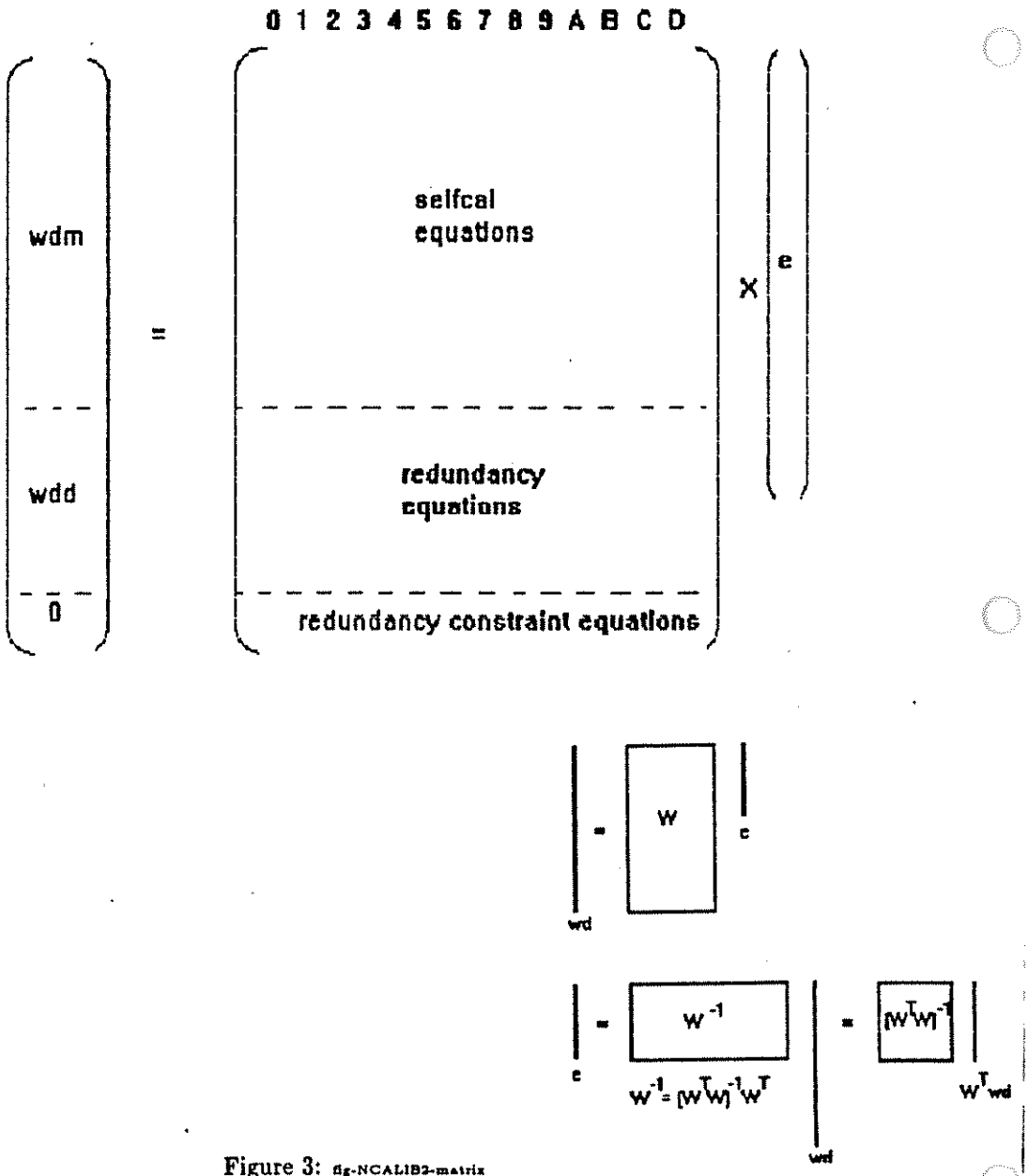


Figure 3: fig-NCALIB2-matrix

Schematic view of the least-squares fitting of telescope-based errors by means of matrix inversion.

The **problem** *is formulated by the matrix equation* $[wd] = W \times [e]$*, in which* $[wd]$ *is a vector of known values, and* $[e]$ *is a vector of unkown telescope errors. Each row (equation) in the matrix equation is multiplied by a weight factor, which determines its relative influence on the solution. The same matrix can contain a mixture of Selfcal equations and Redundancy equations, which have the same general form. If there are only Redundancy equations, extra constraint equations are needed to supply the missing information about the absolute gain (flux) and/or the absolute phase gradient over the array (position in the sky).*

The matrix W *is rectangular because their are more equations than unknowns. A least-squares* **solution** *is obtained by the inversion of* W*:*

$[e] = W^{-1} \times [wd] = (W^T W)^{-1} W^T \times [wd]$

To save space and time, the NEWSTAR *implementation uses the much smaller matrix* $(W^T W)^{-1}$ *and the vector* $W^T[wd]$*, which is of course mathematically equivalent.*

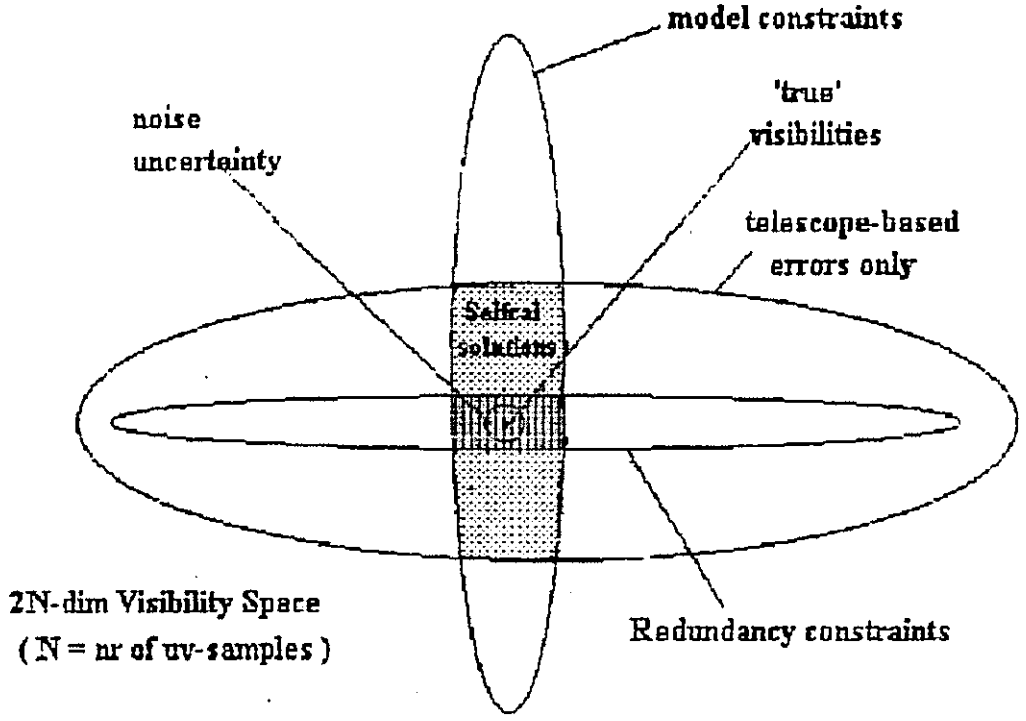


Figure 4: ag-NCALIB2-Vispace

A somewhat more abstract description of Selfcal, with or without Redundancy constraints. A perfect instrument would give visibility values that are represented by a point ('true' visibilities) in a 2N-dimensional 'visibility space' (Vispace). N is the number of measured uv-samples. The actual visibility values, which will be corrupted by instrumental errors, are represented by some other point in Vispace. In a calibration process, a priori knowledge is used to constrain the volume of Vispace in which this other point can lie. In Selfcal, the data are compared with a model of the source (model constraints), under the assumption that the only instrumental errors are telescope-based gain and phase errors (instrumental constraints). The points in Vispace that represent Selfcal Solutions must lie in the shaded volume that is bounded by the 'model constraints' and the 'telescope-based error errors only' constraints. The technique works because the two constraint volumes are largely 'orthogonal' to each other, so that the intersection volume is relatively small. Every extra constraint (more telescopes, redundant spacings) will make this intersection volume smaller.

The NEWSTAR program NCALIB
Calibration/Correction of uv-data
Part 3: Option POLAR: Polarisation corrections

Editor: J.E.Noordam

January 17, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]prog4_NCALIB.tex

1 NCALIB part 3: Polarisation corrections

The POLAR option of the program NCALIB is invoked by:

```
>   exe ncalib
    NCALIB$1 is started at 29-DEC-92 14:44:56
⊗ OPTION (REDUNDANCY,POLAR,SET,SHOW,QUIT) = QUIT:  polar
⊗ POLAR_OPTION (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT:  show      Select option
```

An overview of the various sub-options of the NCALIB option POLAR can be found in the table of contents of this section. They will be treated individually after a short treatment of the formalism.

1.1 WSRT polarisation calibration strategy

In the following, it will be assumed that the dipoles of the WSRT telescopes are *parallel* (++). The usual calibration strategy consists of the following steps:

1. The complex gain factors G_{ij} (i.e. dipole gain errors (q_i) and phase errors (p_i)) are determined by observing a strong calibrator, for which an accurate Selfcal model exists. The determination is done separately for the X-dipoles and the Y-dipoles, using equations nr (4). *Note that any inaccuracy in the assumed value for Q will be interpreted as a gain error. Also note that G_{xy} and G_{yx} will be inaccurate by the value of the "phase-zero difference (PZD)" between the X and Y dipoles.*

2. The dipole angle errors Δ_i and ellipticities Θ_i are determined by observing a strong *unpolarised* calibrator. This is done by means of NCALIB option POLAR CALC in this section, using equations nr (5). The measured corrections are stored (as POLC) in the *set header* of the SCN-file because they only change slowly in time.

3. The missing 'phase-zero difference (PZD) between the X- and Y-dipoles is determined with the help of a calibrator with strong U. The equations nr (5) are used again. It is assumed that ϵ_{xy} and ϵ_{yx}, which determine the 'leakage' of the large I-term into the result, are negligible in this stage, and that $V = 0$. If this assumption is incorrect, this will translate into a spurious V in later observations.

4. Ionospheric Faraday rotation may vary on much shorter timescales than the length of the observation. This can be calibrated to a large extent by introducing external information, obtained by ionosonde measurements. (see NCALIB part 1, option SET FARADAY).

5. Ionospheric refraction affects the apparent position of the observed source. It can be corrected with a SELFCAL model.

6. The WSRT instrumental polarisation is small, due to the placement of the feed on the axis of the antenna paraboloid. The effect can be eliminated with the help of the multi-parameter NEWSTAR source model (see NMODEL).

there is on-axis ϵ, θ, off-axis f(radius arcmin)

For the best results, it may be necessary to iterate two or three times, because one pair of dipole errors (angle/ellipticity) may affect the determination of the other pair (gain/phase). However, the process will iterate to the correct result, because, *in the case of parallel dipoles*, all four types of dipole errors are mathematically orthogonal to each other. This means that one type of dipole error cannot be interpreted as another type, and still give a consistent result. Therefore, all four types can be determined independently.

file user5:[noordam.newries.cookbook]FORM_POLAR.TEX

1.2 POLARisation equations

Each of the 14 WSRT telescopes has two perpendicular linear dipoles, X and Y. In the existing frontends, the XY-dipole unit can be rotated over an arbitrary angle. The dipole position angle (ϕ) is defined from North ($\phi = 0$ *degr*) through East ($\phi = 90$ *degr*). The complex visibilty $\mathcal{V}_{12}$ that is measured with an interferometer consisting of two dipoles with position angles ϕ_1 and ϕ_2 can be written as (see Weiler, 1973): [1]

$$\begin{aligned}\mathcal{V}_{12} = \quad & G_{12} \quad (\; I \; [cos(\phi_1 - \phi_2) \; - \; \epsilon_{12} \; sin(\phi_1 - \phi_2)] \\ & + \quad Q \; [cos(\phi_1 + \phi_2) \; - \; \eta_{12} \; sin(\phi_1 + \phi_2)] \\ & + \quad U \; [sin(\phi_1 + \phi_2) \; + \; \eta_{12} \; cos(\phi_1 + \phi_2)] \\ & - \quad i \; V \; [sin(\phi_1 - \phi_2) \; + \; \epsilon_{12} \; cos(\phi_1 - \phi_2)] \;) \end{aligned} \tag{1}$$

in which I, Q, U and V are the Fourier transforms of the corresponding Stokes parameters of the observed source, and the G, ϵ and η factors contain the four types of dipole errors: phase(p), gain($g = \log(g)$), dipole angle error (Δ) and ellipticity(Θ). In the ideal case, they are all zero. For small values of Δ and Θ, second-order terms can be ignored, and we can write: [2]

$$\begin{aligned}\epsilon_{12} &= (\Delta_1 \; - \; \Delta_2) \; - \; i \; (\Theta_1 + \Theta_2) \\ \eta_{12} &= (\Delta_1 \; + \; \Delta_2) \; - \; i \; (\Theta_1 - \Theta_2) \end{aligned} \tag{2}$$

$$G_{12} \; = \; g_1 \; g_2 \; \exp(-i(p_1 - p_2)) \; = \; \exp(g_1 + g_2 - i(p_1 - p_2)) \tag{3}$$

1.2.1 Parallel dipoles (++)

In the 'normal' position (+) of the dipole unit, $\phi_x = 90$ *degr* (east), and $\phi_y = 180$ *degr* (south). Usually the dipole units in all WSRT telescopes are set 'parallel' (++) to each other. In this case, the equations reduce to a particularly simple form. Again ignoring second-order terms, we get: [3]

$$\begin{aligned}\mathcal{V}_{xx} &= G_{xx} \; (I - Q) \\ \mathcal{V}_{yy} &= G_{yy} \; (I + Q) \end{aligned} \tag{4}$$

$$\begin{aligned}\mathcal{V}_{xy} &= G_{xy} \; (-U - iV - \epsilon_{xy} \; I) \\ \mathcal{V}_{yx} &= G_{yx} \; (-U + iV + \epsilon_{yx} \; I) \end{aligned} \tag{5}$$

in which $\mathcal{V}_{xx}$ is the visibility measured between the X-dipoles of telescopes i and j, etc. After calibration, $G = 1$ and $\epsilon = 0$, and the complex Stokes values can be calculated from the observed visibilities: [4]

$$\begin{aligned}I &= +(\mathcal{V}_{xx} + \mathcal{V}_{yy})/2 \\ Q &= -(\mathcal{V}_{xx} - \mathcal{V}_{yy})/2 \\ U &= -(\mathcal{V}_{xy} + \mathcal{V}_{yx})/2 \\ iV &= -(\mathcal{V}_{xy} - \mathcal{V}_{yx})/2 \end{aligned} \tag{6}$$

[1] Equ 1 has LaTeX label equ-pol-002
[2] Equs 2,3 have LaTeX labels equ-pol-004,006
[3] Equs 4,5 have LaTeX labels equ-pol-008,010
[4] Equs 6 have LaTeX label equ-pol-014

In the parallel mode (++), the two sets of dipole errors (phase/gain and angle/ellipticity) are mathematically *orthogonal* to each other. This means that they can be determined separately, without influencing each other. Nowadays (after about 1983), the dipoles are almost always parallel, and in the new frontends (after 1995), the dipoles can not even be rotated any more.

1.2.2 X-Y Phase Zero Difference (PZD)

In the case of parallel dipoles (++), the complex gain errors of the X-dipoles and those of the Y-dipoles are calibrated separately ($\mathcal{V}_{xy}$ and $\mathcal{V}_{yx}$ will usually not have enough signal for calibration, because their dipoles are perpendicular). Since we only calibrate phase *differences*, an arbitrary phase zero is assigned to the X-dipoles (p_{x0}) and the Y-dipoles (p_{y0}). These phase zeroes will cancel out for G_{xx} and G_{yy}, but not G_{xy} G_{yx}. Thus, the latter will be multiplied by an unknown phase factor ψ, the socalled *XY Phase Zero Difference (PZD).* [5]

$$\begin{aligned} G_{xx} &= g_x\, g_x\, \exp(-i(p_x - p_x)) \\ G_{yy} &= g_y\, g_y\, \exp(-i(p_y - p_y)) \end{aligned} \tag{7}$$

$$\begin{aligned} G_{xy} &= g_x\, g_y\, \exp(-i(p_x - p_y + \psi)) \\ G_{yx} &= g_y\, g_x\, \exp(-i(p_y - p_x - \psi)) \end{aligned} \tag{8}$$

in which $\psi = p_{x0} - p_{y0} = PZD$, and p_{x0} is the common phase zero of all the X-dipoles. Of course G_{xy} is the complex gain of the interferometer made up of the X-dipole of telescope i and the Y-dipole of telescope j, etc.

1.2.3 Crossed dipoles (+×)

Before 1983, polarisation measurements with the WSRT were usually carried out with dipoles in the fixed telescopes (0-9) in the 'normal' position, and the dipoles in the movable telescopes (A-D) rotated over 45 degr (×), with $\phi_x = 45\ degr$ and $\phi_y = 135\ degr$. The 40 standard (fixed-movable) interferometers were said to have 'crossed' (+×) dipoles. In this case, the visbility equations take the following form: [6]

$$\begin{aligned} \mathcal{V}_{xx} &= G_{xx}\,(I(1-\epsilon_{xx}) - Q(1+\eta_{xx}) + U(1-\eta_{xx}) + iV(1-\epsilon_{xx})) \\ \mathcal{V}_{yy} &= G_{yy}\,(I(1+\epsilon_{yy}) + Q(1+\eta_{yy}) - U(1-\eta_{yy}) + iV(1-\epsilon_{yy})) \\ \mathcal{V}_{xy} &= -G_{xy}\,(I(1-\epsilon_{xy}) + Q(1-\eta_{xy}) + U(1+\eta_{xy}) - iV(1+\epsilon_{xy})) \\ \mathcal{V}_{yx} &= G_{yx}\,(I(1-\epsilon_{yx}) - Q(1-\eta_{yx}) - U(1+\eta_{yx}) - iV(1+\epsilon_{yx})) \end{aligned} \tag{9}$$

With crossed dipoles, the signal-to-noise ratio of the xy/yx terms is similar to that of the xx/yy terms. This makes it possible to make a Redundancy calibration solution for all 28 dipoles simultaneously, *but only if we may assume that* $V = 0$. In principle, this would open the way to continuous polarisation calibration on the object itself, provided it has enough flux.

However, crossed dipoles introduce a number of new problems that make them less attractive:

- The two sets of dipole errors (phase/gain and angle/ellipticity) are no longer mathematically *orthogonal* to each other. This means that the phase solution will influence the angle error solution, and the gain solution will influence the ellipticity solution.

[5] Equs 7,8 have LaTeX labels **equ-pol-016,018**
[6] Equs 9 have LaTeX label **equ-pol-022**

- For part of the telescopes, the four-petal clover-leaf pattern of instrumental polarisation will be rotated by 45 degrees with respect to the sky and with respect to the four legs that support the focus box. This complicates any calibration schemes for instrumental polarisation.

These problems, combined with the fact that the phase-zero problem is not really avoided (since a Redundancy solution is only possible if it has to be assumed that $V = 0$) have led to the practice to use only parallel dipoles.

1.2.4 Faraday rotation

When radiation passes through a charged medium (like the ionosphere), the plane of linear polarisation will be rotated by the Faraday angle χ: [7]

$$\begin{aligned} Q &= Q_{obs}\cos(2\chi) + U_{obs}\sin(2\chi) \\ U &= U_{obs}\cos(2\chi) - Q_{obs}\sin(2\chi) \end{aligned} \tag{10}$$

Note that I, V and $P = \sqrt{Q^2 + U^2}$ are independent of χ.

The Faraday effect is strongly frequency-dependent: $\chi \div frequ^{-2}$.

[7] Equs 10 have LaTeX label equ-pol-024

1.3 POLAR SHOW: Show corrections

The 56 correction factors (2 for each of the 28 dipoles) for angle errors and ellipticities that are stored in the *set header* (as POLC, see SCN-file description) can be viewed in a digestible form:

⊗ **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: `show`
⊗ **SCAN_NODE** (node name) = 3C147: `<CR>`
⊗ **SETS** (sets to do) = "#0": `<CR>`

For example

Set: #0

	Position		Ellipticity		Rotation	Orthog.
	X(%)	Y(%)	X(%)	Y(%)	(deg)	(deg)
0	0.49	0.07	0.29	-0.23	0.16	-0.24
1	-0.48	-0.32	0.31	-0.40	-0.23	0.09
2	-0.35	-0.02	-0.12	0.16	-0.11	0.19
3	1.19	1.07	-0.26	0.22	0.65	-0.07
4	-0.46	-0.67	0.39	-0.49	-0.32	-0.12
5	1.54	1.74	-1.04	1.11	0.94	0.11
6	0.04	-0.10	-0.13	0.15	-0.02	-0.08
7	0.69	0.88	0.02	0.46	0.45	0.11
8	-1.11	-0.91	0.89	-0.70	-0.58	0.12
9	0.16	-0.41	-0.30	0.11	-0.07	-0.33
A	-0.61	-0.40	0.38	-0.29	-0.29	0.12
B	-0.80	-1.67	0.35	-0.33	-0.71	-0.50
C	0.06	0.02	-0.06	0.25	0.02	-0.02
D	0.02	0.32	-0.28	0.44	0.10	0.17

57.3

The first two columns (position) give the dipole angle error, expressed as a percentage of ~~360~~ degrees. In the last two columns, these same numbers are interpreted as a position error (Rotation) of the entire XY-dipole assembly in degrees, and a deviation from the nominal Orthogonality between the X and Y dipole. This is useful, since the entire XY-dipole assemply can be rotated as a whole for each WSRT telescope.

The ellipticities are also given as percentages of 360 degrees.

1.4 POLAR CALC: Calculate corrections

The dipole angle *errors* Δ_i and ellipticities Θ_i are calculated using equations nr (5). A strong calibrator is observed, which is known to be unpolarised ($U = 0$ *and* $V = 0$). It is assumed that gain and phase have already been calibrated by other means (e.g. Selfcal): $G_{xy} = G_{yx} = 1$. Thus, the equations (5) reduce to:

$$\mathcal{V}_{12}/I = \epsilon_{12} = (\Delta_1 - \Delta_2) - i(\Theta_1 + \Theta_2) \tag{11}$$

with $I = (\mathcal{V}_{11} + \mathcal{V}_{22})/2$. The system of linear equations (one for each $\mathcal{V}_{xy}$ and $\mathcal{V}_{yx}$) can be solved in a manner that is entirely analogous to the Redundancy solution for gain and phase errors (see NCALIB part 2). The separate solutions for the real and imaginary parts now give the Δ_i and Θ_i respectively.

Since the S/N of the $\mathcal{V}_{xy}$ and $\mathcal{V}_{yx}$ will be small, the least-squares solution will be more accurate if more data (Sets and HA-range) are used. However, it must of course be assumed that the Δ_i and Θ_i values are the same for all these data. This is a fairly safe assumption, since the causes for these dipole errors are 'mechanical', and vary only slowly in time. If the estimated values are to be useful for correction subsequent observations, they must at least be constant for the duration of the calibrator observation.

⊗ **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: `calc`
⊗ **SCAN_NODE** (node name) = "": `3c147`
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = "": `#0` *The more data, the better S/N*
⊗ **HA_RANGE** (DEG) = *: `<CR>` *The more data, the better S/N*

```
All cross interferometers pre-selected
```

⊗ **SELECT_IFRS** (Select/deselect Ifrs) = "": `P`

```
  0123456789ABCD
0 -+++++++++++++
1  -++++++++++++
2   -+++++++++++
3    -++++++++++
4     -+++++++++
5      -++++++++
6       -+++++++
7        -++++++
8         -+++++
9          -++++
A           -+++
B            -++
C             -+
D              -
```

⊗ **SELECT_IFRS** (Select/deselect Ifrs) = "": `<CR>`
⊗ **BASEL_CHECK** (M) = 0.5 M: `<CR>` *Redundancy criterion*

```
Set: #0
Gain (X,Y) constraints:
          1   0   1   0   1   0   1   0   1   0   1   0   1   0   1   0   1   0
          1   0   0   1   0   1   0   1   0   1
          0   1   0   1   0   1   0   1   0   1   0   1   0   1   0   1   0   1
          0   1   1   0   1   0   1   0   1   0
Phase (X,Y) constraints:
         -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0
         -1   0   0   1   0   1   0   1   0   1
          0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1   0  -1
          0  -1   1   0   1   0   1   0   1   0
```

The only difference with a 'normal' (gain/phase) Redundancy solution is that these solutions are for all 28 dipoles simultaneously. Therefore, the necessary constraint equations have 28 coefficients. The words 'gain' and 'phase' refer to the separate Real and Imaginary solutions. The succession of dipoles is 0X,0Y,1X,1Y,...etc.

The constraint equations arbitrarily set the *average ellipticity* and the *linear gradient of the dipole angle errors* over the array to zero. Without further information, this may be the most reasonable value. But if it is wrong it could affect the observations that are calibrated with the results.

The result of CALC looks very much like the ouput of SHOW (see above), except that the estimated accuracy (mean error) of the numbers is given in brackets.

	Position		Ellipticity		Rotation	Orthog.
	X(%)	Y(%)	X(%)	Y(%)	(deg)	(deg)
0	0.49(0)	0.07(0)	0.29(0)	-0.23(0)	0.16	-0.24
1	-0.48(0)	-0.32(0)	0.31(0)	-0.40(0)	-0.23	0.09
2	-0.35(0)	-0.02(0)	-0.12(0)	0.16(0)	-0.11	0.19
3	1.19(0)	1.07(0)	-0.26(0)	0.22(0)	0.65	-0.07
4	-0.46(0)	-0.67(0)	0.39(0)	-0.49(0)	-0.32	-0.12
5	1.54(0)	1.74(0)	-1.04(0)	1.11(0)	0.94	0.11
6	0.04(0)	-0.10(0)	-0.13(0)	0.15(0)	-0.02	-0.08
7	0.69(0)	0.88(0)	0.02(0)	0.46(0)	0.45	0.11
8	-1.11(0)	-0.91(0)	0.89(0)	-0.70(0)	-0.58	0.12
9	0.16(0)	-0.41(0)	-0.30(0)	0.11(0)	-0.07	-0.33
A	-0.61(0)	-0.40(0)	0.38(0)	-0.29(0)	-0.29	0.12
B	-0.80(0)	-1.67(0)	0.35(0)	-0.33(0)	-0.71	-0.50
C	0.06(0)	0.02(0)	-0.06(0)	0.25(0)	0.02	-0.02
D	0.02(0.04)	0.32(0.04)	-0.28(0.06)	0.44(0.06)	0.10	0.17

The dipole corrections estimated by CALC will be *added* to the corrections that were applied to the data when they were read in.

Note that for 'normalization' ? reasons the errors for Telescope D are always higher. This is not the case, in fact.

1.5 POLAR SET: Set corrections manually

The user may specify values for the dipole angle errors Δ_i and ellipticity Θ_i manually. The numbers given by the user are converted to internal format, and stored (as POLC) in the headers of the given range of Sets. The default values are zero.

⊗ POLAR_OPTION (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: set
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ SETS (sets to do) = "#0": <CR> *Only one Set in this example*
⊗ POL_ROTAN (dipole p...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: ?
Specify the dipole positions (rotation angles) in degrees.
⊗ POL_ROTAN (dipole p...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>
⊗ POL_ORTHOG (dipole o...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0: ?
Specify the dipole orthogonalities in degrees.
⊗ POL_ORTHOG (dipole o...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0: <CR>
⊗ POL_X_ELLIPS (X ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: ?
Specify the X dipole ellipticity in %.
⊗ POL_X_ELLIPS (X ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>
⊗ POL_Y_ELLIPS (Y ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: ?
Specify the Y dipole ellipticity in %.
⊗ POL_Y_ELLIPS (Y ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>

```
Set: #0
```

1.6 POLAR EDIT: Edit corrections

This is similar to POLAR SET above, except that the default values are the corrections (POLC) that are already stored in the Set headers. Thus, the existing POLC corrections in each Set header of the given range can be edited separately:

⊗ POLAR_OPTION (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: edit
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ SETS (sets to do) = "#0": <CR> *Only one Set in this example*

```
Set: #0
```

⊗ POL_ROTAN (dipole p...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>
⊗ POL_ORTHOG (dipole o...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0: <CR>
⊗ POL_X_ELLIPS (X ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>
⊗ POL_Y_ELLIPS (Y ellipt...) = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0: <CR>

1.7 POLAR ZERO: Zero corrections

For the specified range of sets, the POLC corrections in the set header are set to zero:

⊗ POLAR_OPTION (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: zero
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ SETS (sets to do) = "0": <CR> *All sets in job 0*

```
Set: 0.0.0.0.0
Set: 0.0.0.0.1
```

1.8 POLAR COPY: Copy corrections from somewhere else

The polarisation corrections Δ_i and Θ_i are calculated with the help of a strong, unpolarised calibrator source, using the option POLAR CALC (see above). In order to use these corrections to correct a real observation, they must be transferred (copied) from the Set header of the calibrator to the Set header(s) of the observed object.

There are two possibilities: The calibrator observation (and thus the desired corrections) may be stored in a separate SCN-file (node), or they may be stored in another 'job' of the same SCN-file as the observerved object. Below, an example is given for both situations:

If the calibrator observation is stored in a separate SCN-file:

⊗ **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: `copy`
⊗ **SCAN_NODE** (node name) = 3C147: `<CR>` *The observed source*
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = "": `0` *All sets of job 0*
⊗ **INPUT_NODE** (Input node name) = *: `?`

```
Specify the node name from which the corrections should be calculated.
* indicates the same as the output node name.
```

⊗ **INPUT_NODE** (Input node name) = *: `3C48` *Calibrator SCN-file*
⊗ **INPUT_SETS** (sets to use) = "": `#0` *First set of calibrator SCN-file*

```
Set: 0.0.0.0.0
Set: 0.0.0.0.1
```

If the calibrator observation is stored in the same SCN-file (e.g. as job nr 1, while the observed object is stored as job nr 0), the process runs as follows:

⊗ **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: `copy`
⊗ **SCAN_NODE** (node name) = 3C147: `<CR>` *The observed source*
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = "": `0` *All sets of job 0*
⊗ **INPUT_NODE** (Input node name) = *: `<CR>` *Same SCN-file*
⊗ **INPUT_SETS** (sets to use) = "": `1.0.0.0.0` *First set of job 1*

```
Set: 0.0.0.0.0
Set: 0.0.0.0.1
```

1.9 POLAR VZERO: X-Y Phase Zero Difference, assuming V=0

The VZERO option deals with the determination and manipulation of the *Phase Zero Difference (PZD)* between the X and Y dipoles. It is invoked in the following way:

⊗ **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,Z...) = QUIT: `vzero`
⊗ **VZERO_OPTION** (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: `calc`

1.9.1 POLAR VZERO CALC: Calculate and show

An Phase Zero Difference (PZD) between the X-dipoles an the Y-dipoles affects the phase of the complex gain factors G_{xy} and G_{yx} (but not G_{xx} and G_{yy}). The PZD is determined with the help of a strong calibrator source, which must have a relatively large U-component of linear polarisation, and an accurately known amount of circular polarisation (V). The latter is important, since a wrong value for V will be incorrectly interpreted as a PZD. Since the V is difficult to measure accurately, it is safer to use a calibrator that 'should not' have any circular polarisation: $V = 0$.

The algorith uses equations nr (5) and (8). It is assumed that the 'leakage' of the strong I-term is eliminated by means POLAR CALC (see ??): $\epsilon_{xy} = \epsilon_{yx} = 0$. It is also assumed that the gain and phase errors have also been well-calibrated: $g_i = p_i = 0$. Thus, equations nr (5) reduce to:

$$\begin{aligned} \mathcal{V}_{xy} &= -U \exp(-i\psi) \\ \mathcal{V}_{yx} &= -U \exp(+i\psi) \end{aligned} \tag{12}$$

The calculation of the PZD angle (ψ) is now quite straightforward. Because of the low S/N of the 'cross-terms' $\mathcal{V}_{xy}$, as many data (Sets, HA-range) should be used in the estimation as possible. However, the PZD may change with time.

⊗ **VZERO_OPTION** (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: `calc`
⊗ **SCAN_NODE** (node name) = 3C147: `<CR>`
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = *: `<CR>` *The more data, the lower S/N*
⊗ **HA_RANGE** (DEG) = *: `<CR>` *The more data, the lower S/N*
⊗ **SELECT_IFRS** (Select/deselect Ifrs) = "": `<CR>`

```
Set: 0.0.0.0.0
Set: 0.0.0.0.1
```

The resulting PZD angle is given as follows:

```
A complex angle of 0.02-0.00I(0.01+0.00I) or -0.82 degrees
```

1.9.2 POLAR VZERO APPLY: Calculate, show and apply

The PZD angle is 'applied' by adjusting the phase in the 'other corrections' (OTHC) in the Scan headers. This is done for the entire specified range (Sets and HA-range).

⊗ VZERO_OPTION (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: apply
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ LOOPS (n,incr ....) = "": <CR>
⊗ SETS (sets to do) = "#0": <CR>
⊗ HA_RANGE (DEG) = *: <CR>
⊗ SELECT_IFRS (Select/deselect ifrs) = "": <CR>

```
A complex angle of 0.02-0.00I(0.01+0.00I) or -0.82 degrees
```

1.9.3 POLAR VZERO ASK: Calculate, check with user, and apply

The user may influence the result by giving another PZD, which will be 'applied' for the entire specified range (Sets and HA-range). The default is the calculated value.

⊗ VZERO_OPTION (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: ask
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ LOOPS (n,incr ....) = "": <CR>
⊗ SETS (sets to do) = "#0": <CR>
⊗ HA_RANGE (DEG) = *: <CR>
⊗ SELECT_IFRS (Select/deselect ifrs) = "": <CR>

```
A complex angle of 0.02+0.00I(0.01-0.00I) or 0.19 degrees
```

⊗ VZERO_PHASE (X-Y difference) = 0.1859503: ? *User may give another value here*

```
 Specifies the X-Y dipole phse differenece in degrees.
```

⊗ VZERO_PHASE (X-Y difference) = 0.1859503: <CR> *Use the calculated value*

1.9.4 POLAR VZERO MANUAL: Ask for and apply

The PZD angle given by the user is 'applied' to the entire specified range (Sets and HA-range). The default value is $PZD = 0\ degr$.

⊗ VZERO_OPTION (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: manual
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ LOOPS (n,incr ....) = "": <CR>
⊗ SETS (sets to do) = "#0": <CR>
⊗ HA_RANGE (DEG) = *: <CR>
⊗ VZERO_PHASE (X-Y difference) = 0: -1 *User specifies* $PZD = -1\ degr$

1.9.5 POLAR VZERO SCAN: Calculate and apply on a per-scan basis

The change of the PZD angle as a function of time can be studied by estimating it for each Scan (HA) separately. The S/N of the estimation will necessarily be low.

⊗ VZERO_OPTION (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: scan
⊗ SCAN_NODE (node name) = 3C147: <CR>
⊗ LOOPS (n,incr ....) = "": <CR>
⊗ SETS (sets to do) = "#0": <CR>
⊗ HA_RANGE (DEG) = *: <CR>
⊗ SELECT_IFRS (Select/deselect ifrs) = "": <CR>

1.9.6 POLAR VZERO COPY: Calculate from input and apply to output

The PZD angle is calculated with the help of a strong, unpolarised calibrator source. In order to use it to correct a real observation, it must be transferred (copied) from the Scan header(s) of the calibrator to the Scan header(s) of the observed object.

There are two possibilities: The calibrator observation (and thus the desired corrections) may be stored in a separate SCN-file (node), or it may be stored in another 'job' of the same SCN-file as the observerved object. NB: In the latter case, it could even be the same observation! In both cases, the PZD is calculated from a range of data (Sets and HA-range) in the INPUT_NODE, and 'applied' (i.e. stored as 'other corrections' (OTHC)) to the Scan Headers in the SCAN_NODE. In the following example, the calibrator is stored in the same SCN-file as job nr 1, while the observed object is stored as job nr 0):

⊗ VZERO_OPTION (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT) = QUIT: [copy]
⊗ SCAN_NODE (node name) = 3C147: [<CR>] *SCN-file*
⊗ LOOPS (n,incr ....) = "": [<CR>]
⊗ SETS (sets to do) = "*": [0.*.*.*.*] *Apply PZD to all Sets of Job nr 0*
⊗ INPUT_NODE (input node name) = "": [?]

```
Specify the node name from which the corrections should be calculated.
* indicates the same as the output node name.
```

⊗ INPUT_NODE (input node name) = "": [*] *The same SCN file (node)*
⊗ INPUT_SETS (sets to use) = "*": [1.*.*.*.*] *Use all Sets of job nr 1*
⊗ HA_RANGE (DEG) = *: [<CR>] *Use all Scans in these Sets*

```
  0123456789ABCD
0 -++++++++++++
1  -++++++++++++
2   -++++++++++++
3    -++++++++++
4     -+++++++++
5      -++++++++
6       -+++++++
7        -++++++
8         -+++++
9          -++++
A           -+++
B            -++
C             -+
D              -
```

⊗ SELECT_IFRS (Select/deselect Ifrs) = "": [<CR>] *Use all interferometers*

```
A complex angle of 0.02+0.00I(0.01-0.00I) or 2.49 degrees
```

Warning: Remember that the PZD may partly be an *artifact* of the separate phase/gain calibration of the X-dipoles and the Y-dipoles. This may lead to a different PZD angle for the calibrator and the subsequent observation. The difference may be several degrees, which is much greater than the accuracy that is required for the measurement of very small percentages of circular polarisation. However, the measured V may be more accurate if the observed object is extended, which will often be the case. But the truth of the matter is that, even though the WSRT may offer the best conditions for very accurate polrisation measurements, the PZD problem is essentially unsolved.

Summary of the keywords (parameters) of the program NCALIB Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.naeries.cookbook]prog5_NCALIB.tex

1 NCALIB parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **OPTION** (REDUNDANCY,POLAR,SET,SHOW,QUIT)
  Specify action to perform:
  - REDUNDANCY: make redundancy and/or align or selfcalibration solution
  - POLAR: polarisation corrections
  - SET: set (or zero), redundancy,... corrections
  - SHOW: show (on printer) average corrections in specified set(s)
  - QUIT: finish

- **POLAR_OPTION** (CALC,SHOW,SET,COPY,EDIT,ZERO,VZERO,QUIT)
  Specify action to performon polarisation corrections:
  - CALC: calculate corrections
  - SHOW: show corrections
  - SET: set corrections manually
  - COPY: copy corrections from somewhere else
  - EDIT: edit corrections
  - ZERO: zero corrections
  - VZERO: calculate etc X-Y phase difference assuming V=0
  - QUIT: finish

- **VZERO_OPTION** (CALC,APPLY,ASK,MANUAL,SCAN,COPY,QUIT)
  Specify action to performon polarisation corrections:
  - CALC: calculate and show X-Y phase difference
  - APPLY: calculate, show and apply X-Y phase difference
  - ASK: calculate, check with user and apply X-Y phase difference
  - MANUAL: ask for and apply X-Y phase difference
  - SCAN: calculate and apply on a per-scan basis the phase difference
  - COPY: calculate from input and apply to output the phase difference
  - QUIT: finish

- **SET_OPTION** (ZERO,MANUAL,COPY,LINE,EXTINCT,REFRACT,FARADAY,RENORM,QUIT)

  Specify action to performto set corrections in data:
  - ZERO: zero corrections
  - MANUAL: copy corrections from manual input
  - COPY: copy corrections from somewhere else
  - LINE: copy all corrections from corresponding continuum channel
  - EXTINCT: set extinction
  - REFRACT: set refraction
  - FARADAY: set Faraday rotation
  - RENORM: renormalise telescope corrections
  - QUIT: finish

- **MWEIGHT_TYPE** (STEP,GAUSSIAN,TRIANGLE,ISTEP,IGAUSSIAN,ITRIANGLE)

  Specify the type of weight to be applied as function of baseline:
  - STEP: a block like function
  - GAUSSIAN: a gaussian like function
  - TRIANGLE: a triangular function
  - I...: 1-specified function

  The next question will asked the parameters

- **MWEIGHT_DATA** (centre, halfwidth in m)
  Specify the centre and the halfpower-halfwidth of the model weight function to be applied in meters of baseline.

- **ALIGN_OPTION** (SELFCAL,ALIGN) (type)
  Specify selfcal type to do:
  - SELFCAL: use the model to constrain the redundancy solution
  - ALIGN: use the model to solve constraints after redundancy solution

- **FORCE_FREEDOM** (gain,phase)
  Specifies if you want to force the constraint equations for the align solution, or of the program should determine them from the specified interferometers

- **GAIN_FREEDOM** (gain grouping)
  Specify the gain groups to be solved as constraint. * means one constraint (equal to 1,1,...); to solve e.g. separate gains for fixed and movable telescopes specify: 1,1,1,1,1,1,1,1,1,1,2,2,2,2

- **PHASE_FREEDOM** (phase grouping)
  Specify the phase groups to be solved as constraint (i.e. as slopes). * means one constraint (equal to 1,1,...); to solve e.g. separate phase slopes for fixed and movable telescopes specify: 1,1,1,1,1,1,1,1,1,1,2,2,2,2

- **GAIN_NORM** (gain telescopes)
  Specify the telescopes to use as a base for renormalising the telescope gains. * means all. Telescopes selected should have a value >0 at selected position. E.g. 0,1,0,1,0 means telescopes 1 and 3

- **PHASE_NORM** (phase telescopes)
  Specify the telescopes to use as a base for renormalising the telescope phases. * means all. Telescopes selected should have a value >0 at selected position. E.g. 0,1,0,1,0 means telescopes 1 and 3

- **HA_INTEGRATION** (Scans to combine)
  Specify the number of scans to combine before solving.
  Default value(s): 0 /ASK

- **GAIN_X** (gain corrections X)
  Specify the gain corrections for X as a factor (1= no correction). The corrections are additive to any existing

- **GAIN_Y** (gain corrections Y)
  Specify the gain corrections for Y as a factor (1= no correction). The corrections are additive to any existing

- **PHASE_X** (phase corrections X)
  Specify the phase corrections for X in degrees. The corrections are additive to any existing

- **PHASE_Y** (phase corrections Y)
  Specify the phase corrections for Y in degrees. The corrections are additive to any existing

- **EXTINCTION** (extinction-1, function freq(GHz))
  Specifies the extinction coefficient (or rather 1-coefficient) as a function of frequency in GHz. The three values are the constant term, the linear and the quadratic term.
  Default value(s): .00557,.00461,-.000544

- **REFRACTION** (refraction-1, function freq(GHz))
  Specifies the refraction coefficient (or rather 1-coefficient) as a function of frequency in GHz. The three values are the constant term, the linear and the quadratic term.
  Default value(s): .00031,0.,0.

- **FARADAY_FILE** (Faraday rotation data file)
  Specify the name of a file with Faraday rotation data if you want to have Faraday corrections. The file is a normal ASCII file. Each line should have 2 numbers separated by a comma. The first is an hour-angle in degrees, the second the Faraday rotation in degrees at 1 GHz.

- **POL_ROTAN** (dipole position)
  Specify the dipole positions (rotation angles) in degrees.

- **POL_ORTHOG** (dipole orthogonality)
  Specify the dipole orthogonalities in degrees.

- **POL_X_ELLIPS** (X ellipticity)
  Specify the X dipole ellipticity in %.

- **POL_Y_ELLIPS** (Y ellipticity)
  Specify the Y dipole ellipticity in %.

- **VZERO_PHASE** (X-Y difference)
  Specifies the X-Y dipole phse differenece in degrees.

- **QDETAILS** (more details?)
  Specify if you want to specify some details

- **BASEL_CHECK** (M) (Baseline deviation allowed)
  Specify the maximum deviation between baselines for them to be considered the 'same'.
  Default value(s): .5

- **WEIGHT_MIN** (Minimum weight accepted)
  Specify the minimum relative weight of a data point that is still acceptable. The weight is relative to the maximum weight in the same scan, and in most cases can be seen as the minimum data amplitude accepted as fraction of the maximum in the scan.
  Default value(s): 0.01

- **FORCE_PHASE** (DEG) (Force phase-zeroes)
  Force for all telescopes an initial phase zero. This is useful for pathological cases.
  Default value(s): 0,0,0,0,0,0,0,0,0,0,0,0,0,0

- **CONTINUITY** (Continuity in solution (Y/N))
  Specify if continuity in gain solution wanted. If not, the initial guess for the solution will be 0 (gain) and the forced phases (phase), else the solution found in a previous scan.
  Default value(s): YES

- **SOLVE** (Solve for gain, phase (Y/N))
  Specify if solution for gain and/or phase wanted.
  Default value(s): YES,YES

- **COMPLEX** (Complex solution (Y/N))
  Specify if a complex solution of gains wanted
  Default value(s): YES

- **CHECKS** (Maximum deviations)
  Specify (not used in the programs):
  -the maximum phase deviation per individual data point, specified relative to the solution mean error
  -the allowable phase mean error relative to the gain mean error for the same scan solution
  -the mean error allowed per scan, relative to the average mean error for all scans already solved.
  Default value(s): 5,5,3

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

- **MODEL_OPTION**
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000
  - BEAM_FACTORS
  - DELETE_LEVEL

The NEWSTAR program NMODEL
Interaction with source model

Editor: J.E.Noordam

January 20, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user8:[noordam.aseries.cookbook]prog1_NMODEL.tex

1 The program NMODEL

The program NMODEL allows the user to manipulate the Selfcal source model. The latter consists of a list of multi-parameter source components and CLEAN components. In general, the model will be stored in an .MDL file, which may also contain the observing frequency and the coordinates of the field centre. But when a source model is Fourier transformed to its uv-representation and stored in a uv-data (SCN) file, a copy of the the component model is also stored with it. For a more detailed description of model components, see the section 'Description of the MDL file' in this cookbook.

1.1 Overview of NMODEL options

The main 'actions' (keyword **ACTION**) of the program NMODEL are listed below. On the adjoining page is a list of options for general model 'handling', which are available in any program that uses models from .MDL files. The distinction between the two is not always very clear.

- **HANDLE:** general model handling (see **MODEL_OPTION** on the next page).
- **HELP:** some explanation on model lists.
- **FIND:** find point sources in map(s) in WMP file(s).
- **UPDATE:** improve positions and intensity of source components (but not CLEAN components) automatically, using uv-data and the uv-model in a SCN file.
- **XUPDATE:** like UPDATE, but for parameters of extended sources.
- **FROM_OLD:** convert old (R-series) format source list (use B1950 or Apparent by preference)
- **TO_OLD:** convert to old (R-series) format source list
- **CONVERT:** convert source list from epoch to epoch, or to a different coordinate system. If the coordinates of the field centre are not known (local mode), they may be obtained from the relevant SCN file.
- **BEAM:** correct model components for primary beam attenuation. See NMODEL keywords for default 'beam factors'. If the observing frequency is not known (local mode), it may be obtained from the relevant SCN file.
- **DEBEAM:** the reverse operation of BEAM (above)
- **SAVE:** save model data in SCN file
- **GET:** get model from SCN file
- **NVS:** convert a MDL node to a newer version if a program change necessitates it (you are warned when this is the case).
- **CVXL:** convert a MDL node into the current machine's data format. NB: if a MDL node is copied from one machine to another in FTP, always set *binary (=image)* mode
- **QUIT:** quit the program NMODEL

1.2 Overview of model 'handling' options.

In various NEWSTAR programs (NMODEL, NCALIB, NMAP etc) the user is able to handle the list of model components in various ways. The keyword often used is **MODEL_OPTION**:

- **READ/WRITE:** Read/write from/to an external MDL file
- **CLEAR:** Clear the source component list, while resetting reference coordinates
- **ZERO:** Empty the source component list, but keep the coordinates of the field centre and frequency.
- **SHOW/LIST:** Show source list on terminal screen, or both terminal and LOG-file.
- **RSHOW/RLIST:** Show source list in RA/DEC coordinates
- **TOT:** Show source list statistics
- **ADD:** Add sources to the list by hand.
- **CALIB:** Convert the source list by scaling intensities and/or moving l,m positions.
- **EDIT:** Edit the sources in the list (an amplitude of zero will delete the source)
- **FEDIT:** Edit a 'field' (parameter) for a range of sources
- **MERGE:** Combine sources components that have the same position
- **SORT:** Sort the source list in decreasing amplitude (sorting will always precede a write)
- **FSORT:** Sort on a specified 'field' (parameter) in the source list
- **DEL:** Delete sources
- **DNCLOW:** Delete non-CLEAN components with low amplitudes
- **DCLOW:** Delete CLEAN components with low amplitudes
- **DAREA:** Delete sources in specified area

Summary of the keywords (parameters) of the program NMODEL Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aperlea.cookbook]prog3_NMODEL.tex

1 NMODEL parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- ACTION
  Specify action to perform:
  - HANDLE: general model handling
  - HELP: some explanation on model lists
  - FIND: find sources in map
  - UPDATE: update sources
  - XUPDATE: update source extensions
  FROM_OLD convert old format source list (use B1950 or Apparent by preference)
  - TO_OLD: convert to old format source list
  - CONVERT: convert source list from epoch to epoch or coordinate to coordinate
  - BEAM: correct sources for primary beam
  - DEBEAM: de-correct sources for primary beam
  - SAVE: save model data in SCN file
  - GET: get model from scan file
  - NVS: make new version of model file if necessary
  - CVXL: convert formats between machines
  - QUIT: finish

- MODEL_OPTION
  Specify action:
  - READ/WRITE: Read/write from/to an external source file
  - CLEAR: Empty source list, with resetting reference coordinates
  - ZERO: Empty source list, without resetting reference coordinates
  - SHOW/LIST: Show source list on terminal, or both terminal and LOG
  - RSHOW/RLIST: Show source list in RA/DEC coordinates
  - TOT: Show source list statistics
  - ADD: Add sources to list
  - CALIB: Calibrate the source list to new ampl/position
  - EDIT: Edit source list (an amplitude of zero will delete the source)
  - FEDIT: Edit a field in a range of sources
  - MERGE: Combine sources at same position
  - SORT: Sort source list in decreasing amplitude (sorting will always
  - : precede a write)
  - FSORT: Sort on a specified field in source list
  - DEL: Delete sources
  - DNCLOW: Delete non-clean sources with low amplitudes
  - DCLOW: Delete clean components with low amplitudes
  - DAREA: Delete sources in specified area
  - QUIT: Ready

- CONVERT_TO (B1950,APPARENT,LOCAL)
  Specify the output source list type.

- MODEL_ACTION
  Give 4 answers to specify the combined model/scan action. The first answer can be:
  - MERGE: merge the model saved in the scan with the model specified,
  - : and save the given model in the scan
  - ADD: use the model saved in scan, and add the model specified,
  - : and save the sum of the two in the scan
  - NEW: save the model specified in the scan
  - TEMP: use the model specified only, and do not save in scan

- INCR: use the model saved in scan, and add the model specified,
- : without saving anything in scan
The second answer can be:
- BAND: correct model for bandsmearing (except if clean component)
- NOBAND: do not correct for bandsmearing
The third answer can be:
- TIME: correct model for time smearing (except if clean component)
- NOTIME: do not correct for bandsmearing
The fourth answer can be:
- INPOL: correct model for instrumental polarisation
- NOINPOL: do not correct for instrumental polarisation

- **SORT_TYPE** (INCREASING,DECREASING)
Specify the type of sorting wanted on the model list.

- **SORT_FIELD** (I,L,M,LM,ML,ID,Q,U,V,SI,RM,LA,SA,PA,BITS,TYP,CC,TP2,DIST,POL)

Specify the field on which to sort:
- I: amplitude
- L,M: l or m
- LM,ML: l and m, or m and l
- ID: identification
- Q,U,V: Q or U or V
- SI,RM: spectral index or rotation measure
- LA,SA,PA: long or short axis, position angle
- BITS: bits (i.e. if source is extended or has polarisation)
- TYP: source type (0 is the standard)
- CC: clean component
- TP2: reserved
- DIST: distance to a specified centre
- POL: polarised intensity

- **SORT_CENTRE** (Centre l,m for SORT)
Specify the centre l,m values (in arcsec) for a distance sort.

- **EDIT_FIELD** (I,L,M,ID,Q,U,V,SI,RM,LA,SA,PA,BITS,TYP,CC,TP2)
Specify the field to edit:
- I: amplitude
- L,M: l or m
- ID: identification
- Q,U,V: Q or U or V
- SI,RM: spectral index or rotation measure
- LA,SA,PA: long or short axis, position angle
- BITS: bits (i.e. if source is extended or has polarisation)
- TYP: source type (0 is the standard)
- CC: clean component
- TP2: reserved

- **EDIT_VALUE** (edit value)
Specify the value to set in specified edit field.

- **SOURCE_NUMBER** (Source number)
Specify the number of the source to be acted upon.

- **SOURCE_LIST** (Source number list)
Specify a list of source numbers to be acted upon. If * a range will be asked for.

- **SOURCE_RANGE** (Source number range)
Specify the start and end source number. * indicates all.

- **SOURCE** (I,l,m,id,Q,U,V,lax,sax,pa,si,rm,rs)
Specify the source parameters:

- I: in W.U.
- l offset: in arcsec
- m offset: in arcsec
- id: identification number
- Q: in %
- U: in %
- V: in %
- long axis width: full halfwidth in arcsec
- short axis width: full halfwidth in arcsec
- pa: position angle long axis in degrees (N thru E)
- si: spectral index (f**si)
- rm: rotation measure (rot = rm*((c/f)**2)) in rad/m**2
- rs: reserved

- **SOURCE_FACTORS** (An/Ao,dl,dm)
  Specify the calibration parameters:
  - An/Ao: new/old amplitude
  - dl: new-old l in arcsec
  - dm: new-old m in arcsec

- **DELETE_LEVEL** (low level)
  Specify the level below which (absolutely) the source should be deleted

- **DELETE_AREA** (l,m,dl,dm arcsec)
  Specify the area in arcsec in which all sources should be deleted

- **BEAM_FACTORS** (Beam factors)
  Specify factors to be used in (de-)beaming. 6 pairs should be specified. The first of each pair is a frequency in MHz up to which the factor holds (frequencies should increase), the second the factor to be used in the formula: cos(factor*freq(MHz)*angle(degrees))**6

  Default value(s): 500,.0629,1000,0.065,2000,0.065,4000,0.065,8000

- **INPOLQ_100** (Instrumental pol.)
  Specify the instrumental polarisation. INPOL* asks for Q, U and iV. The value following the _ gives the maximum frequency (MHz) for which the values hold. The formula is used:
  - : A0+A1(a1 sin p +b1 cos p)+A2(a2 sin 2p + b2 cos 2p), with:
  - : Ai=Ci sin(Di.f.r)**2
  The first 6 values give the Ci,Di pairs, the remaining 4 the ai,bi pairs. p is the position angle from North through East; r the distance from the beam centre in degrees, f the frequency in MHz
  Default value(s): -.003137,.0024037,.01159,.0038832,.02139,.00278
  INPOL keywords are:
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000

- **OLD_FILE** (old model filename)
  Specify the filename of an old format RMODEL file that will be read (FROM) or written (TO). Give the full file name, possibly as obtained from DATAB if on the VAX, and after creation use DATAB on the VAX to connect it to node.

- **CALIBRATORS** (List of sources or *)
  Specify the source numbers to be used as calibrators.

- **REFERENCE_DATA** (RA, DEC, Freq, rot.)
  Specify (in degrees) the RA, DEC and precession rotation angle for the new reference source list position, and the spectral index reference frequency (MHz).

- **REFERENCE_FREQ** (MHz)
  Specify the spectral index reference frequency (MHz).

- **OUTPUT_NODE** (Optional output model node)
  Specify the name of the node in which the updated model should be written. If none given, the program will ask for user action.

- **MAP_LIMIT** (relative limit)
  Specify the lowest limit with respect to the maximum in the map that will be considered a valid source

- **MAX_NUMBER** (maximum number to add)
  Specify the maximum number of sources that will be found

- **ID_START** (identification number)
  Specify the start of the ID number to be used in the source list.

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

Description of the program NGCALC
Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]prog1_NGCALC.tex

1 The program NGCALC

1.1 Overview of NGCALC options

The NGCALC keyword **ACTION** can have the following responses:

- **NODE:** switch data node
- **EXTRACT:** extract information from SCN file into NGF file
- **SHOW:** show information in NGF file
- **BRIEF:** show one line information of plots in NGF file
- **MERGE:** merge a number of plot sets into a single NGF file
- **COMBINE:** combine info in NGF file(s) into new NGF file
- **TRANS:** interchange frequency and HA axes (rough version, in which bands are translated into HA's and vv
- **CALC:** do some calculation on an NGF file
- **COPY:** copy info from other NGF file
- **MONGO:** produce NGF file info into a MONGO readable file
- **PLOT:** plot NGF file(s)
- **DELETE:** delete NGF file(s)
- **CVX:** convert NGF file from other machine's format to local format
- **NVS:** update to latest NGF file format
- **QUIT:** leave the program

1.2 ...

Summary of the keywords (parameters) of the program NGCALC Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aseries.cookbook]prog3_NGCALC.tex

1 NGCALC parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **ACTION**
  Specify the action to be performed:
  - NODE: switch data node
  - EXTRACT: extract information from SCN file into NGF file
  - SHOW: show information in NGF file
  - BRIEF: show one line information of plots in NGF file
  - MERGE: merge a number of plot sets into a single NGF file
  - COMBINE: combine info in NGF file(s) into new NGF file
  - TRANS: interchange frequency and HA axes (rough version, in which bands are
  - : translated into HA's and vv
  - CALC: do some calculation on an NGF file
  - COPY: copy info from other NGF file
  - MONGO: produce NGF file info into a MONGO readable file
  - PLOT: plot NGF file(s)
  - DELETE: delete NGF file(s)
  - CVX: convert NGF file from other machine's format to local format
  - NVS: update to latest NGF file format
  - QUIT: leave
- **OUT_NODE** (Name of output data node)
  Specify the name of the node to which the data should be copied. An * indicates the same node as the input node. Note that the node name can be preceded with dev:[dir..] (or /dev/dir... if Unix). The specified dev and/or directories will become the default database for all node related names. Atstart of program the default is the current directory. The database specified should be an existing directory. By placing part of the node name in parentheses (), the string defined in such a way can in subsequent node questions be referenced with a #. Hence the following inputs will translate as:
  - ngc1204.21cm.long: ngc1204.21cm.long
  - [dwl.ger.ngc1204]ngc1204.90cm.short: [dwl.ger.ngc1204]ngc1204.90cm.short
  - (ngc1204.21cm.)long: [dwl.ger.ngc1204]ngc1204.21cm.long
  - a.#long: [dwl.ger.ngc1204]a.ngc1204.21cm.long
- **POLAR** (XYX,XY,YX,Y,X) (polarisation info)
  Specify the polarisation info to obtain:
  - X: XX only
  - Y: YY only
  - XY: XX and YY
  - YX: XY and YX
  - XYX: all four combinations
- **EXTRACT_TYPE** (TCOR,ICOR,DATA,MODEL,WEIGHT,QUIT) (data type)
  Specify the datatype to extract from SCN file:
  - TCOR: Telescope based corrections
  - ICOR: Interferometer corrections
  - DATA: Data from scan
  - MODEL: Model from scan
  - WEIGHT: Weight from scan
  - QUIT: leave
- **CALC_TYPE** (AVER,SMOOTH,POLY,DPOLY,NULL,QUIT) (calculation type)
  Specify the action to be performed on plot(s):
  - AVER: average a plot file

- SMOOTH: triangular smoothing of plot data
- POLY: calculate polynomial through plot data and output residuals
- DPOLY: output residuals if known polynomial subtracted from data
- NULL: set datapoints to deleted in plot(s)
- QUIT: leave
- Note:: All calculations are done with complex numbers. To use e.g. only
- : amplitude, convert it first with an expression= AMPL(#..)

- **HA_WIDTH** (DEG) (smoothing width)
  Specify smoothing halfwidth of triangular smoothing function in degrees HA.

- **POLY_N** (polynomial degree)
  Specify degree of polynomial to solve.

- **POLY_COEF** (polynomial coefficients)
  Specify polynomial coefficients.

- **PLOTTER** (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP,BIT1,BIT2,BIT3,X11,USE1,USE2)

  (plotter to use)
  - QMS: QMS laser printer in landscape orientation
  - QMSP: QMS laser printer in portrait orientation
  - REGIS: graphics VT terminal
  - FREGIS: (*) REGIS to file
  - EPS: encapsulated PostScript for use in textprocessors etc
  - EPP: EPS in portrait mode
  - PSL: Postscript (do not use often, especially not for halftone: slow!)
  - PSP: PostScript in portrait mode
  - BIT1: (*) bitmap for 100 dpi
  - BIT2: (*) bitmap for 200 dpi
  - BIT3: (*) bitmap for 300 bpi
  - X11: (*) X11 terminal

- **HA_SCALE** (HA plot scale degree/cm)
  Specify the HA scale in degree/cm.
  Default value(s): 15.

- **SCALE** (plot scale units/mm)
  Specify the scale in units/mm.
  Default value(s): 10.

- **OFFSET** (plot offset in units)
  Specify the plot zero point offset in units (e.g. 1 will draw (data-1) in stead of (data)).
  Default value(s): 0.

- **TELESCOPES** (Telescope(s) to select)
  Specify the telescopes to be selected: Select 1 or more telescopes by typing a string of characters consisting
  - : of a combination of 0123.....ABCD
  For all telescopes: *
  *Default value(s): **

- **PLOTS** (plots to do)
  Specify the plots to do:
  - #k: e.g. #32 do the specified sub-plot
  - i.j...: e.g. 0.1.2 do the specified plot
  All of the k,i,j,.. can be an *, indicating all at that level. Non-specified sub-levels are assumed to be *. i,j.. can also be specified as a loop: n1-n2 or n1- or n1-* means all plots in range n1-n2 inclusive or n1-infinite n1[-..]:n3 means the specified range (or -* if omitted) with an increment n3 (default increment is 1): .5-.1-7:2 means all of 0.5.1, 0.5.3, 0.5.5, 0.5.7, 0.6.1, ...

- **PLOT_SET** (plot to use)
  Specify the plot to use in the expression:
  - #k: e.g. #32 use the specified sub-plot
  - i.j...: e.g. 0.1.2 use the specified plot
  All of the k,i,j,.. can be an *, indicating all at that level. Non-specified sub-levels are assumed to be *. NOTE: In the expression only the logical first plot will be used

- **EXPRESSION**
  Specify the expression defining the new output plot. In general the expression should be enclosed in "". Elements of an expression can be: constants (e.g. 5, -1.23E-12), #nn (e.g.#20) to indicate a plot with an internal identification of 20, ##nn (e.g.##20) as #nn without loop update, functions, or expression enclosed in (). Operators, in order of decreasing priority (results of logical and relation expressions: 0. (False), 1. (True)): +,-; **; *,/; +,-; >=,<=,=,<,>,<>; <>; &; !. Functions (angles always in degrees): ATAN(x), ATAN(sin,cos), SIN(x), COS(x), ASIN(x), ACOS(x), EXP(x), EXP10(x), EXP2(x), LOG(x), LOG10(x), LOG2(x), PI, EE, HA, UT, ABS(x), FLOOR(x), CEIL(x), ROUND(x), INT(x), FRACT(x), REAL(x), IMAG(x), IMUL(x), AMPL(x), PHASE(x), SQRT(x). Examples:
  -(SQRT(REAL(#0)**2+IMAG(#0)**2)+.5E3)/PI
  -2*(#12<=0)+2*(#12>0) (will clip to -2 and +2 levels)
  Note: All calculations done on complex data; comparisons on absolute value. Loops are incorporated in standard way.

- **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT)
  Specify action to perform:
  - LAYOUT: give layout of file
  - SHOW: display all fields in file header
  - EDIT: edit fields in file header
  - CONT: continue with set headers
  - QUIT: finish this node

- **PLOT_ACTION** (NEXT,SHOW,EDIT,CONT,QUIT)
  Specify action to perform:
  - NEXT: do next specified plot
  - SHOW: display all fields in plot header
  - EDIT: edit fields in plot header
  - CONT: continue with plot data
  - QUIT: finish this plot

- **DATA_ACTION** (S,A,P,Q)
  Specify action to perform:
  - S[how]: show detailed plot data
  - A[mpl]: show amplitude of plot data
  - P[hase]: show phase of plot data
  - Q[uit]: quit data part

- **PLOT_TYPE** (COS,SIN,AMPL,PHASE)
  Specify the data type of the output plot:
  - COS: real part
  - SIN: imaginary part
  - AMPL: ABS(data)
  - PHASE: ARG(data)
  valid abbreviated options

- **MONGO_FILE** (Mongo file name)
  Specify the name of the file to be used in Mongo plotting.

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN

- DATAB
- INFIX
- APPLY
- DE_APPLY
- LOOPS
- DELETE_NODE

- **MODEL_OPTION**
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000
  - BEAM_FACTORS
  - DELETE_LEVEL

The NEWSTAR program NMAP
Making and manipulating maps (WMP-files)

Editor: J.E.Noordam

January 19, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aseries.cookbook]prog1_NMAP.tex

1 The program NMAP

The NEWSTAR program NMAP makes maps (images) from the uv-data in a SCN-file, and stores them in a WMP-file. The input data may be combined in many different ways, to produce maps that range from simple to quite exotic. As a by-product, antenna patterns may be calculated (and stored in the same WMP-file), to be used in NCLEAN.

The program NMAP also allows the user to manipulate maps in a WMP file, and to convert them to other formats.

A WMP-file may contain various kinds of maps that are *related in some way*, i.e. different frequency channels, polarisations, pointing centres (mosaickink), antenna patterns, residuals, and even gridded uv-data. Each item ('Set') in a WMP file (node) consists of a 2-dimensional array of pixel values, and is linked to a map header with auxiliary information. A 'hypercube' of maps in a WMP file can have maps of different sizes in all dimensions. For more information about the structure of the WMP file, see the section on WMP file description.

1.1 Overview of NMAP options

The program NMAP offers the following main options:

- **MAKE:** Make map(s) and/or antenna patterns
- **SHOW:** Show/edit map (header) data. See section on WMP File Description.
- **FIDDLE:** Perform all kinds of operations on maps.
- **W16FITS:** Write FITS tape/disk with 16 bits data.
- **W32FITS:** Write FITS tape/disk with 32 bits data.
- **FROM_OLD:** Convert from old (R-series) format to WMP format.
- **TO_OLD:** Convert from WMP format to old (R-series) format.
- **CVX:** Convert a WMP-file from other machine's format to local machine's.
- **NVS:** Convert a WMP file to newest version. This option should be run if indicated by the program).
- **QUIT:** Quit the program NMAP.

1.2 Option MAKE: Making maps

1.2.1 Types of output maps

The program NMAP produces (multiple) maps of the following types:

- **MAP:** Normal map of the uv-data or the uv-model (the uv-representation of the Selfcal model) from the SCN-file. Various linear combinations of the four measured polarisations can be specified with the keyword **MAP_POLAR**, to produce:
  - **XX, YY, XY or YX-maps:** Use XX, YY, XY or YX data only
  - **I-map:** (XX+YY)/2
  - **Q-map:** (-XX+YY)/2
  - **U-map:** (-XY+YX)/2
  - **V-map:** (XY+YX)*i/2
  - **L-map:** XX or YY or (XX+YY)/2 if both present
  - ***I-map:** any of the above, but multiplied with $\sqrt{-1}$

  NB: Note that parallel dipoles (++) are assumed here. Observations with 'crossed' dipoles (+x) require linear combinations of all four polarisations. This can be done in the map plane (see NMAP option **FIDDLE_OPTION**).
- **AP:** Antenna pattern (Replace uv-data by 1's)
- **COS:** Assume input sines to be zero
- **SIN:** Assume input cosines to be zero
- **AMPL:** Assume input phases to be zero
- **PHASE:** Assume input amplitudes to be one

It is also possible to store **uv-data** in a WMP file, for display purposes. The uv-data is 'gridded' (convolved onto a rectangular grid).

- **COVER:** Gridded uv-coverage (data replaced by 1's)
- **REAL:** Real part of the gridded uv-data
- **IMAG:** Imaginary part of the gridded uv-data
- **AMPL:** Amplitude of the gridded uv-data
- **PHASE:** Phases of the gridded uv-data

1.2.2 Operations on the input data

- **Data selection:** The uv-data that go into a map may be selected in various ways:
  - One or more SCN-files.
  - Sets within a SCN-file (e.g. frequ channels, pointing centres).
  - HA-Scans within each Set (HA-range).
  - Polarisations (XX,YY,XY,YX).
  - Individual iterferometers
  - An area in the uv-plane
  - Clip-level
- **Data correction:** Various correction factors are stored in the Set and Scan headers of the SCN-file. They may be applied to the uv-data at the moment that they are read from disk for processing. Application of correction is controlled by the NGEN keywords **APPLY** and **DE_APPLY** (see the section 'Common Features of NEW-STAR Programs' in this Cookbook).
- **Data conversion:** The uv-data may be converted in various ways:
  - Subtraction of a source model.
  - Combination of different polarisations (XX,YY,XY,YX).
  - Conversion to amplitudes or phases.
- **Data weighting:**
  - NATURAL: Take each individual measured point separately, without weighting for the UV track covered by it.
  - STANDARD: Weight each observed point with the track length covered on the UV plane, and average redundant baselines on a per set basis. *(default)*
  - FULL: Weight each point according to the actual UV point density. In this case care is taken of all local UV plane density enhancements, but it necessitates an extra pass through the data.
- **Data tapering:**
  - GAUSS: $\exp(-baseline^2)$ *(default)*
  - LINEAR: max(0,1-baseline/taper_value)
  - NATURAL: no taper
  - OVERR: $1/baseline$
  - RGAUSS: $\exp(-baseline^2)/baseline$
- **Data convolution:** (onto a rectangular grid in the uv-plane)
  - GAUSS: Gaussian type with 4 × 4 grid points
  - P4ROL: Prolate spheroidal function with 4 × 4 grid points
  - P6ROL: Prolate spheroidal function with 6 × 6 grid points
  - EXPSINC: *sinc* × exp on 6 × 6 grid points *(default)*
  - BOX: A square box

At this point, the data are either stored in the WMP file as gridded uv-data (usually for display purposes), or Fouries transformed into an image.

1.3 Option MAKE: example

The following is an example of making a 'normal' map (and its antenna pattern), using the program defaults. This is usually sufficient to get a satisfactory result. Experienced users may experiment with some of the more advanced options.

```
> exe nmap

    NMAP$1 is started at 10-JAN-93 16:53:14
⊗ OPTION (MAKE,SHOW,FIDDLE,W16FITS,W32FITS,WRLFITS,FROM_OLD, ...) = QUIT:  [make]
⊗ LOOPS (n,incr ....) = "":  [<CR>]

Input data:
⊗ SCAN_NODE (input node name) = "":  [3c147]                    SCN-file
⊗ SETS (sets to do) = "":  [0.0.0.0.0]
⊗ HA_RANGE (HA range) = *:  [<CR>]
All fixed/movable interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  [<CR>]
⊗ SCAN_NODE (input node name) = "":  [<CR>]          if multiple input SCN-files
⊗ USER_COMMENT (map comment) = "":  [redun/align, subtracted 3500,0,0]
                    NB: take the trouble to think of a good descriptive comment!

Map properties:
⊗ UV_COORDINATES (UV,BASHA,IPRHA) = UV:  [<CR>]
⊗ FT_SIZE (FFT size) = 512,512:  [<CR>]
⊗ OUT_SIZE (Output size) = 512,512:  [<CR>]
⊗ FIELD_SIZE (DEG) = 0.6 DEG,0.6 DEG:  [<CR>]
⊗ QMAPS (More map details?) = NO:  [<CR>]                       See 1.3.1 below

Data manipulations:
⊗ QDATAS (More data handling details?) = NO:  [<CR>]            See 1.3.2 below
⊗ SUBTRACT (Source subtraction?) = NO:  [y]
                    If yes, pecify model (not shown here)

1 sources in list
⊗ MODEL_ACTION (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL:  [<CR>]

Output files:
⊗ MAP_POLAR (XX,XY,YX,YY,I,Q,U,V,L,XXI,XYI,YXI,YYI,II,QI,UI, ...) = XX:  [<CR>]
⊗ MAP_COORD (B1950_J2000,APPARENT,REFERENCE,AREFERENCE) = B1950_J2000:  [<CR>]
⊗ OUTPUT (MAP,AP,COVER,REAL,IMAG,AMPL,PHASE) = MAP,AP:  [<CR>]
⊗ OUTPUT_MAP (output node name) = "":  [3c147.sub1]
```

```
Sorting at 16:56:59 (Wall: 00:00:00.00  CPU: 00:00:00.00  I/O: 0  P/F: 0)
Scan node 3C147 started at 16:56:59
Set 0.0.0.0 started at 16:57:25
Scan node 3C147 started at 16:58:11
Set 0.0.0.0 started at 16:58:29
Convolving at 16:59:16 (Wall: 00:02:17.02  CPU: 00:00:38.38  I/O: 1243  P/F: 369
Transposing at 17:00:37 (Wall: 00:03:38.73  CPU: 00:01:05.37  I/O: 1418  P/F: 39

Description of the map produced:

0.0.0.0.0.0(#0) type MAP in node 3C147.SUB1

Field: 3C147                          User comment: REDUN
RA: 84.68127 deg  Dec: 49.82856 deg  Epoch: 1950.0  Frequency: 1401 MHz

RA (1950)     84.68127 deg                          Obs.day          362
Dec(1950)     49.82856 deg                          Obs.year          84
Frequency   1401.39062 MHz                          Epoch        1984.99
Bandwidth     80.00000 MHz                          Map epoch    1950.00
```

```
Type: MAP(XX)                          Size: 512*512     FFT size: 512*512
Fieldsize: 0.5988*0.5988 deg           Grid step: 4.22*4.22 arcsec
                                       Fieldshift: 0.00*0.00 arcsec
Maximum: 27.27 W.U. at 3,3             Minimum: -26.02 W.U.at -4,-2

Input baselines: 80                    Input sets: 2     Input points: 56280
Normalisation: 35920.7                 Noise: 0.577 W.U.

Gaussian taper; Expsinc convolution( corrected); Not clipped;  subtractions; 0 d

0.0.0.0.0.0(#0) type MAP in node 3C147.SUB1

Finished at 17:01:23 (Wall: 00:04:24.66  CPU: 00:01:20.22  I/O: 1548  P/F: 4401)

Description of the antenna pattern produced:

0.0.0.0.1.0(#1) type AP in node 3C147.SUB1

Field: 3C147                           User comment: REDUN
RA: 84.68127 deg  Dec: 49.82856 deg    Epoch: 1950.0  Frequency: 1401 MHz

RA (1950)     84.68127 deg                          Obs.day          362
Dec(1950)     49.82856 deg                          Obs.year          84
Frequency   1401.39062 MHz                          Epoch         1984.99
Bandwidth     80.00000 MHz                          Map epoch     1950.00

Type: AP(XX)                           Size: 512*512     FFT size: 512*512
Fieldsize: 0.5988*0.5988 deg           Grid step: 4.22*4.22 arcsec
                                       Fieldshift: 0.00*0.00 arcsec
Maximum: 1.00 W.U. at 0,0              Minimum: -0.12 W.U.at 3,-1

Input baselines: 80                    Input sets: 2     Input points: 56280
Normalisation: 35920.7                 Noise: 0.000 W.U.

Gaussian taper; Expsinc convolution( corrected); Not clipped;  subtractions; 0 d

0.0.0.0.1.0(#1) type AP in node 3C147.SUB1

Finished at 17:02:05 (Wall: 00:05:06.07  CPU: 00:01:33.77  I/O: 1699  P/F: 4681)

End at 17:02:06 (Wall: 00:05:07.05  CPU: 00:01:33.85  I/O: 1704  P/F: 4707)
      NMAP$1 is ended at 17:02:11              STATUS=SUCCESS
```

1.3.1 QMAPS: Hidden map options

The more advanced map-making options are hidden behind the NMAP keyword **QMAPS**. If skipped, the (context-sensitive) default values will give a satisfactory result in most cases. Their values will be printed in the NMAP log-file. For more information on each of these keywords, see the on-line Help text (type '?'), which is also printed in the 'Summary of NMAP keywords' in this Cookbook.

⊗ **QMAPS** (More map details?) = NO: [y]
⊗ **UNIFORM** (NATURAL,STANDARD,PULL) = STANDARD: [<CR>]
⊗ **TAPER** (GAUSS,LINEAR,NATURAL,OVERR,RGAUSS) = GAUSS: [<CR>]
⊗ **TAPER_VALUE** (M) = 2548 M: [<CR>]
⊗ **CWEIGHT_TYPE** (GAUSS,LINEAR,NATURAL) = NATURAL: [<CR>]
⊗ **CONVOLVE** (GAUSS,BOX,P4ROL,P6ROL,EXPSINC) = EXPSINC: [<CR>]
⊗ **DECONVOLVE** (Correct for convolution?) = YES: [<CR>]

1.3.2 QDATAS: Hidden map options

The more advanced data-selection options are hidden behind the NMAP keyword **QDATA**. If skipped, the (context-sensitive) default values will give a satisfactory result in most cases. Their values will be printed in the NMAP log-file. For more information on each of these keywords, see the on-line Help text (type '?'), which is also printed in the 'Summary of NMAP keywords' in this Cookbook.

⊗ **QDATAS** (More data handling details?) = NO: [y]

(To be added later).

1.4 Option FIDDLE: Operations on maps

The NMAP keyword **FIDDLE_OPTION** offers the user a wide range of possibilities to perform oparations on maps in WMP files:

All relevant data are copied from the first (or only) input map. If 2 maps are required (ADD,AVER,POL,ANGLE) all pairs of SETS_1 and SETS_2 will produce an output map. SUM will average all SETS_1 maps. MOSCOM will produce a single output map from all specified input maps. The others will produce an output for each SETS_1. BEAM, DEBEAM and FACTOR will overwrite the input maps. F is a specified multiplication factor.

- **ADD:** $F1 \times map1 + F2 \times map2$
- **AVER:** $(F1 \times map1 + F2 \times map2)/(abs(F1) + abs(F2))$
- **SUM:** Various averages of map's. In all cases the summation produces a weighted average map over all SETS_1, the weight depending on the type:
  - **SUM:** weight(i)= 1
  - **NSUM:** weight(i)= normalisation factor of map
  - **BSUM:** weight(i)= bandwidth
  - **BNSUM:** weight(i)= bandwidth × normalisation factor
  - **FSUM:** weight(i)= factor given by the user. Up to 8 factors can be given, which will be used in a cyclic fashion if more needed
  - **NSSUM:** weight(i)= $1/(mapnoise^2)$
  - **QUIT:** finished
- **POL:** $\sqrt{map1^2 + map2^2}$, unless $< F1$, then 0
- **ANGLE:** 0.5 × arctan $(map1/map2)$ (radians), unless $POL < F1$, then 0
- **EXTRACT:** extract an area of map's
- **COPY:** copy map's
- **BEAM:** correct map's for primary beam
- **DEBEAM:** de-correct map's for primary beam
- **FACTOR:** $F1 \times map$
- **MOSCOM:** combine all specified maps (referenced to same mosaic position) into one output map. The noise of the individual maps may be used as weight.
- **QUIT:** quit the fiddle option.

1.5 Option FITS: Conversion to FITS format

There are two separate NMAP options (**W16FITS** and **W32FITS**.

Summary of the keywords (parameters) of the program NMAP Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.nseries.cookbook]prog3_NMAP.tex

1 NMAP parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **OPTION** (MAKE,SHOW,FIDDLE,W16FITS,W32FITS,FROM_OLD,TO_OLD,CVX,NVS,QUIT)

  Specify action to perform:
  - MAKE: make map(s)
  - SHOW: show/edit map data
  - FIDDLE: combine or change maps
  - W16FITS: write FITS tape/disk with 16 bits data
  - W32FITS: write FITS tape/disk with 32 bits data
  - FROM_OLD: convert from old format
  - TO_OLD: convert to old format
  - CVX: convert a map file from other machine's format to local
  - : machine's
  - NVS: convert a map file to newest version. Should be run if indicated
  - : by program
  - QUIT: finish

- **FIDDLE_OPTION**
  Specify action to perform. All relevant data are copied from the first (or only) input map. If 2 maps required (ADD,AVER,POL,ANGLE) all pairs of SETS_1 and SETS_2 will produce an output map. SUM will average all SETS_1 maps. MOSCOM will produce a single output map from all specified input maps. The others will produce an output for each SETS_1. BEAM, DEBEAM and FACTOR will overwrite the input maps. F is a specified factor.
  - ADD: F1*map1 + F2*map2
  - AVER: (F1*map1 + F2*map2)/(abs(F1)+abs(F2))
  - SUM: Various averages of map1's
  - POL: sqrt(map1**2 + map2**2), unless < F1, then 0
  - ANGLE: 0.5*atan(map1/map2) (radians) unless POL < F1, then 0
  - EXTRACT: extract an area of map1's
  - COPY: copy map1's
  - BEAM: correct map1's for primary beam
  - DEBEAM: de-correct map1's for primary beam
  - FACTOR: F1*map1
  - MOSCOM: combine all specified maps (referenced to same mosaic position)
  - : into one output map
  - QUIT: finish

- **USE_NOISE** (Weight with noise?)
  Specify if you want to use the noise of the individual maps to be used as a weight in the MOSCOM combination

- **SUM_OPTION** (SUM,NSUM,BSUM,BNSUM,FSUM,NSSUM,QUIT)
  Specify the type of summation to perform. In all cases the summation produces a weighted average map over all SETS_1, the weight depending on the type:
  - SUM: weight(i)= 1
  - NSUM: weight(i)= normalisation factor of map
  - BSUM: weight(i)= bandwidth
  - BNSUM: weight(i)= bandwidth*normalisation factor
  - FSUM: weight(i)= given factor. Up to 8 factors can be given,
  - : which will be used in a cyclic fashion if more needed
  - NSSUM: weight(i)= 1/(map noise **2)
  - QUIT: finish

- **MAP_FACTORS** (Map factors)
  Specify the factors by which the input maps have to be multiplied.

- **SUM_FACTORS** (Sum factors)
  Specify the factors by which the input maps have to be multiplied. Up to 8 can be given. If more needed they will used in a cyclic fashion. E.g. 1,-1 will take the average of the difference of the odd and even specified sets.

- **MAP_LEVEL** (Pol. level)
  Specify the minimum level that is still to be considered valid polarisation. If polarisation is less than this level it is assumed to be zero.

- **COMMENT** (Tape comment)
  The given text will be included as COMMENT in FITS output.

- **FITS_SCALE** (JY,WU)
  Specify the output units of the FITS data: Jy/beam or W.U.

- **QMAPS** (More map details?)
  Specify if you want to specify more details of the map (else defaults taken)

- **QDATAS** (More data handling details?)
  Specify if you want to specify more details of the data handling (else defaults taken)

- **INPUT_FILE** (input file name)
  Specify the full file name of the file to be converted.

- **FILENAME** (filename)
  Specify the file name without an extension to be used in creating an output file name.

- **OUTPUT_LABEL** (output label)
  Specify the first output tape label. * or 0 indicates at the end of the tape.

- **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT)
  Specify action to perform:
  - LAYOUT: give layout of file
  - SHOW: display all fields in file header
  - EDIT: edit fields in file header
  - CONT: continue with set headers
  - QUIT: finish this node

- **MAP_ACTION** (NEXT,SHOW,EDIT,CONT,QUIT)
  Specify action to perform:
  - NEXT: do next specified map
  - SHOW: display all fields in map header
  - EDIT: edit fields in map header
  - CONT: continue with map data
  - QUIT: finish this map

- **DATA_ACTION** (S,N,O,Q)
  Specify action to perform:
  - S[how]: show detailed map data
  - N[oise]: calculate noise
  O[ffset] calculate noise and offset
  - Q[uit]: quit data part

- **USER_COMMENT** (map comment)
  Give, optionally, a decsriptive comment for the maps.

- **MAP_POLAR** (XX,XY,YX,YY,I,Q,U,V,L,XXI,XYI,YXI,YYI,II,QI,UI,VI,LI)
  (polarisation info) Specify up to four polarisations of the maps to make:
  - XX: XX only
  - YY: YY only
  - XY: XY only

- YX: YX only
- I: (XX+YY)/2
- Q: (-XX+YY)/2
- U: (-XY+YX)/2
- V: (XY+YX)*i/2
- L: XX or YY or (XX+YY)/2 if both present
- *I: same as above, but multiplied with sqrt(-1)

- **MAP_COORD** (B1950,APPARENT,REFERENCE,AREFERENCE) (type of map coord.)
  Specify the coordinates of the map:
  - B1950: in 1950.0 coordinates
  - APPAR: in apparent coordinates
  - REFER: use a 1950 reference coordinate (e.g. the centre of mozaic area)
  - AREFER: same, but apparent

- **REF_COORD** (reference RA,DEC)
  Specify (in degrees) the RA and DEC of the reference coordinates to use in producing the map.

- **UV_COORDINATES** (UV,BASHA,IFRHA) (coordinate type)
  Specify the type of UV coordinates wanted for UV plane type output (COVER etc.)
  - UV: standard UV coordinates
  - BASHA: baseline (as U) vs hour-angle
  - IFRHA: IFR number (01,02,...,0D,12,13,...,CD) as U vs hour-angle

- **HA_RESOLUTION** (DEG) (HA bin width)
  Specify the width of one HA UV point for BASHA and IFRHA.
  - Note:: Points are separated by multiples of 10 UT seconds, not ST

- **BAS_RESOLUTION** (baseline bin in m)
  Specify the width of one UV point in meters for BASHA

- **IFR_RESOLUTION** (interferometer separation)
  Specify the separation in grid points between interferometers for IFRHA

- **USER_DATA** (STANDARD,MODEL) (Data to use)
  Specify the data to use in the program:
  - STANDARD: use the observed cosines/sines
  - MODEL: use the source model as defined later with type=0 sources

- **FT_SIZE** (FFT size)
  Specify the size of the Fourier transform in the l and m direction. If the size is <= 17, a DFT in stead of an FFT will be done.

- **OUT_SIZE** (Output size)
  Specify the size of the output map(s) in the l and m direction.

- **OUT_CENTRE** (Output centre)
  Specify the centre of the output map in the l and m direction; in pixels with respect to the mosaic reference position. Specifying an * will ask for l,m and RA,DEC position

- **LM_CENTRE** (Output centre)
  Specify the centre of the output map in the l and m direction; in arcsec with respect to the mosaic reference position. Specifying an * will ask for RA,DEC position

- **RADEC_CENTRE** (Output centre)
  Specify the centre of the output map in RA and DEC ; in degrees.

- **FIELD_SIZE** (DEG) (Fieldsize)
  Specify the l and m field size of the map to be transformed.

- **UNIFORM** (NATURAL,STANDARD,FULL) (Uniform coverage)
  Specify the way the UV coverage should be determined: NATURAL: Take each individual measured point separately, without weighting for the UV track covered by it STANDARD: Weight each observed point with the track length covered on the UV plane, and average redundant baselines on a per set basis FULL: Weight each point according to the actual UV point density. In this case care is taken of all local UV plane density enhancements, but it necessitates an extra pass through the data.

- **TAPER** (GAUSS,LINEAR,NATURAL,OVERR,RGAUSS) (Taper type)
  Specify the type of taper to apply:
  - GAUSS: exp(-baseline**2)
  - LINEAR: max(0,1-baseline/taper_value)
  - NATURAL: no taper
  - OVERR: 1/baseline
  - RGAUSS: exp(-baseline**2)/baseline

- **CWEIGHT_TYPE** (GAUSS,LINEAR,NATURAL) (Circular weight type)
  Specify the type of circular weight to apply:
  - GAUSS: exp(-baseline**2)
  - LINEAR: max(0,1-baseline/taper_value)
  - NATURAL: no weight

- **TAPER_VALUE** (M) (Taper width)
  Specify the half-width of the taper function. The default gives 25% taper at 3000 m.
  Default value(s): 2548.

- **CWEIGHT_VALUE** (M) (Circ. weight width)
  Specify the half-width of the circular weight function. The default gives 25% taper at 3000 m.
  Default value(s): 2548.

- **CONVOLVE** (GAUSS,BOX,P4ROL,P6ROL,EXPSINC) (Convolution type)
  Specify the type of convolution in the UV plane:
  - GAUSS: Gaussian type with 4*4 grid points
  - P4ROL: Prolate spheroidal function with 4*4 grid points
  - P6ROL: Prolate spheroidal function with 6*6 grid points
  - EXPSINC: Sinc*exp on 6*6 grid points
  - BOX: A square box

- **CONVOL_WIDTH** (Convolution width)
  Specify the width of the convolution function in the l and m coordinate direction.

- **DECONVOLVE** (Correct for convolution?)
  Specify if you want to correct for the convolution tapering in your map/ap.

- **UV_AREA** (M) (Select UV area)
  Specify the (circular) UV plane radii (in m) between which you want to include the data

- **CLIPPING** (Clipping?)
  Specify YES if you want to clip data, i.e. if for a certain part of the UV plane you want to discard data between two amplitude levels, e.g. to suppress the influence of interference

- **CLIP_AREA** (M)
  Specify the baselines (in m) between which clipping has to occur

- **CLIP_LEVELS**
  Specify the levels (in W.U.) of the amplitude range you want to discard

- **SUBTRACT** (Source subtraction?)
  Specify YES if you want to subtract sources

- **FIELD_SHIFT** (Field shift)
  Specify (in arcsec) the field centre shift wanted for the l and m coordinates
  Default value(s): 0.,0. /ASK

- **DATA_TYPE** (NORMAL,COS,SIN,AMPL,PHASE) (Use of data)
  Specify the use of the input data:
  - NORMAL: Use the complex input data
  - COS: Assume input sines to be zero
  - SIN: Assume input cosines to be zero
  - AMPL: Assume input phases to be zero
  - PHASE: Assume input amplitudes to be one

  valid abbreviated options

- **OUTPUT** (MAP,AP,COVER,REAL,IMAG,AMPL,PHASE) (Output types)
  Specify one or more output types:
  - MAP: Produce an output map
  - AP: Produce an antenna pattern
  - COVER: Produce the 'antenna-pattern' convolved UV plane
  - REAL: Produce the real part of the convolved UV plane
  - IMAG: Produce the imaginary part of the convolved UV plane
  - AMPL: Produce the amplitude part of the convolved UV plane
  - PHASE: Produce the convolved UV plane in phase format

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

- **MODEL_OPTION**
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000
  - BEAM_FACTORS
  - DELETE_LEVEL

Description of the program NCLEAN
Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.useries.cookbook]prog1_NCLEAN.tex

1 The program NCLEAN

This program is used to CLEAN NEWSTAR maps in NEWSTAR .WMP files.

1.1 Overview of NCLEAN options

- **HISTO:** Produce only a histogram of the map and/or beam.
- **BEAM:** Use the map and the beam to clean. Only a limited part, depending on the workarea size, can be cleaned.
- **UVCOVER:** Use the map and the beam transform to clean.
- **COMPON:** Get highest clean components in map. The clean level will depend on the beam shape, the type of map data and the memory size.
- **UREST:** Use a CLEAN component list and a map to restore the clean components in the map.
- **QUIT:** leave the program NCLEAN.

1.2 ...

Summary of the keywords (parameters) of the program NCLEAN Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.nseries.cookbook]prog3_NCLEAN.tex

1 NCLEAN parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **OPTION** (HISTO,BEAM,UVCOVER,COMPON,UREST,QUIT) (Clean option)
  Specify the type of clean to do:
  - HISTO: Produce only a histogram of the map and/or beam
  - BEAM: Use the map and the beam to clean. Only a limited part, depending
  - : on the workarea size, can be cleaned
  - UVCOVER: Use the map and the beam transform to clean
  - COMPON: Get highest clean components in map. The clean level will depend
  - : on the beam shape, the type of map data and the memory size.
  - UREST: Use a clean component list and a map to restore the clean
  - : components in the map.
  - QUIT: leave program.
- **BEAM_TYPE** (FULL,PATCH) (Beam action option)
  Specify the type of clean to do:
  - PATCH: Use in the clean a contiguous patch area of the beam. This produces
  - : a faster minor cycle, but maybe more major cycles if there are
  - : grating rings present
  - FULL: Use in the clean all points in the beam above a certain level.
- **DECONVOLUTION** (Play with deconvolution?)
  YES will in general produce a result with less aliasing, NO the reverse.
  Default value(s): NO /NOASK
- **COMPON_LOG** (Type,print components found interval)
  Specify if you want typed on terminal and/or printed a review of the clean components found. You can specify an interval step, e.g. 2,20 specifies that every 2nd component found will be typed, and every 20th printed in the log. A value of 0 suppresses the listing.
- **CYCLE_DEPTH** (Major cycle depth)
  Specify the level to which you want to clean in one major cycle.
  Default value(s): .05 /ASK
- **MAP_FACTOR** (Multiplication factor)
  Specify the factor by which to multiply the input map before restoration takes place (e.g. 0).
  Default value(s): 1. /ASK
- **CLEAN_LIMIT** (Limit in fraction of map max.)
  Specify the level to which to clean in fraction of the map maximum.
  Default value(s): .1 /ASK
- **COMPON_LIMIT** (Max. number of components to find)
  Specify the maximum number of components that are allowed to be found
  Default value(s): 100 /ASK
- **LOOP_GAIN** (Clean loop factor)
  Specify the clean loop factor
  Default value(s): .4 /ASK
- **GRATING_FACTOR** (Grating correction factor)
  The clean method used can find sources on the intersections of grating rings occasionally before the source itself. This can be suppressed by specifying a correction factor. 1 means that no correction is necesary, 2 e.g. that you expect intersections of grating rings of sources with equal intensity.

- **PRUSSIAN_HAT** (Prussian hat value)
  For extended sources a prussian hat (i.e. a nominal additional peak value on the central value of the antenna pattern, see Cornwell) may give better clean results. Values of .1 to .4 coulde be tried.

- **RESIDUAL** (residual map?)
  Specify if a residual map should be output after the clean cycle.

- **RESTORE** (restored map?)
  Specify if a restored map should be output after the clean cycle.

- **DEFAULT_BEAM** (use default restore beam)
  Specify if the default restore beam can be used or not. A default beam will not be skew. It will be calculated from the antenna pattern, if present and given, or be taken as 12*1400/(frequency in MHz) arcsec.

- **RESTORE_BEAM** (dl, dm arcsec, pa deg)
  Specify the restore beam width:
  - dl: width of beam in arcsec (full-halfwidth)
  - dm: width of beam in arcsec (full-halfwidth)
  - pa: position angle of skewed beam in degrees
  - : (anti-clock, 0 deg: dl along l)

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

- **MODEL_OPTION**
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400
  - INPOLQ,U,V_1000
  - INPOLQ,U,V_2000
  - INPOLQ,U,V_4000
  - INPOLQ,U,V_10000
  - INPOLQ,U,V_100000
  - BEAM_FACTORS
  - DELETE_LEVEL

The NEWSTAR program NPLOT
Plotting uv-data, maps and uv-models

Editor: J.E.Noordam

January 20, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

SPECTRAL .

file user6:[noordam.nseries.cookbook]prog1_NPLOT.tex

1 The program NPLOT

1.1 Overview of NPLOT options

The program NPLOT has the following main options:

- **MAP:** Plot map(s) of various types (from WMP-file)
- **DATA:** Plot uv-data from SCN-file
- **MODEL:** Plot uv-model from SCN-file (i.e. the uv-representation of a Selfcal model, calculated for the uv-coordinates of the uv-data in the SCN-file)
- **TELESCOPE:** Plot telescope errors
- **RESIDUAL:** Plot interferometer residuals (i.e. the residual scatter of redundant spacing data after a Redundancy solution, or the residual differences with the source model after a Selfcal solution.
- **QUIT:** Quit the program NPLOT

1.2 NPLOT output

After selecting an NPLOT option, the **PLOTTER** question is asked:

⊗ **PLOTTER** (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = QMS: [?]

Produces QMS format files, and prints them automatically (if you are lucky) on the old (and dying) QMS laser printer in Dwingeloo computer room:

```
QMS    QMS laser printer in landscape orientation
QMSP   QMS laser printer in portrait orientation
```

Produces output on VT terminal:

```
REGIS  graphics VT terminal (e.g. VT330)
FREGIS (*) REGIS to file
```

Produces PostScript files on disk (which may fill up quickly!):

```
EPS    encapsulated PostScript for use in textprocessors etc
EPP    EPS in portrait mode
EAL    encapsulated PostScript A3 plotter landscape
EAP    EPS A3 in portrait mode
```

Produces PostScript files, and prints them automatically (if you are lucky) on the New PostScript laser printer in the Dwingeloo computer room:

```
PSL    Postscript (do not use halftone: slow!)
PSP    PostScript in portrait mode
PAL    Postscript A3 (do not use halftone: slow!)
PAP    PostScript A3 in portrait mode
```

Bit maps:

```
BIT1   (*) bitmap for 100 dpi
BIT2   (*) bitmap for 200 dpi
BIT3   (*) bitmap for 300 bpi
```

Produces a X11-plot on the Workstation (or X-terminal) screen.

```
X11    X11 terminal (partly available, do not use halftone !!)
```

(*) = not implemented yet.
All plot file names start with the 3-4 letters of the selected option (e.g. EPS), followed by a unique combination of characters based on the time and the date. All plot files have the extension .PLT.

1.3 Plotting a map from a WMP file

NPLOT option MAP can be used to make **contour plots** or **greyscale plots** of 2-dimensional pixel-arrays stored in a WMP-file. These may be maps, antenna-patterns, residuals etc, but also gridded uv-data (see the description of the program NMAP, and the WMP file). They can also be displayed (and analysed) as a color image on the X11 screen with the program NGIDS.

```
>   exe nplot
     NPLOT$1 is started at 10-JAN-93 17:03:03
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: [map]
⊗ PLOTTER (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = EPS: [psp]
⊗ MAP_NODE (input node name) = "3C147.SUB1": [<CR>]                    WMP file
⊗ LOOPS (n,incr ....) = "": [<CR>]
⊗ SETS (sets to do) = "0.0.0.0.0.0": [<CR>]
Set: 0.0.0.0.0.0
⊗ PLOT_TYPE (CONT,HALF,POL,RULE) = CONT,HALF: [c]                      Contour plot
⊗ DATA_TYPE (DATA,SLOPE) = DATA: [<CR>]
⊗ AREA (Area l,m,dl,dm) = 0,0,512,512: [0,0,128,128]

Area(s) selected:
Total    : l= 0, m= 0, dl= 128, dm= 128

Noise= 0.7464 W.U.

Range: -26.02485, 27.27408
⊗ SIZE (plot size) = 1,1: [1.3,1.3]                                    Fits on A4 sheet
⊗ FULL_CONT (contour values) = "": [2,4,6,8,10,20]                     full contours
⊗ DOT_CONT (contour values) = "": [-2,-4,-6,-8,-10,-20]                dotted contours
⊗ COORD (NONE,DLM,LM,DRADEC,RADEC,DDEGREE,DEGREE) = NONE: [lm]
⊗ COORD_TYPE (TICK,DOTTED,FULL) = TICK: [<CR>]
⊗ PLOT_POSITIONS (NO,YES) = NO: [y]                                    of model sources
⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT) [read]
                                                                       specify model (not shown here)

1 sources in list
```

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

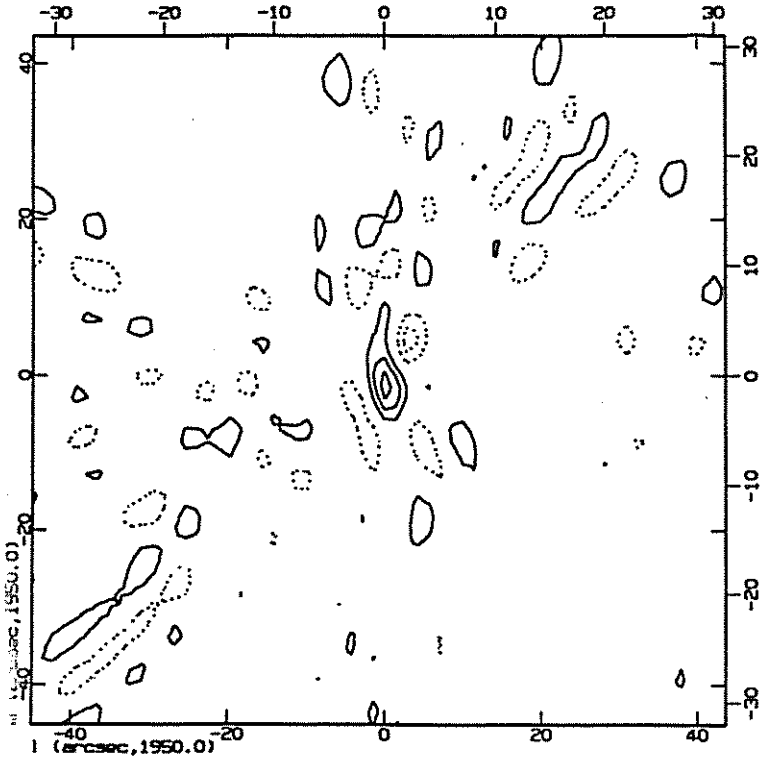
```
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: [<CR>]
     NPLOT$1 is ended at 17:07:01                    STATUS=SUCCESS
```

It is also possible to make 'ruled-surface' plots, halftone plots and plots in polar coordinates (gridded uv-data).

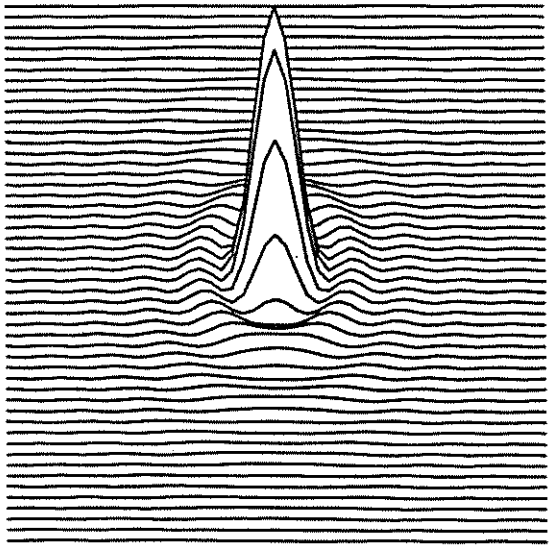
20-Jan-1993 17:06 MAP (DATA) 0.0/0.0

Obs: SUB16 File: SUB16.MAP
Sel: 80 Field: MAP48 Polarisation: I

Full contours: 0.05000, 0.1000, 0.1500
Dotted contours: -0.05000, -0.1000



28-Jan-1992 09:49 Map Ja



Example of a ruled-surface plot option MAP, PLOT_TYPE : Rule

1.4 Plotting uv-data from a SCN-file

1.4.1 NORMAL: interferometers vs time (HA)

```
>  exe nplot
    NPLOT$1 is started at 10-JAN-93 16:29:32
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  data
⊗ PLOTTER (QMS,QMSP,REGIS,PREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = EPS:  psl
⊗ PLUVO (PLUVO,NORMAL) = NORMAL:  <CR>                          ifrs vs time
⊗ SCAN_NODE (input node name) = "":  3c147                      SCN-file
⊗ LOOPS (n,incr ....) = "":  <CR>
⊗ SETS (sets to do) = "":  0.0.0.0.0
⊗ HA_RANGE (HA range) = *:  <CR>
⊗ POLARISATION (XYX,XY,YX,Y,X) = XY:  <CR>                      XX and YY
All cross interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  <CR>
⊗ DATA_TYPES (AMPLITUDE,PHASE,COSINE,SINE) = AMPLITUDE,PHASE:  <CR>
⊗ SCALE_AMPL (plot scale W.U./mm or %/mm) = 100:  <CR>
⊗ SCALE_PHASE (plot scale in W.U./mm or deg./mm) = 10:  <CR>
⊗ HA_SCALE (HA plot scale degree/cm) = 15:  <CR>                 12 hrs fit on A4 plot
```

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

```
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  <CR>
    NPLOT$1 is ended at 16:34:14                    STATUS=SUCCESS
```

1.4.2 PLUVO: frequ channels vs time (HA)

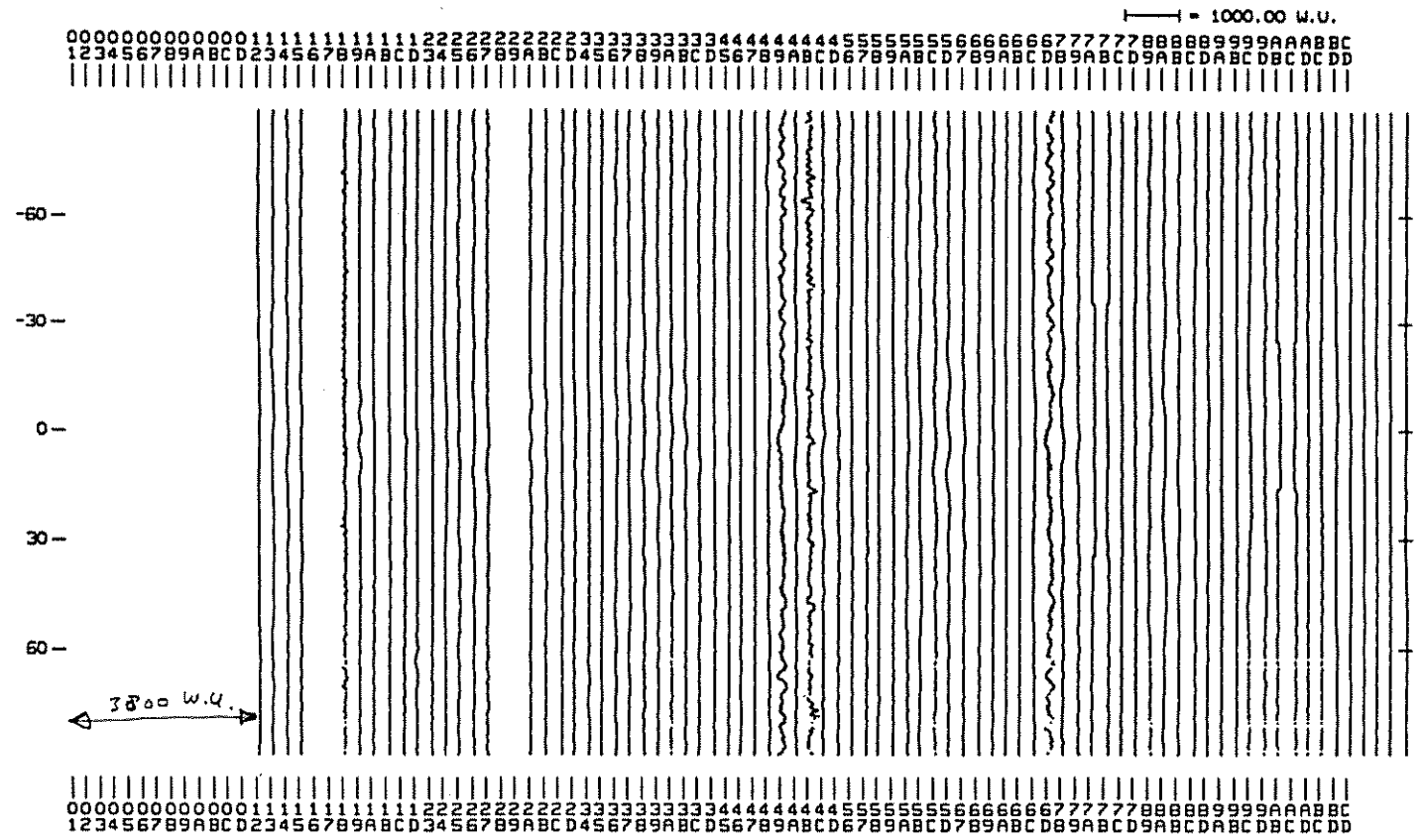
```
>  exe nplot
    NPLOT$1 is started at 10-JAN-93 16:29:32
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  data
⊗ PLOTTER (QMS,QMSP,REGIS,PREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = EPS:  psl
⊗ PLUVO (PLUVO,NORMAL) = NORMAL:  pluvo
```

Etc...

10-Jan-1993 16:30 AMPLITUDE (R+A+O) data 0.0/0.0

Node: 3C147
Set range: 0.0.0.0.0 - 0.0.0.0.0

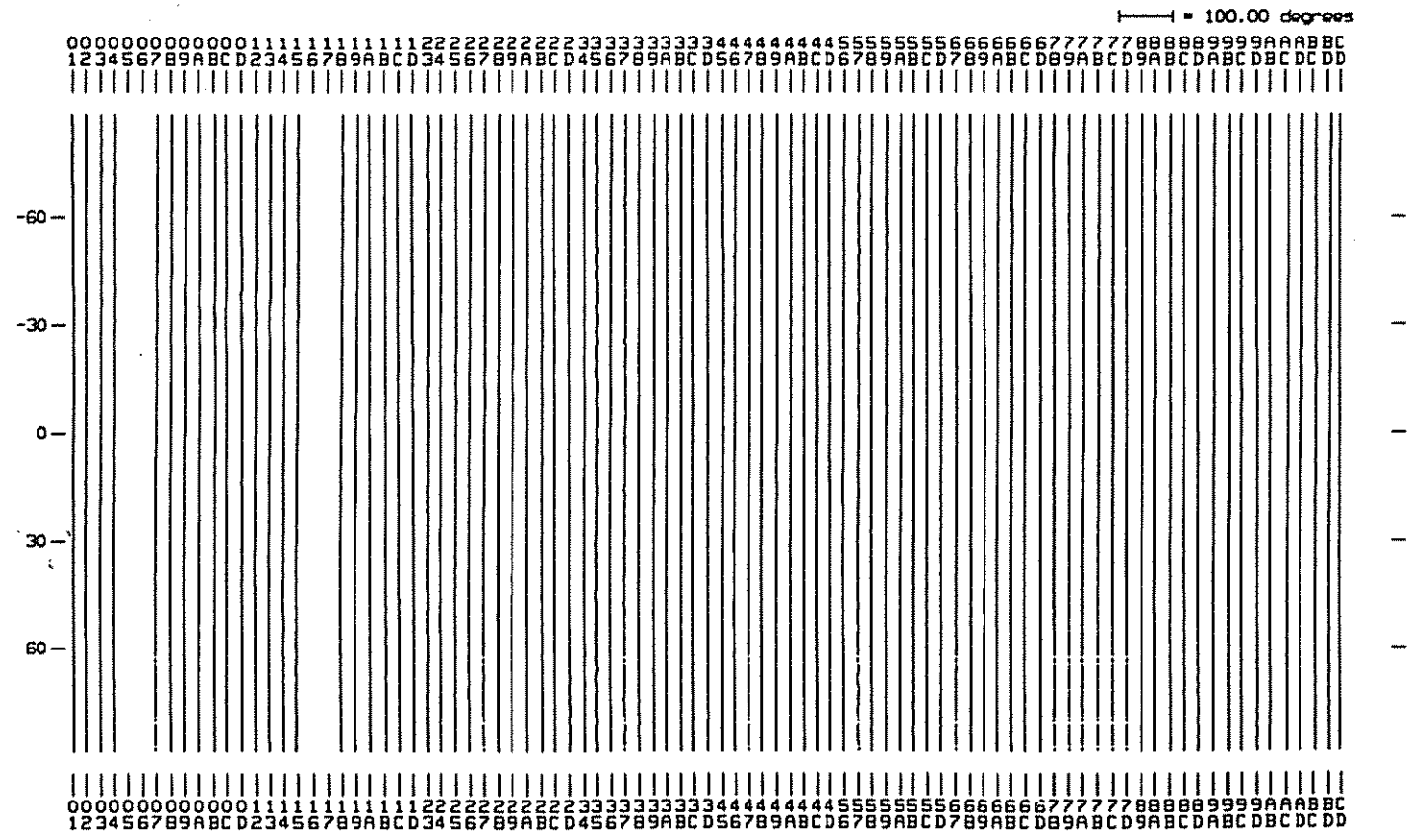
File: 3C147.SCN
Field: 3C147 Polarisation: XX



10-Jan-1993 16:40 PHASE (R+A+O) data 0.0/0.0

Node: 3C147
Set range: 0.0.0.0.0 - 0.0.0.0.0

File: 3C147.SCN
Field: 3C147 Polarisation: XX



1.5 Plotting a uv-model from a SCN-file

The same as plotting uv-data (see 1.4).

⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: `model`
⊗ **PLOTTER** (QMS,QMSP,REGIS,PREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = PSL: `psp`
⊗ **PLUVO** (PLUVO,NORMAL) = NORMAL: `<CR>` *ifrs vs time*
⊗ **SCAN_NODE** (Input node name) = 3C147: `<CR>` *SCN-file*
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = "0.0.0.0.0": `<CR>`
⊗ **HA_RANGE** (HA range) = *: `<CR>`
⊗ **POLARISATION** (XYX,XY,YX,Y,X) = X: `<CR>`
⊗ **MODEL_OPTION** (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT) `read`
⊗ **MODEL_NODE** (Model node) = "": `3c147` *MDL file*
rest of model specification not shown here

```
2 sources in list
```

⊗ **MODEL_ACTION** (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL: `<CR>`

```
All fixed/movable interferometers pre-selected
```

⊗ **SELECT_IFRS** (Select/deselect ifrs) = "": `<CR>`
⊗ **DATA_TYPES** (AMPLITUDE,PHASE,COSINE,SINE) = AMPLITUDE,PHASE: `<CR>`
⊗ **SCALE_AMPL** (plot scale W.U./mm or %/mm) = 100: `<CR>`
⊗ **SCALE_PHASE** (plot scale in W.U./mm or deg./mm) = 10: `<CR>`
⊗ **HA_SCALE** (HA plot scale degree/cm) = 15: `<CR>`

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: `<CR>`

```
NPLOT$1 is ended at 15:43:25                STATUS=SUCCESS
```

1.6 Plotting telescope corrections

The telescope corrections are stored in the HA-Scan headers of the SCN-file. Plotted are the **total** telescope corrections: Redundancy (REDC) plus Align (ALGC) plus 'other' (OTHC).

1.6.1 NORMAL: telescopes vs time (HA)

⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: `tel`
⊗ **PLOTTER** (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = PSL: `<CR>`
⊗ **PLUVO** (PLUVO,NORMAL) = NORMAL: `<CR>`
⊗ **SCAN_NODE** (input node name) = 3C147: `<CR>`
⊗ **LOOPS** (n,incr ....) = "": `<CR>`
⊗ **SETS** (sets to do) = "0.0.0.0.0": `<CR>`
⊗ **HA_RANGE** (HA range) = *: `<CR>`
⊗ **POLARISATION** (XYX,XY,YX,Y,X) = X: `<CR>`
⊗ **TELESCOPES** (Telescope(s) = *: `<CR>`
⊗ **DATA_TYPES** (AMPLITUDE,PHASE,COSINE,SINE) = AMPLITUDE,PHASE: `<CR>`
⊗ **SCALE_AMPL** (plot scale W.U./mm or %/mm) = 4: `<CR>` *here: %/mm*
⊗ **SCALE_PHASE** (plot scale in W.U./mm or deg./mm) = 2: `20` *here: degr/mm*
⊗ **HA_SCALE** (HA plot scale degree/cm) = 15: `<CR>`

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

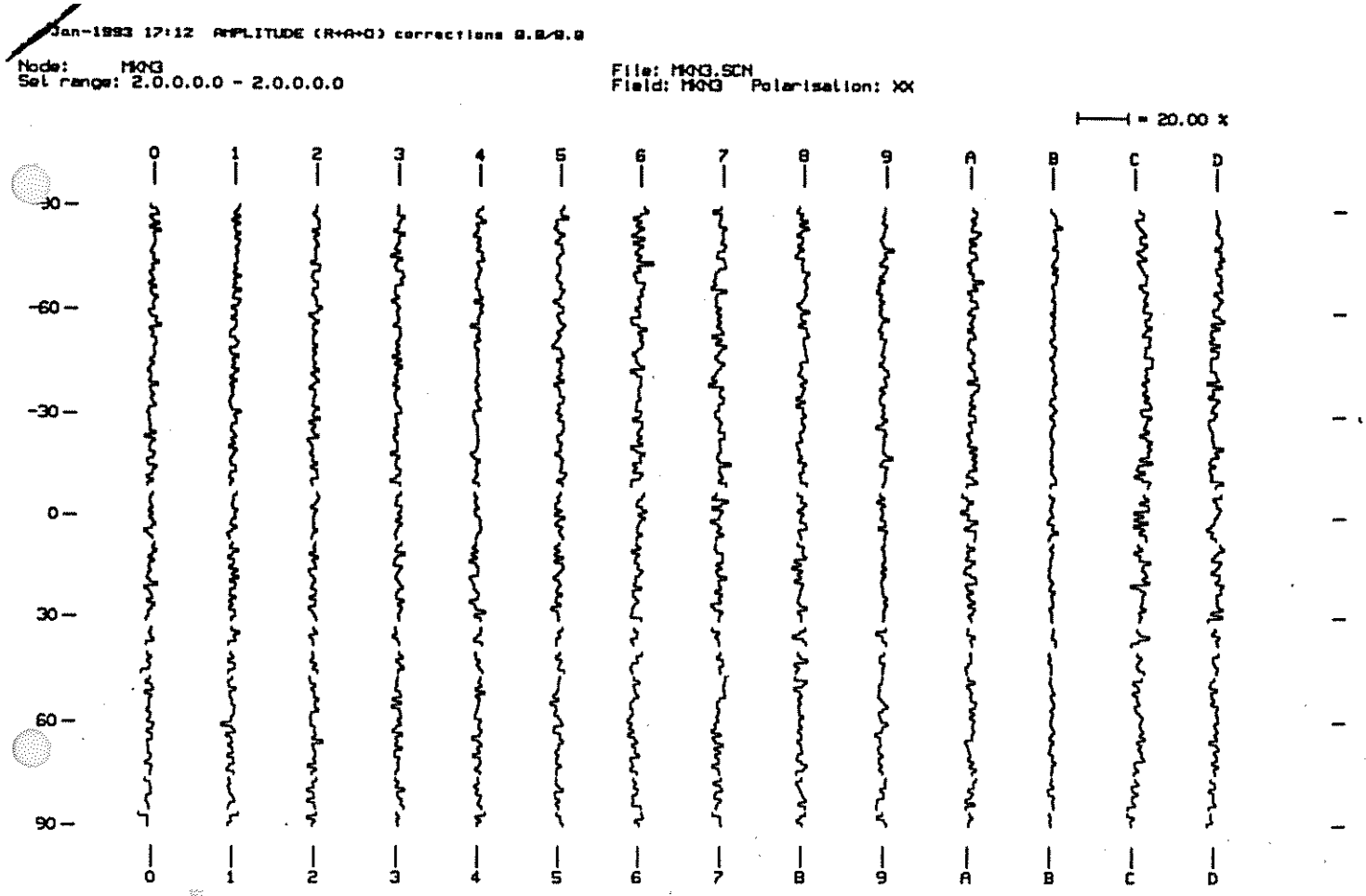
⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: `<CR>`

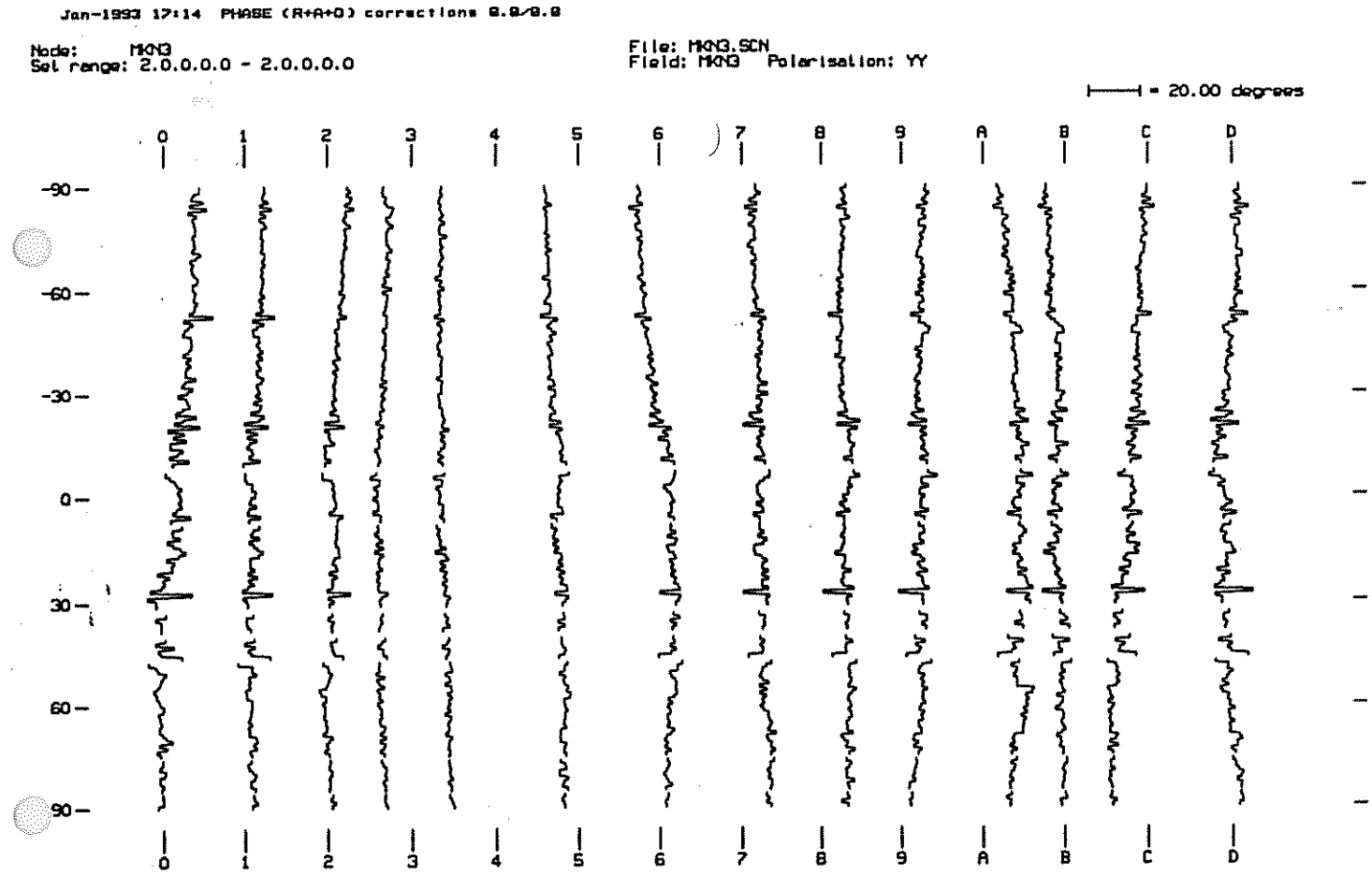
```
NPLOT$1 is ended at 16:21:18                    STATUS=ERROR
```

1.6.2 PLUVO: frequ channels vs time (HA)

⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: `tel`
⊗ **PLOTTER** (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = PSL: `<CR>`
⊗ **PLUVO** (PLUVO,NORMAL) = NORMAL: `pluvo`

Etc...


Jan-1993 17:12 AMPLITUDE (R+A+O) corrections 0.0/0.0
Node: MKN3
Set range: 2.0.0.0.0 - 2.0.0.0.0
File: MKN3.SCN
Field: MKN3 Polarisation: XX
= 20.00 x
90
60
30
0
-30
-60
-90
0 1 2 3 4 5 6 7 8 9 A B C D


Jan-1993 17:14 PHASE (R+A+O) corrections 0.0/0.0
Node: MKN3
Set range: 2.0.0.0.0 - 2.0.0.0.0
File: MKN3.SCN
Field: MKN3 Polarisation: YY
= 20.00 degrees
90
60
30
0
-30
-60
-90
0 1 2 3 4 5 6 7 8 9 A B C D

1.7 Plotting NCALIB residuals from SCN-file

The residuals are calculated by going though the SCN-file and subtracting a uv-model from the uv-data (Selfcal rediduals), or the average of redundant spacings (Redundancy residuals).

1.7.1 Redundancy residuals

```
>  exe nplot
    NPLOT$1 is started at 10-JAN-93 16:11:39
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  res
⊗ PLOTTER (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = EPS:  psl
⊗ PLUVO (PLUVO,NORMAL) = NORMAL:  <CR>                          ifrs vs time (HA)
⊗ SCAN_NODE (input node name) = "":  3c147                      SCN-file
⊗ LOOPS (n,incr ....) = "":  <CR>
⊗ SETS (sets to do) = "":  0.0.0.0.0
⊗ HA_RANGE (HA range) = *:  <CR>
⊗ POLARISATION (XYX,XY,YX,Y,X) = XY:  <CR>                      XX and YY
⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT)  <CR>
                                                  No model: Redundancy selected

0 sources in list

Redundancy residuals selected
All cross interferometers pre-selected
⊗ SELECT_IFRS (Select/deselect ifrs) = "":  <CR>
⊗ DATA_TYPES (AMPLITUDE,PHASE,COSINE,SINE) = AMPLITUDE,PHASE:  <CR>
⊗ SCALE_AMPL (plot scale W.U./mm or %/mm) = 65.87241:  <CR>      Here: W.U./mm
⊗ SCALE_PHASE (plot scale in W.U./mm or deg./mm) = 112.7789:  <CR>  Here: W.U./mm
⊗ HA_SCALE (HA plot scale degree/cm) = 15:  <CR>                 12 h fits on A4 plot
```

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

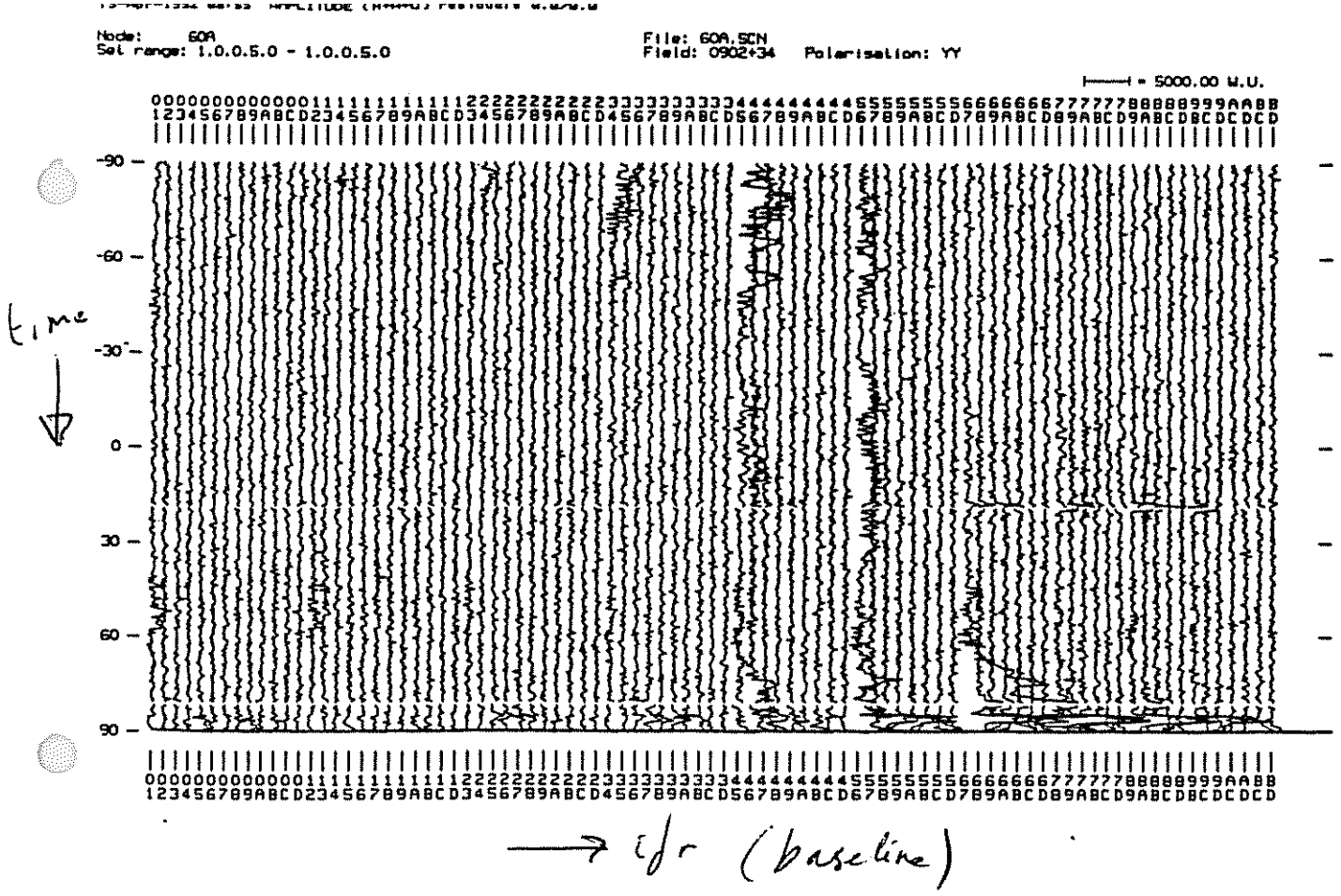
```
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  <CR>
    NPLOT$1 is ended at 17:07:01                    STATUS=SUCCESS
```

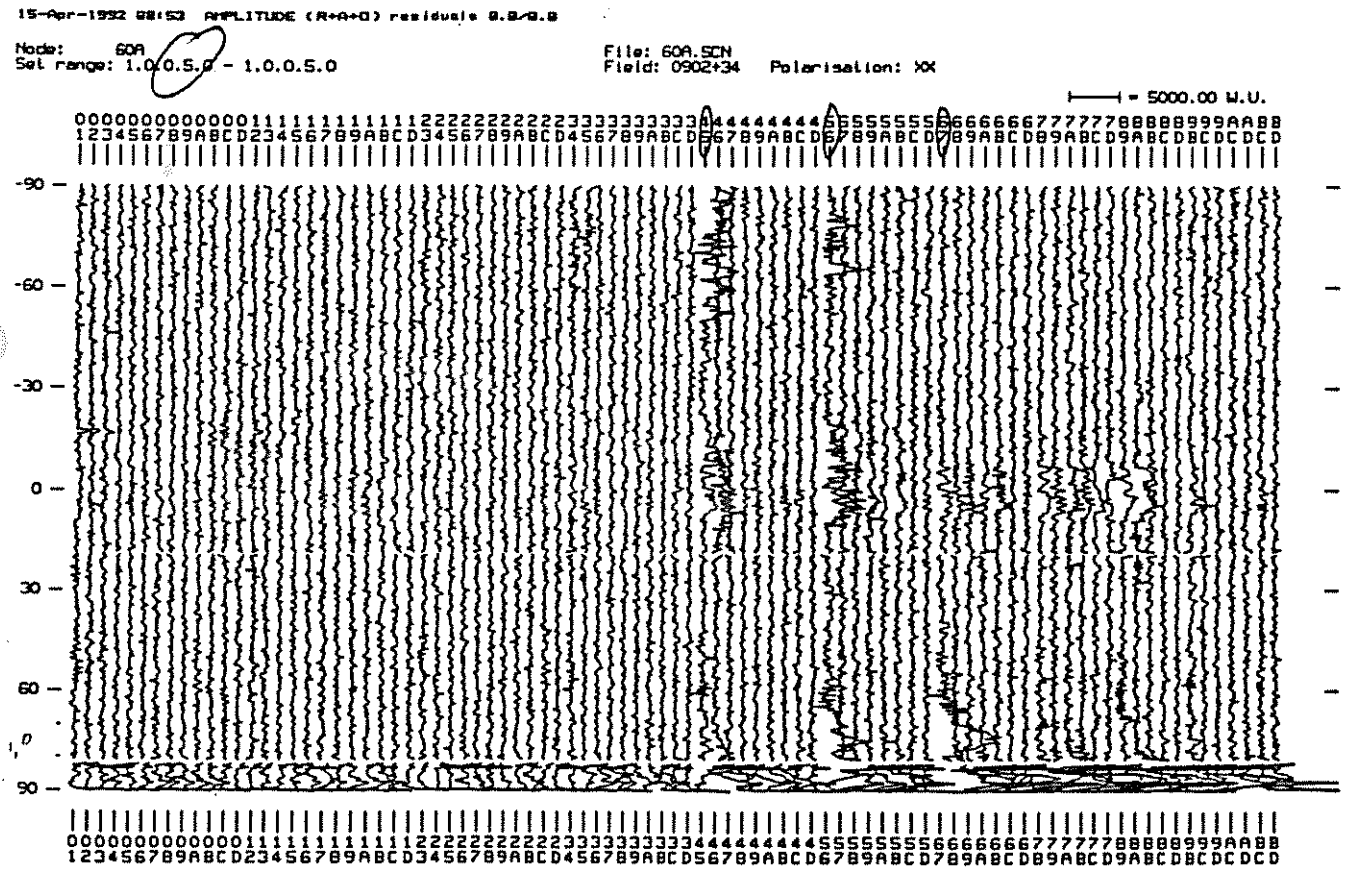
1.7.2 Selfcal residuals

```
>  exe nplot
    NPLOT$1 is started at 10-JAN-93 15:46:36
⊗ OPTION (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT:  res
⊗ PLOTTER (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP, EAL,EAP, ...) = EPS:  psl
⊗ PLUVO (PLUVO,NORMAL) = NORMAL:  <CR>
⊗ SCAN_NODE (input node name) = "":  3c147                      SCN-file
⊗ LOOPS (n,incr ....) = "":  <CR>
⊗ SETS (sets to do) = "":  0.0.0.0.0
⊗ HA_RANGE (HA range) = *:  <CR>
⊗ POLARISATION (XYX,XY,YX,Y,X) = XY:  <CR>
⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT)  read
⊗ MODEL_NODE (Model node) = "":  3c147                          MDL file
                                          specification of model not shown here

1 sources in list

⊗ MODEL_ACTION (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL:  <CR>
All fixed/movable interferometers pre-selected
```





NPLOT plot frequency channel 5

Amplitude residuals after SELFCAL/REDUN

⊗ **SELECT_IFRS** (Select/deselect ifrs) = "": <CR>
⊗ **DATA_TYPES** (AMPLITUDE,PHASE,COSINE,SINE) = AMPLITUDE,PHASE: <CR>
⊗ **SCALE_AMPL** (plot scale W.U./mm or %/mm) = 132.2784: <CR> *Here: W.U./mm*
⊗ **SCALE_PHASE** (plot scale in W.U./mm or deg./mm) = 594.4136: <CR> *Here: W.U./mm*
⊗ **HA_SCALE** (HA plot scale degree/cm) = 15: <CR> *12 hr fits on A4 plot*

The plot is now made, and sent to the laser plotter automatically if that has been specified. The program will then prompt the user for the next plot to be specified.

⊗ **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT) = QUIT: <CR>

```
NPLOT$1 is ended at 15:55:16                    STATUS=SUCCESS
```

Summary of the keywords (parameters) of the program NPLOT Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.neeries.cookbook]prog3_NPLOT.tex

1 NPLOT parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **PLOTTER** (QMS,QMSP,REGIS,FREGIS,EPS,EPP,PSL,PSP,BIT1,BIT2,BIT3,X11,USE1,USE2)

  (plotter to use)
  - QMS: QMS laser printer in landscape orientation
  - QMSP: QMS laser printer in portrait orientation
  - REGIS: graphics VT terminal
  - FREGIS: (*) REGIS to file
  - EPS: encapsulated PostScript for use in textprocessors etc
  - EPP: EPS in portrait mode
  - PSL: Postscript (do not use often, especially not for halftone: slow!)
  - PSP: PostScript in portrait mode
  - BIT1: (*) bitmap for 100 dpi
  - BIT2: (*) bitmap for 200 dpi
  - BIT3: (*) bitmap for 300 bpi
  - X11: (*) X11 terminal

- **OPTION** (MAP,DATA,MODEL,TELESCOPE,RESIDUAL,QUIT)
  Specify action to perform:
  - MAP: plot map(s)
  - DATA: plot scan data
  - MODEL: plot model data
  - TELESCOPE: plot telescope errors
  - RESIDUAL: plot interferometer residuals
  - QUIT: finish

- **PLUVO** (PLUVO,NORMAL)
  Specify action to perform:
  - PLUVO: plot scan data as function of channel and HA
  - NORMAL: plot scan data as function of interferometer and HA

- **ANGLE_SET** (pol. angle set)
  Specify the setset with the polarisation angles

- **TELESCOPES** (Telescope(s) to select)
  Specify the telescopes to be selected: Select 1 or more telescopes by typing a string of characters consisting
  - : of a combination of 0123.....ABCD
  For all telescopes: *
  *Default value(s): **

- **DATA_TYPES** (AMPLITUDE,PHASE,COSINE,SINE) (data types to plot)
  Specify the data type(s) to be plotted. An * will plot amplitude and phase
  Default value(s): AMPLITUDE

- **DATA_TYPE** (DATA,SLOPE) (data types to plot)
  Specify the data type(s) to be plotted. An * will plot data
  - DATA: plot the data as given in the map
  - SLOPE: a first trial only!!
  - : plot the horizontal slope of the data
  Default value(s): DATA /ASK

- **PLOT_TYPE** (CONT,HALF,POL,RULE) (plot types)
  Specify the way(s) the data should be plotted. An * will plot contours and halftone

- **SCALE_AMPL** (plot scale W.U./mm or %/mm)
  Specify the amplitude (and/or cosine, sine) scale in W.U./mm for residuals, raw data and model data , or in percent/mm for telescope corrections.
  Default value(s): 10.

- **SCALE_PHASE** (plot scale in W.U./mm or deg./mm)
  Specify the phase scale in W.U./mm for residuals, and in degrees/mm for raw data, model data and for telescope corrections.
  Default value(s): 1.

- **HA_SCALE** (HA plot scale degree/cm)
  Specify the HA scale in degree/cm.
  Default value(s): 1.

- **POL_SCALE** (pol. plot scale W.U./cm)
  Specify the polarisation scale in W.U./cm.

- **RULE_SCALE** (ruled plot scale W.U./cm)
  Specify the ruled surface plot scale in W.U./cm.

- **SIZE** (plot size)
  Specify the size of the plot. A value of 1 will produce a plot that will fit on one page (i.e. a QMS plot of a power of 2 length will be 12.8 cm). The value given will multiply the size of this standard plot. The first value is the horizontal direction, the second the vertical direction.
  Default value(s): 1.,1. /ASK

- **FULL_CONT** (contour values)
  Specify the values of the contours to be drawn as full lines. A maximum of 32 values is allowed.

- **DOT_CONT** (contour values)
  Specify the values of the contours to be drawn as dotted lines. A maximum of 32 values is allowed.

- **HALFTONE** (NONE,CONTINUE,STEP,PATTERN) (halftone type)
  Specify the type of halftone wanted (or NONE).
  - CONTINUE: a continuous variation within the specified range
  - STEP: a stepped variation within the specified range
  - PATTERN: a set of patterns
  Default value(s): NONE /ASK

- **RANGE** (halftone range)
  Specify the maximum value that has to be white, and the minimum value that has to be black. Values outside the range will always be white.

- **POL_RANGE** (pol. range)
  Specify the value below which no polarisation line will be drawn and the maximum polarisation value that has to be plotted

- **POL_TYPE** (POL,MAG) (pol. type)
  Specify if polarisation (POL) or magnetic field (MAG) should be plotted

- **RULE_RANGE** (ruled range)
  Specify the minimum and maximum cut-off values for ruled surface plots

- **TRANSFORM** (transmission curve)
  Specify a possible non-linear intensity response curve. Consider the IN and OUT intensities of the procedure. The IN are in the range 0 through 1, where 0 corresponds to the lower value of the range given, 1 to the high value.
  - : Specify now first an IN range over which the formula should hold,
  and then up to three values which will produce:
  - : OUT=v1+v2*IN+v3*IN*IN.
  Examples:

- : standard linear: 0,1,0,1
- : absolute values in halftone: 0,.5,1,-2
- : .5,1,-1,2
- : four steps in blackness: 0,.25,0
- : .25,.5,.25
- : .5,.75,.5
- : .75,1,1

- **COORD** (NONE,DLM,LM,DRADEC,RADEC,DDEGREE,DEGREE) (axis annotation)
  - NONE: no annotation
  - DLM: l, m in arcsec with respect to centre of plot
  - : (or annotation for UV-plane plots)
  - LM: l, m in arcsec with respect to map centre
  - : (or annotation for UV-plane plots)
  - DRADEC: right ascension and declination offset with respect to centre of plot
  - RADEC: right ascension and declination
  - DDEGREE: decimal right ascension and declination with respect to plot centre
  - DEGREE: decimal right ascension and declination

  Default value(s): NONE /ASK

- **COORD_TYPE** (TICK,DOTTED,FULL) (axis type)
  - TICK: give ticks along edges
  - DOTTED: draw dotted coordinate lines
  - FULL: draw full coordinate lines

  Default value(s): TICK /ASK
  valid abbreviated options

- **PLOT_POSITIONS** (NO,YES) (show sources)
  Specify if model sources have to be drawn in plot
  Default value(s): NO /ASK

- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

- **MODEL_OPTION**
  See NMODEL keyword descriptions
  MODEL keywords are:
  - MODEL_OPTION
  - MODEL_ACTION
  - SORT_TYPE
  - SORT_FIELD
  - SORT_CENTRE
  - EDIT_FIELD
  - EDIT_VALUE
  - CONVERT_TO
  - SOURCE
  - SOURCE_NUMBER
  - SOURCE_LIST
  - SOURCE_RANGE
  - SOURCE_FACTORS
  - INPOLQ,U,V_100
  - INPOLQ,U,V_400

- INPOLQ,U,V_1000
- INPOLQ,U,V_2000
- INPOLQ,U,V_4000
- INPOLQ,U,V_10000
- INPOLQ,U,V_100000
- BEAM_FACTORS
- DELETE_LEVEL

Summary of the keywords (parameters) of the NEWSTAR program NGIDS Image display on screen

Editor: J.E.Noordam

July 23, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aseries.cookbook]prog3_NGIDS.tex

1 NGIDS parameters (keywords)

The following description of the program parameters is also available as on-line HELP.

1.1 Specific NGIDS keywords

- **OPTION** (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR,QUIT)
  Specify action to perform:
  - LOAD: Load map(s) from WMP-files, for GIDS display
  - GCLEAR: Remove saved maps from GIDS display
  - POINTS: Type info about map points clicked by the mouse.
  NB: only for the last map loaded into GIDS (i.e. record 0).
  MB1 will give the information about a point in the GIDS display
  MB3 will cause exit of the display mode
  MB2 will fix the first corner of a rectangle, MB1 the second
  - FLAG: Interactive specification of uv-data to be flagged.
  Works on uv-data in WMP file, created by NMAP with IFRHA option.
  The specified points (or areas) are stored in an internal flag list
  MB1, 2, 3 as above
  - UFLAG: Specify the flag type to be used (default type: MANUAL)
  - UNLOAD: Save the internal flag list in an FLF-file, to be used in NFLAG
  - WRITE: Save the internal flag list in an ASCII file (see NFLAG)
  - CLEAR: Clear the internal flag list
  - QUIT: Leave NGIDS

- **MAP_COMPRESS** (factor)
  For a value N, the lines and pixels in a N*N box are averaged to give a single data point. A 1024*1024 map loaded with N=2 will result in a 512*512 image.
  Default value(s): 1

- **MAP_RANGE** (minimum,maximum data value)
  Cut-off values for the scaling of the data values to the range of pixel values. By choosing the proper cut-offs you can study a particular range of intensities in the map. The prompt values are the minimum and maximum data values in the map.

- **MAP_SEQUENCES** (minimum,maximum sequence number)
  A range of contiguous sequence numbers to be deleted from the GIDS display memory.

- **OUTPUT_FILE** (output filename)
  Specify the full name for the output disk-file.

- **USER_FLAG** (NONE,ALL,MAN,OLD,CLIP,NOISE,ADD,SHAD,U1,U2,U3)
  (specify flags to use)
  Specify the type(s) of flags to be used.
  More than one type may be specified, separated by comma's.
  NB: The selected flag type(s) OVERRIDE the default type (e.g. MANUAL) unless NONE is specified.

  - NONE: no flag type specified (i.e. use default types)
  - ALL or *: use all flag types
  - OLD: use the flag type for the 'OLD' class (i.e. flagged before 930609, and converted with NVS option)
  - MAN: use the flag type for the MANUAL class of operations
  - CLIP: use the flag type for the CLIP class of operations
  - NOISE: use the flag type for the NOISE class of operations

- SHAD: use the flag type for the SHADOW class of operations
- ADD: use the flag type for the ADDITIVE class of operations
- U1: use a separate flag for some user-defined operations
- U2: use a separate flag for some user-defined operations
- U3: use a separate flag for some user-defined operations

Explanation: Each uv-data sample in a SCN file has 8 flag-bits, which may be set or reset independently. Each of these flags corresponds to a certain class of flagging operations, which then become independent of each other.

- **AREA** (l,m,dl,dm)
  Specify an area of a map:
  - l: l-coord of centre of area (0 is map centre, <0 in direction of RA)
  - m: m-coord of centre of area (0 is map centre, >0 in direction of DEC)
  - dl: width of area in grid points
  - dm: height of area in grid points

1.2 Common keywords

The following keywords are common to many NEWSTAR programs. Some of them are used in NGIDS They are described in more detail in other sections of this Cookbook (see below).

- **NGEN (general) NEWSTAR keywords**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **LOG** (SPOOL,YES,NO,CATEN) (log-file action)
  - **RUN** (run mode)
  - **DATAB** (default directory name)
  - **INFIX** (node name shorthand)
  - **APPLY**
  - **DE_APPLY**
  - **UFLAG** (NONE,ALL,MAN,OLD,CLIP,NOIS,ADD,SHAD,U1,U2,U3) (specify un-flag details)
  - **DELETE_NODE** (yes/no)
  - **DISPLAY** (Output Display)

- **Keywords for medium (unit) selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **UNIT** (0,1,2,3,4,5,6,7,8,9,D) ('tape' unit)
  - **INPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (input 'tape' unit)
  - **OUTPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (output 'tape' unit)

- **Keywords for data-file (node) selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **SCN_NODE** (input/output 'node' name)
  - **WMP_NODE** (input/output 'node' name)
  - **MDL_NODE** (input/output 'node' name)
  - **NGF_NODE** (input/output 'node' name)
  - **FLF_NODE** (input/output 'node' name)
  - **INPUT_SCN_NODE** (input 'node' name)
  - **INPUT_WMP_NODE** (input 'node' name)
  - **INPUT_MDL_NODE** (input 'node' name)
  - **INPUT_NGF_NODE** (input 'node' name)
  - **INPUT_FLF_NODE** (input 'node' name)
  - **OUTPUT_SCN_NODE** (output 'node' name)
  - **OUTPUT_WMP_NODE** (output 'node' name)
  - **OUTPUT_MDL_NODE** (output 'node' name)
  - **OUTPUT_NGF_NODE** (output 'node' name)
  - **OUTPUT_FLF_NODE** (output 'node' name)

- **Keywords for Set selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **SCN_SETS** (Set(s) to do: g.o.f.c.s)
  - **SCN_GROUPS** (1st index: group(s))
  - **SCN_OBSS** (2nd index: observation(s))
  - **SCN_FIELDS** (3rd index: field(s))
  - **SCN_CHANNELS** (4th index: channel(s))
  - **SCN_SECTORS** (5th index: sector(s))
  - **WMP_SETS** (Set(s) to do: g.f.c.p.t.m)
  - **WMP_GROUPS** (1st index: group(s))
  - **WMP_FIELDS** (2nd index: field(s))
  - **WMP_CHANNELS** (3rd index: frequ channel(s))
  - **WMP_POLARS** (4th index: polarisation(s))
  - **WMP_TYPES** (5th index: type(s))
  - **WMP_MAPS** (6th index: map(s))
  - **MDL_SETS** (Set(s) to do)
  - **NGF_SETS** (Set(s) to do)
  - **NGF_GROUPS** (1st index: group(s))
  - **NGF_FIELDS** (2nd index: field(s))
  - **NGF_CHANNELS** (3rd index: frequ channel(s))
  - **NGF_POLARS** (4th index: polarisation(s))
  - **NGF_IFRS** (5th index: ifr(s)/tel(s))
  - **NGF_CUTS** (6th index: cut(s))
  - **LOOPS** (niter,Setincr ....)

- **Keywords for uv-data selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **POLARISATION** (XYX,XY,Y,X,YX) (polarisation(s))
  - **HA_RANGE** (DEG) (HA range)
  - **SELECT_IFRS** (Select/de-select ifrs)
  - **SELECT_TEL** (Select/de-select tels)

Summary of the keywords (parameters) of the program NGIDS Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 9, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aseries.cookbook]prog3_NGIDS.tex

2 NGIDS parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **MAP_NODE** (input node name)
  See GENERAL keyword descriptions: NODE
- **SETS** (sets to do)
  See GENERAL keyword descriptions: SETS
- **LOG** (SPOOL,YES,NO,CATEN) (logging type)
  See GENERAL keyword descriptions
  Default value(s): NO /NOASK
- **RUN** (run/test type)
  See GENERAL keyword descriptions
  Default value(s): YES /NOASK
- **DATAB** (database)
  See GENERAL keyword descriptions
  Default value(s): "" /NOASK
- **INFIX** (node shorthand)
  See GENERAL keyword descriptions
  Default value(s): "" /NOASK
- **APPLY**
  See GENERAL keyword descriptions
  *Default value(s): * /NOASK*
- **DE_APPLY**
  See GENERAL keyword descriptions
  Default value(s): NONE /NOASK
- **LOOPS** (n,incr ....)
  See GENERAL keyword descriptions
  Default value(s): "" /ASK
- **MAP_COMPRESS** <factor>
  For a value N, the lines and pixels in a N*N box are averaged to give a single data point. A 1024*1024 map loaded with N=2 will result in a 512*512 image. Answer CTRL/Z to get back to the MAIN_OPTION question.
  Default value(s): 1
- **MAP_RANGE** <minimum,maximum data value>
  Cut-off values for the scaling of the data values to the range of pixel values. By choosing the proper cut-offs you can study a particular range of intensities in the map. The prompt values are the minimum and maximum data values in the map. Answer CTRL/Z to get back to the MAIN_OPTION question.
- **MAP_WINDOW** <xmin,ymin,xmax,ymax>
  The window in the map to be loaded. You will be prompted with the maximum window (whole map). Answer CTRL/Z to get back to the MAIN_OPTION question.

Description of the program NATNF
Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.aseries.cookbook]prog1_NATNF.tex

1 The program NATNF

This program controls the reading of uv-data from the 6 km East-West array (the socalled 'compact array') of the Australia Telescope (AT) into a NEWSTAR .SCN file.

1.1 Overview of NATNF options

1.2 ...

Summary of the keywords (parameters) of the program NATNF Part of the NEWSTAR Cookbook

J.E.Noordam (editor)

September 10, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.naeries.cookbook]prog3_NATNF.tex

1 NATNF parameters (keywords)

The following description of the program parameters is also available as on-line HELP. The text has been slightly modified (clarified) where necessary.

- **OPTION** (LOAD,QUIT)
  Specify action to perform:
  - LOAD: load RPFITS data into scan file
  - QUIT: finish
- **INPUT_FILE** (input filename)
  Specify the input filename (without an extension for the LOAD from disk option).
- **INPUT_LABELS** (input labels)
  Specify the tape labels to be read. * specifies all labels on the tape
- **NGEN keywords, subset of COMMON keywords**
  See COMMON keyword descriptions: NGEN subset
  NGEN keywords are:
  - LOG
  - RUN
  - DATAB
  - INFIX
  - APPLY
  - DE_APPLY
  - LOOPS
  - DELETE_NODE

Summary of the keywords (parameters) of the NEWSTAR program NFLAG Inspection and flagging of uv-data

Editor: J.E.Noordam

July 23, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.aseries.cookbook]prog3_NFLAG.tex

1 NFLAG parameters (keywords)

The following description of the program parameters is also available as on-line HELP.

1.1 Specific NFLAG keywords

- **OPTION** (SHOW,FLAG,QUIT)
  Specify the action to be performed by the program NFLAG:

  - SHOW: show/edit data and header information in SCN-file
  - FLAG: (un-)flag uv-data in SCN-file
  - QUIT: leave the program NFLAG

- **INPUT_FILE** (input filename)
  Specify the full name of the input disk-file.

- **OUTPUT_FILE** (output filename)
  Specify the full name for the output disk-file.

- **FLAG_OPTION** (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT)
  Specify the flagging option to be performed:

  - FLAG: Start actual flagging operations
  - CLEAR: Clear the internal flag list
  - LOAD/UNLOAD: Transfer the internal flag list from/to an FLF-file
  - READ/WRITE: Transfer the internal flag list from/to an ASCII file
  - QUIT: Leave flagging option

- **FLAG_MODE**
  Specify certain operational modes, which remain active from here onwards.
  At least one of the modes has to be selected (e.g. FLAG).
  All modes can be changed, but only one at a time.
  CONTINUE when ready. A summary of the current mode setting will be shown.

  - FLAG/UNFLAG: Flags will be set or reset.
  - CORR/NOCORR: Apply corrections or not (incl flagging!) to the uv-data, before using them (default: NOCORR)
  - SHOW/NOSHOW: Show for each Scan when flag is changed (default: NOSHOW)

  - UFLAG: Select one or more flag types to be used. This overrides the default flag type(s) that would otherwise be used for a particular (un)flagging operation.

  It is possible to (un)flag entire Scans by using the flags in the Scan header.
  - FHEAD: Force flag setting in Scan header always, even if only one uv-data point in the Scan is (un)flagged.
  - FDATA: Force flag setting in individual uv-data always. For operations that work on entire Scans, all uv-data are (un)flagged.
  - NOFORCE: (default) Use the default choice for each operation.

  Escape options:
  - CONT: Down: Continue to actual flagging operations
  - NEXT: Up: Select the next Set of Sectors
  - NODE: Up: Select the next SCN node
  - QUIT: Up: Select next NFLAG option

Default value(s): CONT

- **OPERATION_0**

Escape options:
- CONT: Down: Continue to other flagging operations
- MODE: Up: Modify the flagging mode
- NEXT: Up: Select the next Set of Sectors
- NODE: Up: Select next SCN node
- QUIT: Up: Escape to NFLAG options

At this level, there are three categories of flagging operations.
Remember that only the specified uv-data 'hypercube' is affected.
Make sure that you understand which of the 8 flag types are affected!

1) Manual flagging operations. The default flag type is MANUAL (!).

Manual operations that affect ENTIRE SCANS only:
- ALL: UNFLAG all the hypercube HA-scans (not allowed for FLAG)
- HA: (UN-)FLAG the hypercube HA-scans for a selected HA-range

NB: This operation has its own HA-range selection.
- ><: (UN-)FLAG the HA-scan at the next(>) or previous(<) HA.

NB: This works on all the Sectors in the hypercube!
Manual operations that affect INDIVIDUAL UV-DATA only:
- IFR: (UN-)FLAG the uv-data for selected interferometer(s)

NB: This operation has its own ifr-selection. The affected data
lie in the cross-section of this and the specified hypercube.

2) Operations using criterion(s) from the Scan header. A single upper LIMIT value is asked. If mode=FLAG, a flag is set if MAX(crit)>LIMIT, or if MIN(crit)<0. See below for mode=UNFLAG. The default flag type is NOISE:

- ANOISE: Criterions are the 4 values for ALGNS (WU), i.e. the overall Align/Selfcal noise values for Xgain, Xphase, Ygain and Yphase.
- XAN/YAN: As ANOISE, but only looked at the X or Y ALGNS (for gain,phase).
- RNOISE: Criterions are the 4 values for REDNS (WU), i.e. the overall Redundancy noise values for Xgain, Xphase, Ygain and Yphase.
- XRN/YRN: As RNOISE, but only looked at the X or Y REDNS (for gain,phase).

3) Operations using a criterion derived from individual uv-data.
Two LIMIT values are asked: lower, upper. If mode=FLAG, a flag is set if lower<crit<upper. See below for mode=UNFLAG. The default flag type is CLIP:

- CLIP: Criterion is the AMPLITUDE of the individual uv-data.

NB: This works best for corrected uv-data (mode=CORR).
- MAX: Criterion is MAX(ABS(COS),ABS(SIN)) over ALL the uv-data in the Scan. NB: This value is taken from the Scan header.
- RRESID: Criterion is the AMPLITUDE of the Redundancy residuals for the individual uv-data.

NB: The flag-bytes associated with the uv-data (or Scans) are modified by means of logical operations with a 'FLAGBYTE'. In the latter, 1-8 set bits represent the active flag types. Each operation sets a default flag bit in the FLAGBYTE, unless the user has overridden this by using the UFLAG option at FLAG_MODE.
- If mode=FLAG, affected flags are modified by: flagbyte.IOR.FLAGBYTE.
- If mode=UNFLAG, affected flags are modified by: flagbyte.IAND.NOT(FLAGBYTE).

With 'affected flags' are meant those uv-data (or Scans) that meet the criterion specified for a particular (un)flagging operation. Note that a

flagging operation is exactly reversed by setting mode=UNFLAG and doing the same operation with the same values for the criterion LIMIT(S).

NB: The operations ALL, HA, >, < and MAX work only on entire Scans, i.e. the flags are set in the Scan header. The other operations (except IFR?) also do this if mode=FHEAD (force flags into header). The various NOISE operations work on the entire Scan if mode=FILLED (i.e. all ifrs and all pols are specified in the hypercube). In all other cases, the (selected!) uv-data are flagged individually.

- **OPERATION_1** (TOTEL,TODATA,TOHEAD,GET,PUT,CONT,MODE,NEXT,NODE,QUIT)

Escape options:
- CONT: Down: Continue to other operations
- MODE: Up: Modify the flagging mode
- NEXT: Up: Select the next Set of Sectors
- NODE: Up: Select next SCN node
- QUIT: Up: Escape to NFLAG options

At this level, there are 3 categories of operations.
Remember that only the specified data 'hypercube' is affected.
Make sure that you understand which of the 8 flag types are affected!

) Operations to copy flags between data/headers and the internal flag list.

- GET: The flags that are set with the uv-data in the specified hypercube may be collected as entries in an internal flag list of NFLAG. Successive GET operation's will add entries to the list, which may be CLEARed and manipulated at a higher level (FLAG_OPTION). The default flag type is ALL.
- PUT: The entries in the internal list are used to set flags for specified uv-data or Scans. Flag types are given by the list entries.

2) Operations to copy flags between individual uv-data samples:

- TOTEL: Transfer flags from interferometers to telescopes; i.e. (UN-)FLAG all uv-data (in a Scan) that share a common RECEPTOR (i.e. X or Y channel of a telescope) with an interferometer that has been flagged with the selected flag.

3) Operations to copy flags between Scan headers and individual uv-data:

- TOHEAD: Look for the selected flag(s) in all specified data. If the the same flag(s) are set in the Scan header (i.e. the entire Scan is flagged). NB: The flags remain set in the uv-data!
- TODATA: Look for the selected flag(s) in the Scan header. If one or more of those flags are present, delete them from the Scan header, and add them to the specified data.

NB: The Scan header has 8 flag bits, with exactly the same meaning as the 8 flag-bits for each individual uv-data point. Setting flags in the header is equivalent to setting the same flags for all data points in that Scan. It speeds up the running of NEWSTAR programs if many Scans are flagged entirely. However, beginning users may ignore the distinction.

- **OPERATION_2** (SHAD,RED,NORED,CONT,MODE,NEXT,NODE,QUIT)

Escape options:
- CONT: Continue to other (?) flagging operations
- MODE: Up: Modify the flagging mode

- NEXT: Up: Select the next Set of Sectors
- NODE: Up: Select next SCN node
- QUIT: Up: Escape to NFLAG options

Remember that only the specified uv-data 'hypercube' is affected.
Make sure that you understand which of the 8 flag types are affected!

Flagging operations with 'deterministic' algorithms, i.e. algoritms that depend on telescope position or HA. The default flag type is SHADOW:

- SHAD: (UN-)FLAG the specified uv-data if telescope 'shadowing' occurs.

NB: Since the telescope diameter can be specified, this is also a way to flag baselines smaller than a certain projected size.

- RED: (UN-)FLAG the specified uv-data if the baseline is redundant
- NORED: (UN-)FLAG the specified uv-data if the baseline is non-redundant

- **USER_FLAG** (NONE,ALL,MAN,OLD,CLIP,NOISE,ADD,SHAD,U1,U2,U3)
(specify flags to use)
Specify the type(s) of flags to be used in the current (un-)flagging operation. More than one type may be specified, separated by comma's.
NB: The selected flag type(s) OVERRIDE the default types for the NFLAG flagging operations (except if NONE is specified).

  - NONE: no flag type specified (i.e. use default types)
  - ALL or *: use all flag types
  - OLD: use the flag type for the 'OLD' class (i.e. flagged before 930609, and converted with NVS option)
  - MAN (80): use the flag type for the MANUAL class of operations
  - CLIP (40): use the flag type for the CLIP class of operations
  - NOISE(20): use the flag type for the NOISE class of operations
  - SHAD (10): use the flag type for the SHADOW class of operations
  - ADD (08): use the flag type for the ADDITIVE class of operations
  - U1 (04): use a separate flag for some user-defined operations
  - U2 (02): use a separate flag for some user-defined operations
  - U3 (01): use a separate flag for some user-defined operations

  Explanation: Each uv-data sample in a SCN file has 8 flag-bits, which may be set or reset independently. Each of these flags corresponds to a certain class of flagging operations, which then become independent of each other. Shown between parenteses is the internal code for each flag type.

- **PUT_RANGE** (chan,HA,ifr,pol)
The entries in the internal flag list can be used to (un)flag uv-data in the SCN-file, using the PUT operation. Each list entry contains information about channel nr, HA, interferometer and polarisation, and thus indicates a (range of) uv-data points, or 'fields', in the SCN-file. The PUT operation is made very powerful by the possibility to (un)flag a 4-dimensional volume of uv-data around each field indicated by a list entry. This volume is of course limited by the boundaries of the specified hypercube in which the PUT operation is active. The volume is specified by specifying RANGES along the 4 axes: channel, HA, interferometer and polarisation.

  The range along each axis can be:
  . : just the fields indicated by the list entries
  * : the maximum possible range along this axis
  n[C] : n fields centered on indicated fields (n=even: one more on the left)
  nL : n fields to the 'left' of (and incl.) indicated fields
  nR : n fields to the 'right' of (and incl.) indicated fields

Notes:
n=0 : no flagging (volume of hypercube is zero)
n=1 : identical to . (indicated field only)
* : identical to .,.,.,. (this use of * is not consistent here!)
The interferometer range works on telescope (or rather receptor) basis,
i.e. the given range is valid for both receptors (e.g. 0Y and AX)

Example: if a particular ifr is flagged in one of the frequency channels, one may wish to flag the uv-data in all the other channels for that ifr. The put-range then becomes: *,.,.,.

- **HA** (DEG) (HA(s) to do)
Specify the start and end of the HA-range in which the action(s) should occur. If only one value is given, the end value will be the same.

- **LIMIT** ((un-)flag limit value)
Specify the limit (threshold) value for the (un-)flag criterion. The unit (e.g. W.U.) depends on the criterion under consideration.

- **LIMITS** ((un-)flag limits)
Specify the LOWER and UPPER limiting values for the (un-)flag criterion. The unit (e.g. W.U.) depends on the criterion under consideration.

- **FLAG_LIMIT** (minimum flag count)
If more than a certain number of uv-data in a Scan is flagged with the selected flag(s), the flag(s) are set in the Scan header (thus flagging the entire Scan). Specify this minimum number.

- **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT)
Specify interaction with the SCN-file header:
  - LAYOUT: show the layout (contents) of the file
  - SHOW: display all fields (values) in the File header
  - EDIT: edit fields (values) in the File header
  - CONT: go down one level, to interact with Sector headers
  - QUIT: go up one level, i.e. escape

- **SECTOR_ACTION** (NEXT,IFRS,NAME,FLAGS,SHOW,EDIT,CONT,QUIT)
Specify interaction with Sector header(s):
  - NEXT: go to the next Sector (of the specified Set)
  - IFRS: show the interferometer table
  - NAME: show index 'name' of the current Sector (if #nr specified)
  - FLAGS: show the nr of flags/ifr that are set in the current Sector
  - SHOW: display all fields (values) in the Sector header
  - EDIT: edit fields (values) in the Sector header by name
  - CONT: go down one level, to interact with Scan headers and uv-data
  - QUIT: go up one level, to interact with the File header

- **SCAN_ACTION** (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,T,E,Q)
Specify interaction with HA-Scan header(s) and/or their uv-data:
  - XX,etc: Select polarisation mode (active from thereon)
    (initial default: XX)
  - nn.nn: Select an Hour Angle (degr, e.g. -12.36)
    (initial default: first HA-scan in this Sector)
  - >[n]: Select the n-th next HA-scan (default=next, i.e. n=1)
  - <[n]: Select the n-th preceding HA-scan (default=last, i.e. n=1)
  - S[how]: Show detailed Scan header values at the current HA
  - E[dit]: Edit individual Scan header values at the current HA
  - D[ata]: Show Real/Imag (Cos/Sin) uv-data for the current HA and polar.
  - A[mpl]: Idem, but converted to ampl/phase
  - W[eight]: Show uv-data Weight factors for the current HA and polarisation, and the current data flag(s)

- T[el] Calculate and show a guess (based on a point source model) for telescope gain/phases. This also works when the phase errors are so large that NCALIB Selfcal/Redundancy gets stuck in 180 degree phase ambiguities. The calculated values may be transferred to the Scan header by hand (NCALIB SET MANUAL).
- Q[uit]: Go up one level, to interact with Sector header(s)

- **EDIT** (Edit: name [,first] , val [,...])
  The values ('fields') in the various headers (File, Sector, Scan) may be edited individually by hand. This is laborious, BUT IT IS POSSIBLE!

  There are two forms of EDIT specification:
  - For a SINGLE-VALUE field, give the pair: fieldname,value
    Example: RA,85.90 will change the current value of RA to 85.90 (degr)
  - For a MULTIPLE-VALUE field (vector), give:
  fieldname [,first element to be changed, 0=first], value [,value,...]
    Example: .....?

  The correct (edit) names of the 'fields' are shown when the full headers are displayed with the SHOW option at the various levels. Value input depends on the field. In general the input should be in the same format as the output (string vs number, number radix, deg vs h:m:s).
  - Strings have to be enclosed in "" if they contain special characters or separators like spaces or comma's (,,).
  - Numbers [+-] [i][.[i]] [eExXdD[+-[i]]] the number should be given in the same radix as shown, but can be changed by preceding a number with % and BODX (e.g. %X-1.a2dX+2 is a valid number)
  - Angles can be expressed as [+-][h]:[m][:[s[.tht]]] or [+-][d].[m].[s[.tht]]
  - Complex numbers are given as [number] [+/- numberI]

1.2 Common keywords

The following keywords are common to many NEWSTAR programs. Some of them are used in NFLAG They are described in more detail in other sections of this Cookbook (see below).

- **NGEN (general) NEWSTAR keywords**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **LOG** (SPOOL,YES,NO,CATEN) (log-file action)
  - **RUN** (run mode)
  - **DATAB** (default directory name)
  - **INFIX** (node name shorthand)
  - **APPLY**
  - **DE_APPLY**
  - **UFLAG** (NONE,ALL,MAN,OLD,CLIP,NOIS,ADD,SHAD,U1,U2,U3) (specify un-flag details)
  - **DELETE_NODE** (yes/no)
  - **DISPLAY** (Output Display)

- **Keywords for medium (unit) selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **UNIT** (0,1,2,3,4,5,6,7,8,9,D) ('tape' unit)
  - **INPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (input 'tape' unit)
  - **OUTPUT_UNIT** (0,1,2,3,4,5,6,7,8,9,D) (output 'tape' unit)

- **Keywords for data-file (node) selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **SCN_NODE** (input/output 'node' name)
  - **WMP_NODE** (input/output 'node' name)
  - **MDL_NODE** (input/output 'node' name)

- **NGF_NODE** (input/output 'node' name)
- **FLF_NODE** (input/output 'node' name)
- **INPUT_SCN_NODE** (input 'node' name)
- **INPUT_WMP_NODE** (input 'node' name)
- **INPUT_MDL_NODE** (input 'node' name)
- **INPUT_NGF_NODE** (input 'node' name)
- **INPUT_FLF_NODE** (input 'node' name)
- **OUTPUT_SCN_NODE** (output 'node' name)
- **OUTPUT_WMP_NODE** (output 'node' name)
- **OUTPUT_MDL_NODE** (output 'node' name)
- **OUTPUT_NGF_NODE** (output 'node' name)
- **OUTPUT_FLF_NODE** (output 'node' name)

- **Keywords for Set selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **SCN_SETS** (Set(s) to do: g.o.f.c.s)
  - **SCN_GROUPS** (1st index: group(s))
  - **SCN_OBSS** (2nd index: observation(s))
  - **SCN_FIELDS** (3rd index: field(s))
  - **SCN_CHANNELS** (4th index: channel(s))
  - **SCN_SECTORS** (5th index: sector(s))
  - **WMP_SETS** (Set(s) to do: g.f.c.p.t.m)
  - **WMP_GROUPS** (1st index: group(s))
  - **WMP_FIELDS** (2nd index: field(s))
  - **WMP_CHANNELS** (3rd index: frequ channel(s))
  - **WMP_POLARS** (4th index: polarisation(s))
  - **WMP_TYPES** (5th index: type(s))
  - **WMP_MAPS** (6th index: map(s))
  - **MDL_SETS** (Set(s) to do)
  - **NGF_SETS** (Set(s) to do)
  - **NGF_GROUPS** (1st index: group(s))
  - **NGF_FIELDS** (2nd index: field(s))
  - **NGF_CHANNELS** (3rd index: frequ channel(s))
  - **NGF_POLARS** (4th index: polarisation(s))
  - **NGF_IFRS** (5th index: ifr(s)/tel(s))
  - **NGF_CUTS** (6th index: cut(s))
  - **LOOPS** (niter,Setincr ....)

- **Keywords for uv-data selection**
  (For details, see the description of COMMON NEWSTAR keywords.)
  - **POLARISATION** (XYX,XY,Y,X,YX) (polarisation(s))
  - **HA_RANGE** (DEG) (HA range)
  - **SELECT_IFRS** (Select/de-select ifrs)
  - **SELECT_TEL** (Select/de-select tels)

- **Model handling keywords** *(For details, see the description of NMODEL keywords.)*
  - MODEL_OPTION
  - CONVERT_TO (B1950,J2000,APPARENT,LOCAL)
  - MODEL_ACTION
  - SORT_TYPE (INCREASING,DECREASING)
  - SORT_FIELD (I,L,M,LM,ML,ID,Q,U,V,SI,RM,LA,SA,PA,BITS,TYP,CC,TP2,DIST.POL)
  - SORT_CENTRE (Centre l,m for SORT)
  - EDIT_FIELD (I,L,M,ID,Q,U,V,SI,RM,LA,SA,PA,BITS,TYP,CC,TP2)
  - EDIT_VALUE (edit value)
  - SOURCE_NUMBER (Source number)
  - SOURCE_LIST (Source number list)
  - SOURCE_RANGE (Source number range)
  - SOURCE (I,l,m,id,Q,U,V,lax,sax,pa,si,rm,rs)

- **SOURCE_FACTORS** (An/Ao,dl,dm)
- **DELETE_LEVEL** (low level)
- **DELETE_AREA** (l,m,dl,dm arcsec)
- **BEAM_FACTORS** (Beam factors)

Instrumental polarisation keywords:

- **INPOLQ,U,V_100**
- **INPOLQ,U,V_400**
- **INPOLQ,U,V_1000**
- **INPOLQ,U,V_2000**
- **INPOLQ,U,V_4000**
- **INPOLQ,U,V_10000**
- **INPOLQ,U,V_100000**

The NEWSTAR program NFLAG
Inspection and flagging of uv-data

Editor: J.E.Noordam

July 24, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5 [noordam.aperies.cookbook]prog1_NFLAG.tex

1 The program NFLAG

1.1 Overview of NFLAG options

The program NFLAG consists of two options that have been 'split off' the program NSCAN in June/July 1993. The the option FLAG allows the user to 'flag' uv-data in various ways. This option has grown out of the NSCAN option DELETE, and has greatly expanded functionality. The option SHOW allows the user to inspect the contents of a SCN-file.

1.2 NFLAG option FLAG

1.2.1 NEWSTAR phlagging philosophy

The NEWSTAR uv-data flagging scheme gives the user unprecedented flexibility for reversible data-editing. However, its very power (and the fact that it is different from AIPS) may be a little confusing, especially to the beginning NEWSTAR user. We hope that this section of the Cookbook will be a help

The most important thing to remember is that there are **8 types of flag**, which can be set and reset independently of each other. So, the user must always be aware which particular flag types have been selected. Each of the 8 flag types represents a certain class of (un)flagging operations:
- MANUAL (80): Manual, by the user.
- CLIP (40): Criterion is a data value (e.g. sin,cos,ampl,residue).
- NOISE (20): Criterion is an overall noise value (e.g. REDNS, ALGNS).
- ADDITIVE (10): Criterion is an overall average (e.g. DC offset).
- SHADOWING (08): Deterministic algorithms (e.g. shadowing).
- U3 (04): User-definable class
- U2 (02): User-definable class
- U1 (01): User-definable class

The eight types of flag are implemented as bits in a 'flag-byte', which is attached to each uv-data sample in a SCN-file, and to each Scan header. Flag settings may be inspected with the SHOW option of the program NFLAG. The sum of the internal codes indicates unambiguously which flags are set (e.g. MAN+SHAD+U1=89).

Flags are set and reset with the various operations behind the FLAG option of the program NFLAG (see block diagram NFLAG_002.fig). This includes the flags that have been specified interactively with a mouse, using the program NGIDS. The latter are transferred to the program NFLAG by means of a flag-file (see block diagram NFLAG_001.fig).

NB: The concept of a flag-file, which may also be edited, is potentially quite powerful. The information in such a flag-file specifies flag type, and ranges for frequency channel, HA, ifr and polarisation. Since these ranges may be wild-cards (*), the flag-files may be used to copy flags from a calibrator observation to a real observation.

The use of data flags by NEWSTAR programs is entirely analogous to the use of uv-data corrections: they are applied 'on-the-fly' whenever uv-data are read in for processing. The default is that all 8 flag types are tested for. But the user may specify that one or more flag types are to be ignored, by means of the general (NGEN) keyword UFLAG. This is of course analogous to the use of the NGEN keywords APPLY and DE_APPLY for on-the-fly corrections.

The advantage of the NEWSTAR approach is that different flagging operations can be treated independently of each other. For instance, a selection by laborious manual editing is not undone by a subsequent automatic clipping operation that has gone wrong. The 3 user-definable flag-types (or indeed any of the eight flags) can be used for experiments with different selections of uv-data, in the same way as different flag-tables are used in AIPS.

The flags in the Scan header have the same 8 types as the individual data-flags. They are used to (un)flag entire Scans, and is logically equivalent to using the flags of all the individual uv-data in the Scan. Since it saves testing time, it speeds up NEWSTAR programs. It is also possible to flag an entire Scan temporarily, while retaining its internal flag distribution. But the disadvantage is that the user can be confused by a technical distinction, which could be completely transparent. This issue is not resolved yet, and in the meantime the user must contend with flagging modes like FDATA, FHEAD and NOFORCE, and flagging operations like TOHEAD and TODATA.

The user is urged to work through the examples in this section before starting to use NFLAG. Quite a lot of valuable (and sometimes essential) information is available in the on-line HELP text.

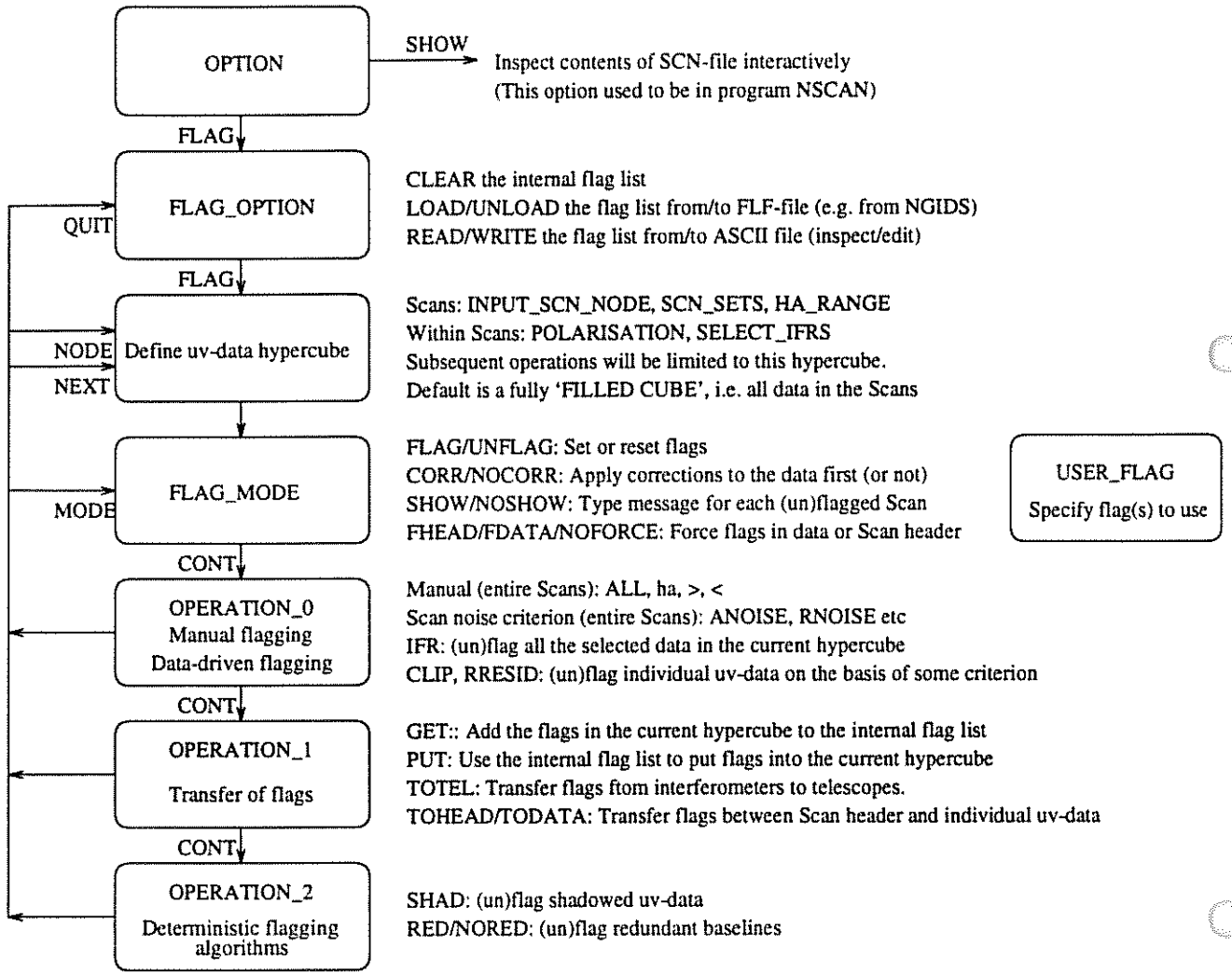


Fig ....: Schematic block diagram of the use of the FLAG option of the program NFLAG

1.2.2 Example: two flagging operations

This example has been annotated, so that it may serve as a first introduction to the use of NFLAG option FLAG. Starting with an unflagged SCN-file, two successive flagging operations are shown, using two different flag types (MANUAL and SHADOW). The results of this particular example are shown in the description of NFLAG option SHOW, later in this section.

```
> rzmws0 [101] % dwe nflag                                  Start program NFLAG
    NFLAG$1 is started at 15-JUL-93 16:25:02
⊗ OPTION (SHOW,FLAG,QUIT) = QUIT:  flag
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = FLAG:  <CR>
```

The other FLAG_OPTIONs deal with the NFLAG internal flag list: Clearing it and copying flags to/from the two types of flag files. This will be dealt with in a later example of GET/PUT. Now the uv-data hypercube will be defined, to which the subsequent (un)flagging operations will be limited:

```
⊗ INPUT_SCN_NODE (input 'node' name) = "":  3c147                SCN-file
⊗ SCN_SETS (Set(s) = "":  0.0.0.3.0                              Frequ channel 3
⊗ POLARISATION (XYX,XY,Y,X,YX) = XYX:  x                         XX only
 All auto/cross interferometers pre-selected
⊗ SELECT_IFRS (Select/de-select ifrs) = "":  <CR>                All ifrs
⊗ HA_RANGE (DEG) = *:  <CR>                                      All HA-Scans
```

In an FILLED data hypercube, all interferometers and all polarisations are selected. Some operations will then set flags in the Scan header, rather than in the individual uv-data. The user should be aware of this when inspecting flags, and in subsequent unflagging operations.

```
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = FLAG:  <CR>
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT:  show
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT:  <CR>
 Current modes: FLAG      SHOW      UNFILLED  NOCORRECT
```

At least one FLAG_MODE must be specified here, e.g. FLAG. The specified FLAG_MODEs will be active from here onwards. NOCORRECT indicates that no corrections will be applied to the uv-data when they are read in. In the following (un)flagging operations, their default flag types will be used, since we have not used the option UFLAG to override them.

First, some individual interferometers will be flagged, using the manual operation IFR. Its default flag has type MANUAL. This operation has its own interferometer selection, which can only narrow *the hypercude specified above.*

```
⊗ OPERATION_0 (ALL,HA,>,<,MAX...) = MODE:  ifr   Default flag type: MANUAL
 No  interferometers pre-selected
⊗ SELECT_IFRS (Select/de-select ifrs) = "":  7*,9A
```

```
   0123456789ABCD
 0 -------+------
 1  ------+------
 2   -----+------
 3    ----+------
 4     ---+------
 5      --+------
 6       -+------
 7        -++++++
 8         ------
 9          -+---
 A           ----
 B            ---
```

```
 C              --
 D               -
⊗ SELECT_IFRS (Select/de-select ifrs) = "": <CR>
 HA  -88.99 interferometers flagged for sector 0.0.0.3.0
 HA  -88.74 interferometers flagged for sector 0.0.0.3.0
 HA  -88.49 interferometers flagged for sector 0.0.0.3.0
 HA  -88.24 interferometers flagged for sector 0.0.0.3.0
:
 HA   88.24 interferometers flagged for sector 0.0.0.3.0
 HA   88.50 interferometers flagged for sector 0.0.0.3.0
 HA   88.75 interferometers flagged for sector 0.0.0.3.0
```

The message per (affected) Scan is the result of the flag-mode SHOW, which will now be turned off. The second operation that will be shown here is the flagging of uv-data that are affected by 'shadowing', i.e. the partial blocking of the field of view of one telescope by another. This is called a 'deterministic algorithm', since it only uses known instrumental parameters like HA, elevation and telescope position. The operation has the default flag type SHADOW.

```
OPERATION_0 (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YAN,
        IFR,CLIP,RRESID,CONT,MODE,NEXT,NODE,QUIT) = MODE:
⊗ OPERATION_0 (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YA...) = MODE:  <CR>
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT:  noshow
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT:  <CR>
 Current modes: FLAG     NOSHOW     UNFILLED  NOCORRECT
⊗ OPERATION_0 (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YA...) = MODE:  cont
⊗ OPERATION_1 (TOTEL,TODATA,TOHEAD,GET,PUT,CONT,MODE,NEXT, ...) = MODE:  cont
⊗ OPERATION_2 (SHAD,RED,NORED,CONT,MODE,NEXT...) = MODE:  shad
⊗ LIMIT ((un-) = 25:  200                          Give telescope diameter (m)
```

For real flagging of shadowed data, the actual diameter of 25m for WSRT telescopes should of course be given here. This example is to demonstrate a serendipitous alternative use of this operation, i.e. the flagging of all baselines with a projected *size of less than 200m.*

The result of these two flagging operations may now be verified with NFLAG option SHOW. This is done, for this particular example, in the description of option SHOW at the end of this section.

```
⊗ OPERATION_2 (SHAD,RED,NORED,CONT,MODE,NEXT...) = MODE:  quit
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = QUIT:  <CR>
⊗ OPTION (SHOW,FLAG,QUIT) = QUIT:  show                NFLAG option SHOW
```

1.2.3 Example: resetting all flags

The following example is very important for users who have lost their way in the many options and operations of NEWSTAR flagging, and want to start with a clean slate. It also emphasizes the two questions that users should constantly ask themselves: Which flag types are being affected, and are they affected in the Scan header or in the individual uv-data?

⊗ **OPTION** (SHOW,FLAG,QUIT) = QUIT: `flag`
⊗ **FLAG_OPTION** (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = FLAG: `<CR>`
⊗ **INPUT_SCN_NODE** (input 'node' name) = "": `3c147` *SCN-file*
⊗ **SCN_SETS** (Set(s) = "": `*` *All Sectors*
⊗ **POLARISATION** (XYX,XY,Y,X,YX) = XYX: `<CR>` *All polarisations*
`All auto/cross interferometers pre-selected`
⊗ **SELECT_IFRS** (Select/de-select ifrs) = "": `<CR>` *All ifrs*
⊗ **HA_RANGE** (DEG) = *: `<CR>` *All HA-Scans*

⊗ **FLAG_MODE** (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = FLAG: `unflag`
⊗ **FLAG_MODE** (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT: `<CR>`
⊗ **FLAG_MODE** (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT: `uflag`
⊗ **USER_FLAG** (NONE,ALL,MAN,OLD,CLIP,NOISE,ADD,SHAD,U1,U2,U3) = NONE: `all`
⊗ **FLAG_MODE** (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT: `<CR>`
`Current modes: UNFLAG NOSHOW FILLED NOCORRECT`

Note the use of UFLAG to select all (8) flag types, This overrides the default flag types of any subsequent flagging operations.

⊗ **OPERATION_0** (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YA, ...) = MODE: `ifr`
`No interferometers pre-selected`
⊗ **SELECT_IFRS** (Select/de-select ifrs) = "": `*` *All ifrs*

```
  0123456789ABCD
0 -+++++++++++++
1  -++++++++++++
2   -+++++++++++
3    -++++++++++
4     -+++++++++
5      -++++++++
6       -+++++++
7        -++++++
8         -+++++
9          -++++
A           -+++
B            -++
C             -+
D              -
```

⊗ **SELECT_IFRS** (Select/de-select ifrs) = "": `<CR>`

This takes care of the flags in the individual uv-data. To make sure that also the flags in the Scan headers are reset, another operation must be performed:

⊗ **OPERATION_0** (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YA, ...) = MODE: `all`
⊗ **HA** (DEG) (HA(s) to do) = "": `*` *All HA-Scans*

Now all flags in this SCN-file have been reset. This may be verified with NFLAG option SHOW.

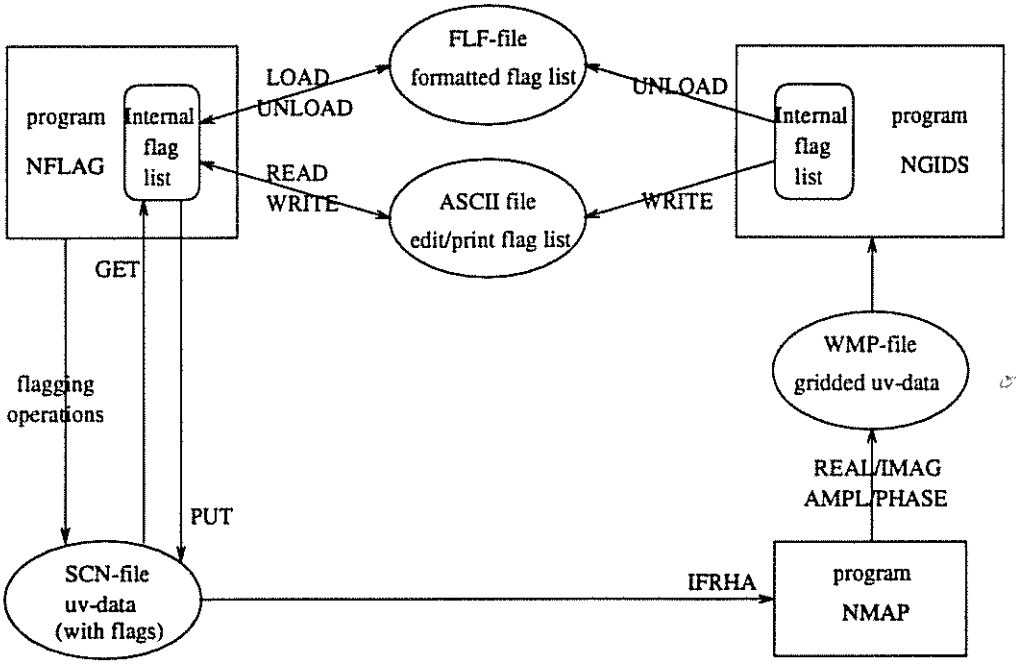


Fig .....: Schematic relationship of the various uv-data 'flagging' operations and concepts.
Each uv-data sample in a SCN-file has 8 different flags, which can be set or reset independently.
Different flags are assigned to various classes of flagging operations of the programs NFLAG and NGIDS.
Flags can be transferred (copied) between the uv-data and the outside world via an internal flag-list
of the program NFLAG. The two kinds of intermediate files contain the same flag-information, but
in different formats. The ASCII file is larger than the FLF-file, but can be inspected and edited.

1.2.4 The internal flag list: GET and PUT

The program NFLAG uses an internal flag list to store flags temporarily in the process of copying them from one place to another. This is illustrated in the block diagram NFLAG_001.fig.

The flag list can be filled in either of two ways:

1. From the uv-data in the SCN-file, in one or more GET operations. The user should keep in mind that each GET operation will add entries to the list. the list should be explicitly CLEARed when necessary.
2. From one of the two kinds of flag file, i.e. an FLF-file (LOAD) or an ASCII flag file (READ). The reverse operations (UNLOAD and WRITE) are also possible. Thus, to store flags that are set in a SCN-file in a flag-file, they must first be GOT into the internal list, and then UNLOADED or WRITTEN to a flag-file.

The PUT operation, with which flags are copied from the internal flag list to the uv-data is very powerful, because it is possible to specify a socalled PUT_RANGE. This is a range in the four 'dimensions' channel, HA, ifr and polarisation. For each entry in the flag list, not only the corresponding uv-data point will be flagged, but a 4-dimensional hypercube around it. For example: if some automatic algorithm has detected interference in one frequency channel, it is possible to flag all frequency channels at that particular HA and ifr in this way.

1.2.5 The two kinds of flag files

Flags may be stored into two different files: the FLF-file is (much) smaller because the information is stored in a compact format. The same information can also be stored in an ASCII file (default name FLAG.LOG), which can be inspected and edited with a normal text editor. It is up to the user to choose between these two kinds of file.

When printed, the ASCII file looks as follows:

```
> rzmws0 [102] % more FLAG.LOG                                    UNIX command

!+ Flagging file FLAG.LOG
!  Created by NOORDAM on 930715 at 17:06:18 at rzmws0
!  Flags:
!       MAN : 80  CLIP: 40  NOIS: 20  ADD : 10
!       SHAD: 08  U3  : 04  U2  : 02  U1  : 01
!  Types:
!       00:  Interprete Ifr field as interferometer
!       01:  Interprete Ifr field as baselines in m
!  Data following an ! are seen as comments
!  Remaining fields have format:
!       *:          all values
!       value:      single value
!       val1=val2:  value range (inclusive)
!
!-
!Flag  Type  Channel       Hour-angle          Ifr         Pol
40     00    0             -88.74              *           *
40     00    0             -88.49              *           *
40     00    0             -88.24              *           *
40     00    0             -87.99              *           *
80     00    0             -10.02              *           *
80     00    0             -9.77               *           *
80     00    0             -9.52               *           *
80     00    0             -9.27               *           *
80     00    0             -9.02               *           *
80     00    0             -8.77               *           *
```

Note the two flag types, MANUAL (80) and NOISE (40), which have obviously been set in two different flagging operations. *(NB: the use of the word 'type' for the second column is a bit cunfusing here: it has no relation to the flag type!)*

1.3 Interactive flagging using NGIDS

In the following, a complete sequence of steps will be shown:
- Making a 'map' of uv-data in a WMP file, using NMAP
- Displaying the uv-data on the screen, using NGIDS
- Specifying areas of uv-data to be flagged, using mouse and NGIDS
- Writing the flags from NGIDS to an ASCII flag-file
- Printing the ASCII flag-file to inspect it
- Reading the flags into the NFLAG internal flag list
- Copying the flags to the uv-data in the SCN-file, using PUT

See the block diagram NFLAG_001.fig.

1.3.1 Putting the uv-data into a WMP file

```
> rzmws0 [101] % dwe nmap

     NMAP$1 is started at 24-JUL-93 16:59:38
⊗ OPTION (MAKE,SHOW,FIDDLE,W16FITS,W32FITS,WRLFITS,FROM_OLD, ...) = QUIT:  make
⊗ LOOPS (niter,Setincr ....) = "":  <CR>
 Input data:
⊗ SCN_NODE (Input/output 'node' name) = "":  3c147          SCN-file
⊗ SCN_SETS (Set(s) = "":  0.0.0.3.0                         Channel 3 only
⊗ HA_RANGE (DEG) = *:  <CR>                                 All HA-Scans
 All fixed/movable interferometers pre-selected
⊗ SELECT_IFRS (Select/de-select ifrs) = "":  *              All ifrs!
   0123456789ABCD
 0 -++++++++++++
 1  -+++++++++++
 2   -++++++++++
 3    -+++++++++
 4     -++++++++
 5      -+++++++
 6       -++++++
 7        -+++++
 8         -++++
 9          -+++
 A           -++
 B            -+
 C             -
 D
⊗ SELECT_IFRS (Select/de-select ifrs) = "":  <CR>
⊗ SCN_NODE (Input/output 'node' name) = "":  <CR>           If 2nd input SCN-file
⊗ USER_COMMENT (map comment) = "":  Cookbook demo
 Map properties:
⊗ UV_COORDINATES (UV,BASHA,IFRHA) = UV:  ifrha              Recommended!
⊗ HA_RESOLUTION (DEG) = 0.50137 DEG:  <CR>
⊗ IFR_RESOLUTION (interferometer separation) = 1:  <CR>
⊗ FT_SIZE (FFT size) = 211,361:  <CR>
⊗ OUT_SIZE (Output size) = 211,361:  <CR>
⊗ QMAPS (More map details?) = NO:  <CR>
 Data manipulations:
⊗ QDATAS (More data handling details?) = NO:  <CR>
⊗ SUBTRACT (Source subtraction?) = NO:  y                   Recommended if possible
⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT)  read
```

⊗ INPUT_MDL_NODE (input 'node' name) = "": [3c147] *MDL-file*
⊗ MODEL_OPTION (READ,WRITE,CLEAR,ZERO,SHOW,LIST,RSHOW,RLIST, ...) A,QUIT) [<CR>]

```
 1 sources in list
```

⊗ MODEL_ACTION (MERGE,AD...) = MERGE,BAND,TIME,NOINPOL: [<CR>]

```
 Output files:
```

⊗ MAP_POLAR (XX,XY,YX,YY,I,Q,U,V,L,XXI,XYI,YXI,YYI,II,QI,UI, ...) = XX: [XX,YY]
⊗ MAP_COORD (B1950_J2000,APPARENT,REFERENCE,AREFERENCE) = B1950_J2000: [<CR>]
⊗ OUTPUT (MAP,AP,COVER,REAL,IMAG,AMPL,PHASE) = AMPL,PHASE: [real,imag]
⊗ OUTPUT_WMP_NODE (output 'node' name) = "": [demo]

```
 Creating node DEMO
0.0.0.0.3.0(#0) type REAL in node DEMO
0.0.0.0.4.0(#1) type IMAG in node DEMO
0.0.0.1.3.0(#2) type REAL in node DEMO
0.0.0.1.4.0(#3) type IMAG in node DEMO
     NMAP$1 is ended at 17:06:32                STATUS=SUCCESS
```

1.3.2 Displaying the WMP file with NGIDS

```
 > rzmws0 [102] % dwe ngids
     NGIDS$1 is started at 24-JUL-93 17:10:14
remove tvdevices?
```

⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = LOAD: [<CR>]
⊗ INPUT_WMP_NODE (input 'node' name) = "": [demo]
⊗ WMP_SETS (Set(s) = "": [*]
⊗ AREA (l,m,dl,dm) = 0,0,364,106: [<CR>]

```
 Area(s) selected:
 Total    : l= 0, m= 0, dl= 364, dm= 106
```

⊗ MAP_COMPRESS (factor) = 1: [<CR>]
⊗ MAP_RANGE (minimum,maximum data value) = -10108,9213: [<CR>]

```
 Set 0.0.0.0.3.0 will be loaded
 Set 0.0.0.0.4.0 will be loaded
 Set 0.0.0.1.3.0 will be loaded
 Set 0.0.0.1.4.0 will be loaded
 Could not record map 4
```

⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = QUIT: [points]

```
    16,    41:   9166.00 WU at (9D,       8.0)
   -55,    34:   2277.00 WU at (8B,     -27.6)
    54,   -29:   3119.00 WU at (1B,      27.1)
```

1.3.3 Interactively specifying flags in NGIDS

⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = QUIT: [flag]

```
  -125,    23:  -3089.00 WU at (6D,      -62.7)
(   46,    41,    62,     1)
(    5,     3,   122,    64)
```

1.3.4 Transferring flags with an ASCII flag file

⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = QUIT: [unload]
⊗ OUTPUT_FLF_NODE (output 'node' name) = "": [demo]

```
 5 entries in FLF node DEMO

⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = QUIT:  write
⊗ OUTPUT_FILE (output filename) = "FLAG.LOG":  demoflag.log
 3 entries in file DEMOFLAG.LOG
⊗ OPTION (LOAD,GCLEAR,POINTS,FLAG,UFLAG,UNLOAD,WRITE,CLEAR, ...) = QUIT:  <CR>
     NGIDS$1 is ended at 17:18:40                STATUS=SUCCESS

> rzmws0 [104] % more DEMOFLAG.LOG                NEWSTAR makes capitals!

!+ Flagging file DEMOFLAG.LOG
!  Created by NOORDAM on 930724 at 17:18:00 at rzmws0
!  Flags:
!       MAN : 80  CLIP: 40  NOIS: 20  ADD : 10
!       SHAD: 08  U3  : 04  U2  : 02  U1  : 01
!  Types:
!       00:  Interprete Ifr field as interferometer
!       01:  Interprete Ifr field as baselines in m
!  Data following an ! are seen as comments
!  Remaining fields have format:
!       *:           all values
!       value:       single value
!       val1=val2:   value range (inclusive)
!
!-
!Flag  Type  Channel       Hour-angle          Ifr         Pol
80      00    *             -62.67              6D          *
80      00    *             7.52=   38.10       9D          *
80      00    *             -28.08=   32.59     1B=8B       *

> rzmws0 [105] % dwe nflag
    NFLAG$1 is started at 24-JUL-93 17:19:50
⊗ OPTION (SHOW,FLAG,QUIT) = QUIT:  flag
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = FLAG:  read
⊗ INPUT_FILE (input filename) = "FLAG.LOG":  demoflag.log
 5 entries in list
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = FLAG:  <CR>
⊗ INPUT_SCN_NODE (input 'node' name) = "":  3c147
⊗ SCN_SETS (Set(s) = "":  0.0.0.4.0
⊗ POLARISATION (XYX,XY,Y,X,YX) = XYX:  <CR>
 All auto/cross interferometers pre-selected
⊗ SELECT_IFRS (Select/de-select ifrs) = "":  <CR>
⊗ HA_RANGE (DEG) = *:  <CR>
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = FLAG:  show
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = FLAG:  uflag
⊗ USER_FLAG (NONE,ALL,MAN,OLD,CLIP,NOISE,ADD,SHAD,U1,U2,U3) = NONE:  U1
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = FLAG:  <CR>
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT:  <CR>
 Current modes: FLAG     SHOW      FILLED    NOCORRECT
⊗ OPERATION_0 (ALL,HA,>,<,MAX,ANOISE,RNOISE,XRN,YRN,XAN,YA. ..) = MODE:  cont
⊗ OPERATION_1 (TOTEL,TODATA,TOHEAD,GET,PUT,CONT,MODE,NEXT, ...) = MODE:  put
⊗ PUT_RANGE (chan,HA,ifr,pol) = ",",",",",",",":  <CR>
 HA  -62.67 interferometers flagged for sector 0.0.0.4.0
 HA  -28.07 interferometers flagged for sector 0.0.0.4.0
 HA  -27.82 interferometers flagged for sector 0.0.0.4.0
```

```
   HA  -27.57 interferometers flagged for sector 0.0.0.4.0
:
   HA   37.10 interferometers flagged for sector 0.0.0.4.0
   HA   37.36 interferometers flagged for sector 0.0.0.4.0
   HA   37.61 interferometers flagged for sector 0.0.0.4.0
   HA   37.86 interferometers flagged for sector 0.0.0.4.0
   HA   38.11 interferometers flagged for sector 0.0.0.4.0
   Current modes: FLAG     SHOW      FILLED    NOCORRECT
```

Now the reverse operation: GET, WRITE and inspect the ASCII file:

⊗ OPERATION_1 (TOTEL,TODATA,TOHEAD,GET,PUT,CONT,MODE,NEXT, ...) = MODE: [get]
Current modes: FLAG SHOW FILLED NOCORRECT
⊗ OPERATION_1 (TOTEL,TODATA,TOHEAD,GET,PUT,CONT,MODE,NEXT, ...) = MODE: [<CR>]
⊗ FLAG_MODE (FLAG,UNFLAG,CORR,NOCORR,SHOW,NOSHOW,FHEAD,FDATA, ...) = CONT: [quit]
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = QUIT: [write]
⊗ OUTPUT_FILE (output filename) = "FLAG.LOG": [<CR>]
3 entries in file FLAG.LOG
⊗ FLAG_OPTION (FLAG,CLEAR,LOAD,UNLOAD,WRITE,READ,QUIT) = QUIT: [<CR>]
⊗ OPTION (SHOW,FLAG,QUIT) = QUIT: [<CR>]
NFLAG$1 is ended at 17:25:57 STATUS=SUCCESS

```
> rzmws0 [106] % more FLAG.LOG

!+ Flagging file FLAG.LOG
!  Created by NOORDAM on 930724 at 17:25:42 at rzmws0
!  Flags:
!       MAN : 80  CLIP: 40  NOIS: 20  ADD : 10
!       SHAD: 08  U3  : 04  U2  : 02  U1  : 01
!  Types:
!       00:  Interprete Ifr field as interferometer
!       01:  Interprete Ifr field as baselines in m
!  Data following an ! are seen as comments
!  Remaining fields have format:
!       *:          all values
!       value:      single value
!       val1=val2:  value range (inclusive)
!
!-
!Flag  Type  Channel       Hour-angle          Ifr         Pol
80     00    *             -62.67              6D          *
80     00    *             7.52=   38.10       9D          *
80     00    *             -28.08=   32.59     1B=8B       *
```

This is not quite the result we expected. A GET operation produces a separate entry for every HA-Scan. What we see here is the compact notation produced by NGIDS. This is partly explainable: we have forgotten to CLEAR the internal flag list of NFLAG before doing the GET operation (remember that each GET adds *entries to the list). But we must conclude that GET has not added any entries to it in this case, so there must be something wrong.....*

1.4 NFLAG option SHOW

The option SHOW allows the user to inspect the contents of a SCN-file: the general layout, the contents of headers at the various levels (i.e. file header, Sector header and Scan header), and the uv-data itself. Until july 1993, this functionality used to be part of the program NSCAN. Its use is demonstrated in some detail in another section of this Cookbook: 'Description of the NEWSTAR SCN-file'. In this section, we will only draw attention to the use of the SHOW option to inspect flags that are set in the uv-data or in the Scan header.

1.4.1 A short tour of SHOW

The NFLAG option SHOW allows the user to inspect the SCN-file at successively deeper levels: File layout and header, Sector header, Scan header (including header flags) and finally Scan data (including weights and flags):

⊗ **OPTION** (SHOW,FLAG,QUIT) = QUIT: `show`
⊗ **INPUT_SCN_NODE** (input 'node' name) = "": `3c147`

```
File description of node 3C147:

Created: 17-Sep-1992 12:15  Revision(84): 23-Jul-1993 16:21
File contains 9 datasectors in 1 groups and has version 1
```

⊗ **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT: `<CR>`
⊗ **SCN_SETS** (Set(s) = "": `0.0.0.3.0`

```
Sector 0.0.0.3.0(#3) - 3C147 - Channel 3 - 711 scans - 4 polarisations
```

⊗ **SECTOR_ACTION** (NEXT,IFRS,NAME,FLAGS,SHOW,EDIT,CONT,QUIT) = CONT: `flags`
⊗ **SCAN_ACTION** (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,T,E,Q) = ">": `w`

1.4.2 Inspecting flags per Sector

It is possible to inspect the total number of flags for each interferometer in a Sector (i.e. many consecutive Scans). Each flagged uv-data point is counted for 'one', even if more than one of the 8 flag types have been set.

⊗ **OPTION** (SHOW,FLAG,QUIT) = QUIT: `show`
⊗ **INPUT_SCN_NODE** (input 'node' name) = "": `3c147`
⊗ **FILE_ACTION** (LAYOUT,SHOW,EDIT,CONT,QUIT) = CONT: `<CR>`
⊗ **SCN_SETS** (Set(s) = "": `0.0.0.3.0`
⊗ **SECTOR_ACTION** (NEXT,IFRS,NAME,FLAGS,SHOW,EDIT,CONT,QUIT) = CONT: `flags`
⊗ **HA_RANGE** (DEG) = *: `<CR>` *All HA-Scans*
⊗ **POLARISATION** (XYX,XY,Y,X,YX) = XYX: `<CR>` *All pols*

Flag count

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	.
	.	710	.	.	.	.	.	645	.	.	.	.	.	.	0
0	1420	.	710	.	.	.	.	.	.	.	.	.	.	.	1
1	1420	1420	.	710	.	.	.	645	.	.	.	.	.	.	2
2	1420	1420	1420	.	710	.	.	645	.	.	.	.	.	.	3
3	1420	1420	1420	1420	.	710	.	645	.	.	.	.	.	.	4
4	.	1420	1420	1420	1420	.	710	645	.	.	.	.	.	.	5
5	.	.	1420	1420	1420	1420	.	645	.	.	.	.	.	.	6
6	1290	.	1290	1290	1290	1290	1290	.	645	645	645	645	645	645	7
7	1420	1420	1420	1420	1420	1420	1420	1290	.	710	283	.	.	.	8
8	1420	1420	1420	1420	1420	1420	1420	1290	1420	.	710	710	.	.	9
9	2840	2840	2840	2840	2840	2840	2840	2580	2840	2840	.	710	.	.	A
A	2840	2840	2840	2840	2840	2840	2840	2580	2840	2840	1420	.	.	.	B

```
B  2840 2840 2840 2840 2840 2840 2840 2580 2840 2840 1420 1420    .   710 C
C  2840 2840 2840 2840 2840 2840 2840 2580 2840 2840 1420 1420 1420    .  D
     0    1    2    3    4    5    6    7    8    9    A    B    C    D  .
                              Data count
```

1.4.3 Inspecting flags per Scan

The flag-bytes of individual uv-data can be inspected too:

⊗ **SCN_SETS** (Set(s) = ""): [0.0.0.3.0]

⊗ **SECTOR_ACTION** (NEXT,IFRS,NAME,FLAGS,SHOW,EDIT,CONT,QUIT) = CONT: [<CR>]

⊗ **SCAN_ACTION** (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,T,E,Q) = ">": [W] *weights/flags*

```
                          XX Data weight (*0.1000)
     0    1    2    3    4    5    6    7    8    9    A    B    C    D  .
     .  -80   80   80   80    .    .  -80   80   80  200  150  180  220  0
0    .    .  -70   70   80   70    .    .   80   70  180  140  170  200  1
1   08    .    .  -70   80   70   80  -70   80   70  190  150  170  210  2
2    .   08    .    .  -80   70   80  -70   80   70  180  140  170  200  3
3    .    .   08    .    .  -80   80  -80   80   80  200  150  180  220  4
4    .    .    .   08    .    .  -80  -70   80   70  190  150  170  200  5
5    .    .    .    .   08    .    .  -80   80   70  190  150  180  210  6
6    .    .    .    .    .   08    .    .  -80  -70 -180 -150 -170 -200  7
7   80    .   80   80   80   80   88    .    .  -80 -200  160  180  220  8
8    .    .    .    .    .    .    .   88    .    .  -180 -140  170  200  9
9    .    .    .    .    .    .    .   80   08    .    .  -370  430  520  A
A    .    .    .    .    .    .    .   80   08   88    .    .  340  410  B
B    .    .    .    .    .    .    .   80    .   08   08    .    .  -480  C
C    .    .    .    .    .    .    .   80    .    .    .    .    .    .  D
     0    1    2    3    4    5    6    7    8    9    A    B    C    D  .
                          XX Data flags
```

one line missing

The flag code is the sum of the codes of all the flag types that are set for a particular uv-data sample. Thus, for some only the flag of type MANUAL (80) is set, for others only type SHADOW(08), and a few have them both (88).

Note that the weights of all uv-data that are flagged (any flag type) have been made negative. This is used internally, to speed up NEWSTAR programs.

1.4.4 Inspecting flags in the Scan header

If flags are set in the Scan header, they may be inspected as follows:

⊗ **SCN_SETS** (Set(s) = ""): [<CR>]

⊗ **SECTOR_ACTION** (NEXT,IFRS,NAME,FLAGS,SHOW,EDIT,CONT,QUIT) = CONT: [<CR>]

⊗ **SCAN_ACTION** (XX,XY,YX,YY,ha,>[n],<[n],S,D,A,W,T,E,Q) = ">": [S]

```
 Scan header description

HA          0.0000000      MAX            0.000      SCAL          0.000000
REDNS         15.907        29.774       13.912       30.621
ALGNS          0.000         0.000        0.000        0.000
OTHNS          0.000         0.000        0.000        0.000
BITS        00008000       CLKC       0.000000 s     ACLKC       0.000000 s
IREF              0.00 deg/kEXT          0.00000      REFR         0.00000
```

The flag (type MANUAL, code 80) is visible in BITS.

1.5 Ideas for improvements

Efficient data editing is now the highest priority for NEWSTAR. The program NFLAG, and the interactive flagging option in the program NGIDS, have been created in a very short time, to meet the needs of important new NEWSTAR users (notably the WHISP project and the operational groups in Westerbork and Dwingeloo). Although some of the most important options have been implemented (and partly tested), there is ample room for improvement.

The user is invited to add ideas and suggestions to the list below. This list is not necessarily in order of importance, and certain things will not get done very soon. Some ideas can be implemented rapidly: the program NFLAG has been designed as a frame-work, in which a wide range of flagging operations can be relatively easily provided and/or modified. Other improvements are clearly desirable, but the required effort may not be justifiable for an interim package (waiting for AIPS++). However, the list below will play an important role in determining the list of priorities for the fall of 1993.

- Flagging operation ARESID (comparable to RRESID at level OPERATION_0). The Align/Selfcal residue for each indivudual uv-data point is used as a criterion to flag that data point. The advantage over RRESID is that all interferometers are involved. The disadvantage is that a good source model is required. The most urgent application is the treatment of calibrator observations (for which good models exist) by the operational NFRA groups.
- Flagging operation DCOFFSET, to detect DC offsets caused by the ageing correlator.
- Automatic batch procedures (ABP) for certain flagging operations, for use by the operational groups in Westerbork and Dwingeloo. Ideally, they should lead to a succinct Quality Report for a particular observation, possibly with suggestions for action if necessary (Expert System).
- Easy availability of the necessary information to set limit values for flagging criteria. For instance, the overall Redundancy noise for the entire Sector when using the operation RRESID.
- The possibility of using 'general' limit values for flagging criteria, instead of explicit numeric values. For instance: upper limit is 2*REDNS, or 3.5*SIGMA etc.
- More detailed inspection tools for the flags that are set in the uv-data. At this moment there is only the possibility to show all flags (of any type) per interferometer for a particular Sector. This is only useful to see whether or not a certain interferometer has been flagged. Moreover, it is cumbersome to have to leave NFLAG option FLAG, and to enter NFLAG option SHOW.
- A SHOW option to view the internal flag list. Perhaps this could be gradually extended to a full set of tools to edit the flag list, comparable to the tools for a list of source model components.
- More user-friendly interactive flagging with the NGIDS display. This includes better use of available GIDS functionality to indicate data to be flagged, and improvement of the display of vital reference data (ifr, baseline, HA, channel, value etc) at all times.
- At this moment, the NFLAG flagging operations work directly on the uv-data in the SCN-file. It might be preferable to collect the flags in the internal flag list first, to be copied to the uv-data (PUT) or an external file after inspection.
- Reading the uv-data into NGIDS directly from the SCN-file. The present implementation requires the intermediairy steps via the program NMAP and the WMP file. This was easier to implement, but is more cumbersome for the user. Ideally, the user would of course like to interact directly with the uv-data in the SCN-file, in both directions (like in AIPS TVFLAG).
- In some ways, it may be easier to determine which uv-data are 'bad' from the the line graphics plots produced by NPLOT.

- Filtering operations on frequency-spectra, comparable to the AIPS option UVLIN. The latter simply calculates and subtracts the average from the spectrum of each individual uv-sample, to 'remove' the continuum, and thus to enhance the line. This technique could be refined to subtracting low-order polynomials from the spectra. Moreover, the effects of strong sources that are far outside the primary beam could be filtered out by removing high-frequency ripples from the spectra.
- Solar fringe detection/removal or shift to a set of positions
-
-
-
-
-

NEWSTAR appendix D:
AUTOMATIC BATCH PROCESSING (ABP)

J.E.Noordam (editor)

September 21, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

file user5:[noordam.nseries.cookbook]apx1_D.tex

1 APPENDIX D: Automatic Batch Processing (ABP)

1.1 ..

NEWSTAR appendix E: MONGO GRAPHICS

Editor: J.E.Noordam

January 11, 1993

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.neeries.cookbook]apx1_E.tex

1 APPENDIX E: MONGO graphics

In many cases, the user will need the results of NEWSTAR processing in graphical form. In order to make this easier, the output of programs like NGCALC can be specified to be in 'MONGO' format. These are ASCII files, organised in columns, which can be processed by the MONGO graphics program. Because it is ASCII, the knowledgeable user can easily edit the file with a standard editor, to fine-tune the resulting representation.

MONGO ©is an interactive graphics program, written by John L. Tonry. For a detailed description, see the MONGO manual ([?]). In this section, a few simple examples are given to give the user the general idea, and to get him/her over the initial barrier.

In Dwingeloo, MONGO is only available on the VAX. Create the MONGO command by typing

```
MONGO :== @user5:[mongo]mongo.com
```

MONGO can be used interactively, or by means of a command "macro" (.MAC). The data usually reside in an ASCII data file, which may be edited by the user.

1.1 A simple MONGO plot

Create a MONGO data file (extension .DAT). This is an ASCII, in which the data is lined up in columns. In this case, replace the string "....." in the first column with 0,1,2,3 etc to indicate channel nr.

Then make a MONGO command macro (ext .MAC):

```
data gain64a.dat     MONGO data file
limits -5 70 -10 10
xcol 1               x-values in 1st column of data file
ycol 2               y-values in 2nd column of data file
box                  draw a box around the plot
ptype 4 3
connect
xlabel channel #     horizontal axis label string
ylabel ....          vertical axis label string
id
```

Alternative: use MONGO interactively.

NEWSTAR appendix F: SELECTED PAPERS

Editor: J.E.Noordam

December 14, 1992

This document represents an intermediate update of a part of the NEWSTAR (Netherlands East-West Synthesis Telescope Array Reduction) Cookbook, and has been generated at the date printed above. It should be kept with your latest full release of the Cookbook until the next one appears. Cross-references to other parts of the Cookbook are not possible here of course, and have been replaced by '?'.

Contents

file user5:[noordam.nseries.cookbook]apx1_F.tex

1 APPENDIX F: SELECTED PAPERS

There are some papers that are closely related to the material covered in this cookbook, and may therefore be interesting to the user. Copies of these papers are attached here. Others are mentioned in the bibliography of this Cookbook.

1.1 NFRA ITR 198a by W.N.Brouw

The N-series redundancy programs. Part 1 of 1: General introduction.
by W.N.Brouw
NFRA Internal Techinical Report nr 198a, 2 sept 1992

1.2 Nature 1982 by J.E.Noordam and A.G.de Bruyn

High dynamic range mapping of strong radio sources, with application to 3C84.
by J.E.Noordam and A.G.de Bruyn
Nature, Vol 299, No 5884, pp 597-600, 14 october 1982

1.3 Experimental Astronomy 1992 by M.H.Wieringa

An investigation of the telescope based calibration methods 'Redundancy' and 'Self-cal'
by M.H.Wieringa
Experimental Astronomy, Vol 2, pp 203-225, 1992

1.4 ...

1.5 ...

NFRA/ITR-198a

NETHERLANDS FOUNDATION FOR RESEARCH IN ASTRONOMY

INTERNAL TECHNICAL REPORT 198a

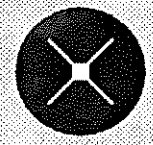
The N-series redundancy programs

Part 1 of 1: General introduction

by

W.N. Brouw

Initial version: 25 May 1991
Last update: 2 September 1992
File: USER5:[WIM.ITR]ITR2A.WNB



Netherlands Foundation for Research in Astronomy
Stichting Astronomisch Onderzoek in Nederland

1 Introduction

The N-series redundancy programs replace the R-series. The major difference between the two sets of programs is the ability to handle mozaicking, and to write AIPS UV-fits data. The programs run with the DWARF user interface on both VMS and Unix. Actual Unix implementations are available for teh Alliant (Summer 1991) and will be available on DEC and SUN workstations (Winter 1991). The implementation for Convexes is completely prepared.

The following programs are available:

NSCAN handle WSRT data (replaces RWTAPE and RSCAN)
NCALIB calibrate WSRT data (replaces REDUN and RLIGN)
NMODEL handle models of the sky (replaces RMODEL)
NMAP handle maps of WSRT data (replaces RMAP and RMFID)
NCLEAN clean maps (replaces RCLEAN)
NPLOT plot data, maps (replaces RPLOT and RMPLOT)
NGCALC do a variety of calculations on data (replaces RGPLOT)
NATNF reads tapes from the Australia Telescope Compact Array (ATCA)

The following dummy program (only for user interface purposes) is present:

NGEN to specify some general user parameters

Appendix A gives a short overview of the possibilities of the programs. For details the Cookbook (see J.E. Noordam) should be referenced.

Data is organised in files (called *nodes* in the programs). The following files are present:

data WSRT data; the file names end in SCN
maps sky maps; the file names end in WMP
model sky model; the file names end in MDL

Files can be found in directories (named sometimes *database* in programs), normally in the current user program.

Within a file sub-items are called sets (e.g. one spoke at one frequency of one field for a mozaick observation, or a map at one frequency). Sets are numbered contiguously, but are normally addressed by an index.

The details of the way the programs were written and how the files are organized can be found in the ITR197 series.

2 Running programs

A full explanation of all the DWARF possibilities can be found in the *"DWARF User's Guide"*. The explanation in this Chapter is concentrated on running the N-programs.

Before running the programs some symbols and logicals (aliases) should be known. Ask the System manager about the proper login commands, or just cp (COPY on VMS) .login and .cshrc (LOGIN.COM) from ~wnb/ (USER5:[WNB]).

Programs are normally run by typing (note that in the above, and in the following the Unix version is given; where different the VMS one is given in ()):

dwe program-name e.g. *dwe nscan*

or:

exe program-name e.g. *exe nscan*

Note: exe stands for execute, and can, and this is preferred, also be given as *dwe* or *dwexe*.

The program starts running, and asks questions that will specify the action of the program. All questions have some prompt information, and more help on the specific question can be obtained by typing a *?* as answer. If you still do not understand the question, see the cookbook and/or tell me to improve the help text.

The type of answer (value, name, option, yesIno, ...) depends on the question. Some answer should be a single value, some should be a list separated by commas. In the case of a numeric list, the answer can also be of the form: *start-value* [by *increment*] [to *end-value*]
There are three special answers that can always be given:

# or ^D (^Z)	end-of-file. The general action is to restart the asking of questions at a higher logical level in the program. In the case of options it indicates in general the *QUIT* option.
""	empty answer. In general this is taken to mean that no answer is given. Depending on the keyword this can mean to by-pass a certain action, or to go back to another level of questions. If the answer is essential, it will be repeated.
*	all. If interpretable it is taken as meaning "all possible values", otherwise the question is repeated, or a special default value is taken.

In general (but with the exception of obvious interactive programs) all questions for the program are asked and checked for consistency, availability of files etc, before the program starts actually executing.

After the program is finished a log of the program, including the questions and their answers and all data produced by the program, is spooled to the lineprinter.
On this log you will see the answers to some questions that were never asked. These hidden questions need normally no change, and are, therefore, not asked. Ways to change them are given in later sections.

2.1 Streams

It is sometimes useful to run the same program in parallel, or to run a program regularly with a specified set of answers. To be able to differentiate between the programs, programs can be run in different *streams*. For all practical purposes the program (and its parameters) have different names. Streamnames can be any alpha-numeric string, although for practical reasons integers may be preferred by the user:

'program-name$stream' (program-name$stream) e.g. dwe 'nscan$5' (dwe nscan$5)

The zero stream has a special meaning in specifying values across streams.

2.2 Asking all

All questions (including the hidden ones) will be asked if you run the program as:

dwe name /ask e.g. dwe nscan/ask

Asking can also be enabled for all runs of all programs by:

dws dwarf/nomenu
: ask=yes *: ask=yes*
: :)

2.3 Specifying answers

In addition to answering questions asked, answers can be also be specified before hand. This is especially handy if you want to run programs in Batch mode, or you want to make sure that for a set of programs the answer will be the same.

Answers can be specified before a program is run, by using the *dws* or *specify* command. To answer all questions:

dws prog-name e.g. *dws nscan*

Answering can be stopped at any time by giving ^D (^Z) as answer.

If you only want to specify a few answers:

dws prog-name/nomenu
: keyword=value
:

e.g. *dws nscan/nom*
: ha_range=-10,10
:

The prog-name can be replaced by '*prog-name$stream*' (*prog-name$stream*) to specify answers for running in a separate stream (the default stream is in general 1).

Note: the program name and/or stream can contain the wildcard *
Note: some questions are asked more than once in the program. To specify all answers, seperate them in dws with a ;. To make sure that all answers are given it is much safer to run a Dry-run or a Saved-run (see later).

In view of the above and the following sections, it is clear that answers can be given in many different ways. The program acts on all of these as follows:

is there a specification for this question in this stream?	*yes* use it
is there a specification for this question in stream 0?	*yes* use it
is there a specification for this question in NGEN in this stream?	*yes* use it
is there a specification for this question in NGEN for stream 0	*yes* use it
should the program ask the question	*no* use program default
ask user	use it

Note: NGEN only for special questions (next section)

2.4 Maintenance of answers

From the above it is clear that the variety of possibilities to specify answer to questions also makes the possibility of errors quite large. It is therefore recommended to make sure that programs are run as intended. To aid in maintaining the specified answers, run the following regularly:

dwv or *view*	to show all specified answers
dwv 'prog-name[$stream]/extern'* (*dwv prog-name[$stream]/extern)	to show all specifications for specified program and stream
dwc 'prog-name[$stream]'* (*dwc prog-name[$stream])	to clear all specifications for specified program and stream

2.5 Special questions

There are a set of special questions that are available for all programs. The answers to this questions are normally taken by default, but they can be specified. There are a number of ways they can be specified:

```
    dwe 'prog-name[$stream]/keyw[=value]'
or: dwe 'prog-name[$stream]/nokeyw'      will set the keyword for specified program and stream
    dws 'prog-name[$stream]/nomenu'      will set the keyword for specified program and stream
    dws 'prog-name$0/nomenu'             will set the keyword for specified program and all streams
    dws 'ngen[$stream]/nomenu'           will set the keyword for all programs in the stream
    dws 'ngen$0/nomenu'                  will set the keyword for all programs and streams
```

The special keywords are:

/LOG= spool|yes|no|append	specify if program log should be spooled (the name will be the first three letters of the program name (capital), followed by date, time, letter (capital), .LOG), made but not spooled (name will be PROG-NAME.LOG), not made, appended to PROG-NAME.LOG.
/LOG /LOG=spool	
/NOLOG /LOG=no	
/RUN	run the program
/NORUN	start the program, ask all questions, save all the answers (as if specified), specify for the program and stream RUN=YES and stop program. This can be used to prepare a batch run e.g.
/DATAB= directory	specify the directory in which the data reside (normally the current directory, i.e. ./ ([]))
/DATAB /DATAB=""	
/NODATAB /DATAB=""	
/INFIX= node shorthand	specify a part of the node name that is always the same (see later)
/INFIX /INFIX=""	
/NOINFIX /INFIX=""	
/APPLY=list-of-options	specify the corrections to be applied to data (see later for full explanation). The list can contain one or more of: ALL, NONE, RED, ALG, OTH, EXT, POL, FAR, MOD, IFR, MIFR , NORED, ... NOMIFR
/APPLY /APPLY=*	
/NOAPPLY /APPLY=NONE	
/DE_APPLY=list-of-options	
/DE_APPLY /DE_APPLY=NONE	
/NODE_APPLY /DE_APPLY=*	

2.6 Dry run

A program can have a dry-run, in which all questions are asked and all checks are done by:

dwe 'prog-name[$stream]/norun'* (*dwe prog-name[$stream]/norun)

In that case (except for obviously interactive program options) the program will not run, but will specify all answers to all questions for subsequent runs.

2.7 Saved run

All answers in any program run can be saved for later use by:

dwe 'prog-name[$stream]/save'* (*dwe prog-name[$stream]/save)

Note: more than one /name can be given at a dwe call

3 Data formats

The programs recognize three types of data files:

WSRT data (scans, SCN)
sky model data (models, MDL)
maps (maps, WMP)

These files are called *nodes* in a *database* (i.e. files in directory).
Node names are a series of alpha-numeric character strings separated by points, e.g.:

mynode
mynode.21cm.yesterday.s.x.c.d.file.dd

The maximum length is 80 characters (which means maximally 39 points). This node name is converted to a file name by appending .SCN, .MDL or .WMP, converting all . to _, converting all lowercase to uppercase, and replacing the _ nearest but less than 47 to a .. This file name is than prefixed with the current database (default nothing).
The database (e.g. ~wnb/data/ (USER5:[WNB.DATA]; ../other ([-.OTHER]) etc) can be specified as stated earlier, or can be included in the node definition:

~wnb/data/mynode (USER5:[WNB.DATA]mynode)
../other/mynode.21cm.yesterday.s.x.c.d.file.dd ([-.OTHER]MYNODE.21CM....)

If a database is specified in this way, it will be saved in such a way that all subsequent program runs in the same stream that have no explicit database set, will use it, including the currently running program at all its further node questions.
Parts of a node name can be set aside for short-hand use. This can be done by the INFIX keyword (see earlier), or in a node specification by enclosing a part in (). This enclosed part will be from then on available to all programs run in the current stream. Reference to the infix is made by typing a #. E.g.:

mynode.21(cm.yesterday.s.x.c.d.file.d)d	is the same node as above
mynode.92#e	will produce mynode.92cm.yesterday.s.x.c.d.file.de

Each node can contain many *sets* (model nodes will hold at the moment only one set). In SCN nodes a set is the data for one mozaicking field, at one frequency, done as a contiguous observation (a spoke); in WMP nodes a set is one 2-dimensional map.
In creating a node sets are given sequential numbers (0,1,...). The number of sets can be very great (e.g. a 64 point mozaicking with 256 line channels and spokes of 1 minute every hour over 16 settings of ABCD will have 64*256*12*16 3000000 sets. To ease the use, an index is provided. Additional indices can be made by the regroup or index options. This index is a series of 5 (SCN) or 6 (WMP) integers (0,1,...) separated by points, e.g.:

0.2.3.4.5
7.9.0.1.3.2

The index is structured, the different levels are for SCN:

job(e.g. input tape).subjob(e.g. tapelabel).moza-field.channel.spoke

where job, subjob and spoke are just running numbers, moza-field and channel the actual field and channel numbers.

For WMP:
job.moza-field.channel.polarisation.type.number

where all are just running numbers, except type which is 0 for map, 1 for ap, 2 for cover etc.

Sets (maps) are always specified as a list of definitions. The sum of all these definitions define the hypercube of data to be used in the program. Each entry in the list can be:

#value	a simple direct set reference
value.value...	a simple indexed reference. Omitted values are interpreted as .* (at end) or as * (in middle), e.g.:
..2 *.*.2.*.*.*.*.*	

Each value in the above two examples can be:

*	to indicate all possible values
val1-val2	range of val1, val1+1,..., val2
val1- val1-*	
val1-val2:val3	range of val1, val1+val3, ..., val2
val1:val3 val1-*:val3	

An example:
Let us assume we have a mozaicking observations at 4 settings of ABCD of 60 fields, 64 line channels. There will thus be 12 spokes per 12 hour per field. These data are on 4 tapes:
Tape 1: label 1 6 hours, label 2 6 hours, both for 36 m
Tape 2: label 1 12 hours for fields 30-59, label 2 for fields 0-29 both for 72 m
Tape 3: label 1 12 hours all fields for 48 m
Tape 4: label 1 12 hours all fields for 90 m

The set numbers will be unknown, but the indices generated are (if they are read in in order of tape and label):
Tape 1: 0.0.0-59.0-64.0-5, 0.1.0-59.0-64.0-5 Note: continuum channel 0 also present
Tape 2: 1.0.30-59.0-64.0-11, 1.1.0-29.0-64.0-11
Tape 3: 2.0.0-59.0-64.0-11
Tape 4: 3.0.0-59.0-64.0-11

If a map is wanted of field 31 using spacings 36 m and 72 m averaging all odd channels between 17 and 25, it could be specified as:
0..31.17-25:2,1..31.17-25:2
or: 0-1.*.31.17-25:2

A more detailed selection, in addition to database, node, sets, can always be made by specifying an hour-angle range and polarisation selection to select data within sets (spokes), or by setting a delete bit in scans to be skipped.

Sometimes a program should be run more than once for comparable data. In that case loops can be specified. A loop is a list of pairs of values. The first value indicates how often the loop should execute, the second specifies an increment to be given at each run to the specified sets. If in the above example maps were wanted for fields 31,35 and 39 and each one for odd channels 17-25 and 41-49, the loop would be specified (with the same set definition as above) as:

loops=3,..4,2,...24

If for fields 31, 35 and 39 a map had to be made of averaging resp. the odd channels 17-25, 23-31, 29-37:

loops=3,..4.6

3.1 SCN files

A set in a scan file is a contigous in time combination of scans of a certain field and frequency channel. A scan is all the data for all baselines and polarisations at a certain time. The number of baselines depends on the observing configuration, the number of polarisations is 1, 2 or 4 (always interpreted as XX, XX,YY , XX,XY,YX,YY.
A scan has a scan header containing corrections and a delete bit, a set has a set header containing information on polarisation corrections and model data. Each data point can be individually flagged.

Data comes from WSRT or reduction group tapes or is copied from other SCN nodes. In all these cases there can be corrections that are already applied to the data. These corrections are saved in the SCN file. All telescope based gain/phase errors are called *OTHER* corrections; additive interferometer corrections are called *IFR*; multiplicative interferometer corrections are called *MIFR*; model data are called *MOD*; and their are also *FAR*aday rotation, *REF*raction, *EXT*inction, *POL*arisation corrections.

The NCALIB calibration program can use the data as provided to set and/or calculate also the above specified corrections, were *OTH* is split into three parts: *RED*undancy, *ALiG*n and *OTH*er.

Before data is used in a program, the *APPLY* and *DE_APPLY* keywords are used to determine what should be done to the data as saved in the SCN file before it is used.

DE_APPLY	specifies corrections that were applied to the data before it went into the SCN file, and that should be undone (default: NONE)
APPLY	specifies corrections that were determined on the basis of the data in the SCN file, and that should be applied before the data is used (default:ALL)

Note that models are handled differently from one would think at first sight. I consider a sky-model as data that was applied to the cosmic noise before it was put in the SCN file. Hence, to subtract the model from the data specify MOD in DE_APPLY! MOD in APPLY will restore a model that was subtracted from the data before it was put in the SCN file.

3.2 MDL files

Model nodes contain information on sources. Sources have the following parameters:

intensity in WU
Q, U, V in %
l, m in offsets to a central position
extension in 2 perpendicular directions and direction of the major axis (assumed Gaussian shape)
rotation measure
spectral index
identification number
flags (e.g. if clean component or proper source)

Model nodes come in three flavours:

local	unknown central position and frequency
apparent	the central position (and l, m) are in apparent coordinates, and the frequency is known
B1950	the central position (and l, m) are in B1950 coordinates, and the frequency is known

Model nodes can be converted between flavours, in which case the l, m offsets, the spectral index and the rotation measure are changed.

The calculation of model data is time consuming. Therefore, a calculated model is saved in the SCN file for which it is calculated, together with the source list used. In all programs that use the model data it can be specified how the model calculation should be done, and if the calculation should be saved. The relevant question is *MODEL_ACTION*, which expects a list of three answers. The first one can be one of:

merge	replace the model saved in the SCN node with the one specified by the user. However, first compare these two lists, and only add the difference to the saved model data. I.e. make the calculation as short as possible if the new list differs only slightly from the saved one.
add	replace the saved model with the sum of the saved one and the one specified by the user. The model calculated on the basis of the user specified list is added to the saved data.
new	replace the saved model by the model specified by the user, and calculate a completely new set of model data
temporary	do not use any data in the SCN node, or write anything, but use the data based on the list specified by the user
increment	use the saved model data, and add to it the model data based on the user specified list, but do not save anything

The second answer can be *BAND* or *NOBAND*, and specifies if in the model calculation source data should be corrected for bandsmearing to match the actual data better.
The third answer can be *TIME* or *NOTIME* to indicate the use of integration time smearing.

3.3 WMP data

Each map in a WMP node consists of a 2-dimensional array of map values (even length of coordinates), and is connected to a map header. A hypercube of maps can thus have different sizes in all dimensions.

A Program options

The following describes for each program shortly the different options. For details read the Cookbook or run the programs.

A.1 NSCAN

DUMP	dump WSRT tape/optical disk onto disk
LOAD	load WSRT data freom tape/optical disk/disk into SCN node
COPY[1]	copy sets from SCN node to same or other
FROM_OLD	convert R-series SCN files to new format
TO_OLD	convert new format SCN nodes to R-series format
SHOW	show and (optionally) edit any data in SCN node
DELETE	(un-)flag data in SCN node
REGROUP	make another index to sets in SCN node
UVFITS	write SCN node data to tape/disk in AIPS UVfits format
PFITS	print a summary of any FITS tape/disk file, showing all keywords and a limited set of data
CVX	convert a SCN node from one machine's data format into the current machine's dataformat. Note: if a SCN node is copied from one machine to another in FTP, always set *binary* (=*image*) mode
NVS	convert a SCN node to a newer version if a program change necessitates it (you are always warned if you have to do this). Also: calculates MJD and precession rotation angle if not present or in error claculates XY type polarisation from LINOBS Stokes output
WERR	WE0 correct online system < 61 tape mosaic error WE1 correct hour angles in scans

A.2 NMODEL

HELP	some explanation on model lists
FIND	find point sources in map(s)
UPDATE	update source positions and intensity from SCN data
XUPDATE	update source extensions from SCN data
CONVERT	convert model list from one flavour to another
BEAM	correct model list for primary beam attenuation
DEBEAM	apply primary beam attenuation to model list
SAVE	save model and model data in SCN node
GET	get a model list from a SCN node
FROM_OLD	convert R-series MDL files to new format
TO_OLD	convert new format MDL nodes to R-series format
CVX	convert a MDL node from one machine's data format into the current machine's dataformat. Note: if a MDL node is copied from one machine to another in FTP, always set *binary* (=*image*) mode
NVS	convert a MDL node to a newer version if a program change necessitates it (you are always warned if you have to do this)
HANDLE	make a user specified list: READ read MDL node into list WRITE write list to MDL node CLEAR clear list ZERO empty list, but keep central coordinates and frequency

1. Option not yet completely written at time of writing this ITR

SHOW		show list
LIST		show list
RSHOW		show list in RA, DEC i.s.o. l, m positions
RLIST		show list in RA, DEC i.s.o. l, m positions
TOT		show statistics of list
ADD		add sources in list
RADD[2]		add sources in list based on RA, DEC
CALIB		convert the list by scaling intensities and/or moving l, m positions
EDIT		edit source list
FEDIT		edit a field in selected sources in list
SORT		sort list in decreasing intensity
FSORT		sort list on specified field or value
DEL		delete sources in list
DCLOW		delete clean components below a limit
DAREA		delete sources in specified area
DNCLOW		delete non-clean components below a limit

A.3 NCALIB

REDUNDANCY	calculate redundancy, align or selfcalibration corrections. Actual type depends on presence of a model.	
POLAR	polarisation corrections:	
	CALC	calculate polarisation corrections
	SHOW	show polarisation corrections
	COPY	copy polarisation correction from one set to others
	EDIT	edit polarisation corrections
	ZERO	zero corrections
	VZERO[3]	assume V=0 and calculate X-Y phase difference and apply it to corrections
		CALC calculate and show
		APPLY calculate, show and apply
		ASK calculate, confirm, apply
		MANUAL apply user value
		COPY[4] calculate from input, apply to output
		SCAN[5] calculate and apply per scan
SET	set some correction data:	
	ZERO	zero selected corrections
	COPY	copy corrections from sets(s) to set(s)
	MANUAL	copy OTHER corrections from manual input
	LINE	copy all corrections from corresponding continuum channel
	EXTINCT	set extinction correction
	REFRACT	set refraction correction
	FARADAY	set Faraday rotation correction
	RENORM	renormalise telescope based errors

A.4 NMAP

MAKE	make map(s)
SHOW	show and optionally edit any data in map
SHOW	calculate noise in (part of) map

2. Not yet implemented
3. Not fully implemented
4. Not yet implemented
5. Not yet implemented

FIDDLE	play with maps:	
	ADD	add two maps with weighting factors
	AVERAGE	average two maps with weighting factors
	SUM	add a series of maps:
		SUM add maps
		NSUM add maps weighted with normalisation sum
		BSUM add maps weighted with bandwidth
		BNSUM add maps weighted with bandwidth and normalisation sum
		FSUM add maps weighted with user specified factors
		NSSUM add maps with weights based on map noise
	MOSCOM	combine mosaic maps into one
	POL	make (Q +U)
	ANGLE	make ½arctan(Q/U)
	EXTRACT	extract part of map
	COPY	copy map to other map
	BEAM	correct map for primary beam attenuation
	DEBEAM	apply primary beam attenuation to map
	FACTOR	multiply map with factor
W16FITS	write 16-bit FITS tape	
W32FITS	write 32-bit FITS tape	
FROM_OLD	convert R-series map files to new format	
TO_OLD	convert new format map nodes to R-series format	
CVX	convert a map node from one machine's data format into the current machine's dataformat. Note: if a map node is copied from one machine to another in FTP, always set *binary* (=*image*) mode	
NVS	convert a map node to a newer version if a program change necessitates it (you are always warned if you have to do this)	

A.5 NCLEAN

HISTO	produce histogram of map and/or beam
BEAM	do Högbom type clean
UVCOVER	do Clark clean
UREST	restore a source list in map with a clean beam
COMPON	get a list of the largest clean components

A.6 NPLOT

MAP	plot maps
DATA	plot SCN data
MODEL	plot MDL data
TELESCOPE	plot telescope based corrections
RESIDUAL	plot interferometer residual data

A.7 NGCALC

NODE	switch extracted data node
EXTRACT	extract data from scan headers
PLOT	plot extracted data
MONGO	produce MONGO array from extracted data
COPY	copy extracted data sets
BRIEF	index of all extracted data
SHOW	detailed listing of extracted data
MERGE	merge extracted data sets into a new one

TRANS	interchange frequency and HA axes
CALC	do calculations on extracted data:
	AVERage get average value
	SMOOTH smooth (triangular weight) extracted data
	POLYfit and subtract polynomial from data
	DPOLY subtract specified polynomial from data
	NULLlear some data
COMBINE	specify an expression to execute on extracted data files
CVX	convert data format from one machine to another
NVS	get newest version of datasets

A.8 NATNF

LOAD	load ATCA tapes into SCN file

CONTENTS