

EVLA Memo 62

Full primary beam Stokes I,Q,U,V imaging

T.J. Cornwell (tcornwel@nrao.edu)

National Radio Astronomy Observatory, Socorro, NM, USA, 87801

Abstract

Full primary synthesis imaging in Stokes I,Q,U,V requires that the full polarization behavior of the antennas be accounted for in the deconvolution step. The “Hamaker-Bregman-Sault” measurement equation provides a compact description of polarization in synthesis imaging. In AIPS++, the HBS ME is used as the basis for synthesis imaging. In this note I describe and demonstrate a CLEAN-based deconvolution algorithm that incorporates knowledge of the polarized primary beam.

1 Introduction

High accuracy full polarization imaging of the entire primary beam using the VLA has always been difficult because of the two instrumental effects:

Beam squint Since the VLA feed circle is off-axis, the two polarizations (right hand and left hand circular) are inevitably squinted.

Cross-hand leakage Off-axis, radiation will leak from right to left and vice versa. Also, antenna and optics imperfections will cause leakage on and off axis.

Cotton (1994) has measured the polarization behavior of the VLA at L-band and Brisken (2003) has calculated the beam (both unpolarized and polarized) using an antenna simulation program. At the 3dB point, Cotton measured about 2% spurious linear and 10% spurious circular polarization. At the 6dB point, these values roughly doubled. Brisken’s calculations of the fractional linear polarization were 50% lower than measured, perhaps due to the un-modelled effects of the L-band feed lens. The beamsquint at any frequency is 0.065 of the HWHM of the antenna beam - about an arcminute at L-band.

Both these effects are believed to be constant in the antenna frame so the impact on an astronomical source (fixed in right ascension and declination) will be parallactic angle dependent. Thus it is not possible to remedy this problem purely in the image plane after deconvolution except for the special case of a snapshot. In the case of a long synthesis, our current recommendation is evidently that observers should break the data into small chunks of roughly constant parallactic angle, deconvolve each data set separately, and then add the resulting images. Besides being tedious, this procedure has the failing that deconvolutions are performed with small data sets so the accuracy of each is lower. Since the deconvolution

is non-linear, the sum of a set of deconvolution is not the same as (and is usually substantially poorer than) the deconvolution of the sum.

For a long synthesis, then, it is preferable to accommodate the squint and leakage within the deconvolution algorithm. In principle this is not different from mosaicing, where the image-plane sampling patterns of the antennas (*i.e.* different pointings) must be accommodated in the deconvolution. Hence the same basic approach as used in mosaicing must work. In this note, I show that this is in fact correct - algorithms similar to normal mosaicing can be used to form high quality deconvolved images in the presence of squint and leakage.

This topic is scientifically important for the VLA but absolutely vital for the EVLA. Reaching the full continuum sensitivity of the EVLA at L-band will require correction of this effect (and many others!).

The basis of the imaging algorithms described here is the ‘‘Hamaker-Bregman-Sault’’ measurement equation (Hamaker, Bregman & Sault [1996], see also Sault and Cornwell, 1999 for an overview of the HBS ME and its relevance to imaging). The HBS ME forms the foundation of the synthesis (and single dish) calibration and imaging code in AIPS++ (see Noordam 1995, 1996; Cornwell 1995a, 1995b; Cornwell and Wieringa 1996; Cornwell 1996; Cornwell and Wieringa 1997). The advantage of the HBS ME is that it provides a compact notation for expressing the well-known physics of the radio-interferometric measurement of polarization radiation (see *e.g.* Conway and Kronberg, 1969). This notation has no new physics but it is conducive to algorithm development because the notational simplicity allows one to concentrate on details of a new algorithm rather than the algebra of polarization measurements. For example, the HBS ME for polarization radiation is isomorphic to the scalar (non-polarized) standard synthesis measurement equation (with Jones matrices replacing scalars and multiplication replaced by direct products).

In this memo, I pick up from the description of the HBS ME given by Sault and Cornwell (1999).

2 Direction-Dependent Effects

As Sault and Cornwell (1999) describe, where calibration effects are direction dependent, different sources within the field will experience different effects. These effects include antenna gains that vary with position on the sky (*e.g.* non-isoplanaticity) or distance from the pointing center (*e.g.* primary beam gain and off-axis polarimetric response). In these cases, the measured coherency vector cannot be described as a simple multiplication of the true Stokes visibility vector with a single response matrix. If we assume that the source is expressible as a sum of point sources, then the measured visibility coherency vector will be

$$V'_{ij} = (\mathbf{J}_{\text{VIS},i} \otimes \mathbf{J}_{\text{VIS},j}^*) \sum_k (\mathbf{J}_{\text{SKY},i}(\rho_k) \otimes \mathbf{J}_{\text{SKY},j}^*(\rho_k)) \mathbf{S} V_{S,ij,k}, \quad (1)$$

where $\mathbf{J}_{\text{SKY}}(\rho)$, which is a function of direction ρ , is the combination of all Jones matrices which dependent on direction (*e.g.* all non-isoplanatic propagation and primary beam effects), and \mathbf{J}_{VIS} is the combination of all direction-independent effects (*e.g.* telescope

signal path effects). The summation is over the set of point sources, and $V_{S,ij,k}$ is the true Stokes visibility vector for the k th source. Here we assume that all position-dependent effects precede the position-independent ones (remember matrix multiplication is not generally commutative!)

3 Implementation in AIPS++

We have seen above, that for a source expressible as a sum of point sources, the measured coherency vector is given by:

$$V'_{ij} = (\mathbf{J}_{\text{VIS},i} \otimes \mathbf{J}_{\text{VIS},j}^*) \sum_k (\mathbf{J}_{\text{SKY},i}(\rho_k) \otimes \mathbf{J}_{\text{SKY},j}^*(\rho_k)) \mathbf{S} V_{S,ij,k}, \quad (2)$$

The true visibility vector $V_{S,ij,k}$ for the k 'th point source is given by:

$$V_{S,ij,k} = I_{S,k} \exp(i\Phi_{ij,k}) \quad (3)$$

where $I_{S,k}$ is the Stokes vector of the point source (i.e. the vector of the point source I , Q , U and V), and $\Phi_{ij,k}$ is the usual Fourier phase term.

To investigate further the case of direction-dependent effects, let us first turn to imaging in the presence of *a priori* known image-plane calibration terms. A simple example would be a model for the primary beam of the array antennas. Generally, when direction-dependent effects are present, the resultant point-spread function will be shift-variant, and so traditional deconvolution approaches cannot be used. Cornwell [1995] shows that one viable approach is to use an algorithm in which an estimate of the sky brightness is iteratively improved. A plausible update formula can be derived from the gradients of the fit (χ^2) of the predicted coherency vectors to those measured. Using an approximate Newton-Raphson approach in which the diagonal elements of the second derivative of χ^2 with respect to $I_{S,k}$ are taken into account, one can define a generalized residual image:

$$I_{S,k}^D = - \left[\frac{\partial^2 \chi^2}{\partial I_{S,k} \partial I_{S,k}^T} \right]^{-1} \frac{\partial \chi^2}{\partial I_{S,k}} \quad (4)$$

where:

$$\frac{\partial \chi^2}{\partial I_{S,k}} = -2 \Re \left[\sum_{ij} \left[\mathbf{J}_{\text{SKY},i}(\rho_k) \otimes \mathbf{J}_{\text{SKY},j}^*(\rho_k) \right]^{*T} \Lambda_{ij} \Delta V_{ij} \exp(-i\Phi_{ij,k}) \right] \quad (5)$$

and:

$$\frac{\partial^2 \chi^2}{\partial I_{S,k} \partial I_{S,k}^T} = 2 \Re \left[\sum_{ij} S^{*T} \left[\mathbf{J}_{\text{SKY},i}(\rho_k) \otimes \mathbf{J}_{\text{SKY},j}^*(\rho_k) \right]^{*T} \Lambda_{ij} \left[\mathbf{J}_{\text{SKY},i}(\rho_k) \otimes \mathbf{J}_{\text{SKY},j}^*(\rho_k) \right] S \right] \quad (6)$$

where Λ_{ij} is the inverse of the covariance matrix of measurement errors on the coherency vector, and ΔV_{ij} is the residual coherency vectors (i.e. prediction error of the coherency vector). This last equation may be interpreted as the covariance matrix of the errors in the Stokes (I,Q,U,V) representation (since the terms on either side of Λ_{ij} each serve to convert to the Stokes representation). Calling this term Λ_{kk}^S , the noise in the final image may be estimated by the sensitivity function:

$$\frac{1}{\sqrt{\Lambda_{kk}^S}} \quad (7)$$

Thus, the residual image is approximately a Fourier summation of the residual coherency vectors, modified by multiplication with the adjoint of the image-plane effect. The residual is normalized by an appropriate sum of the self-products of the image-plane effects. Although this update formula looks quite fearsome, it is quite straightforward and does reduce to known cases (though with one subtlety described in the next section). For example, if the \mathbf{J}_{SKY} terms represent (only) the electric field reception pattern of the antennas then the normal mosaicing equations (see e.g. Cornwell, Holdaway and Uson, 1993) are obtained. Hence it would be reasonable to think of this as a generalization of mosaicing in which not just the antenna primary beams can be corrected but instead any factorizable image plane effect can be corrected.

Note that since this is just an update formula, it must be used in an iterative algorithm to estimate the sky brightnesses $I_{S,k}$. In general, this formula cannot be reduced to a shift-invariant convolution equation, although an approximate convolution relation may be feasible. If so then an algorithm like the following can be used:

- 1 Estimate starting model image $I_{S,k}$
- 2 Calculate residual image for the current model image using above formula, an approximate point spread function, and a threshold in brightness below which the approximation is no longer valid.
- 3 Clean the residual image using the approximate point spread function down to the threshold.
- 4 If the remaining peak residual is too high return to step 2 and continue.
- 5 Smooth the set of point sources thus found and add the residual image to form a restored image.

In AIPS++, the sky brightness is represented in two ways - via a set of discrete components and via images. Compact bright regions may be modelled by discrete components and more extended emission by images. Thus the calculations of Jones matrix application may be done with high precision (e.g. for every increment in parallactic angle), and those for the images with lesser precision (e.g. binned in parallactic angle).

4 Coupling between I,Q,U, and V

One subtle point was glided over in the previous section - the errors in I,Q,U,V are actually coupled. Physically, this occurs because we measure independent RR and LL, say, and then take linear combinations to obtain I and V. The effect gets worse when the different beams for R and L have different gains, as occurs for squinted beams. This general coupling must show up in the Stokes covariance matrix Λ^S as off-diagonal terms. Coupling occurs for other reasons as well - for example, suppose that some antennas cannot measure one hand of polarization or some antennas measure linear and some circular. Thus in the general case, the true inverse of the Stokes covariance matrix must be calculated for each pixel and applied to the residual image.

For example, suppose that the gain differential between RR and LL beams at a given pixel at a give parallactic angle is a , then the covariance matrix for Stokes I, V is:

$$\begin{pmatrix} g + a & -a \\ -a & g + a \end{pmatrix} \quad (8)$$

As expected, I and V intermix with ratio a/g . This can be a significant effect on V, which is the reason for the inadvisability of measuring circular polarization with a circularly polarized system as the VLA.

While inverting each matrix is certainly feasible, let us first explore the consequences of using a diagonal approximation to Λ^S . In that case, only the diagonal elements need be stored (a savings in disk storage) and the matrix may be inverted trivially (a savings in computing costs). The errors induced by this approximation affect the generalized dirty image defined above. If the generalized dirty image is used as a residual image in an iterative algorithm of ths sort described above, the effects of the coupling will be eventually removed as the iteration proceeds. This occurs because the slight errors made via this approximation are corrected in the full calculation of the residual image - as the residual image drives to zero, the slight errors become less and less important. Put another way, the model allows one to correct V for I and I for V.

Hence the diagonal approximation does not limit an iterative algorithm but it does limit the usefulness of the generalized dirty image. Ideally, then, one should invert the full (4 by 4) matrix for all pixels in the image. I defer this possibility to later work, and proceed for rest of this note with the diagonal approximation.

5 Applying the CLEAN algorithm to mosaics

In this note, we are concerned with the problem of imaging one primary beam with correction for the primary beam. Since the primary beam must be accomodated in the deconvolution, one can (and should) think of this as a single pointing mosaic. One key feature is that in the mosaic image, the noise level must vary according to the local sensitivity (since this is effectively divided out). This is acceptable for *e.g.* Maximum Entropy methods but not for the CLEAN algorithm. If one attempts to CLEAN the residual image (including primary

beam correction) then two errors occur - first, the wrong peaks may be found (amplified noise peaks instead of true signal), and second, the PSF removal will not account for the amplification of sidelobes where the sensitivity is low.

Similar problems occur in applying CLEAN to *e.g.* photon-limited data where the noise level varies across the image. The fix there is to rescale the image to constant noise. Here I adopt the same trick.

The algorithm above then becomes:

- 1 Estimate starting model image $I_{S,k}$
- 2 Calculate residual image for the current model image using above formula, an approximate point spread function, and a threshold in brightness below which the approximation is no longer valid.
- 3 Scale the residual image by the sensitivity function. Any CLEAN mask must be updated to limit the search to regions where the sensitivity is non-zero.
- 4 Clean the rescaled residual image using the approximate point spread function down to the threshold.
- 5 Scale the image of CLEAN components by the **inverse** of the sensitivity function.
- 6 If the remaining peak residual is too high return to step 2 and continue.
- 7 Smooth the set of point sources thus found and add the residual image to form a restored image.

The extra work involved is in rescaling the residual and CLEAN component images. This is usually negligible. In the case of uniform sensitivity, this algorithm reduces (as it must) to standard CLEAN.

6 Code and algorithm details

The CLEAN algorithm described here is invoked by using `imager.clean`, having used `imager.setvp` to turn on primary beam correction together with squint correction. The clean algorithm can be performed using a faceted transform. Cleaning (in steps 3, 4, and 5) then proceeds per facet. Currently only Hogbom and Clark CLEAN can be used, but Multi-scale CLEAN will be accommodated in upcoming revisions.

The core C++ code used here has been present in AIPS++ since about 1997. The structure of the C++ classes was described by Cornwell and Wieringa (1997). In the C++ code, the `SkyEquation` class is responsible for applying image-plane effects, using any of a variety of components that can be plugged in. Mark Holdaway wrote a set of C++ classes `PBMath*` for applying the primary beam. These form a quite complete set of one-dimensional models (see the class implementation `PBMath.cc` for more details). In

addition, the two beams for right and left hand circular can be offset from each other by a given squint.

I have added support for two-dimensional beam models via a specialized class `PB-Math2DImage`. This class will regrid the beam model from Az-El to Ra-Dec coordinates before application. This must be done for every parallactic angle bin and can therefore be quite expensive.

The computing costs of this algorithm are substantial. Since the bulk of the costs are in calculating residuals, the computing scales roughly as the number of parallactic angle bins used. For 15deg bins, the computing costs can therefore be an order of magnitude larger than when using an azimuthally symmetric primary beam model.

7 Tests via simulations

I performed two simulations - one using a grid of point sources and one using Conway's "Hydra" model. The latter is a model of a complicated extra-galactic radio galaxy, complete with complex polarization.

7.1 Grid of point sources

A grid of 9 point sources was specified, with separation between the points of 14.5 arcmin. Each source has the same flux: $I = 1.0, Q = 0.5, U = -0.3, V = 0.1$. The script used for the simulation is shown in the appendix.

The observation simulated has the following properties:

- 1.4GHz with 50MHz bandwidth
- VLA D-configuration
- Source at declination +25 degrees.
- Hour angle from -4 to + hours
- Integration times of 60s every 300s.
- No noise was added

The simulations were calculated using a 1D, beam-squinted model of the primary beam, and a 2D model (courtesy of Walter Brisken and Chris Clearfield). The images were then processed using the above CLEAN algorithm, and also once without any primary beam model. The Stokes I,V parts of the images are shown in figures 1, 2, and 3.

The rms errors calculated in between the grid (shown in Table 1) show that the primary beam correction improves the Stoke I dynamic range, as expected. The improvement is moderate in Stokes I but very substantial in Stokes V.

Table 1. *RMS error for I, V images of grid of point sources*

Primary beam model	I	V
	$\mu\text{Jy}/\text{beam}$	
No primary beam	61	514
1D primary beam	39	23
full 2D primary beam	41	25

7.2 Hydra

John Conway’s “Hydra” model provides a good test of this approach. The observation simulated has the following properties:

- 1.4GHz with 50MHz bandwidth
- VLA C-configuration
- Source at declination +75 degrees.
- Hour angle from -4 to + hours
- Integration times of 60s every 300s.
- Binning of 5 degrees in parallactic angle.
- No noise was added.

The model image is shown in Figure 4. This has a number of points sources of strength up to 1Jy in Stokes I, and diffuse emission peaking at about 0.1Jy/beam. Substantial polarization is present. The 1 or 2 percent instrumental cross-polarization should have a noticeable effect on this image, as should the beam squint.

The CLEAN images are shown in Figures 5 (full primary beam model) and 6 (symmetric primary beam model - no squint). In Figure 7, I show the fractional polarization error. This is calculated from the ratio of the RSS of the Q,U errors to the RSS of the Q,U model flux, truncated to where the latter is greater than 1mJy/beam.

The CLEAN with the full primary beam model is clearly superior in the reconstruction. The fractional errors in the reconstructed polarization are mostly less than 10% compared to perhaps at least twice that for the symmetric primary beam case. Table 2 shows the rms errors for the two cases. The full primary beam I noise could probably be decreased by a deeper CLEAN, but the symmetric I noise is improving very slowly with increasing CLEAN number. Note that the V error is a very good diagnostic (as evident from the images).

The timing for the hydra reconstructions are 9272s for the full primary beam (100000 clean components, 4 major cycles) and 2422s (100000 clean components, 34 major cycles) for the symmetric model. Notice that many more major cycles are needed when a poor model of the primary beam is used. Per major cycle, the ratio of times is about 40 times worse

Table 2. *RMS error for I, Q, U, V images of Hydra*

Primary beam model	I	Q	U	V
	$\mu\text{Jy}/\text{beam}$			
Full 2D	308	102	64	24
Symmetric	782	172	143	622

for full primary beam. This can be improved by further optimization of the calculation of equations 5 and 6. It may be possible to get the two overall times to become quite close thus illustrating the importance of using an accurate description of the measurement process.

8 Conclusion

For the EVLA (and for SKA), we can expect that this algorithm will be important for reaching the quality limits of the array. Furthermore, we can expect that similar algorithms that know about image plane effects (pointing, different primary beams per antenna, multi-frequency synthesis) will also be important. The framework in AIPS++ is well-suited to the development of such algorithms. However, the computing costs can become quite large. Roughly speaking, we can expect an order of magnitude increase in computing costs for this level of sophistication. Since we are a long way from the theoretical noise, we have to be concerned that additional computing costs may appear as we try to go deeper and deeper.

A number of improvements to this existing algorithm can yet be made:

- Optimization of the calculation of equations 5 and 6.
- Adaptation to other deconvolution algorithms such as multiscale CLEAN and Maximum Entropy.
- Inclusion of component models in the image plane. Since the primary beam could be applied per component per visibility sample, binning in parallactic angle would not be necessary to deal with the brightest parts of the sky brightness.

Acknowledgements

I thank the Sanjay Bhatnagar, Kumar Golap, and George Moellenbrock for many helpful and enjoyable discussions. I thank Walter Brisken and Chris Clearfield for providing the two dimensional primary beam models, and Rick Perley for helpful comments on this memo.

References

- AIPS++ Project, 2003 <http://aips2.nrao.edu/aips++/docs/html/aips++.html>.
- Brisken, W., 2003, EVLA Memo 58.
- Conway, R.G., & Kronberg, P.P. 1969, *Mon. Not. Roy. Astron. Soc.*, 142, 11–32.
- Cornwell, T.J., Holdaway, M.H., & Uson, J.M. 1993, *Astron. & Astrophys.* 271, 697–713.
- Cornwell, T.J. 1995, *The Generic Instrument: I Overview of Calibration and Imaging*, AIPS++ Note 183.
- Cornwell, T.J. 1995, *The Generic Instrument: II Image solvers*, AIPS++ Note 184.
- Cornwell, T.J. & Wieringa M.H. 1996, *The Generic Instrument: V Design of Calibration and Imaging*, AIPS++ Note 189.
- Cornwell, T.J. 1996, *The Generic Instrument: IV Specifications and Development Plan*, AIPS++ Note 192.
- Cornwell, T.J., & Wieringa M.H., 1996, *The Generic Instrument: V Design of Cross-calibration*, AIPS++ Note 193.
- Cornwell, T.J., & Wieringa, M.H. 1997, in *Astronomical Data Analysis Software and Systems IV* eds. G. Hunt & H.E. Payne (San Francisco: PASP), 10–17.
- Cotton, W.D., 1994, AIPS Memo 86.
- Hamaker, J.P., Bregman, J.D., & Sault, R.J. 1996, *Astron. & Astrophys. Suppl.*, 117, 137–147.
- Hecht, E., & Zajac, A. 1974, *Optics*, Addison-Wesley, Reading, MA.
- Mayer C.H., McCullough T.P., & Sloanaker R.M. 1957, *Astrophys. J.*, 126, 468–470.
- Noordam, J.E. 1995, *Some practical aspects of the matrix-based Measurement Equation of a generic radio telescope*, AIPS++ Note 182.
- Noordam, J.E. 1996, *The Measurement Equation of a Generic Radio Telescope*, AIPS++ Note 185.
- Pawsey J.L., & Bracewell R.N. 1955, *Radio Astronomy*, Oxford Univ. Press, London, England.
- Sault, R.J., Hamaker, J.P., & Bregman, J.D. 1996, *Astron. & Astrophys. Suppl.*, 117, 149–159.

- Sault, R.J., Staveley-Smith, L., & Brouw, W.N. 1996, *Astron. & Astrophys. Suppl.*, 120, 374–384.
- Sault, R.J., and Cornwell, T.J., 1999, *Synthesis Imaging in Radio Astronomy II*, eds. G. Taylor, C. Carilli, and R. Perley, PASP.
- Tinbergen, J. 1996, *Astronomical polarimetry*, Cambridge Univ. Press, Cambridge, Great Britain.
- Turlo, Z., Forkert, T., Sieber, W., & Wilson, W. 1985, *Astron. & Astrophys.*, 142, 181–188.

Appendix: Simulation script

Below I show the AIPS++ script for the simulations.

```
include 'simulator.g';
# pass in an image and simulate away;
const sim:=function(algorithms='wfclean')
{
  include 'logger.g';
  dl.purge(0);

  testdir := 'sim';
  if(sim) {
    note('Cleaning up directory ', testdir);
    ok := shell(paste("rm -fr ", testdir));
    if (ok::status) { throw('Cleanup of ', testdir, 'fails!'); };
    ok := shell(paste("mkdir", testdir));
    if (ok::status) { throw("mkdir", testdir, "fails!") };
  }

  msname := spaste(testdir, '/',testdir, '.ms');
  simmodel := spaste(testdir, '/',testdir, '.model');
  simpbmodel := spaste(testdir, '/',testdir, '.pmodel');
  simsmodel:= spaste(testdir, '/',testdir, '.smodel');
  simtemp := spaste(testdir, '/',testdir, '.temp');
  simpsf := spaste(testdir, '/',testdir, '.psf');
  simempty := spaste(testdir, '/',testdir, '.empty');
  simmask := spaste(testdir, '/',testdir, '.mask');

  dir0 := dm.direction('j2000', '0h0m0.0', '25.00.00.00');
  reftime := dm.epoch('iat', '2003/07/01');

  note('Create the empty measurementset');

  mysim := simulator();

  mysim.setspwindow(row=1, spwname='LBAND', freq='1.4GHz', delt
    freqresolution='50.0MHz', nchannels=1, stokes='LL RR');

  note('Simulating VLA');
  posvla := dm.observatory('vla');
#
# Define VLA C array by hand, local coordinates
```

```

#
  xx := [41.1100006,134.110001,268.309998,439.410004,644.210022,880.309998,
1147.10999,1442.41003,1765.41003,-36.7900009,-121.690002,-244.789993,
-401.190002,-588.48999,-804.690002,-1048.48999,-1318.48999,-1613.98999,
-4.38999987,-11.29,-22.7900009,-37.6899986,-55.3899994,-75.8899994,
-99.0899963,-124.690002,-152.690002];
  yy := [3.51999998,-39.8300018,-102.480003,-182.149994,-277.589996,-387.83999,
-512.119995,-649.76001,-800.450012,-2.58999991,-59.9099998,-142.889999,
-248.410004,-374.690002,-520.599976,-685,-867.099976,-1066.42004,77.1500015,
156.910004,287.980011,457.429993,660.409973,894.700012,1158.82996,1451.43005,
1771.48999];
  zz := [0.25,-0.439999998,-1.46000004,-3.77999997,-5.9000001,-7.28999996,
-8.48999977,-10.5,-9.56000042,0.25,-0.699999988,-1.79999995,-3.28999996,
-4.78999996,-6.48999977,-9.17000008,-12.5299997,-15.3699999,1.25999999,
2.42000008,4.23000002,6.65999985,9.5,12.7700005,16.6800003,21.2299995,
26.3299999];
#
# We want something like D
#
  xx /= 3;
  yy /= 3;
  zz /= 3;

  diam := 0.0 * [1:27] + 25.0;
  mysim.setconfig(telescopename='VLA', x=xx, y=yy, z=zz,
dishdiameter=diam,
mount='alt-az', antname='VLA',
coordsystem='local', referencelocation=posvla);
  mysim.setfield(sourcename='M31SIM', sourcedirection=dir0,
integrations=1, xmospointings=1, ymospointings=1,
mosspacing=1.0);
  mysim.settimes('60s', '300s', T, '-14400s', '+14400s');
  mysim.create(newms=msname, shadowlimit=0.001,
elevationlimit='8.0deg', autocorrwt=0.0);
  mysim.done();

  imsize:=512;
  cell:='15arcsec'
  stokes:="IQUV"

  note('Make an empty image from the MS, and fill it with the points')
  myimager := imager(msname);

```

```

myimager.setdata(mode="none" , nchan=1, start=1, step=1,
  mstart="0km/s" , mstep="0km/s" , spwid=1, fieldid=1);
myimager.setimage(nx=imsize, ny=imsize, cellx=cell , celly=cell ,
  stokes=stokes , fieldid=1, facets=1, doshift=T,
  phasecenter=dir0);
myimager.make(simmodel);
myimager.done();

myimg2 := image(simmodel);

arr1 := myimg2.getchunk();
for (iy in [3*imsize/8+10,imsize/2,5*imsize/8-10]) {
  for (ix in [3*imsize/8+10,imsize/2,5*imsize/8-10]) {
    arr1[ix,iy,,1]:=[1.0,0.5,-0.3,0.1];
  }
}
myimg2.putchunk( arr1 );
myimg2.done();

mysim := simulatorfromms(msname);
mysim.setvp(dovp=T, usedefaultvp=T, dosquint=T, parangleinc='15deg'
mysim.predict(simmodel);
mysim.done();

myimager := imager(msname);
myimager.setdata(mode="none" , nchan=1, start=1, step=1,
  mstart="0km/s" , mstep="0km/s" , spwid=1, fieldid=1);
myimager.setimage(nx=imsize, ny=imsize, cellx=cell , celly=cell ,
  stokes=stokes , fieldid=1, facets=1, doshift=T,
  phasecenter=dir0);

myimager.setvp(dovp=T, usedefaultvp=T, dosquint=T, parangleinc='15d

myimager.weight(type="uniform");

bmaj:=F; bmin:=F; bpa:=F;
myimager.makeimage('psf', simpsf);
myimager.fitpsf(simpsf, bmaj, bmin, bpa);

myimager.make(simempty);

im:=image(simempty);

```

```

shape:=im.shape();
cs:=im.coordsys();
cs.summary();
im.done();
myimager.smooth(simmodel, simtemp, F, bmaj, bmin, bpa, normalize=F);
im:= image(simtemp);
im.regrid(outfile=simsmodel, shape=shape, csys=cs, axes=[1,2]);
im.done();

myimager.pb(simsmodel, simpbmodel);
myimager.regionmask(simmask, drm.quarter());

for (algorithm in algorithms) {

    simimage := spaste(testdir, '/', testdir, '.', algorithm);
    simrest  := spaste(testdir, '/', testdir, '.', algorithm, '.restored');
    simresid := spaste(testdir, '/', testdir, '.', algorithm, '.residual');
    simerror := spaste(testdir, '/', testdir, '.', algorithm, '.error');

    tabledelete(simrest);
    tabledelete(simresid);
    tabledelete(simimage);

    if(algorithm=='wfhogbom') {
        myimager.clean(algorithm='wfhogbom', niter=100, gain=1.0,
            displayprogress=F,
            model=simimage, image=simrest, residual=simresid,
            mask=simmask);
    }
    else {
        myimager.make(simempty);
        myimager.residual(model=simempty, image=simrest);
    }
    if(tableexists(simrest)) {
        imagecalc(simerror, spaste('"', simrest, '" - "', simsmodel, '"')).don
    }
}
myimager.done();
}
sim(algorithms="dirty wfhogbom");

```

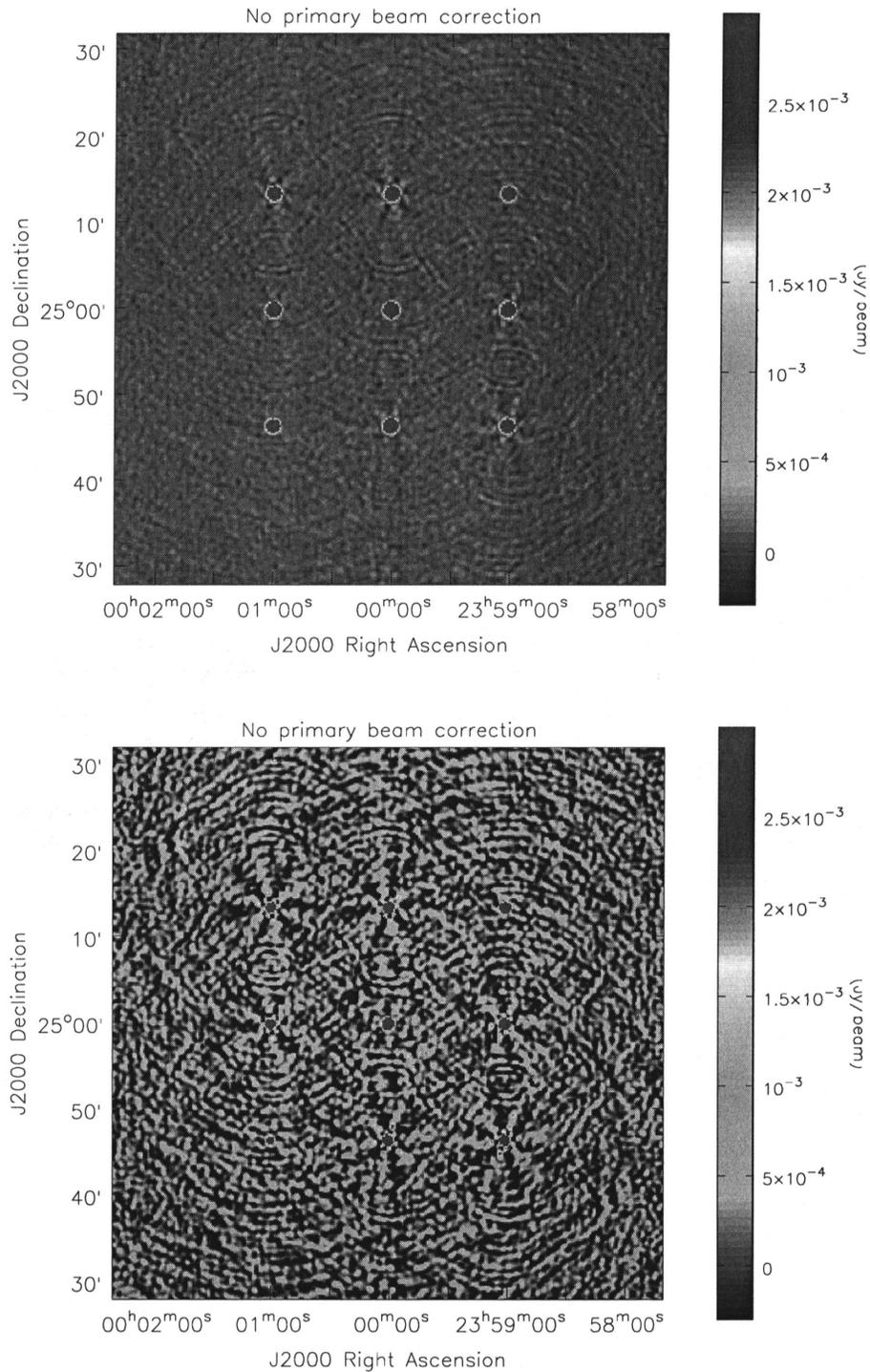


Figure 1: Cleaned image of grid of sources. Data was simulated with a 1D primary beam model (including squint) but imaged without any knowledge of primary beam. Top: Stoke I, Bottom: Stokes V

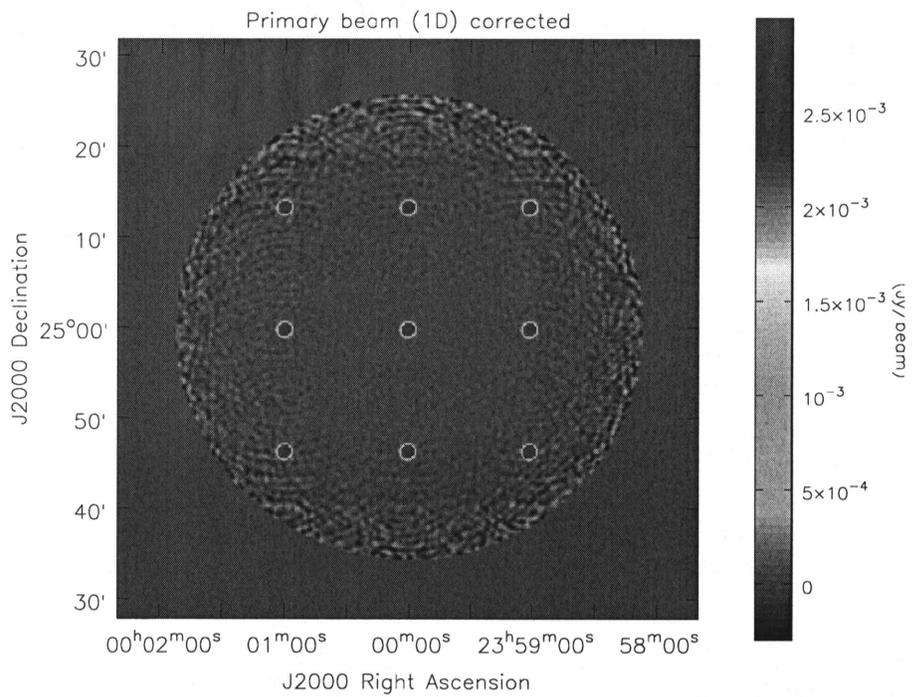
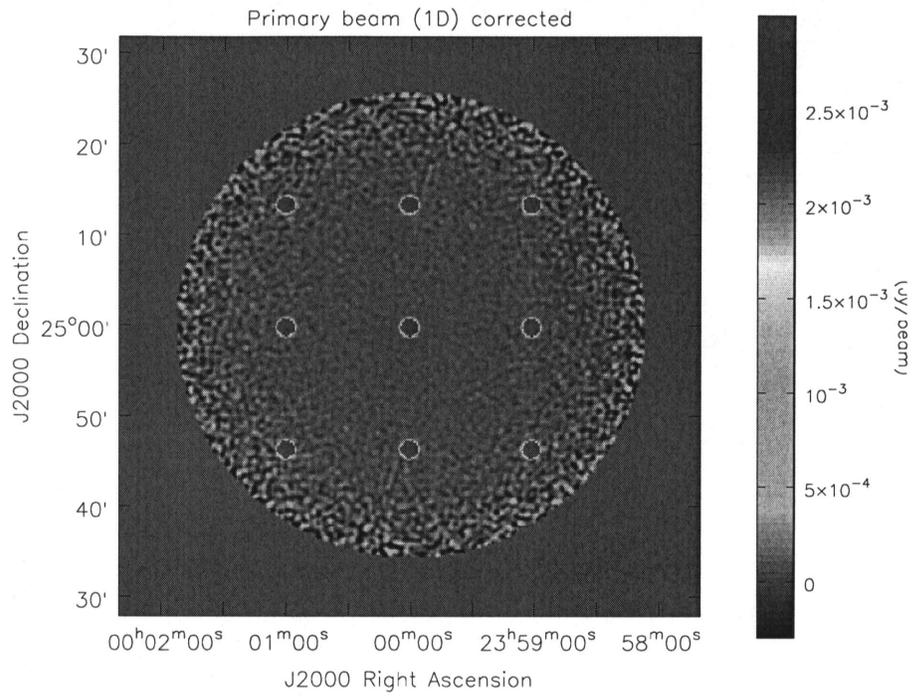


Figure 2: Cleaned image of grid of sources. Data was simulated and imaged with primary beam model (including squint). Top: Stokes I, Bottom: Stokes V

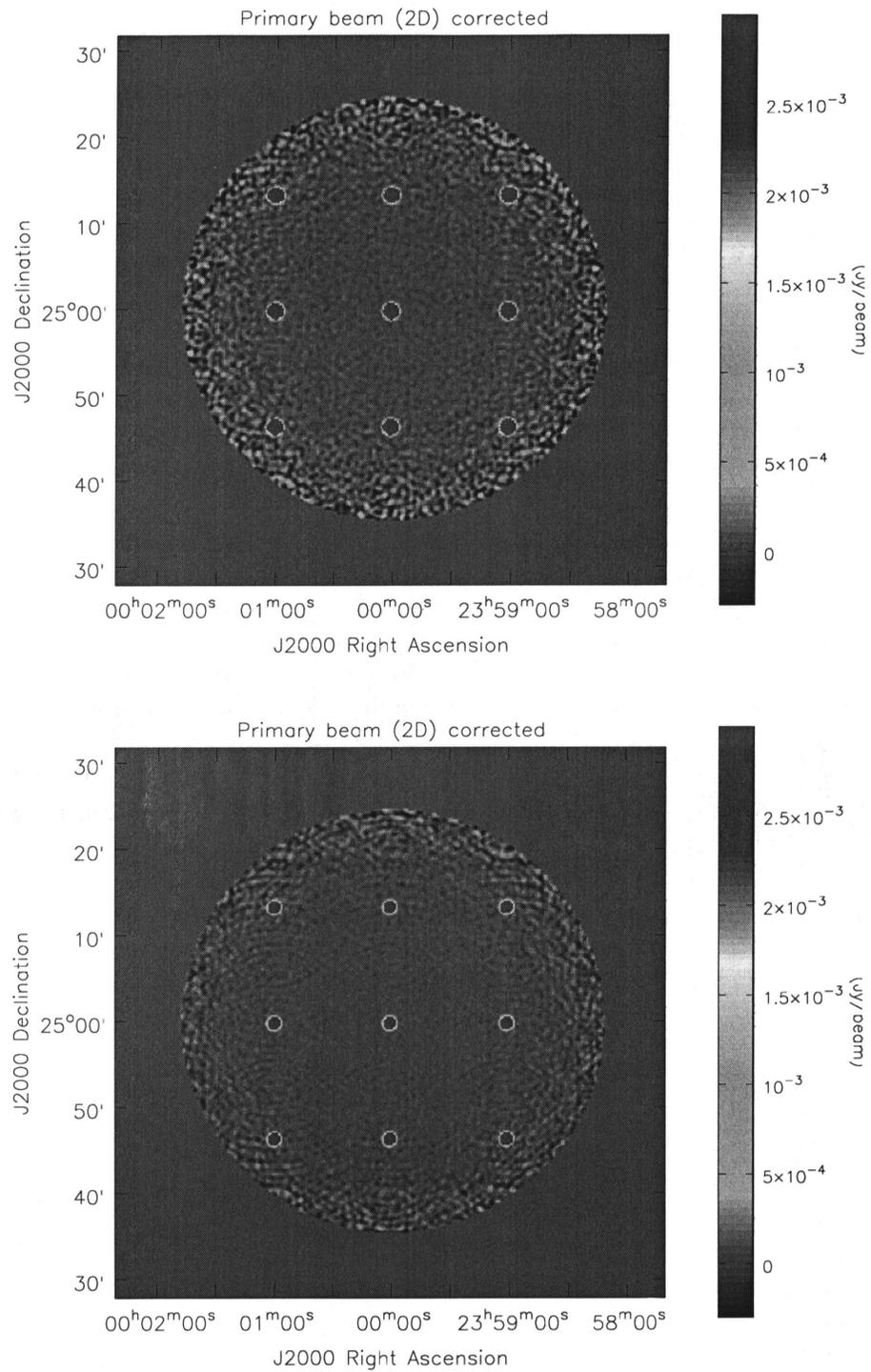


Figure 3: Cleaned image of grid of sources. Data was simulated and imaged with primary beam model. Top: Stokes I, Bottom: Stokes V

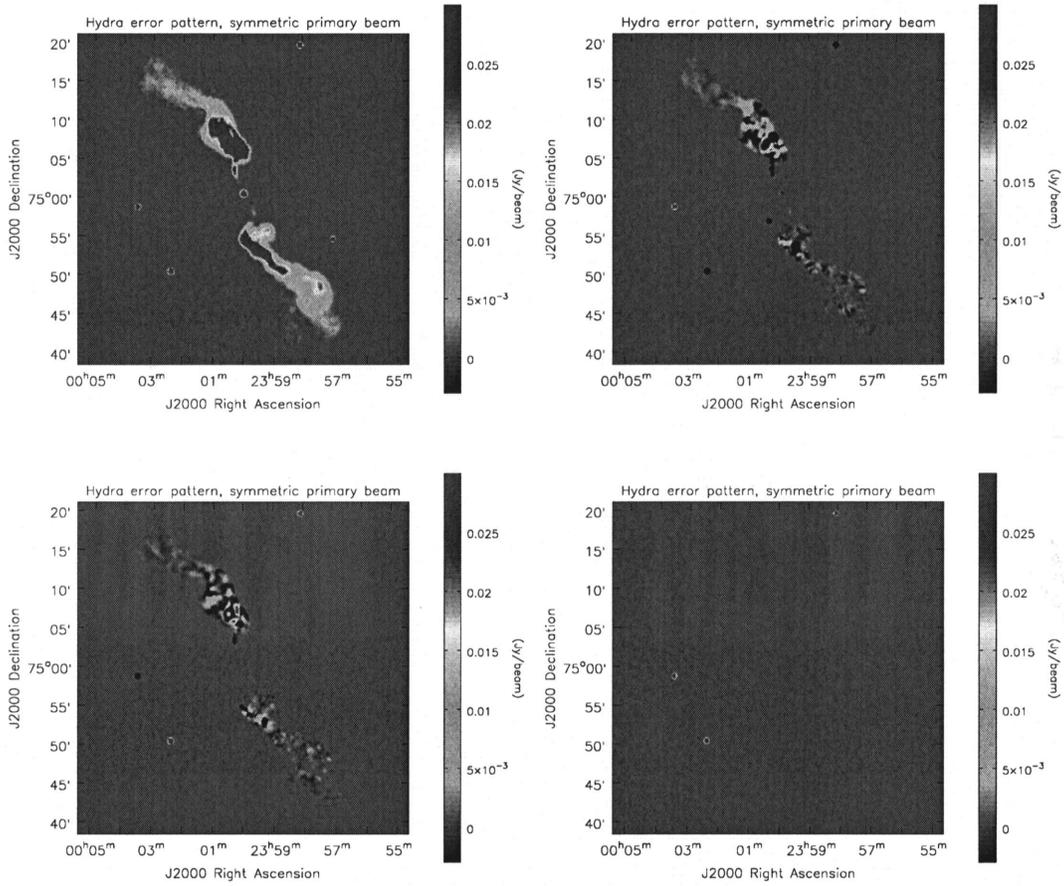


Figure 4: Model image for Conway's Hydra, Stokes I, Q, U, V

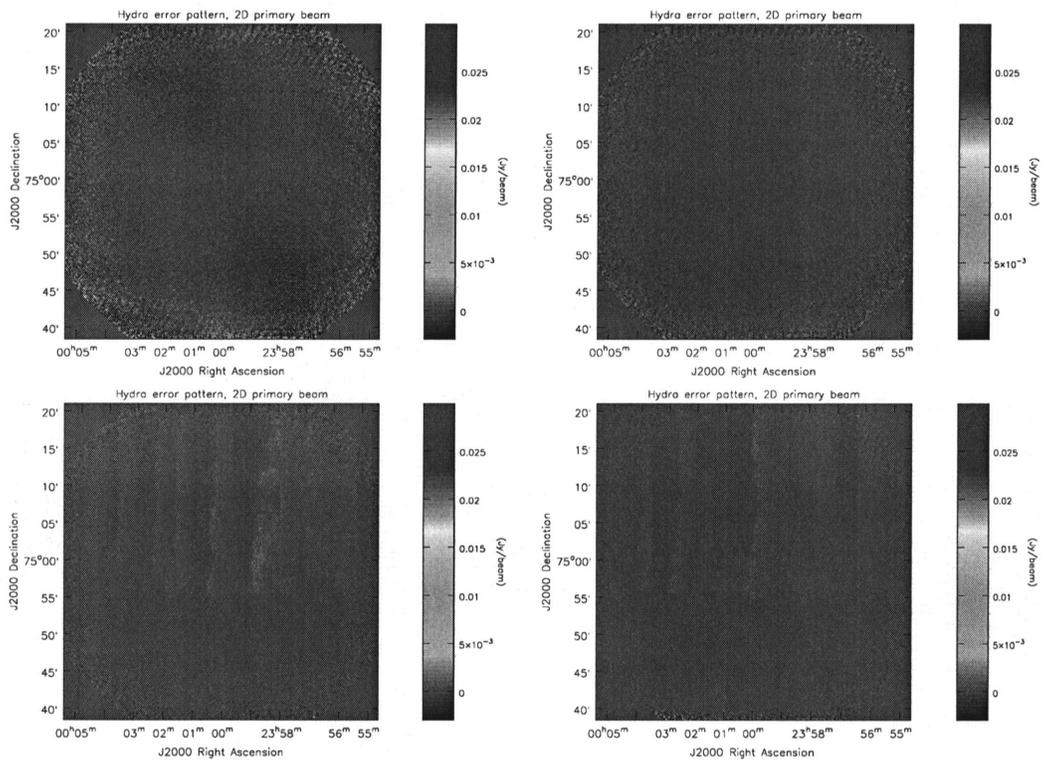


Figure 5: Error in CLEAN image for Conway's Hydra - full primary beam model, Stokes U, V

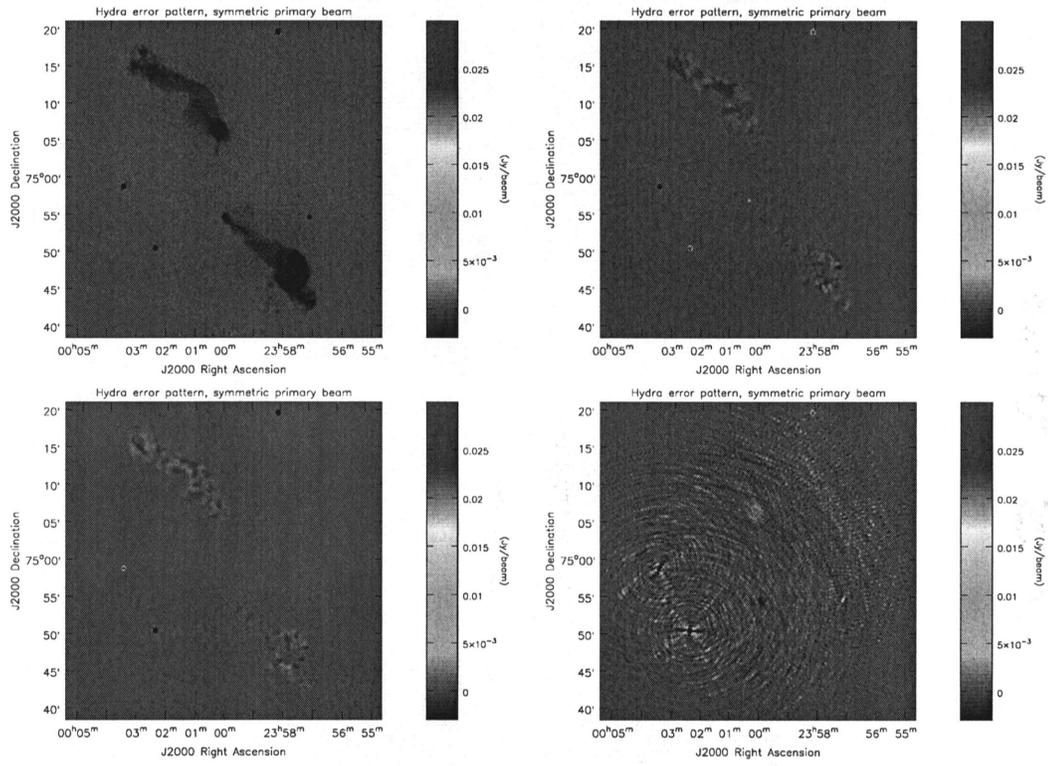


Figure 6: Error in CLEAN image for Conway's Hydra - azimuthally symmetric primary beam model, Stokes I,Q,U,V

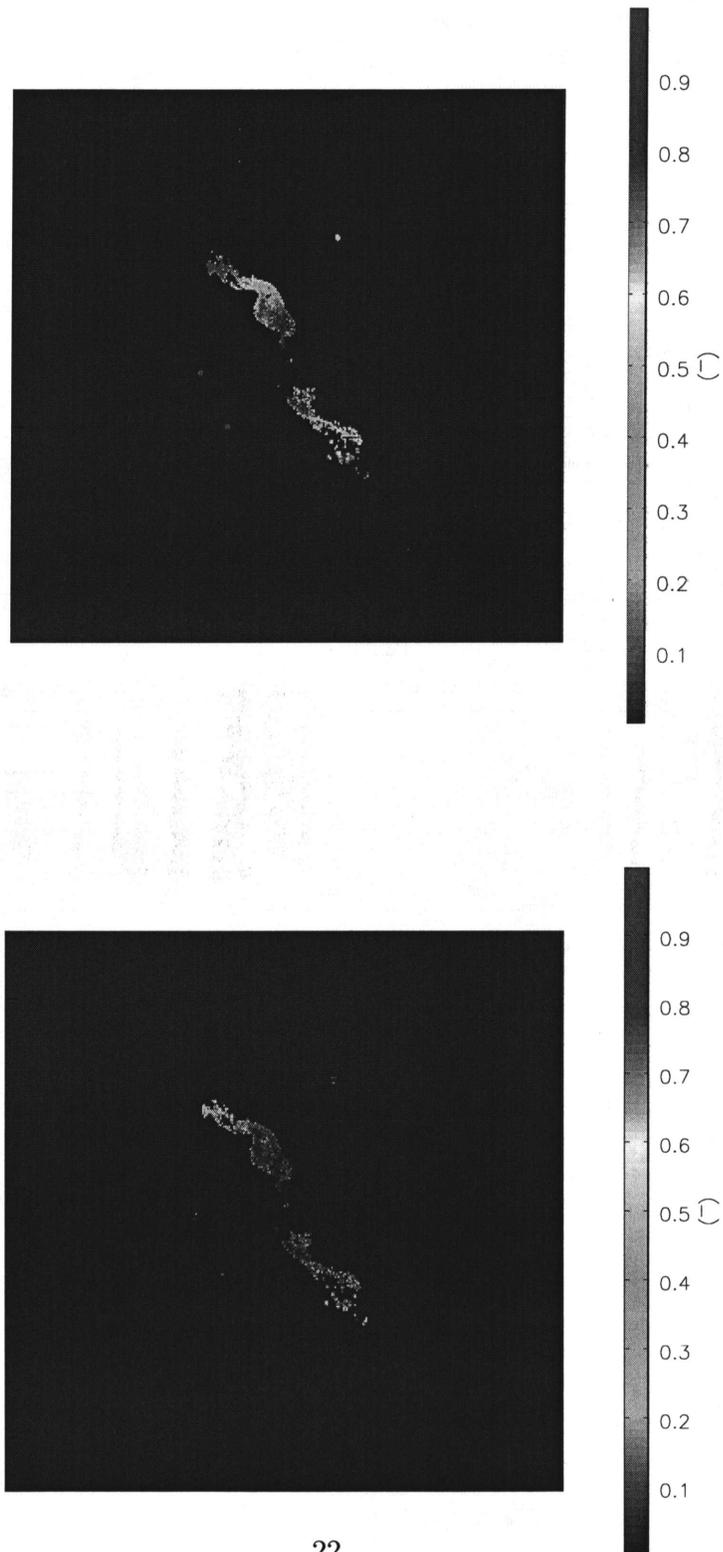


Figure 7: Fractional polarization error in CLEAN images for Conway's Hydromorphically symmetric primary beam model, bottom: full primary beam model. that of the RSS of errors in Q and U to the RSS of the Q and U model values to where the latter is greater than 1mJv/beam.