

# Improvements in wide-field imaging

T.J. Cornwell, NRAO

January 4, 1993

## 1 Introduction

In the last few months, I have made a number of improvements to the SDE software for widefield imaging. I have also made progress in decreasing the noise level of 327 MHz images. As a result of these changes, it is now possible to reach a noise level of about 0.2-0.3 mJy/beam in a long-track B-configuration observation of a fairly empty field away from the Galactic Plane. On the Galactic plane, the noise is considerably higher (up to an order of magnitude). Running on a single IBM RS 6000/560 (aztec), this processing takes about 6-12 hours of wallclock time. Running in parallel on 4 such machines (aztec, apache, navajo, hopi), the efficiency of parallelization is about 75% of the theoretical best case.

The purposes of this memo are: to document the changes in the algorithm, to describe the resulting improvement in noise level, to present my analysis of the effect of confusion on the noise level achieved, and to describe some experiments in parallel processing using the wide-field imaging algorithms.

## 2 Changes in fly

The old wide-field SDE program was called `fly`. It implemented the *polyhedron* algorithm (Cornwell and Perley, 1992) that works as follows:

1. The region of sky to be imaged is split into a *polyhedron* of having a large number of facets, each of which touches the celestial sphere at one point.
2. A residual image is calculated for each facet,
3. If the maximum residual is below a stopping point jump to step 8,
4. A clean limit is chosen as the peak sidelobe level times the peak brightness in any one facet,
5. All facets are cleaned down to the clean limit,

6. The new components for each facet are subtracted from the visibility data to form a residual visibility data set,
7. Return to step 2
8. Smooth clean components for each facet,
9. Sum smoothed clean component images into a large single image having the geometry of the tangent plane of the central facet,
10. Sum residual images into a large single image having the geometry of the tangent plane of the central facet,
11. Correct residual image for overlaps of facets,
12. Add smoothed clean components image and residual image to get final clean image.

The new program in SDE has the alarming name *dragon*. Compared to *fly*, the major changes required were:

**Clean algorithm changes** In the Clark CLEAN, the switch between minor and major cycles is controlled by the sidelobes from the brightest source in the field. For fields dominated by a single point source, this works well, but for the large number of point sources seen at low frequencies, the rule must be modified to sum the sidelobes from all remaining sources incoherently. This can only be done after the deconvolution is finished and one knows the strength and location of all sources. In principle, one could use the source counts to determine this switching but a good enough approximation comes from just summing the effects of the dominant source in all facets brighter than the median. The net effect of this change is to force the algorithm to change to a major cycle more often.

**Targeted slow transforms** Each facet touches the celestial sphere at only one place. Point sources away from this favored location show distinctive error patterns at moderate to high dynamic range. Reducing the facet size helps a lot but at the cost of a considerable increase in computing time. A smarter approach is to perform slow transforms for point sources above some brightness level (about 10 mJy/beam seems to work well).

**Incorporation of self-calibration** The MAP algorithm in the Jodrell Bank system, OLAF, incorporates self-calibration into an MX-type CLEAN with considerable savings in CPU time. The idea is that in the conventional cycle of self-calibration/deconvolution, the components found at high flux levels change very little after self-calibration and so it is pointless to redo the deconvolution step of the brightest points. In the *dragon*

program, the self-calibration is performed every time the maximum residual drops below one of a sequence of thresholds set by the user. Typically these thresholds start just above the first significant error and drop by a factors of two or three down close to the finally stopping point. I wasted a fair amount of time trying various schemes for setting these thresholds automatically but I could not come up with a robust scheme that worked for both point sources and extended objects. The mechanism for setting the thresholds by hand is such that it is easy to specify phase-only solutions for the first few levels, switching to amplitude and phase at a lower flux level. The final self-calibrated dataset is written out to disk as a FITS file and so the program can be restarted with this as the input dataset. If this is done then the image sometimes improves noticeably. The explanation must be that deconvolution errors are made at high flux levels and never corrected subsequently. The same behaviour was apparent in the OLAF MAP program and so a restart with no selfcalibration was often useful in improving the final image quality. Some further work in setting selfcal thresholds might help alleviate this problem.

**Different time-scales for amplitude and phase** Amplitude self-calibration is finicky. A long time scale averages over the effect of any model errors on the amplitude determination but can lead to problems if the phase coherence timescale is too short. For this reason, I solve for and correct phase on a time scale  $\tau_{\text{phase}}$  and then both amplitude and phase on a perhaps longer time scale  $\tau_{\text{amp}}$ .

**RR and LL self-calibration** This is necessary but was not present in the previous self-calibration routines. A nice side benefit is that the Stokes V image for a reference field can be monitored. This usually shows a substantial drop in the first self-calibration cycles followed by smaller but still significant subsequent drops.

**Parallel processing** The general *polyhedron* approach to wide-field imaging is clearly well-suited to implementation on coarsely-grained parallel processing computers. Using the Parallel Virtual Machine (PVM) package from Oak Ridge National Laboratory, I have included parallel processing in *dragon*. Most computation is performed on a single host, with the exception of the transforms to and from the (ungridded) visibility data and interpolations between different coordinate systems, all of which are farmed out to a number of server processes that can run on any machine for which SDE is available. Two of new routines are listed in Appendix C. Overall about 1200 new lines of code were required. The PVM package is nicely put together and I found that incorporating it into SDE was quite straightforward. PVM is most appropriate for systems such as SDE that have a large amount of existing code. For new systems, a higher level system is available from the same people. This is called the Heterogeneous

Computing Environment (HeNCE) and is built on top of PVM.

Since `dragon` itself runs most quickly on the IBMs, the parallel version is best suited to our network of IBMs rather than to the more numerous SPARCs. When running on all four I see a reduction in wallclock time approaching the theoretical best case (see below for more on the timings).

### 3 The importance of large fields

In both `fly` and `dragon`, the computing cost is roughly proportional to the solid angle of the cleaned field of view. Hence minimizing the field of view helps control the computing time required. Since the primary beam at 327MHz is filled with confusing sources, the area to be cleaned must be of the same order as the primary beam size. Both time-averaging and bandwidth smearing also provide suppression of point sources far from the phase center and so might also provide some help in limiting the field of view that must be cleaned. Some time ago Barry Clark calculated some of these effects and came up with some answers that seemed quite encouraging in the size of the field of view and the number of clean components required. Since these conclusions did not seem to be in accord with my experience in making low noise images, I have redone the calculation in a slightly more elaborate way using MathCad. The analysis is quite straightforward. Each uncleaned source in the primary beam contributes to the noise level in the final image. Assuming that the sidelobes add incoherently, it is straightforward to sum the effect of all sources described by the source counts, adjusting for any which might have been subtracted using `clean`. A detailed description is presented in Appendix C. The main conclusions are:

1. The 16 fields of the MX algorithm in AIPS reduce the noise by a factor of 4 (to somewhere in the range 1-5 mJy/beam). *For images of compact sources that are dynamic range limited to this level or above, MX is entirely satisfactory.*
2. Many hundreds of point sources must be subtracted to remove completely the noise power due to sidelobes of confusing sources and attain a noise level close to the theoretical limit. Only the `dragon` program can currently do this properly for C and B configurations.
3. Bandwidth and time-averaging smearing do not help as much as might be hoped. This disappointing behaviour is due to the fact that both of these effects merely smear flux (instead of removing it as does the primary beam).
4. In B-configuration, the noise achieved at 327MHz could be limited by the sidelobes of confusing sources far outside the main lobe of the primary beam. However, the accuracy of my calculations is probably worse than

the factor of 2-3 discrepancy between the theoretical and observed noise. Increasing the accuracy of this prediction requires:

- (a) better knowledge of the primary beam (something that will be hard given the effects of confusion!). At the moment this is by far the dominant cause of uncertainty.
- (b) better knowledge of the source counts (something available from B-configuration observations)
- (c) better numerical calculations of smearing effects. Simulations presented in the next section address this point further.

## 4 Examples

I show four examples of images processed using `dragon`:

**D-configuration blank field** To give some idea of a typical 327MHz primary beam field, I show a nondescript field observed in D-configuration (see figure 1) The image was made from a 7 by 7 grid where each facet was 100 by 100 pixels each of cellsize 60 arcsec. The resolution is about 3 arcmin. The noise level reached in a facet far from the phase center was 2.3 mJy/beam. Using a loop gain of 0.1 and cleaning to a stopping level of 10 mJy/beam, 5608 components totalling 13.11 Jy were found. The field of view is much larger than necessary to reach this noise level and was chosen to reveal the size of the region over which sources may be found.

**A simulation** I used the 327 MHz number counts as described in Appendix C to construct an artificial sky at arcsecond resolution. The SDE program `vissim` was then used to construct simulated data, performing an exact Fourier summation for all brightness. Data were simulated for an object at declination 30 degrees for hour angles ranging from -6 hours to 6 hours. Calibration errors of 30 degrees in phase and 20% in amplitude were then added to each antenna for each integration time of 180 seconds, and the correct level of noise was added. The errors were independent in RR and LL, which is probably a bad assumption in practice but provides a good test. The final noise was 0.21 mJy/beam in Stokes I, and 0.11 mJy/beam in Stokes V. The maximum brightness in the image was about 2.7 Jy/beam, so the final dynamic range exceeds 12000. I include only a detail of this image since it all looks much the same (see Figure 2).

I also plot the noise achieved as a function of the field of view (Figure 3). Note that just as in the analysis presented in Appendix C, this curve shows lumps corresponding to the sidelobes of the primary beam. The trend of the curve for large fields of view shows some signs of converging to a lower number closer to the Stokes V noise of 0.11 mJy/Beam.

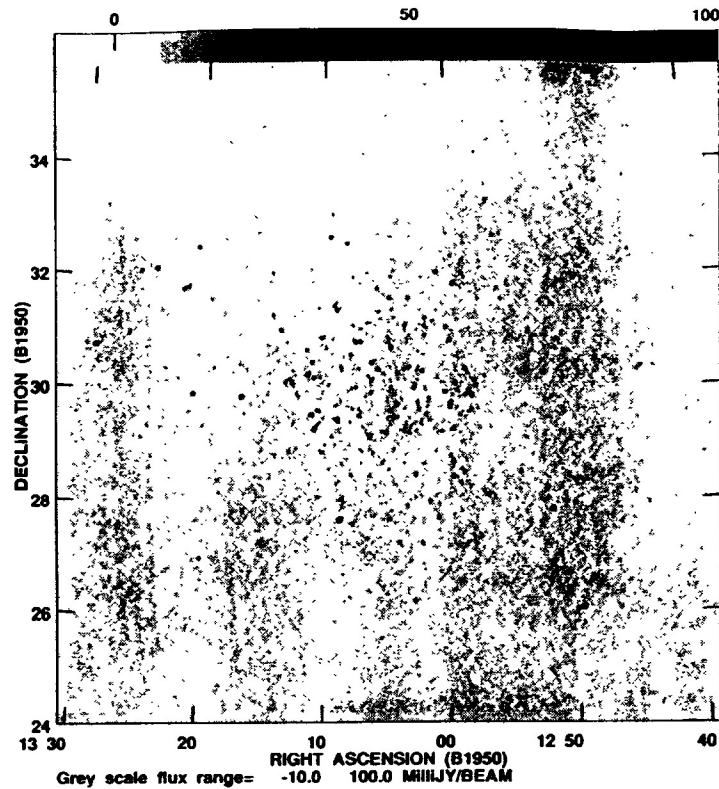


Figure 1: Full image of a nondescript field observed in D-configuration. The grey scale runs from -10 to 100 mJy/beam. The noise level is 2.3 mJy/beam.

This example illustrates some of the pitfalls of parallelization. After the transforms to and from the visibility data, the next most time consuming single step in the processing is the interpolation of all facets onto a single tangent plane. If the interpolation is done serially, dragon takes 8 hours and 40 minutes (real time) on one machine and 4 hours on four machines. Thus the parallelization efficiency is about 54%.<sup>1</sup> Excluding the interpolation step, the wall-clock time drops to 6 hours 40 minutes (serial) and 2 hours (parallel) minutes and so the efficiency rises to about 83%. With this in mind, I parallelized the interpolation step so that the overall parallel run time (now including interpolation) became 3 hours. Comparing

<sup>1</sup>I define the parallelization efficiency to be ratio between wall-clock times divided by the number of processors:  $T_{\text{serial}}/(N_{\text{proc}}T_{\text{parallel}})$ .

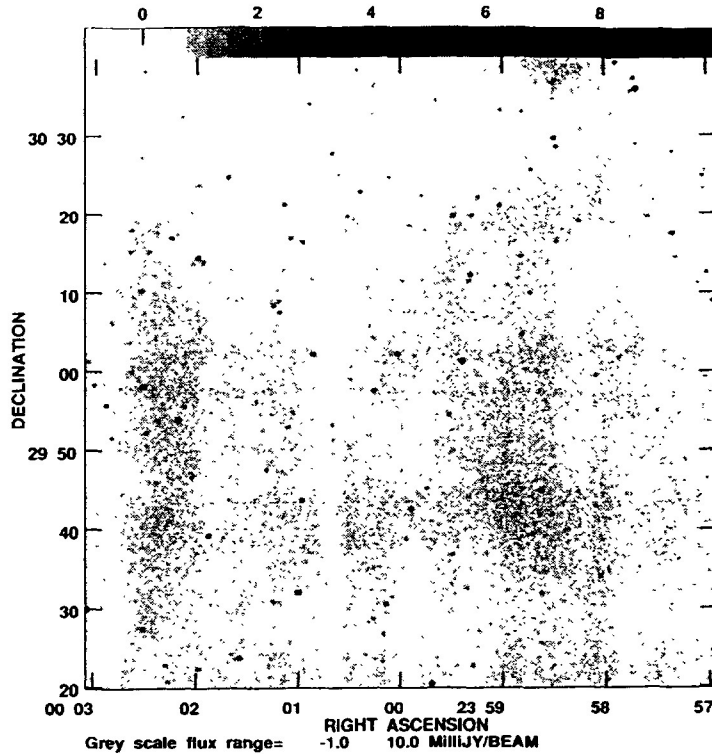


Figure 2: Detail of a simulated field observed in B-configuration. The grey scale runs from -1 to 10 mJy/beam. The noise level is 0.21 mJy/beam.

this to the number of 8 hours 40 minutes, I find a final efficiency of 72%. So effectively, one cpu of the four is consumed in handling the overhead due to running in parallel. I think that with more work this could be improved, probably up to the 85% level seen if interpolation is excluded.

**M30** I have processed a large number of real observations using *dragon*. A detail of an image of the M30 globular cluster is shown in Figure 4. The initial field of view (13 by 13 facets each of 210 by 210 pixels of cellsize 6 arcsec) was too small and the sidelobes of a confusing source were seen. I then had to expand to 15 by 15 facets. The noise level achieved was 0.25 mJy/beam. The brightest single source in this field is only about 400 mJy/beam so the dynamic range is slightly less than 2000. The large field of view made this example particularly time-consuming. Even after

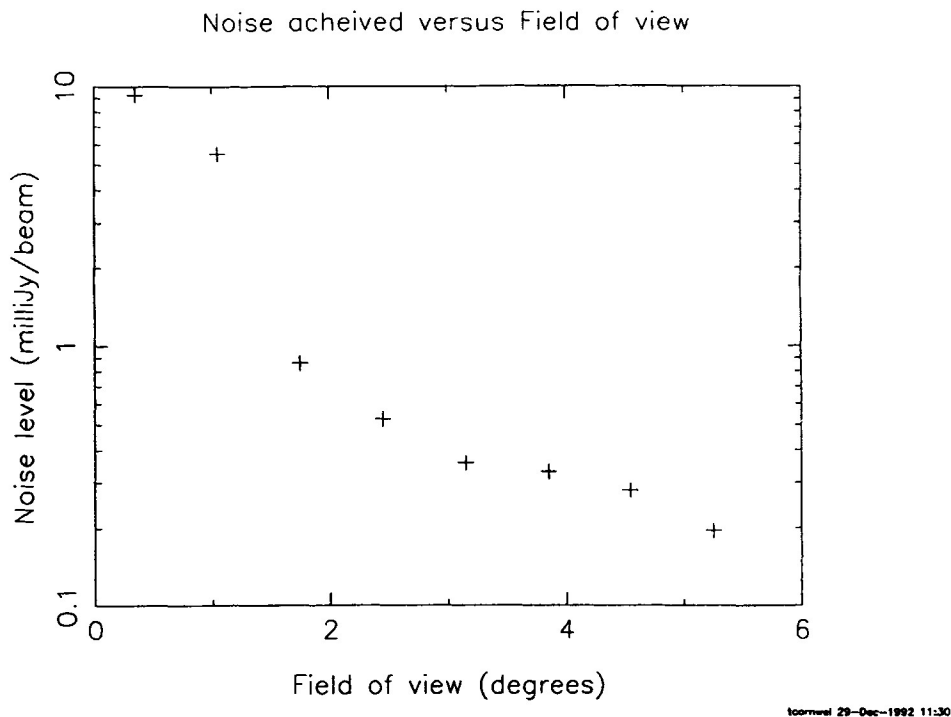


Figure 3: Noise as a function of cleaned FOV for the image shown in Figure 2. The field of view is defined as the length of one side of the image.

averaging the data to 3 minutes integration time, this run of dragon took about 7 hours 30 minutes of wall-clock time to complete, processing in parallel on two IBM RS/6000 560 machines. The short integration time helps beat down the effects of confusing sources, but it does require a stable ionosphere. If a shorter integration were necessary then the computing time could grow proportionately larger.

**A Galactic plane source** The two previous cases were images of a large number of point sources. Dale Frail has data on G5.4-1.2 in B-C hybrid configuration. This image shows a number of quite extended objects (Figure 5). The noise level achieved here, only 2-3 mJy/beam, is substantially poorer than can be achieved in many empty fields away from the Galactic plane but, for comparison, using multiple fields in MX, the best noise achieved was 20mJy/beam.



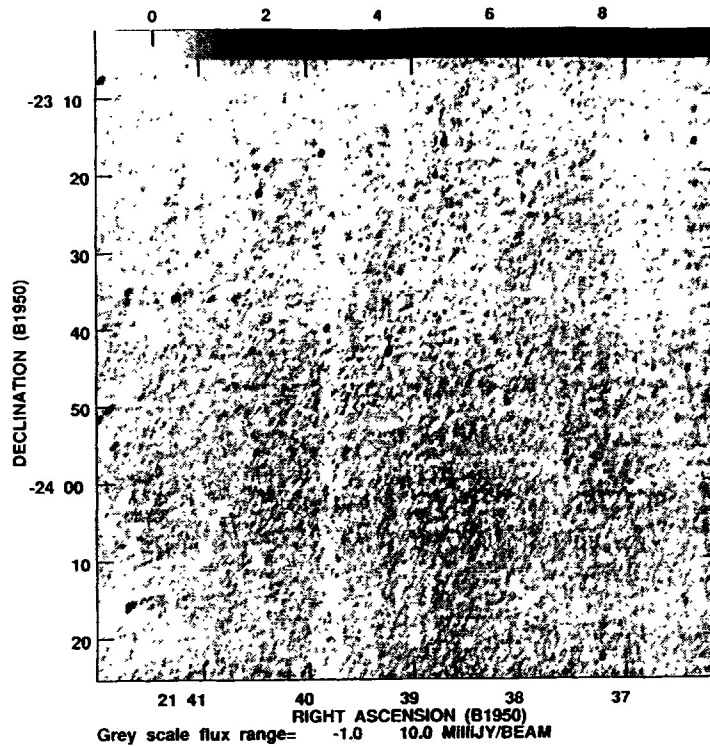


Figure 4: Image of the M30 field.

## 5 Discussion

Wide field imaging seems to be necessary to obtain noise levels approaching 0.2-0.3 mJy/beam in C and B configurations. Brighter sources that are dynamic range limited to noise levels in the range 1-5 mJy/beam can be imaged using standard small field methods since the sidelobes of the usual population of confusing sources are then insignificant.

After proper analysis of the effect of sidelobes of confusing sources, I found that the size of the field of view that must be cleaned of such sources is discouragingly large. The most important consequence is that obtaining low noise images in A-configuration will be hard. I had harbored hopes that processing of A configuration data would be possible using *dragon*. Unfortunately, with the need to clean a large field of view the image sizes become far too large:

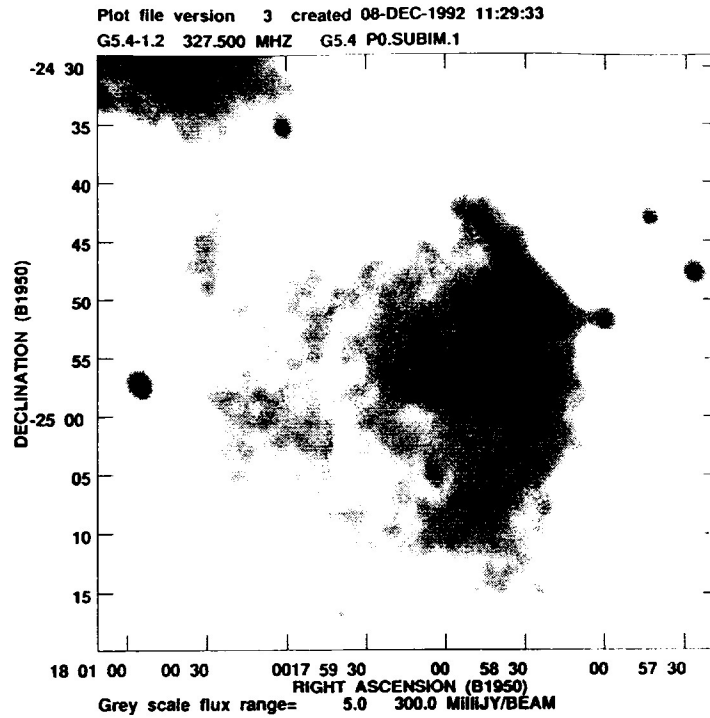


Figure 5: Detail from an image of G5.4-1.2. The grey scale runs from 5 to 300 mJy/beam.

probably 10000 by 10000. It is inefficient to clean this many pixels when most are empty so I plan to try a scheme in which a large number of small patches are placed only where a lower resolution image shows a point source. This will be a "super-MX" in which the number of patches could be several thousand.

I have seen no convincing evidence of non-isoplanatism at 327MHz in B, C and D configurations. This means that sidelobes of all sources in the primary beam can be removed successfully by the straightforward self-calibration scheme used in dragon. On going to lower frequencies, non-isoplanatism must occur for some range of parameters. In the absence of a non-isoplanatic self-calibration algorithm, the noise will then be limited by the sidelobes of confusing sources. At 74 MHz, I compute this sea of noise to be in the range 0.1-1 Jy/beam (for 27 antennas). Since decreasing the sidelobe level helps proportionately, multi-

frequency synthesis (Conway, Cornwell, and Wilkinson, 1990) may help if clear bands can be found.

## Acknowledgements

I thank Tony Willis for bringing PVM to my attention. Rick Perley helped tremendously by arranging for the allocation of computing resources for this development.

## References

- Conway, J.E., Cornwell, T.J., and Wilkinson, P.N., 1990, MNRAS, **246**, 490-509.  
Cornwell, T. J., Perley, R. A., 1992, A&A 261, 353  
Högbom, J., 1974, ApJS 15, 417

## Appendix A: Using the SDE software

SDE is designed for development of algorithms like that described here. Hence it is not very friendly in things like help files, format of inputs and so on. Nevertheless, it is fairly simple to use for some operations. I encourage those interested to play around with the `dragon` program either on their own data or on a standard dataset in the SDE area.

1. The environment must first be set up correctly to allow the SDE programs to be invoked. On a workstation in the Korn shell do:  

```
. /zia/u/sde/sdeini
```
2. Export your visibility data to SDE in FITS format, with a single IF, continuum only.
3. For testing, you can use the dataset `/zia/u/sde/data/BF7`. The inputs file for this dataset is in `/zia/u/sde/test/dragon.cur`. Copy the data file and the inputs file to your current directory if you want to try this out.
4. Type `dragon` to start the program. `inputs` will reveal the current parameters. `go` goes.
5. The program runs for a long time, and then writes an output FITS image, and, optionally, the self-calibrated dataset.
6. You can use the program `saoimage` to view the image.

7. The program `flydoc` advises on sensible inputs for various configurations.
8. To try out the parallel processing, see me first for further details.

## Appendix B: SDE parallel processing routines

Parallel processing is incorporated into `dragon` using two routines, a host `vistoimg` and a server `vistoimgs`. All routines beginning with the letter `F` are PVM routines. The others are SDE routines. Further optimization of these routines is possible but may not gain very much. For example, some reduction in the network traffic could be gained by reducing the size of the image template that is transmitted.

```

SUBROUTINE VISTOIMGH (NPAR, VIS, SUB, IMG, GRID, PSF)
C
C Parallel interface to VISTOIMG
C
C VISTOIMGH provides a parallel interface to VISTOIMG. A number
C of images can be made from the same data set. If PVM is available
C and the parallel system is enabled then NPROC servers are
C instantiated (NPROC is the number of processors available).
C Each server is primed with visibility data and then fed image
C templates. As each image is calculated, it is sent back and stored
C in the appropriate location. Every time a server returns with a
C result, it is given a new template until all the work is done.
C If PVM is not available or enabled then the calculation is done
C serially.
C
C NPAR INT input Number of images
C VIS CH*(*) input Name of visibility set
C SUB CH*(*) input Name of sub-class to grid e.g. OBS/I
C IMG CH*(*) input Name of output images
C GRID CH*(*) input Name of grids for gridding
C PSF LOG input TRUE for PSF calculation
C
C Audit trail:
C Original version: Audit trail comments go on this line
C and successive lines
C T.J.Cornwell Dec 1 1992
C-----
#include "stdinc.h"
C
C Arguments go here
C

```

```

        INTEGER NPAR
        CHARACTER*(*) VIS, SUB, IMG(*), GRID(*)
        LOGICAL PSF(*)
C
        CHARACTER*(*) ROUTINE
        PARAMETER (ROUTINE = 'VISTOIMGH')
C
        INTEGER I
#if PVM
C
        CHARACTER*(SYSMKNAM) STRM2, STRM3, SERVER
C
        INTEGER ID(1000), INFO, MSGTYPE, HOSTID, J, SID, LEN
        INTEGER MSGTYPES(2), NPROC, IPROC, IDONE, NFORMAT, IPAR
        DATA MSGTYPES /11,999/
C=====
        IF (ERROR) GO TO 999
C
C Is the parallel system enabled?
C
        IF(.NOT.SYSPAR) THEN
            GO TO 800
        ENDIF
C
C Enroll with PVM
C
        CALL FENROLL ('vistoimgh\0', HOSTID)
        IF(HOSTID.LT.0) THEN
            CALL MSGPUT ('PVM failure in fenroll as host, do serially',
$           'I')
            GO TO 800
        ENDIF
C
C Find the number of processors available. We only start one
C server per processor.
C
        CALL FPSTATUS(NPROC, NFORMAT, INFO)
        NPROC = MIN(NPAR, NPROC)
C
C Now set up the NPROC server instances we need
C
        DO 5 IPROC = 1, NPROC
            CALL FINITIAATE('vistoimghs\0', '\0', ID(IPROC))
            IF(ID(IPROC).LT.0) THEN

```

```

        CALL MSGPUT ('PVM failure to initiate, do serially', 'I')
        DO 15 J = 1, I-1
            CALL FTERMINATE ('vistoimgs\0', ID(J), INFO)
15      CONTINUE
        CALL FLEAVE()
        GO TO 800
    ENDIF
5  CONTINUE
C
C Send visibility data globally to all servers as message 1
C
    CALL FINITSEND()
    CALL CRDPWRIT (STRM2(VIS,SUB))
    CALL ARRPPWRIT (STRM3(VIS,SUB,'VIS'))
    CALL ARRPPWRIT (STRM3(VIS,SUB,'WT'))
    CALL ARRPPWRIT (STRM2(VIS,'UU'))
    CALL ARRPPWRIT (STRM2(VIS,'VV'))
    CALL ARRPPWRIT (STRM2(VIS,'WW'))
    CALL FSND ('vistoimgs\0', -1, 1, INFO)
    IF(INFO.LT.0) THEN
        CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM send error')
        GO TO 990
    ENDIF
C
C Prime NPROC servers to do the work by sending each an initial
C image template. We also send the number of each image so that
C when returned we know where to put it.
C
    IPAR=0
    IDONE=0
    DO 8 IPROC = 1, NPROC
        IPAR=IPAR+1
        CALL FINITSEND ()
        CALL FPUTNINT (IPAR, 1, INFO)
        IF(PSF(IPAR)) THEN
            CALL FPUTNINT (1, 1, INFO)
        ELSE
            CALL FPUTNINT (0, 1, INFO)
        ENDIF
        CALL DATPWRT (IMG(IPAR))
        IF(ERROR) GO TO 990
        CALL FSND ('vistoimgs\0', ID(IPROC), 2, INFO)
        IF(INFO.LT.0) THEN
            CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM send error')

```

```

                GO TO 990
            ENDIF
        8    CONTINUE
    C
    C Now start loop: First we look for a result. If ok then we
    C get it and start another using the server that just reported.
    C Message 11 is ok, message 999 means an error has occurred in the
    C server.
    C
    20    CONTINUE
        CALL FRCVMULTI (2, MSGTYPES, MSGTYPE)
        CALL FRCVINFO(LEN, MSGTYPE, SERVER, SID, INFO)
        IF(MSGTYPE.EQ.11) THEN
            CALL FGETNINT (I, 1, INFO)
            CALL ARRPREAD (IMG(I))
            IDONE=IDONE+1
            IF(IPAR.LT.NPAR) THEN
                IPAR=IPAR+1
                CALL FINITSEND ()
                CALL FPUTNINT (IPAR, 1, INFO)
                IF(PSF(IPAR)) THEN
                    CALL FPUTNINT (1, 1, INFO)
                ELSE
                    CALL FPUTNINT (0, 1, INFO)
                ENDIF
                CALL DATPWRT (IMG(IPAR))
                IF(ERROR) GO TO 990
                CALL FSND ('vistoimgs\0', SID, 2, INFO)
                IF(INFO.LT.0) THEN
                    CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM send error')
                    GO TO 990
                ENDIF
            ENDIF
        ELSEIF(MSGTYPE.EQ.999) THEN
            CALL ERRREPOR (ERRLOGIC, ROUTINE, 'vistoimgs failed')
            GO TO 990
        END IF
        IF(ERROR) GOTO 990
        IF(IDONE.LT.NPAR) GOTO 20
    C
    C Trace any SDE errors
    C
    990 IF (ERROR) CALL ERRTRACE (ROUTINE)
    C

```

```

C Kill all servers
C
  DO 30 IPROC = 1, NPROC
    CALL FTERMINATE ('vistoimgs\0', ID(IPROC), INFO)
  30 CONTINUE
C
C Un-enroll from PVM
C
  CALL FLEAVE()
  GO TO 999

C
C Alternate path: Do in series
C
  800 CONTINUE
  DO 10 I = 1, NPAR
    CALL VISTOIMG (VIS, SUB, IMG(I), GRID(I), PSF(I))
  10 CONTINUE
C
  999 CONTINUE
  END

#else
C=====
  IF (ERROR) GO TO 999

C
C No PVM (alas!): do in series
C
  DO 10 I = 1, NPAR
    CALL VISTOIMG (VIS, SUB, IMG(I), GRID(I), PSF(I))
  10 CONTINUE
C
  990 IF (ERROR) CALL ERRTRACE (ROUTINE)
C
  999 CONTINUE
  END

#endif

```

The server is as follows:

```

SUBROUTINE SDEMAIN

C
CD Vistoimg PVM interface: called only by PVM
C
C VISTOIMGS is called by VISTOIMGH as a server. It is first
C fed a visibility dataset and then a sequence of image templates
C are passed. For each template, an image is made using VISTOIMG

```



```

C and the resulting image is passed back. The calling routine
C must explicitly kill this server when it is no longer required.
C
C Audit trail:
C Original version: Audit trail comments go on this line
C and successive lines
C T.J.Cornwell Dec 1 1992
C
C-----
C
C#include "stdinc.h"
C
C      CHARACTER*(*) ROUTINE
C      PARAMETER (ROUTINE = 'VISTOIMGS')
C
C      INTEGER PSF, DIR
C      INTEGER ID, HOSTID, LEN, INFO, MSGTYPE, INST
C      CHARACTER*(SYSMXNAM) HOST
C=====
C
C Turn off message system
C
C      SYSMSG=.FALSE.
C
C Enroll in PVM
C
C      CALL FENROLL('vistoimgs\0', ID)
C      IF(ID.LT.0) THEN
C          CALL ERRREPOR (ERRLOGIC, ROUTINE,
C$          'PVM failure in fenroll as node')
C          GO TO 999
C      ENDIF
C
C First read the visibility data that was broadcast globally as
C message 1
C
C      MSGTYPE = 1
C      CALL FRCV(MSGTYPE, INFO)
C      IF(INFO.LT.0) THEN
C          CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM Receive error')
C          GO TO 990
C      ENDIF
C      CALL FRCVINFO(LEN, MSGTYPE, HOST, HOSTID, INFO)
C      WRITE(MESSAGE, 1000) LEN

```

```

CALL MSGPUT (MESSAGE, 'I')
CALL DATCREAT('Vis')
CALL DATCREAT('Vis/OBS')
CALL DATCREAT('Vis/OBS/I')
CALL CRDPREAD('Vis/OBS/I')
CALL ARRPREAD('Vis/OBS/I/VIS')
CALL ARRPREAD('Vis/OBS/I/WT')
CALL ARRPREAD('Vis/UU')
CALL ARRPREAD('Vis/VV')
CALL ARRPREAD('Vis/WW')
IF(ERROR) GO TO 990
CALL MSGPUT ('Read visibility data', 'I')

```

C

C Now wait for action: Message 2 is an image that is to be  
C calculated.

C

```

1 CONTINUE
MSGTYPE = 2
CALL FRCV(MSGTYPE, INFO)
CALL FRCVINFO(LEN, MSGTYPE, HOST, HOSTID, INFO)
WRITE(MESSAGE, 1000) LEN
1000 FORMAT ('Received ',I7,' bytes')
CALL MSGPUT (MESSAGE, 'I')
IF(INFO.LT.0) THEN
    CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM Receive error')
    GO TO 990
ENDIF

```

C

C Receive instance number. This is needed by the host so that the  
C resulting image can be put in the correct place.

C

```

CALL FGETNINT (INST, 1, INFO)
IF(INFO.LT.0) THEN
    CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM Receive error')
    GO TO 990
ENDIF

```

C

C Make PSF or Dirty Image?

C

```

CALL FGETNINT (PSF, 1, INFO)
IF(INFO.LT.0) THEN
    CALL ERRREPOR (ERRLOGIC, ROUTINE, 'PVM Receive error')
    GO TO 990
ENDIF

```

```
        IF(PSF.EQ.1) THEN
            CALL MSGPUT ('Making PSF', 'I')
        ELSE
            CALL MSGPUT ('Making Dirty image', 'I')
        ENDIF
C
C Read Image template and make FFT template
C
        CALL DATCREAT('Image')
        CALL DATPREAD('Image')
        CALL FFTCONJA ('Image', 'Grid', DIR, 2)
        CALL MSGPUT ('Read image', 'I')
C
C Now make image using standard SDE routine
C
        CALL MSGPUT ('Making Image', 'I')
        CALL VISTOIMG ('Vis', 'OBS/I', 'Image', 'Grid', PSF.EQ.1)
        IF(ERROR) GO TO 990
C
C Now return resulting image as message 11. Precede the image
C by the instance number. Then delete the Image and FFT so that
C we may get the next template.
C
        CALL FINITSEND()
        CALL FPUTNINT (INST, 1, INFO)
        CALL ARRPWRT('Image')
        CALL DATDELET('Image')
        CALL DATDELET('Grid')
        CALL FSND(HOST, HOSTID, 11, INFO)
        CALL MSGPUT ('Completed one pass', 'I')
        GO TO 1
C
C We made an error: return message 999 and close down.
C
990 CONTINUE
        SYSMMSG=.TRUE.
        CALL FINITSEND()
        CALL FPUTNINT (INST, 1, INFO)
        CALL FSND(HOST, HOSTID, 999, INFO)
        CALL FLEAVE()
C
999 CONTINUE
        END
```

## **Appendix C: Effect of confusion on 327 MHz noise**

I used MathCad to produce estimates of the confusion noise at 327MHz in the various configurations. The file is /noggs/tcornwel/mcad/327.MCD. Anyone is welcome to copy this file and insert their own favorite parameters. I append a copy of the postscript output here.

The role of confusion in limiting P-band sensitivity

T. Cornwell NRAO  
November 1992

VLA Scientific  
Memo 164

INTRODUCTION

At the New Mexico Symposium held at the AOC in December 1990, Barry Clark gave a talk on the effects of confusion on P-band noise and on the likely success of various countermeasures. He showed that

- (i) sidelobe confusion produces noise in the range 5-30 mJy/beam going from A to D configuration
- (ii) in A and B configurations, about 350 point sources must be removed to reduce the sidelobe confusion noise to the level of thermal noise
- (iii) using MX to deconvolve the 16 brightest sources will reduce the sidelobe confusion noise by about a factor of 6 (down to somewhere in the range 1-5 mJy/beam).

In addition to these calculations, we have learnt from experience that even when using a wide-field deconvolution algorithm, the noise in B-configuration is limited to about 0.4-0.5 mJy/beam or worse (i.e. several times theoretical). Many hundreds of point sources must be extracted from the full primary beam. This limit is a true saturation: increasing the integration time beyond a few hours brings no improvement, indicating that calibration errors due to, for example, non-isoplanicity are not likely to be important. It also seems that in C-configuration images, the noise level is improved by removing sources in the first sidelobe of the primary beam.

The purpose of this memo is to re-examine these effects in more detail using numerical integration where appropriate to allow the use of more accurate models for confusion, smearing effects and the primary beam sidelobes.

This memo is actually an executable MathCad document and therefore contains a number of strange symbols which may not be self-explanatory. Mostly, however, the calculations should be transparent. This file is available to anyone who wants to change some of the basic assumptions or augment the calculations.

I proceed in three steps:

- (1) Find approximations to the primary beam and smearing beams
- (2) Integrate to find moments of the effective primary beam
- (3) Find moments of the source counts for various cleaning levels

BASIC PARAMETERS AND DEFINITIONS

$\lambda := 0.9$	Wavelength of observation	$i := 0..3$		Solid angle of dirty beam								
$D := 25$	Diameter of VLA dish	$b_i :=$	$\Psi_i := \left(1.2 \cdot \frac{\lambda}{b_i}\right)^2$	$\Psi_i$								
$D_b := 2$	Diameter of subreflector	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td><math>36.7 \cdot 10^3</math></td></tr><tr><td><math>11 \cdot 10^3</math></td></tr><tr><td><math>3.37 \cdot 10^3</math></td></tr><tr><td><math>1.02 \cdot 10^3</math></td></tr></table>	$36.7 \cdot 10^3$	$11 \cdot 10^3$	$3.37 \cdot 10^3$	$1.02 \cdot 10^3$		<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td><math>8.66 \cdot 10^{-10}</math></td></tr><tr><td><math>9.64 \cdot 10^{-9}</math></td></tr><tr><td><math>1.027 \cdot 10^{-7}</math></td></tr><tr><td><math>1.121 \cdot 10^{-6}</math></td></tr></table>	$8.66 \cdot 10^{-10}$	$9.64 \cdot 10^{-9}$	$1.027 \cdot 10^{-7}$	$1.121 \cdot 10^{-6}$
$36.7 \cdot 10^3$												
$11 \cdot 10^3$												
$3.37 \cdot 10^3$												
$1.02 \cdot 10^3$												
$8.66 \cdot 10^{-10}$												
$9.64 \cdot 10^{-9}$												
$1.027 \cdot 10^{-7}$												
$1.121 \cdot 10^{-6}$												
$\eta := 0.01$	Fractional bandwidth											
$TOL := 10^{-6}$	Tolerance for numerical work											
$Jinc(m, x) :=$	$if \left( x=0.0, \frac{J_n(m, TOL)}{TOL^m}, \frac{J_n(m, x)}{x^m} \right)$			Jinc function for primary beam								
$ERF(x) :=$	$if \left( x=0, \frac{erf(TOL)}{TOL}, \frac{erf(x)}{x} \right)$			Needed for smearing effects								
$N := 10$	$j := 0..N$	$S_j := 10^{-3 + \frac{j}{N} \cdot 5}$		Needed for graphs								
$Aconf := 0$	$Bconf := 1$	$Cconf := 2$	$Dconf := 3$	Needed for graphs								

## PRIMARY BEAM MODEL

There are many possible approximations to the primary beam. A uniformly illuminated aperture is incorrect since the dipole illuminates the primary and furthermore there is significant defocus. To approximate the real illumination pattern, I will use a parabolically tapered aperture throughout most of this document but where relevant I will also discuss the effect of a uniformly illuminated aperture. I include the effects of the (uniformly illuminated) blockage of the sub-reflector but not of the feed legs, etc.

$$E(\theta, a, m) := \frac{2 \cdot \pi \cdot a^2}{\lambda} \cdot (1 + \cos(\theta)) \cdot \text{Jinc}\left(m, 2 \cdot \pi \cdot \frac{a}{\lambda} \cdot \sin(\theta)\right)$$

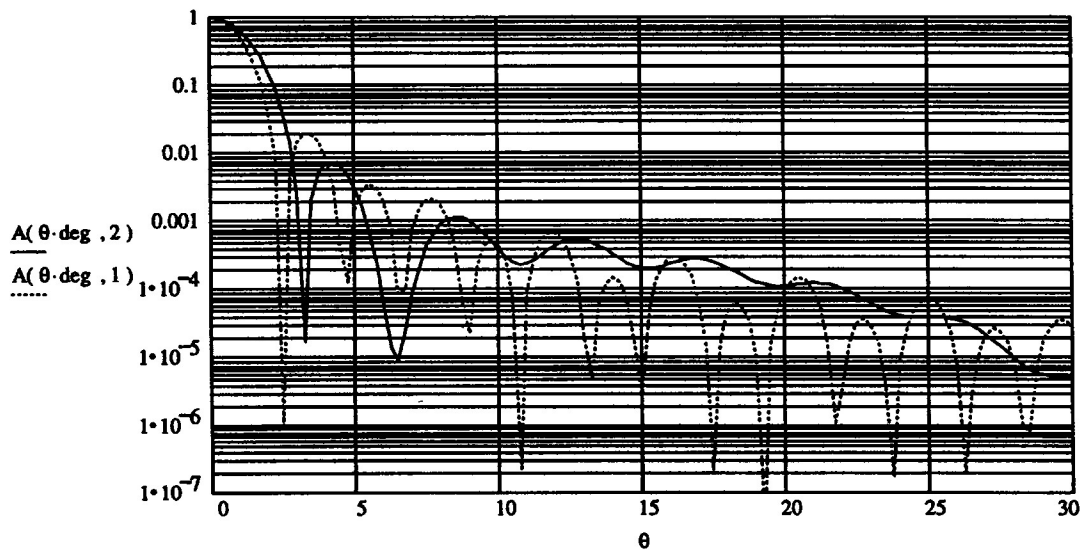
Electric field pattern

$$A(\theta, m) := \frac{\left(E\left(\theta, \frac{D}{2}, m\right) - E\left(\theta, \frac{D_b}{2}, 1\right)\right)^2}{\left(E\left(0, \frac{D}{2}, m\right) - E\left(0, \frac{D_b}{2}, 1\right)\right)^2}$$

Blocked Power pattern

$$\theta := 0, 0.25 \dots 30$$

### Primary beam models for parabolically tapered and uniformly illuminated apertures

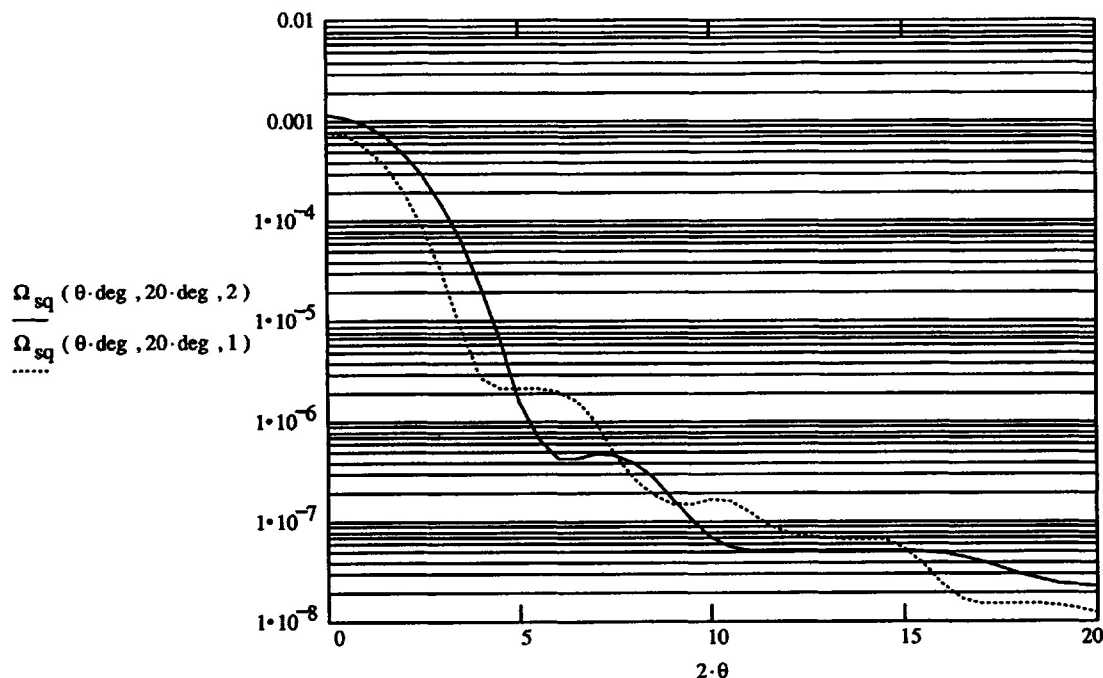


Below I will show that the relevant quantity for assessing the noise behaviour is the second moment of the primary beam. This can be regarded as a figure of merit for minimizing sidelobe confusion. Accordingly I plot this against radius for both types of illumination pattern.

$$\Omega_{sq}(\theta_{\min}, \theta_{\max}, m) := \int_{\theta_{\min}}^{\theta_{\max}} 2 \cdot \pi \cdot \sin(\theta) \cdot A(\theta, m)^2 d\theta$$

$$\theta := 0, 0.25 \dots 10$$

## Second moment of the primary beam as a function of radius



## SMEARING EFFECTS

There are many approximations for the decorrelation due to bandwidth smearing. Here is a formula from the Synthesis Imaging book for square bandpass and Gaussian beam. I will assume that the time-averaging smearing is chosen to match the bandwidth smearing and is thus given by roughly the same formula

$$\gamma := 2 \cdot \sqrt{\ln(2)}$$

$$A_v(\theta, b, \eta) := \frac{\sqrt{\pi}}{2} \cdot \text{ERF}\left(\theta \cdot \gamma \cdot \eta \cdot \frac{b}{2 \cdot \lambda}\right) \quad A_t(\theta, b, \eta, \tau) := \frac{\sqrt{\pi}}{2} \cdot \text{ERF}\left(\theta \cdot \gamma \cdot \eta \cdot \frac{b}{2 \cdot \lambda} \cdot \tau\right)$$

Now we find the first and second moments of the primary beam. For the former, we may ignore the smearing effects since flux is conserved. For the latter, the main effect of the smearing is to increase the number of sources at a given level. As a result only a single power of the relevant "beam" must appear in the calculation of the second moment. One further point is that in the integral of the second moment we ignore all angles greater than 20 degrees (field of view 40 degrees) since the contribution is very small and taxes the numerical integration routines of MathCad to no useful end.

$$\Omega_0(\theta_{\min}, \theta_{\max}, m) := \int_{\theta_{\min}}^{\theta_{\max}} A(\theta, m) \cdot 2 \cdot \pi \cdot \sin(\theta) \, d\theta$$

$$\Omega_{sq}(\theta_{min}, \theta_{max}, m, b, \eta, \tau) := \int_{\theta_{min}}^{\theta_{max}} 2 \cdot \pi \cdot \sin(\theta) \cdot (A(\theta, m)^2 \cdot A_v(\theta, b, \eta) \cdot A_t(\theta, b, \eta, \tau)) d\theta$$

$$\Omega_t := \Omega_0(0 \cdot \text{deg}, 90 \cdot \text{deg}, 2)$$

$$\Omega_t = 0.002$$

Primary beam first moment

$$\Phi_1 := \Omega_{sq}(0 \cdot \text{deg}, 20 \cdot \text{deg}, 2, b_i, 0.01, 1.0)$$

$$\Phi$$

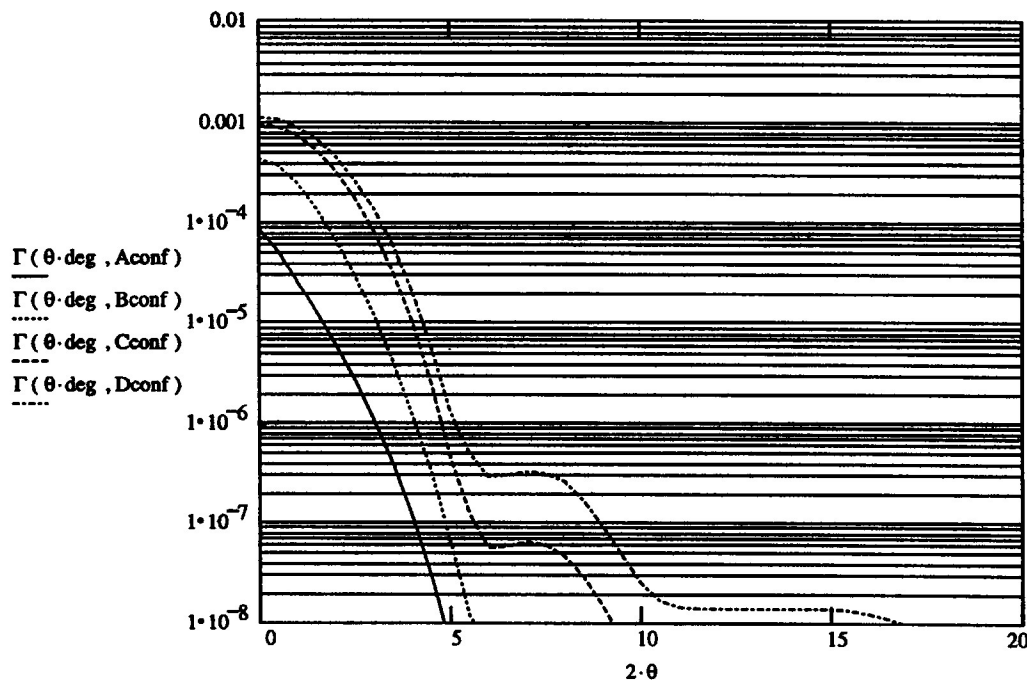
Primary beam second moment for various configurations.

$8.548 \cdot 10^{-5}$
$4.176 \cdot 10^{-4}$
$9.335 \cdot 10^{-4}$
0.001

To illustrate the effects of smearing, I repeat the plot of the second moment of the primary beam, now incorporating smearing effects.

$$\Gamma(\theta, i) := \Omega_{sq}(\theta, 20 \cdot \text{deg}, 2, b_i, 0.01, 1.0)$$

### Second moment of the parabolic taper primary beam as a function of radius



### SOURCE COUNTS

I will use a simple approximation to the source counts of Condon (1984) and Wieringa (1992). This has two segments of differing slopes, joining at a break point close to 1 Jansky. The numbers below 1Jy are required for predicting the effects of cleaning a single field, while the counts above 1Jy are needed to predict the effects of sidelobes from sources located outside the primary field of view

$$S_{max} := 10^3 \quad S_{min} := 10^{-6} \quad \beta := 2.75$$

$$n_0 := 2 \cdot 10^3 \quad S_b := 1 \quad n_u(S) := \left(\frac{S_b}{S}\right)^\beta \cdot n_0$$

High flux source counts from Condon (1984)

$$\alpha := 1.75 \quad n_l(S) := \left(\frac{S_b}{S}\right)^\alpha \cdot n_0$$

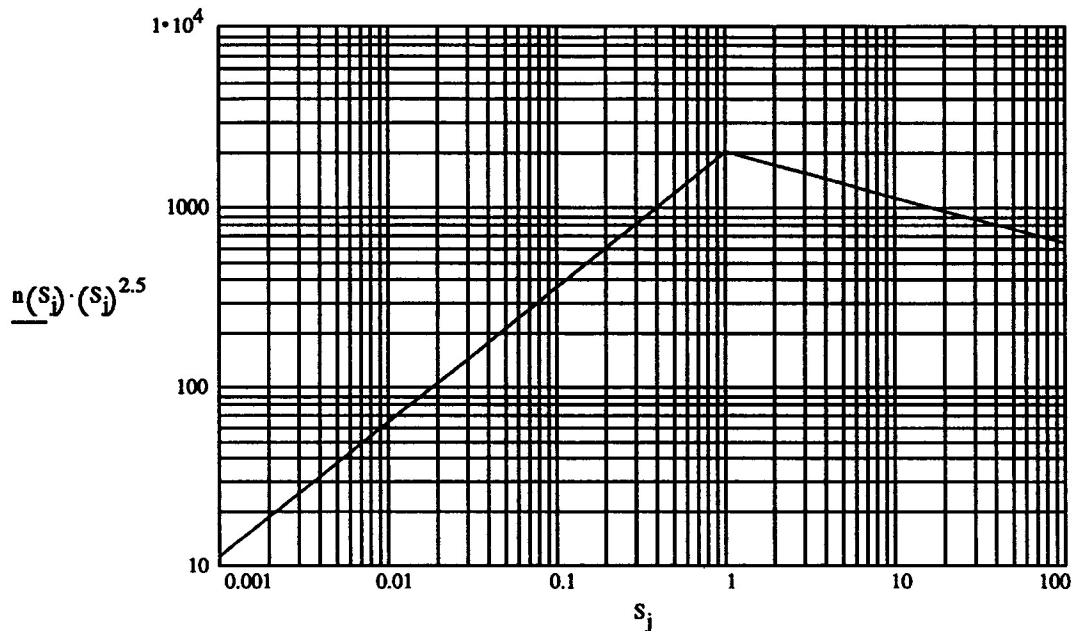
Low flux source counts from Wieringa (1992). Good to about 3 mJy/beam.



$$n(S) := \left[ \text{if} \left[ (S > S_{\min}) \cdot (S \leq S_b), n_l(S), 0 \right] + \text{if} \left[ (S > S_b) \cdot (S < S_{\max}), n_u(S), 0 \right] \right]$$

The shape of the curve (after removing the Euclidean term) is shown in this plot:

### Differential Source Counts



The m'th moment is given by:

$$\Lambda(m, S_l, S_u) := \int_{\ln(S_l)}^{\ln(S_u)} e^{x \cdot (m+1)} \cdot n(e^x) dx$$

We use this form to speed evaluation by MathCad numerical routines.

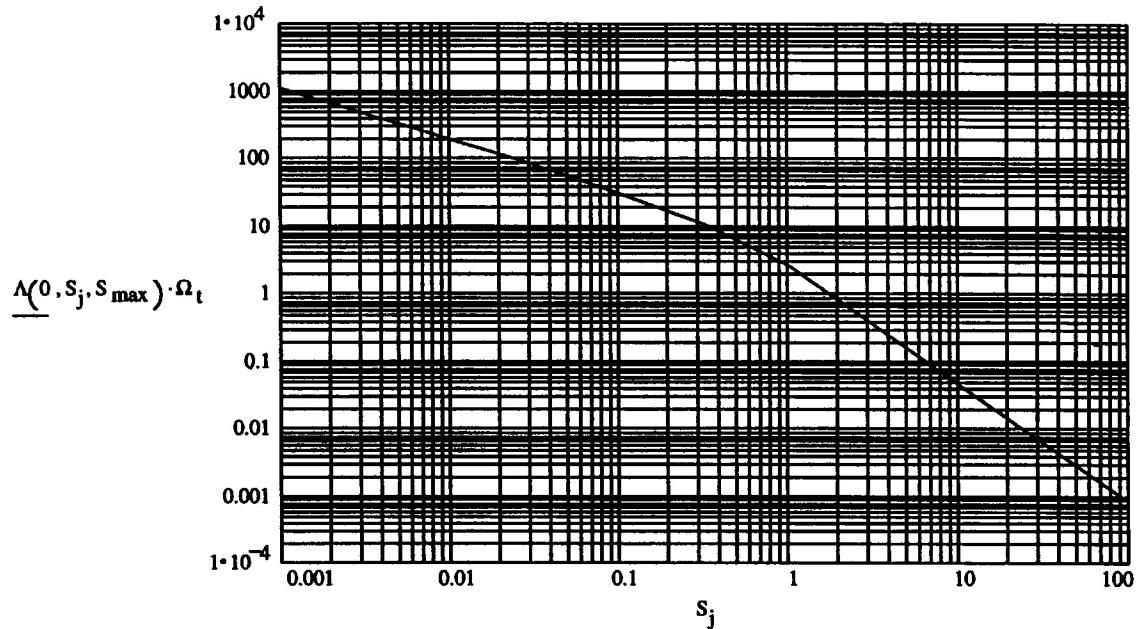
To verify this empirically, I check the prediction against the observed flux of about 12-13 Jy found when cleaning a fairly empty field down to about 10 mJy/beam. Typically a few hundred point sources are found when cleaning down to this level.

$$S_{\text{predicted}} := (\Lambda(1, 0.010, 2.0)) \cdot \Omega_0(0.5 \cdot \text{deg}, 2) \quad S_{\text{predicted}} = 14.095$$

$$N_t := (\Lambda(0, 0.010, 2.0)) \cdot \Omega_0(0.5 \cdot \text{deg}, 2) \quad N_t = 177.425$$

I also show the integral counts, scaled to the solid angle of the primary beam. Note that a couple of 1Jy sources will typically be found in a VLA primary beam.

### Integral Source Counts for the solid angle of one Primary beam



Now I can calculate the noise level due to confusing sources by integrating the effect of all sources. I need to assume a value for the sidelobe level of the synthesized beam. A simulation of a long observation gives

$$\sigma_b := 0.003$$

The noise has three contributions: first, due to the sidelobes of sources below the clean cutoff, second, due to the sidelobes of the remnants of cleaned sources, and third, due to the presence of sources inside the synthesized beam.

$$\sigma_1(S, i) := \sqrt{\Lambda(2, S_{min}, S) \cdot \Phi_1} \sigma_b \quad \text{Sidelobes of confusion}$$

$$\sigma_2(S, i) := \sqrt{[(\Lambda(0, S, S_{max})) \cdot S^2] \cdot \Phi_1} \sigma_b \quad \text{Sidelobes of remnants}$$

$$\sigma_3(S, i) := \sqrt{(\Lambda(2, S_{min}, S) \cdot \Psi_1)} \quad \text{Synthesized beam confusion}$$

$$\sigma_{\chi}(S, i) := \sqrt{\sigma_1(S, i)^2 + \sigma_2(S, i)^2 + \sigma_3(S, i)^2}$$

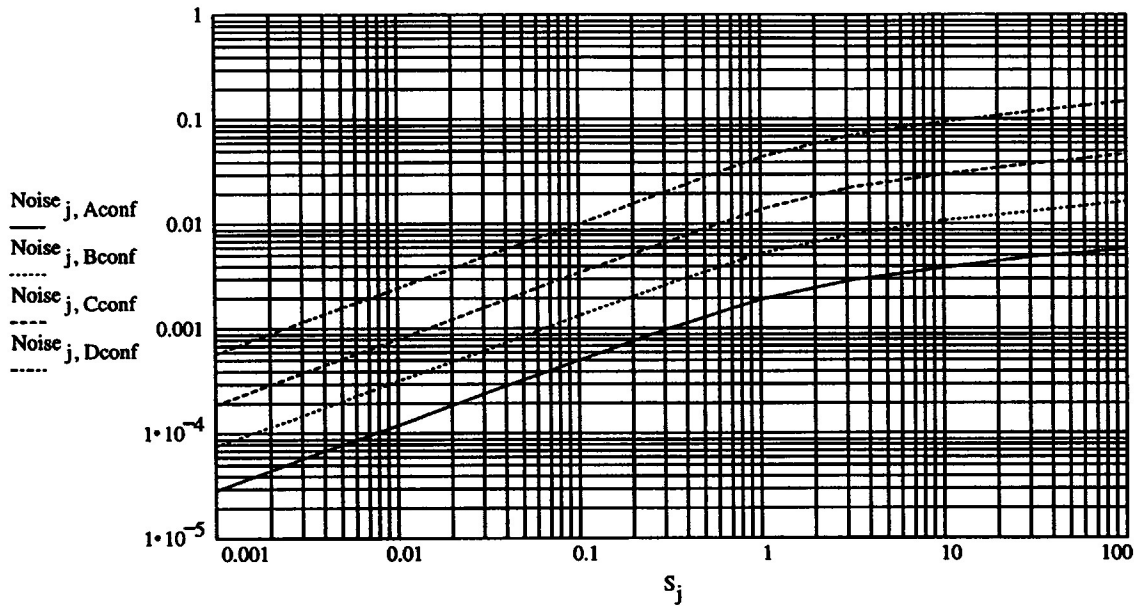
Below I plot the noise achieved against clean limit and number of point sources. The most interesting conclusion is that to reach close to the thermal noise for a long synthesis (0.1-0.2 mJy/beam, say), it is necessary to clean down to a limit of about 1mJy/beam even though the rms sidelobe level of the synthesized beam is about 1%. As one goes deeper the number of sources increases rapidly, making reduction of sidelobe noise more and more difficult.

$$TOL := 10^{-3}$$

$$\text{Number}_j := \Lambda(0, S_j, S_{max}) \cdot \Omega_t \quad \text{Number of sources}$$

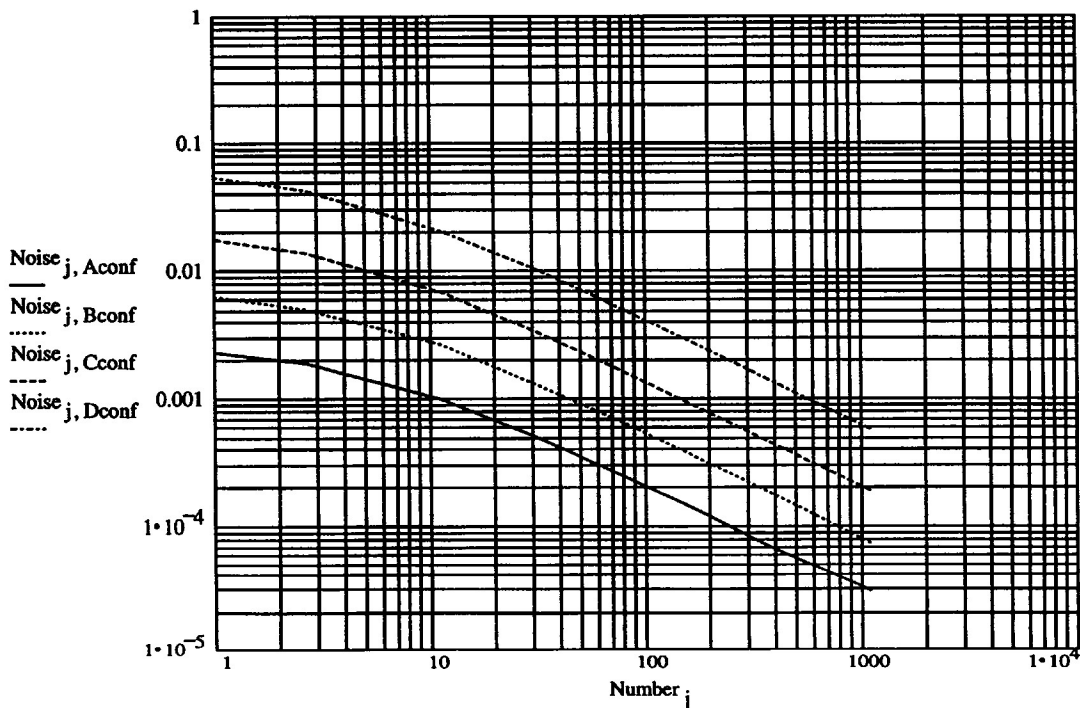
$$\text{Noise}_{j,i} := \sigma_{\chi}(S_j, i) \quad \text{Noise achieved}$$

### Noise achieved versus Clean limit



Next we show the noise achieved for various numbers of point sources subtracted. Note that in A-configuration, bandwidth and time-averaging smearing bring about a factor of two improvement in noise level for a fixed number of components, meaning that to achieve a given noise level, about one third the D-configuration number of clean components are required. Even so, to reach the noise limit for a long synthesis, say about 0.1-0.2 mJy/beam, at least 100 point sources must be subtracted. The 16 fields of MX, if allocated one per point source, will give a noise level of a 800 microJy/beam.

### Noise achieved versus Number of sources subtracted



So far I have not mentioned confusion limits. The meaning of confusion is rather difficult to pin down precisely. It cannot be the rms variation in any one resolution element since this diverges as the maximum flux allowed in the source counts. A reasonable definition is the level at which there is expected to be one source per resolution element. However, even this is somewhat arbitrary. For my analysis there is an obvious definition. I have assumed that separate point sources can be identified and cleaned. The confusion limit can then be defined as the level at which this breaks down. The limit in the number of point sources which can be subtracted comes from the number of synthesized beams in the primary beam. Setting a filling factor of about 20%, we find the following limits:

$$N_{c_i} := 0.2 \cdot \left( \frac{\Phi_i}{\Psi_i} \right)$$

$N_{c_i}$

$1.974 \cdot 10^4$
$8.664 \cdot 10^3$
$1.818 \cdot 10^3$
196.682

A configuration

B configuration

C configuration

D configuration

Hence neither A nor B configurations are confusion limited, C is marginally limited at about the long synthesis limit of 0.1-0.2 mJy/beam, and D configuration is limited at about 2-3 mJy/beam.

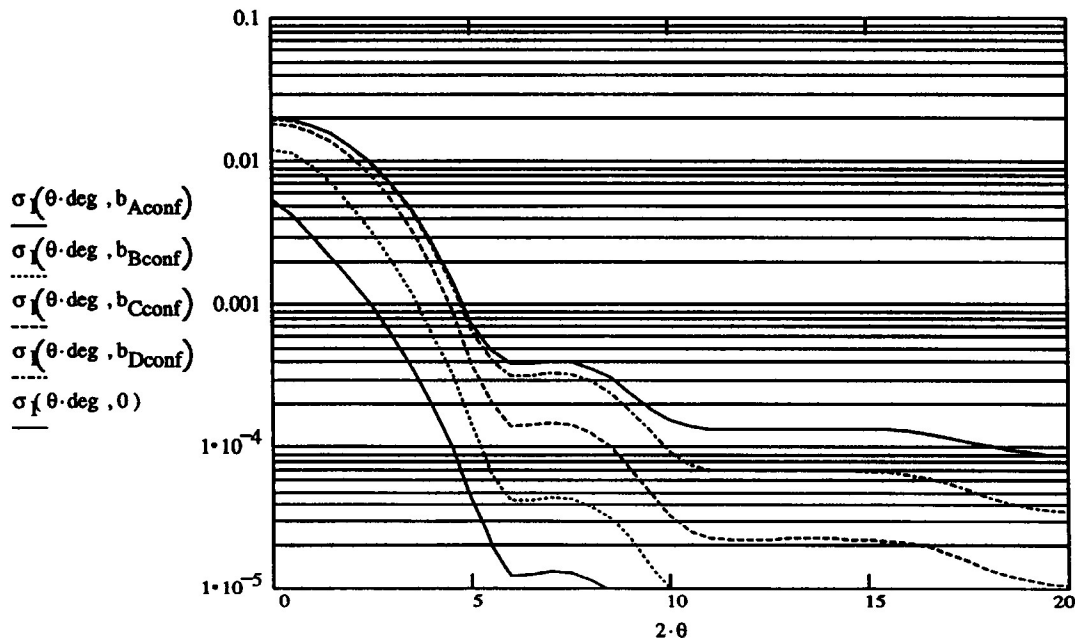
The final question to ask is how big a field must we clean in order to reach this limit? To answer this, we must compute the noise contributed by sources outside the cleaned region. This is given by

$$K := \sqrt{\Lambda(2, S_{\min}, S_{\max})} \cdot \sigma_b \quad \sigma_f(\theta, b) := \sqrt{\Omega_{sq}(\theta, 20\text{-deg}, 2, b, 0.01, 1.0)} \cdot K$$

$$\theta := 0, 0.25 \dots 10$$

$$\text{TOL} := 10^{-6}$$

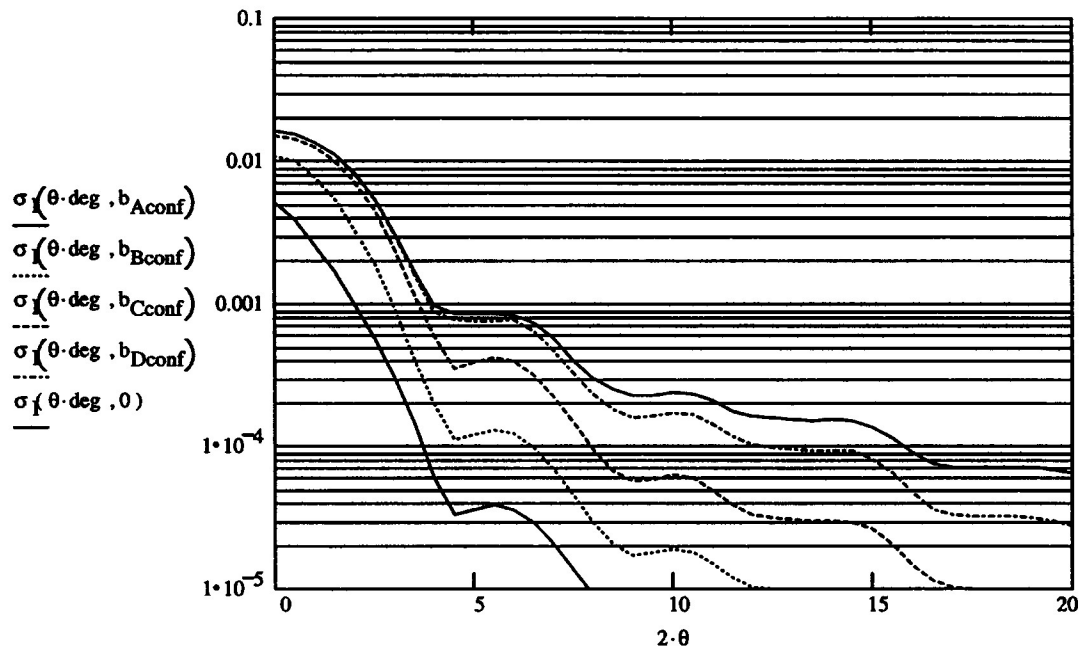
### Noise achieved versus field of view (Parabolic taper)



The uppermost line shows the noise for no smearing at all. To reach long-synthesis noise in this case requires a field of view of about 10 degrees. This curve is affected by the choice of primary beam model. The plot for a uniformly illuminated aperture is given below.

$$\sigma_{\mathbf{I}}(\theta, b) := \sqrt{\Omega_{\text{sq}}(\theta, 20 \cdot \text{deg}, 1, b, 0.01, 1.0)} \cdot K$$

### Noise achieved versus field of view (Uniform illumination)



Hence the importance of increasing the field of view does depend on the actual illumination pattern of the P-band system. The sharp edge of the dish leads to high sidelobes close in and thence to increased sidelobe confusion when uniformly illuminated. The actual illumination is probably closer to parabolic since the feed is a dipole so these two different illuminations probably bracket the real pattern. Actually the most important factor in determining the level of sidelobe confusion is the blockage due to the subreflector. If this is removed, the noise in the first primary beam sidelobe drops by about 80% for parabolic illumination. For uniform illumination, the improvement is only about 20%.

Note that for either illumination pattern it pays to get outside a given sidelobe of the primary beam. For the canonical problem of achieving noise-limited B-configuration images, we see that for uniform illumination it is necessary to clean out beyond the first sidelobe for a total field of view of 8 degrees. Since about 100 point sources must be subtracted, the only viable approach is to map the entire field of view requiring an image of size 3600 by 3600 pixels.

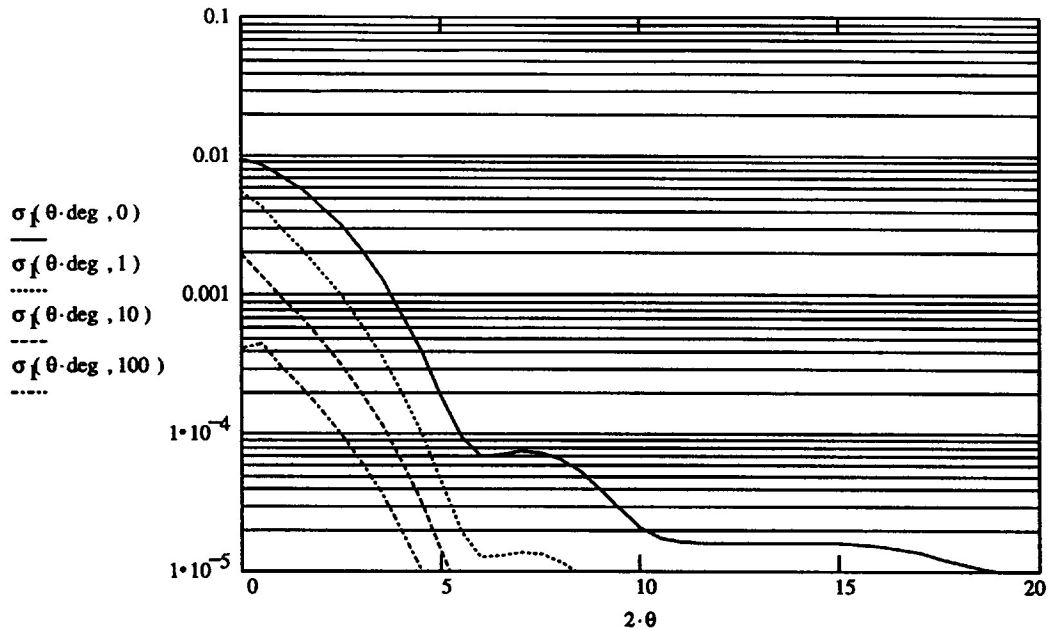
One further important topic is the question of how much time-averaging will improve the situation. So far we have assumed that time-averaging and bandwidth smearing are balanced. The maximum fractional bandwidth at P-band is certainly hard to increase, but we can increase the amount of time-averaging smearing at will. Assuming parabolic tapering:

$$\sigma_f(\theta, \tau) := \sqrt{\Omega_{sq}(\theta, 20 \cdot \text{deg}, 2, b_0, 0.01, \tau)} \cdot K$$

The units of time-averaging are the time taken for time-averaging smearing to balance bandwidth smearing.

$$\theta := 0, 0.25 \dots 10$$

### Noise achieved versus field of view for various levels of time averaging for A-configuration



The field of view shrinks with increasing time-averaging but not particularly quickly. For an averaging time of about 10 minutes, the field of view which must be cleaned is still about 2.5 degrees. An averaging time of ten minutes is almost certainly too long for the ionospheric phase to remain constant so a smaller interval like 1 minute may be better, leading to a field of view of 4 degrees.

### CONCLUSIONS

(1) Many hundreds of point sources must be subtracted to remove the noise power due to sidelobes of confusing sources. Only the dragon program in SDE can currently do this properly for C and B configurations.

(2) The 16 fields of MX reduce the noise by a factor of 4 (to somewhere in the range 1-5 mJy/beam).

(3) Bandwidth and time-averaging smearing help quite a bit but not as much as might be hoped. This disappointing behaviour is due to the fact that both of these effects merely smear flux (instead of removing it as does the primary beam).

(4) In B-configuration P-band noise could be limited by the sidelobes of confusing sources far outside the main lobe of the primary beam. However, the accuracy of my calculations is probably worse than the factor of 2-3 discrepancy between the theoretical and observed noise. Increasing the accuracy of this prediction requires:

(a) better knowledge of the primary beam (something which will be hard given the effects of confusion!). At the moment this is by far the dominant cause of uncertainty.

(b) better knowledge of the source counts (something we could get from B-configuration observations)

(c) better numerical calculations of smearing effects (using simulations).