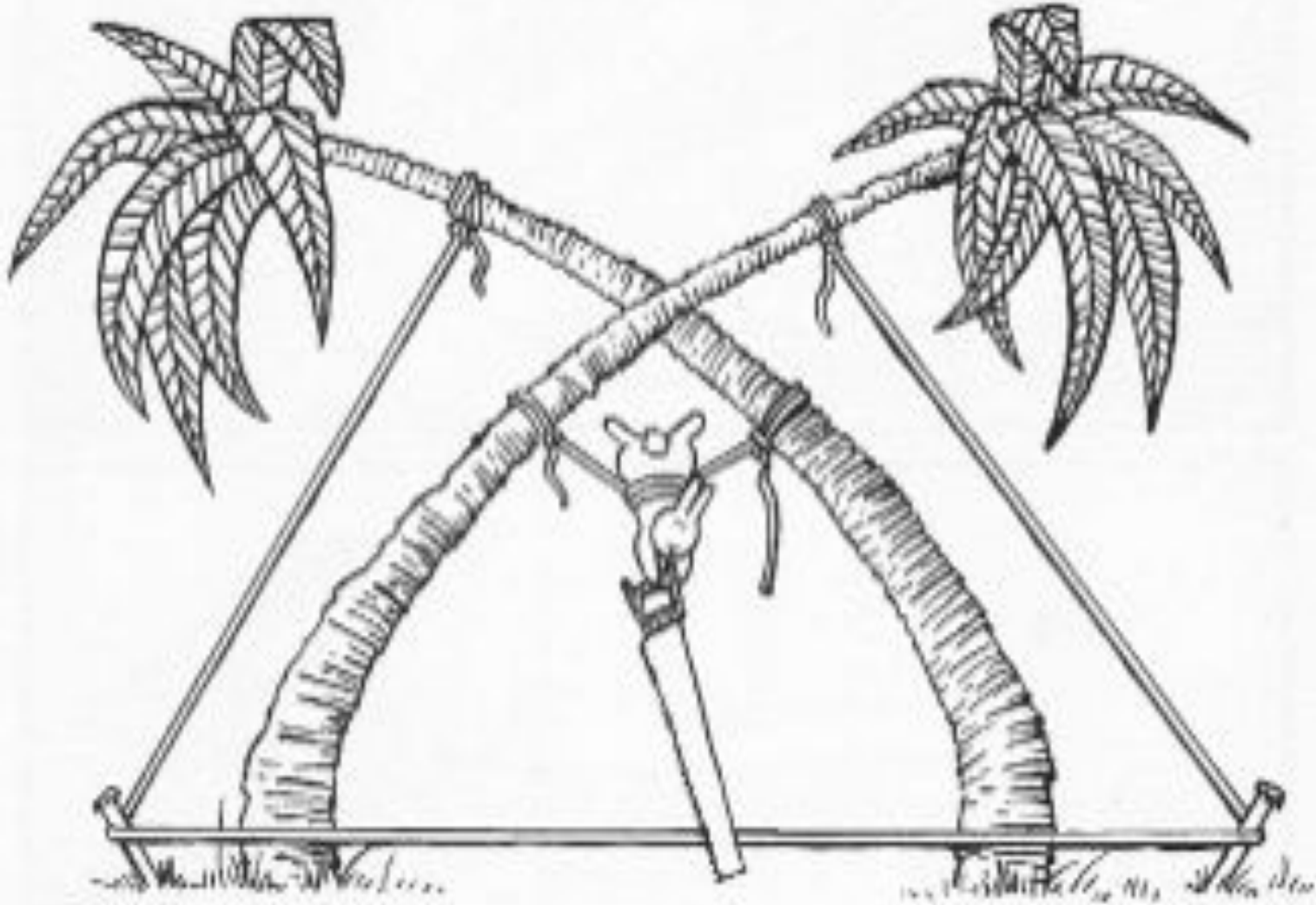


# MeqTrees, Measurement Equations, And All That



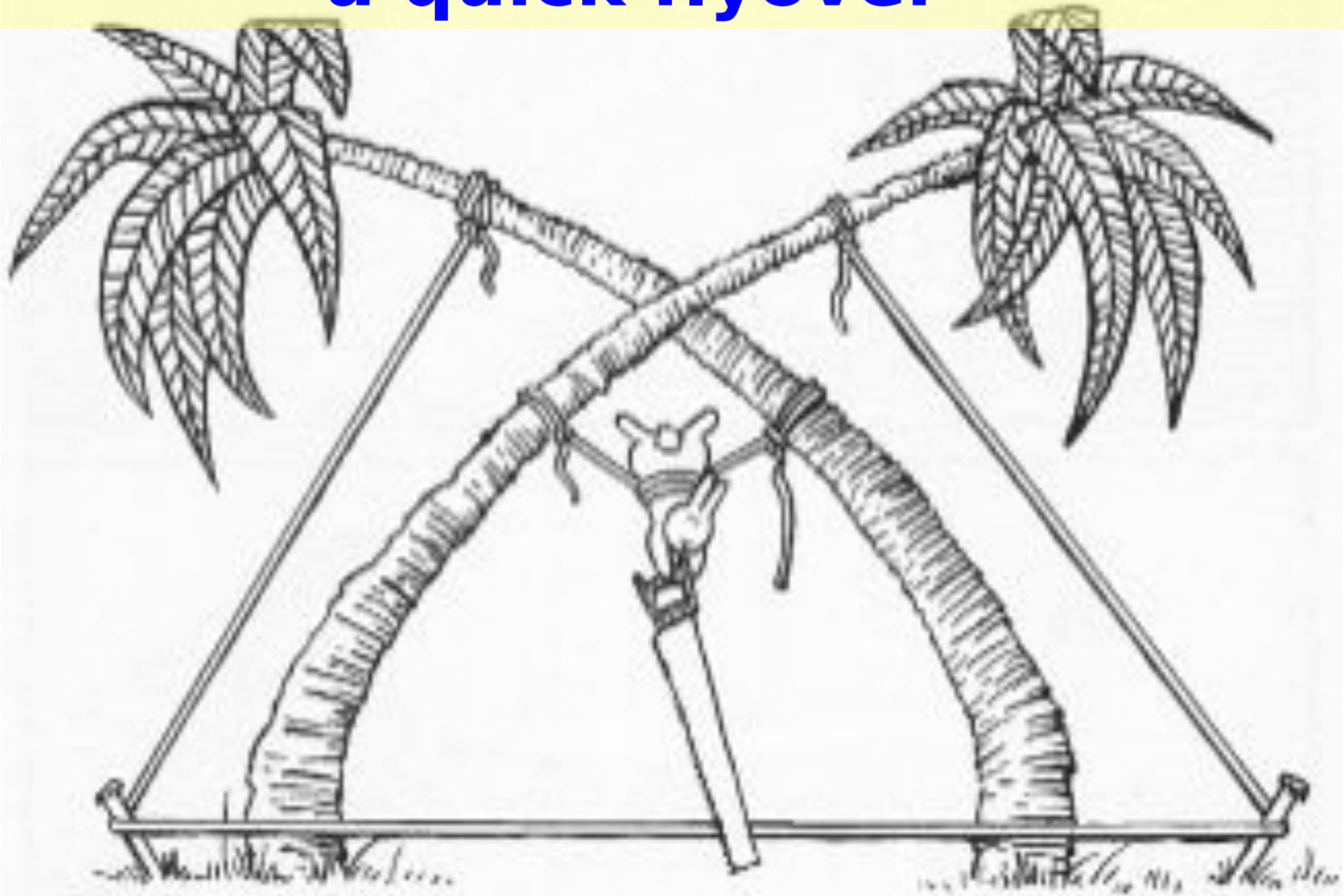
**O. Smirnov (ASTRON)**

# Today's Menu

## (A Fourier Transform Of The Official Program)

- Measurement Equations
- MeqTrees
- Practicals
  - ...all interspersed
  
- Questions welcome at any time

# The Measurement Equation - a quick flyover



# The Measurement Equation (of a generic radio interferometer)

- First formulated by Hamaker, Bregman & Sault (and further developed by Hamaker.)
- A mathematically complete and elegant description of what you actually measure with an interferometer
  - all we had before were hints and approximations
- Absolutely crucial for simulating and calibrating the next generation of radio telescopes; everything literally revolves around it.
- Like most great things, is utterly obvious in hindsight.

# A Wafer-Thin Slice of Physics: EM Field Propagation

Pick an  $xyz$  frame with  $z$  along the direction of propagation.

The EM field can be described by the complex vector  $\vec{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix}$

The fundamental assumption is **LINEARITY**:

1. Propagation through a medium is linear

⇒ can be fully described by a 2x2 complex matrix:

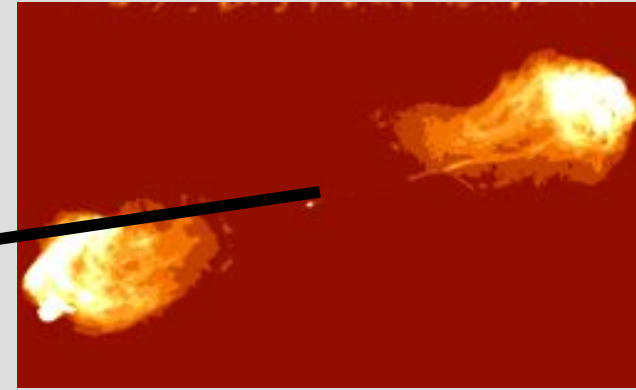
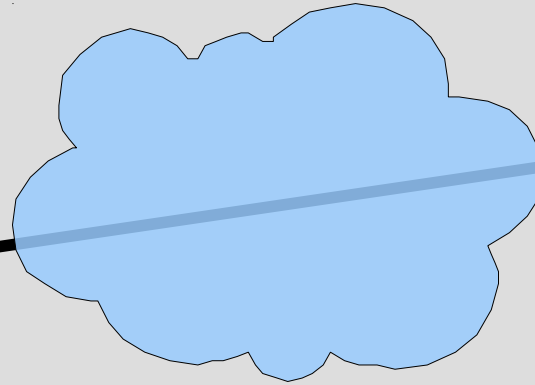
$$\vec{e}' = \mathbf{J} \vec{e} \quad \text{i.e.} \quad \begin{pmatrix} e'_x \\ e'_y \end{pmatrix} = \begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix}$$

2. Receptor voltages  $\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$  are also linear w.r.t.  $\vec{e}$

$$\Rightarrow \vec{v} = \mathbf{J} \vec{e}$$

# Single Dish

$$\vec{v} = \mathbf{J} \vec{e}$$

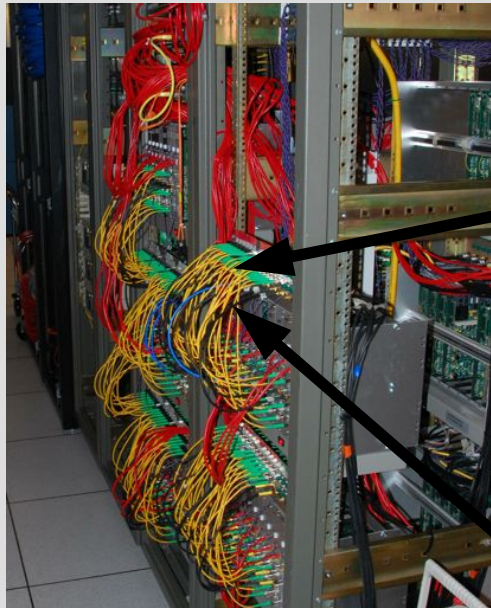


$$\vec{e}$$

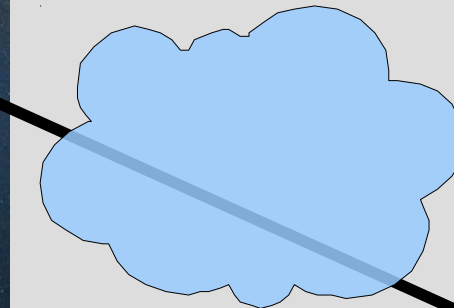
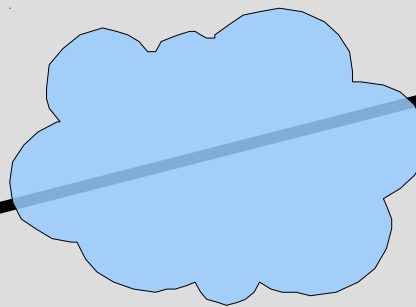
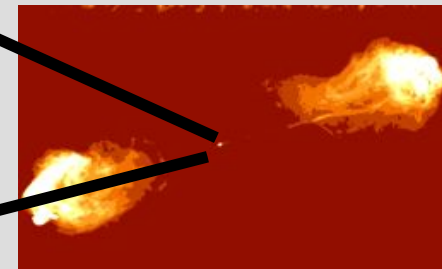
measured voltages are a complex 2-vector  $(v_x, v_y)$  -- we have two polarized *feeds*



# Interferometry



$$\vec{v}_p = \mathbf{J}_p \vec{e}$$


 $\vec{e}$ 


$$\vec{v}_q = \mathbf{J}_q \vec{e}$$

$$V_{xx} = \langle V_{px} V_{qx}^* \rangle$$

$$V_{yy} = \langle V_{py} V_{qy}^* \rangle$$

$$V_{xy} = \langle V_{px} V_{qy}^* \rangle$$

$$V_{yx} = \langle V_{py} V_{qx}^* \rangle$$

# A Wafer-Thin Slice of Physics: Correlations & Visibilities

An interferometer measures *coherencies* btw vectors  $\vec{V}_p, \vec{V}_q$ :

$$V_{xx} = \langle V_{px} V_{qx}^* \rangle, V_{xy} = \langle V_{px} V_{qy}^* \rangle, V_{yx} = \langle V_{py} V_{qx}^* \rangle, V_{yy} = \langle V_{py} V_{qy}^* \rangle$$

It is convenient to represent these as a matrix product:

$$\mathbf{V}_{pq} = 2 \langle \vec{V}_p \vec{V}_q^\dagger \rangle = 2 \left\langle \begin{pmatrix} V_{px} \\ V_{py} \end{pmatrix} \begin{pmatrix} V_{qx}^* & V_{qy}^* \end{pmatrix} \right\rangle = 2 \begin{pmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{pmatrix}$$

( $\langle \rangle$ : time/freq averaging;  $\dagger$ : conjugate-and-transpose)

$\mathbf{V}_{pq}$  is also called the *visibility matrix*.

*Now let's assume that all radiation arrives from a single point, and designate the "source" E.M. vector by  $\vec{e}$ .*



# A Wafer-Thin Slice of Physics: The M.E. Emerges

Antennas  $p, q$  then measure:  $\vec{v}_p = \mathbf{J}_p \vec{e}$ ,  $\vec{v}_q = \mathbf{J}_q \vec{e}$

where  $\mathbf{J}_p, \mathbf{J}_q$  are Jones matrices describing the signal paths from the source to the antennas.

$$\text{Then } \mathbf{V}_{pq} = 2 \langle (\mathbf{J}_p \vec{e})(\mathbf{J}_q \vec{e})^t \rangle = 2 \langle \mathbf{J}_p (\vec{e} \vec{e}^t) \mathbf{J}_q^t \rangle = \mathbf{J}_p 2 \langle \vec{e} \vec{e}^t \rangle \mathbf{J}_q^t$$

(making use of  $(\mathbf{AB})^t = \mathbf{B}^t \mathbf{A}^t$ , and assuming  $\mathbf{J}_p$  is constant over  $\langle \rangle$ )

The inner quantity is known as the *source coherency*:

$$\mathbf{B} = 2 \langle \vec{e} \vec{e}^t \rangle \equiv \begin{pmatrix} I+Q & U \pm iV \\ U \mp iV & I-Q \end{pmatrix} \leftrightarrow (I, Q, U, V)$$

which we can also call the *source brightness*. Thus:

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^t$$

# And That's The Measurement Equation!

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^\dagger$$

- Or in more pragmatic terms:

$$\overbrace{\begin{pmatrix} XX & XY \\ YX & YY \end{pmatrix}}^{\text{measured}} = \overbrace{\begin{pmatrix} j_{xx(p)} & j_{xy(p)} \\ j_{yx(p)} & j_{yy(p)} \end{pmatrix}}^{\mathbf{J}_p} \overbrace{\begin{pmatrix} I+Q & U+iV \\ U-iV & I-Q \end{pmatrix}}^{\text{source}} \overbrace{\begin{pmatrix} j_{xx(q)}^* & j_{yx(q)}^* \\ j_{xy(q)}^* & j_{yy(q)}^* \end{pmatrix}}^{\mathbf{J}_q^\dagger}$$

- NB: it is also possible to write the ME with a circular polarization basis (RR, LL, etc.) We'll use linear polarization throughout.

# Jones Matrices

- $\mathbf{J}$  is called a *Jones* matrix
- Total  $\mathbf{J}$  is a product of individual Jones terms:

$$\vec{v} = \mathbf{J}_n(\mathbf{J}_{n-1}(\dots \mathbf{J}_1 \vec{e})) = \left( \prod_{i=1}^n \mathbf{J}_i \right) \vec{e} = \mathbf{J} \vec{e}$$

where  $\mathbf{J}_1 \dots \mathbf{J}_n$  describes the full signal path.

- Order of  $\mathbf{J}$ s corresponds to the physical order of effects in your signal path.
- Matrices (usually) don't commute!

# Accumulating Jones Terms

If  $\mathbf{J}_p, \mathbf{J}_q$  are products of Jones matrices:

$$\mathbf{J}_p = \mathbf{J}_{pn} \cdots \mathbf{J}_{p1}, \quad \mathbf{J}_q = \mathbf{J}_{qm} \cdots \mathbf{J}_{q1}$$

Since  $(\mathbf{AB})^\dagger = \mathbf{B}^\dagger \mathbf{A}^\dagger$ , the M.E. becomes:

$$\mathbf{V}_{pq} = \mathbf{J}_{pn} \cdots \mathbf{J}_{p2} \mathbf{J}_{p1} \mathbf{B} \mathbf{J}_{q1}^\dagger \mathbf{J}_{q2}^\dagger \cdots \mathbf{J}_{qm}^\dagger$$

or in the "onion form":

$$\mathbf{V}_{pq} = \mathbf{J}_{pn} \left( \cdots \left( \mathbf{J}_{p2} \left( \mathbf{J}_{p1} \mathbf{B} \mathbf{J}_{q1}^\dagger \right) \mathbf{J}_{q2}^\dagger \right) \cdots \right) \mathbf{J}_{qm}^\dagger$$

# Why Is This Great?

- A complete and mathematically elegant framework for describing all kinds of signal propagation effects.
- ...including those at the antenna, e.g.:
  - beam & receiver gain
  - dipole rotation
  - receptor cross-leakage
- Effortlessly incorporates polarization:
  - think in terms of a **B** matrix and never worry about polarization again.
- Applies with equal ease to heterogeneous arrays, by using different Jones chains.

# Why Is This Even Greater?

- Most effects have a very simple Jones representation:

$$\text{gain: } \mathbf{G} = \overbrace{\begin{pmatrix} g_x & 0 \\ 0 & g_y \end{pmatrix}}^{\text{diagonal matrix}} \quad \text{phase delay: } \overbrace{\begin{pmatrix} e^{-i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}}^{\text{scalar matrix}} \equiv e^{-i\phi}$$

$$\text{rotation: } \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \equiv \text{Rot}(\gamma) \quad (\text{rotation matrix})$$

$$\text{e.g. Faraday rotation: } \mathbf{F} = \text{Rot}\left(\frac{RM}{\nu^2}\right)$$



# Three Layers Of Intuition

- **Physical**

- Beam pattern of X and Y dipoles different, causes instrumental polarization of off-center sources
- Parallax angle rotates angle of polarization

- **Geometrical**

- A Jones matrix is also a coordinate transform
- gain is stretching => instrumental polarization
- P.A. is a rotation
- The two do not commute

- **Mathematical:** matrix properties

$$\mathbf{G} = \begin{pmatrix} g_x & 0 \\ 0 & g_y \end{pmatrix} \quad \mathbf{P} = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix}$$

# ME ME ME

- The general formulation above is “The Measurement Equation” (of a generic radio interferometer...)
- When we want to simulate a specific instrument, we put specific Jones terms into the ME, and derive a measurement equation for that instrument.
- We then *implement* that specific m.e. in software (e.g. with MeqTrees)
- Existing packages implicitly use specific m.e.'s of their own.

# Obfuscating The M.E., Part 1: Evil Mueller's Evil 4×4 Formalism

- The M.E. is, at core, very simple
- Unfortunately, some approaches (prevalent in literature!) tend to make it complicated
- Outer products and Mueller matrices is one of them
  - Used in Hamaker et al.'s original ME paper (“Paper I”, 1996)
  - Picked up for Noordam's Note 185
  - Firmly entrenched with the imaging crowd
  - 2x2 version not proposed by Hamaker until Paper IV.

# Evil Mueller's Formalism

Outer (direct, tensor, Kronecker) product:

$$\vec{a} \otimes \vec{b} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix}, \quad \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11} b_{11} & a_{11} b_{12} & a_{12} b_{11} & a_{12} b_{12} \\ a_{11} b_{21} & a_{11} b_{22} & a_{12} b_{21} & a_{12} b_{22} \\ a_{21} b_{11} & a_{21} b_{12} & a_{22} b_{11} & a_{22} b_{12} \\ a_{21} b_{21} & a_{22} b_{22} & a_{22} b_{21} & a_{22} b_{22} \end{pmatrix}$$

The M.E. may be rewritten as:

$$\underbrace{\vec{V}_{pq}}_{\text{visibility vector}} = \begin{pmatrix} V_{xx}^{(pq)} \\ V_{xy}^{(pq)} \\ V_{yx}^{(pq)} \\ V_{yy}^{(pq)} \end{pmatrix} = \underbrace{(\mathbf{G}_p \otimes \mathbf{G}_q^t)(\mathbf{K}_p \otimes \mathbf{K}_q^t)}_{\text{Mueller matrices}} \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 0 & 1 & -i \\ 1 & -1 & 0 & 0 \end{pmatrix}}_{=\mathbf{S}} \underbrace{\begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix}}_{\text{Stokes vector } \vec{I}}$$

# Why Evil Mueller?

- Disadvantages of Mueller formalism are obvious: non-intuitive, too many indices to keep straight
  - Human mind only keeps track of 7 things at once
- Advantages:
  - Emphasizes that observed visibilities are linear w.r.t. input Stokes images
    - And thus beloved by imaging people
  - Makes simple things complicated
    - And thus beloved by the High Priesthood





# MeqTrees, What And Why

- A software system for building numerical models – ***simulation***
- ...and solving for their parameters – ***calibration***
- Models are usually derived via a *measurement equation*
  - (we are, after all, in the *measurement business*)
- ...and specified as *trees*
  - because this is a very flexible way to specify low-level mathematical expressions
  - the high-level user may be (blissfully) oblivious to this



# MeqTree Components

- **meqbrowser**
  - GUI front-end, provides controls & visualization,
- **meqserver**
  - Computational back-end to do the heavy work
- **TDL (Tree Definition Language)**
  - Python-based scripting language to define trees
  - Runs on the browser side
- **Frameworks**
  - High-level TDL frameworks for implementing M.E.s, simulation, calibration, etc.
- Ancillary tools (PURR, etc.)

# MeqTree Clientele

## Group 1: Developers



### Developers:

- overworked
- underpaid
- grouchy
- ...but covered in reflected glory

NB: this is not  
a picture of Oleg

# MeqTree Clientele

## Group 2: Power Users



### Power Users:

- have more fun
- steal glory from developers

NB: this is also not a picture of Oleg

# MeqTree Clientele Group 3: Button-Pushing Astronomers



The ideal  
astronomer GUI  
(Tony Willis):

**GO**

**GO FASTER**

**DO WHAT  
I MEAN!**



# Which Group Are You?



**do not  
reproduce**



**The Future!**

**going  
extinct**



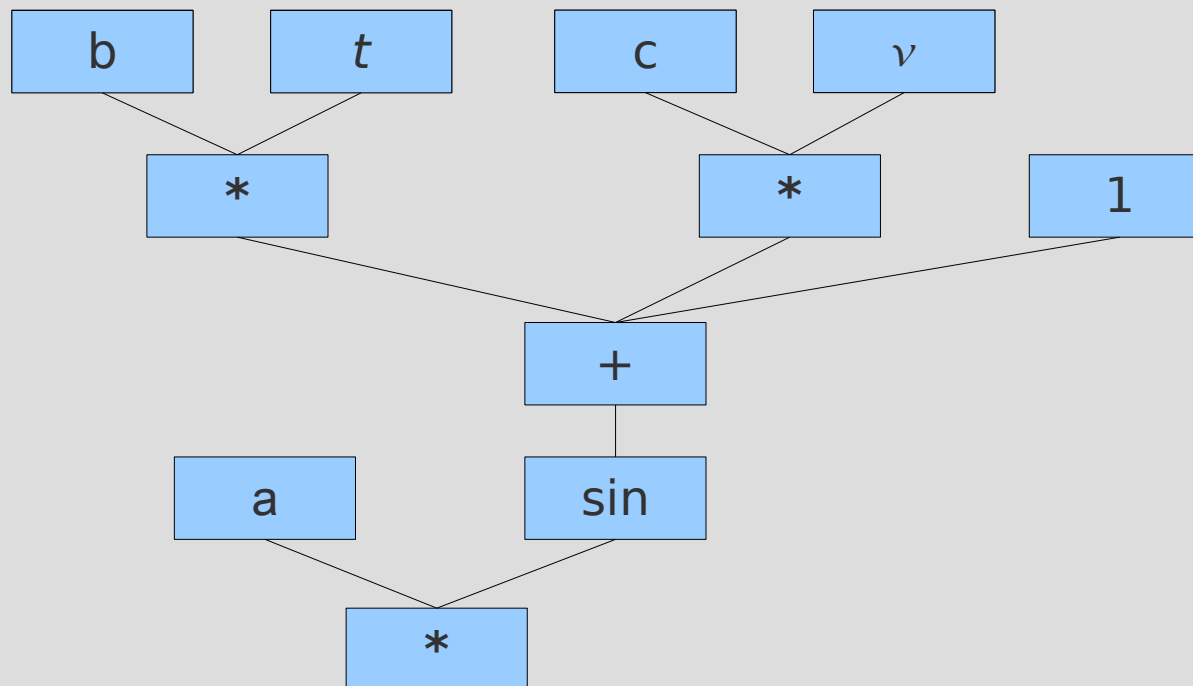




# A Very Basic Tree

- Any mathematical expression can be represented by a tree.

$$f = a * \sin(b * t + c * v + 1)$$



# Exercise 0: A Basic Tree

```
$ cd ~  
$ ./glow-meqtrees-update.sh  
$ cd GLOW2010  
$ svn up  
$ meqbrowser
```

- “Start” to start a meqserver
- TDL | Load TDL script
- Select ex0-basic-tree.py
- Bookmarks | result of 'f'
- “test forest”

# And that's all there is to it!

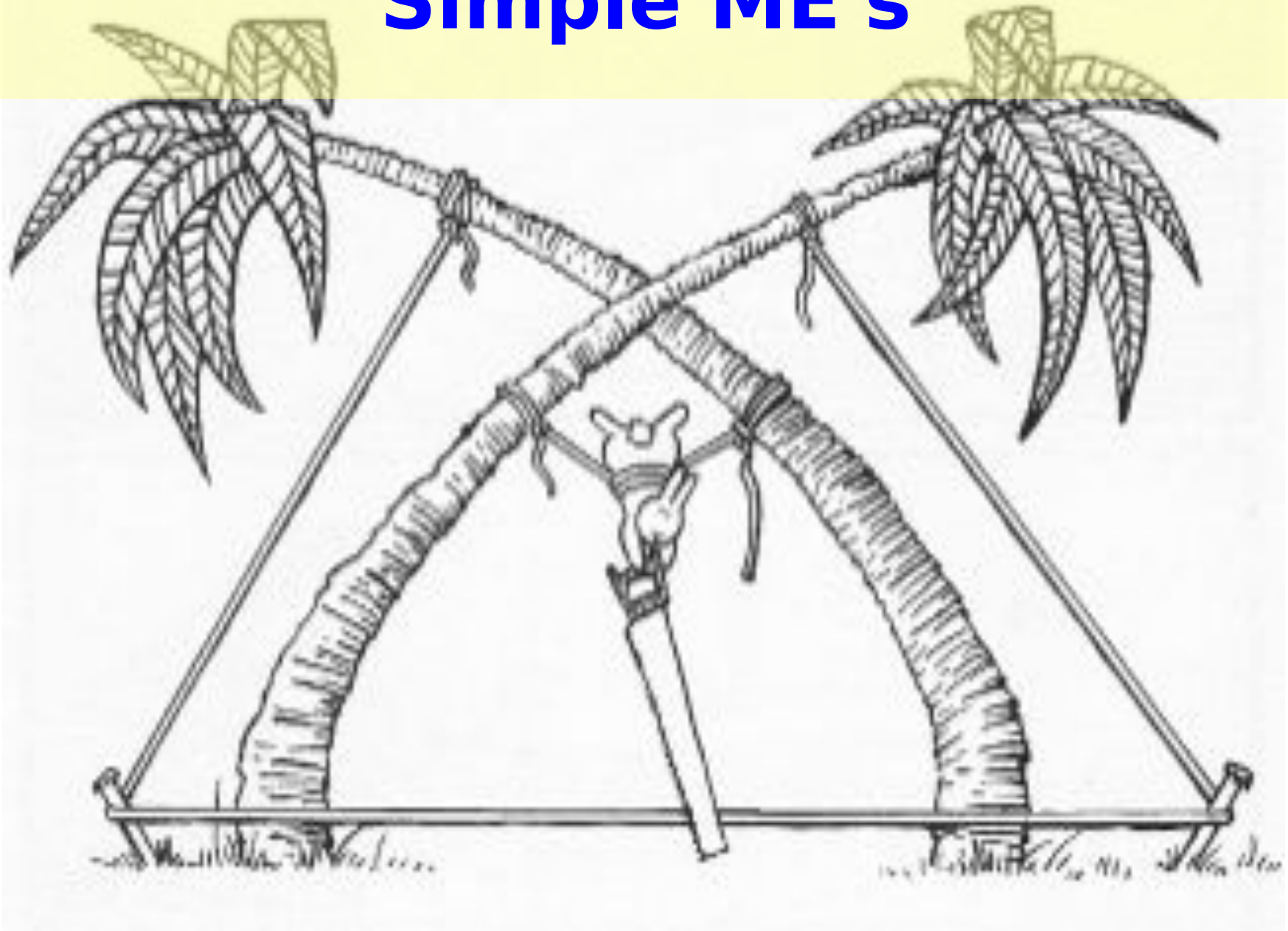
- $t, \nu$  are *variables* (can be arbitrary)
- A node is a function of  $N$  variables
  - (a constant is a trivial kind of function)
- a data source (e.g. MS, FITS image) is also a kind of function:  $V(t, \nu)$  or  $B(x, y)$
- Parent nodes combine their children into compound functions
- The tree as a whole evaluates some complicated function
  - such as a some kind of an M.E...
  - And all intermediate steps can be visualized.

# TDL: Python for trees

```
def _define_forest (ns,**kwargs):
    ns.b << 1;
    ns.c << 2;
    ns.x << Meq.Time;
    ns.y << Meq.Freq;
    a = ns.a << 297.61903062068177;
    ns.f << a*Meq.Sin(ns.b*ns.x + ns.c*ns.y + 1);

def _test_forest (mqs, parent):
    domain = meq.domain(10,20,0,10);
    cells = meq.cells(domain,num_freq=200, num_time=100);
    request = meq.request(cells, rqtype='ev');
    result = mqs.execute('f',request);
```

# Simple ME's





# Observing a point source with a perfect instrument

Even w/o instrumental effects, we still have empty space, so:

$$\mathbf{V}_{pq} = \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger \equiv \underbrace{\mathbf{X}_{pq}}_{\substack{\text{source} \\ \text{coherency}}}$$

$\mathbf{K}_p$  is the *phase shift* term, a **scalar** Jones matrix:

$$\mathbf{K}_p = \begin{pmatrix} e^{-i\phi_p} & 0 \\ 0 & e^{-i\phi_p} \end{pmatrix} \equiv e^{-i\phi_p}$$

- $\mathbf{K}$  accounts for the pathlength difference
  - (and is what makes interferometry possible in the first place...)

# The (familiar?) Scalar Case

'Classic' (scalar) visibility of a source:

$$V_{pq} = I e^{-i\phi_{pq}}$$

where  $\phi_{pq}$  is the *interferometer phase difference*:

$$\phi_{pq} = 2\pi(u_{pq}l + v_{pq}m + w_{pq}(n-1))$$

This can be decomposed into per-antenna phases

by decomposing  $(u_{pq}, v_{pq}, w_{pq}) = \vec{u}_{pq} = \vec{u}_p - \vec{u}_q$ .

$$V_{pq} = I e^{-i(\phi_p - \phi_q)} = e^{-i\phi_p} I (e^{-i\phi_q})^*$$

# Implicit m.e.'s ("What Would AIPS Do?")

- Pre-ME packages use some implicit, specific, form of the ME
- For example, a perfect point source:

$$V_{xx,pq} = (I + Q) e^{-i\phi_{pq}} = e^{-i\phi_p} (I + Q) (e^{-i\phi_q})^*$$

$$V_{yy,pq} = (I - Q) e^{-i\phi_{pq}} = e^{-i\phi_p} (I - Q) (e^{-i\phi_q})^*$$

etc...

compare this to:

$$\mathbf{V}_{pq} = \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger,$$

$$\text{with } \mathbf{B} = \begin{pmatrix} I + Q & 0 \\ 0 & I - Q \end{pmatrix}$$

# Introducing Complex Gains

- The “classic” view: each receiver has a complex amplitude and phase term (troposphere/electronics/etc.)

$$V_{xx,pq} = (I + Q) e^{-i\phi_{pq}} g_{x,p} g_{x,q}^*$$

$$V_{yy,pq} = (I - Q) e^{-i\phi_{pq}} g_{y,p} g_{y,q}^*$$

$$V_{xy,pq} = (U + iV) e^{-i\phi_{pq}} g_{x,p} g_{y,q}^*$$

$$V_{yx,pq} = (U - iV) e^{-i\phi_{pq}} g_{y,p} g_{x,q}^*$$

# Gains: The ME View

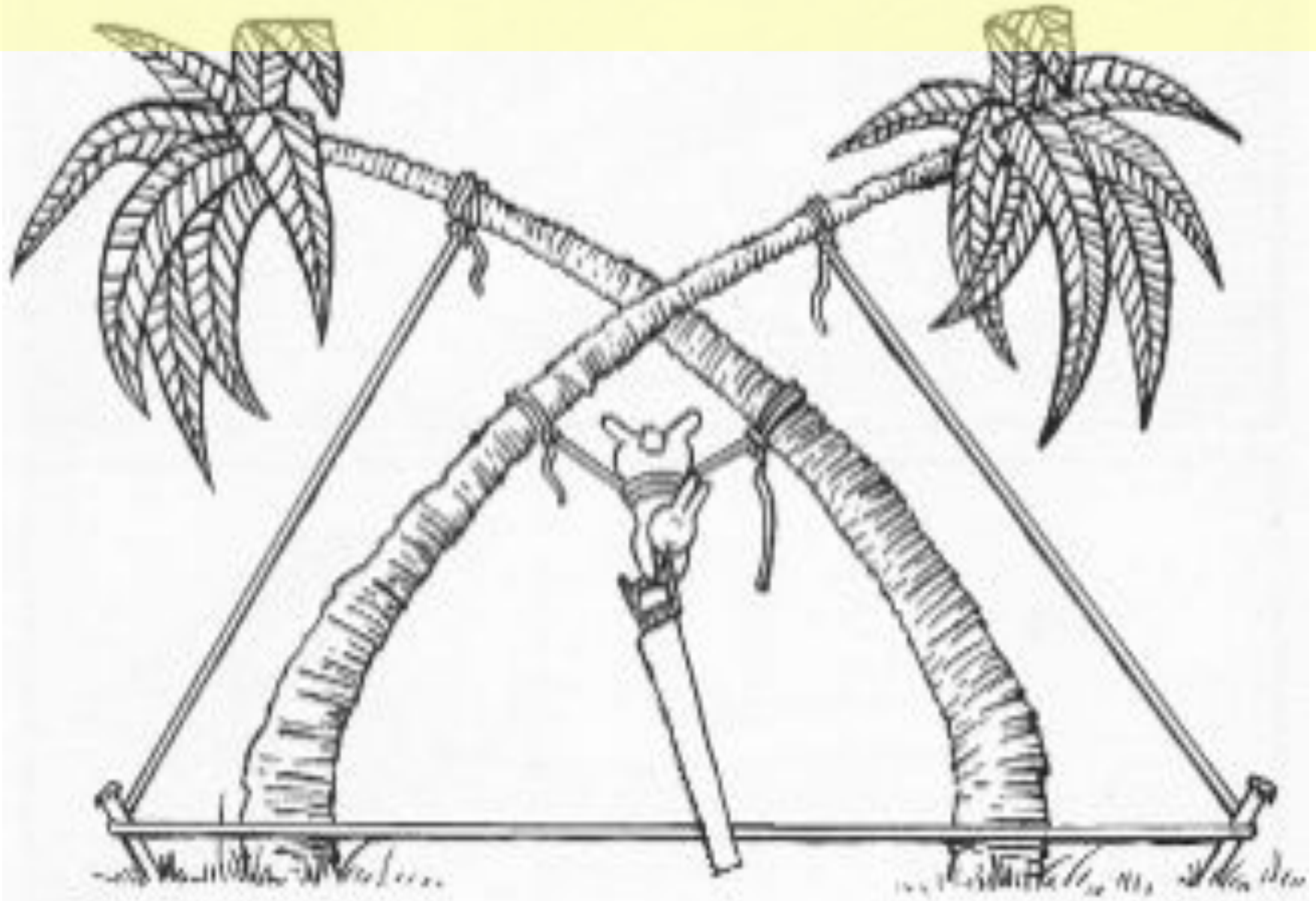
$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{K}_p \mathbf{B} \mathbf{K}_q^t \mathbf{G}_q^t = \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^t$$

$$\mathbf{G}_p = \begin{pmatrix} g_{x,p} & 0 \\ 0 & g_{y,p} \end{pmatrix}$$

and with multiple sources:

$$\begin{aligned} \mathbf{V}_{pq} &= \mathbf{G}_p \left( \sum_s \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)t} \right) \mathbf{G}_q^t = \\ &= \mathbf{G}_p \left( \sum_s \mathbf{X}_{pq}^{(s)} \right) \mathbf{G}_q^t \quad \left( \sum \rightarrow \iint, \text{ which gives the Fourier Transform} \right) \end{aligned}$$

# Simulations Intro



# MeqTree Frameworks

- By having the ME defined in Python, we get endless flexibility...
- ...and also (as it turns out) endless confusion
  - even the simplest ME involves many details to keep track of
  - and not everybody wants to be a programmer
- On the other hand, the same building blocks are reused over and over again
  - sources, Jones matrices, etc.
- Frameworks to the rescue...



# Meow

(Measurement Equation Object frameWork)

software  
bloat





# Meow

- Object-oriented framework for putting together ME-related trees
- Better than “pure TDL”
  - but still (kind of) low-level
  - intended for the *advanced* MeqTree user
- Deals with objects such as *Observation*, *IfrArray*, *SkyComponent*, *PointSource*, *GaussianSource*, etc.
- Base for higher-level frameworks

# Siamese

## (Simulations In Your Sleep)

- Siamese is a Meow-based simulator framework



# Exercise: Make MS

```
$ cd ~/GLOW2010/MS
$ sudo apt-get install makems
$ makems WSRT_makems.cfg
$ mv WSRT.MS_p0 WSRT.MS
```

- The **makems** tool makes empty Measurement Sets that we can then fill with simulated data
- Uses a config file to specify an observation
  - Look inside!
  - (Also need an antenna positions table)

# Exercise: Simulation 1

## (a perfect point source)

```
$ cd ~/GLOW2010/Sim
```

```
$ meqbrowser
```

- TDL | Load TDL Script | example-sim.py
- You see a dialog of “Compile-time options”
  - Click on “Load” and “exercise1”
- Options defined by script itself
  - MeqTrees provides GUI and config file support
- Tour of options:
  - Measurement Set, Local Sky Model, Jones terms
- Press “Compile”



# PURR

- “PURR is Useful for Remembering Reductions”
- Disciplined people keep notes
- Undisciplined people write software

# Using PURR

- The object of PURR is to make note-keeping as effortless as possible
- PURR watches your working directory for new or modified files (“data products”)
  - configuration files, images, screenshots
- Offers to save them to a log
  - ...along with descriptive comments
  - And useful rendering of things like images
- Purrrlogs are natively saved in HTML and may be immediately published or shared

# Example PURR Logs

## **Calibrating 3C147:**

[http://www.astron.nl/meqwiki-data/users/oms/  
3C147-Calibration-Tutorial/purrlog/](http://www.astron.nl/meqwiki-data/users/oms/3C147-Calibration-Tutorial/purrlog/)

## **Enthroned chicken:**

[http://www-astro.physics.ox.ac.uk/~ianh/  
PURRLOGS/enthroned/](http://www-astro.physics.ox.ac.uk/~ianh/PURRLOGS/enthroned/)

# Exercise: Simulation 1

## cont'd

- Compile script
- Bookmarks | Output visibilities inspector
- Jobs & Runtime options
  - All sorts of I/O etc. settings
  - ...and “Jobs” you can execute
- Start “Simulate MS”
- Once it's done, go to “Imaging options: Make a dirty image”
- Admire your first image, and don't forget to save it to a purrlog entry



# Exercise: Simulation 2

## (complex gains)

- We'll throw a **G** Jones into the mix
- The G Jones module provided here implements a simple error model: sine wave
- More realistic error models may be plugged in
  - Implementation is just a bit of Python code
- Rerun script
  - Grid model, 5x5 mJy sources at 5', 1 Jy at center
  - enable G Jones phase error
  - 120 degrees, 2-4 hours
  - Add .1 Jy noise
- Open bookmarks

# Visualization Everywhere

- One of the guiding principles of MeqTrees: everything can be visualized
  - any intermediate calculation or result may be published into the browser and plotted
- But some visualizations are more interesting than others
  - the script (i.e. its author) knows which these are
- Scripts can define “bookmarks” for interesting visualizations
- Run, make image, etc.
  - Set output column to DATA (we'll try to calibrate it later)

# End-To-End Simulations: some practical uses

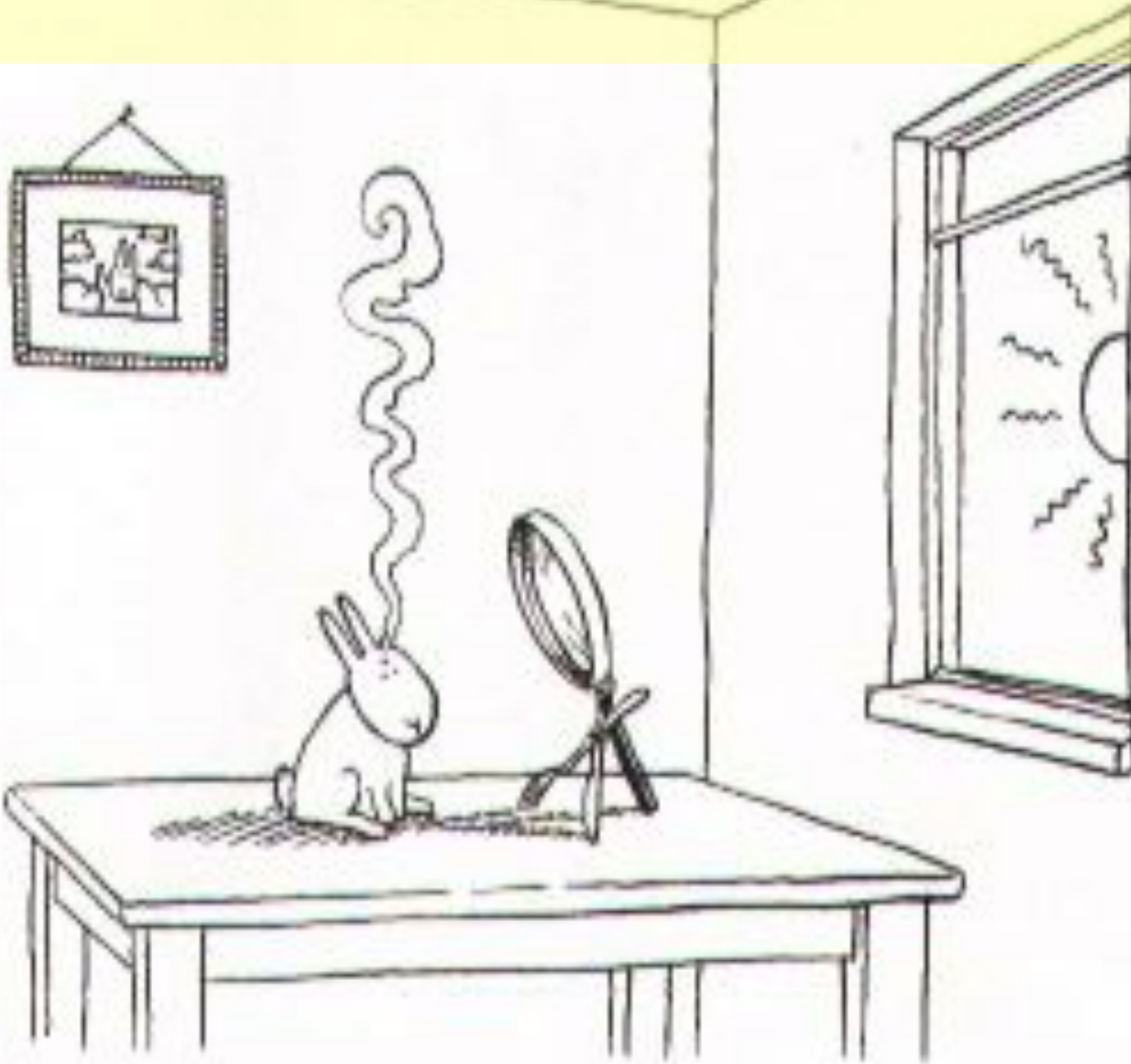
- Appeasing managers
- Getting funding
- Keeping idle PhD students out of one's hair
- Filling up disk space with simulated data
- Honing programming skills

By the time you've finished your e2e simulator, the goalposts have moved and your initial assumptions have become meaningless. And the final instrument is going to be different yet again. So why bother generating garbage “data”?

# Simulations: Some Real Uses

- Study effects in isolation to understand them better
- Stick to small, self-contained simulations to
  - Increase your understanding
  - Explore parameter spaces and boundaries of problems
- Increase sophistication when studying interaction between effects

# Polarization



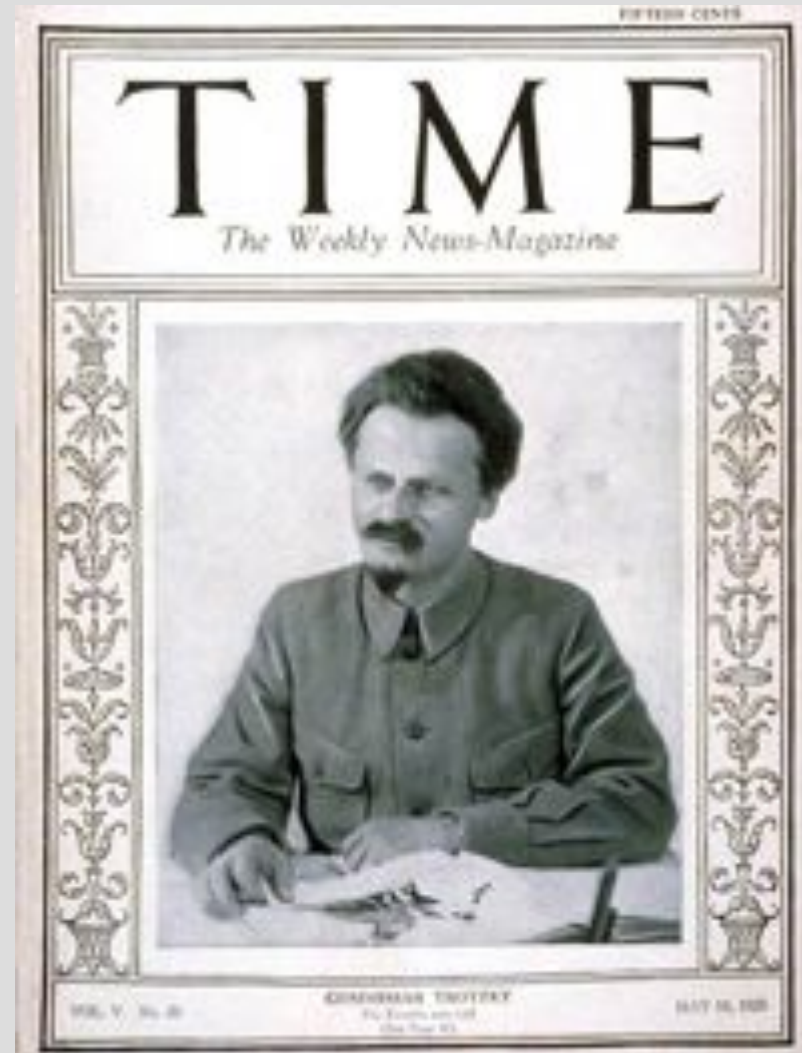
# The Classical Approach To Polarization



## ...and why it doesn't work

*"You may not be interested in the polarization, but the polarization is interested in you."*

*– (wrongly) attributed to Leon Trotsky*





# Example: Differential Faraday Rotation

- Early testing of LOFAR Ef-Ex baseline showed puzzling signal
  - strong XY/YX, dropouts on XX/YY, on unpolarized source
- Eventually realized it was caused by *differential Faraday rotation*
- Was predicted (& forgotten) by Hamaker et al. in original ME paper
- According to James Anderson, was known in the VLBI community during the 1960-70s, hence choice of circularly polarized feeds



# DFR: The Physical View vs. The M.E. View

- Physical view: (lots of handwaving)



# DFR: The Physical View vs. The M.E. View

- Assume a 1 Jy unpolarized source at the phase centre, and no other corruptions:

$$\mathbf{V}_{pq} = \mathbf{F}_p \mathbf{B} \mathbf{F}_q^\dagger$$

$$\mathbf{V}_{pq} = \begin{pmatrix} \cos \phi_p & -\sin \phi_p \\ \sin \phi_p & \cos \phi_p \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi_q & \sin \phi_q \\ -\sin \phi_q & \cos \phi_q \end{pmatrix}$$

and now for  $\phi_p = 0$ ,  $\phi_q = \pi/2$ :

$$\mathbf{V}_{pq} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

# DFR: The M.E. View vs. The Evil Mueller View

$$\vec{V}_{pq} = \begin{pmatrix} \cos \phi_p & -\sin \phi_p \\ \sin \phi_p & \cos \phi_p \end{pmatrix} \otimes \begin{pmatrix} \cos \phi_q & \sin \phi_q \\ -\sin \phi_q & \cos \phi_q \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 0 & 1 & -i \\ -1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} =$$

$$\vec{V}_{pq} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & i \\ 0 & 0 & 1 & -i \\ -1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} =$$

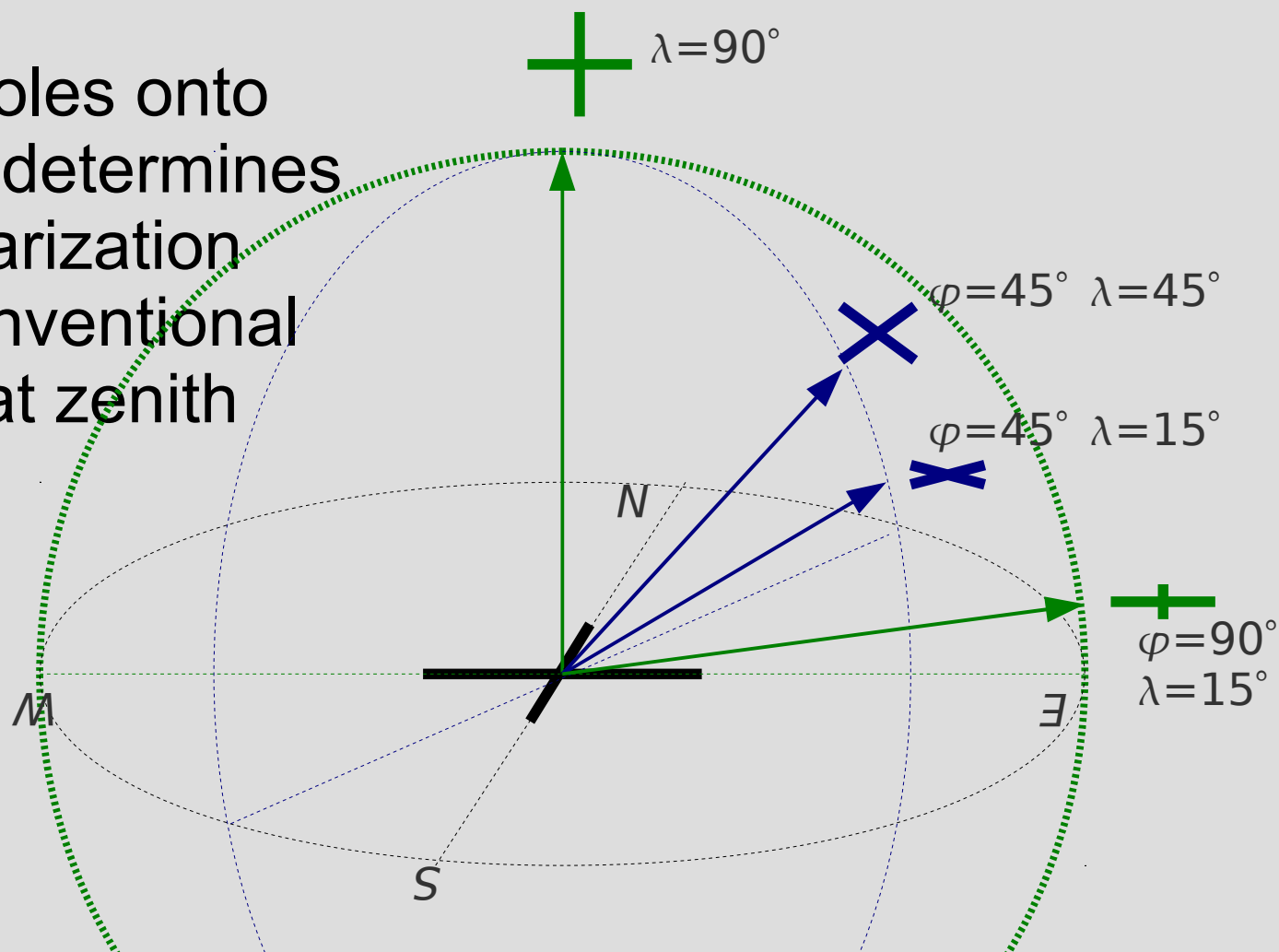
$$\dots = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}$$

# Exercise: Simulating DFR

- Don't have a proper sim for DFR
  - requires a 3D ionosphere and a model for the Earth Magnetic Field
  - Implementations will be gratefully accepted!
- But we can get an idea of the effect by rotating the dipoles around a bit
- Load the simulator, and enable P-Jones
  - Or load the exercise3 profile
- Simulate, make an IQUV image, and try to figure out what's going on

# More Gratuitous Polarization: Dipole Projection

- Aperture array with fixed NS and EW dipoles
- Projection of dipoles onto tangential plane determines sensitivity to polarization
- Equivalent to conventional dipole pair only at zenith



# Dipole Projection Jones Matrix

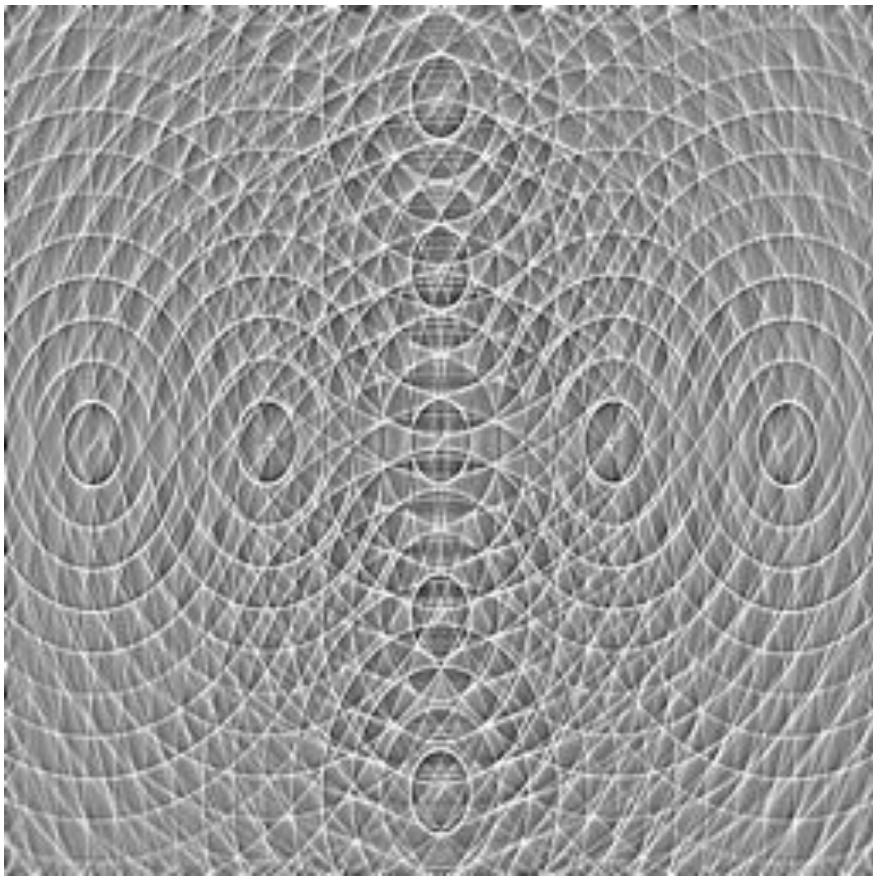
- Projection can be described by a Jones matrix:

$$\mathbf{L}(\varphi, \lambda) = \begin{pmatrix} \cos \varphi & -\sin \varphi \sin \lambda \\ \sin \varphi & \cos \varphi \sin \lambda \end{pmatrix}$$

- Function of azimuth/elevation, so:
  - Varies with time
  - Varies with source position, given a wide field
  - Varies with station position, given a large array

# Exercise: L-Jones Simulation

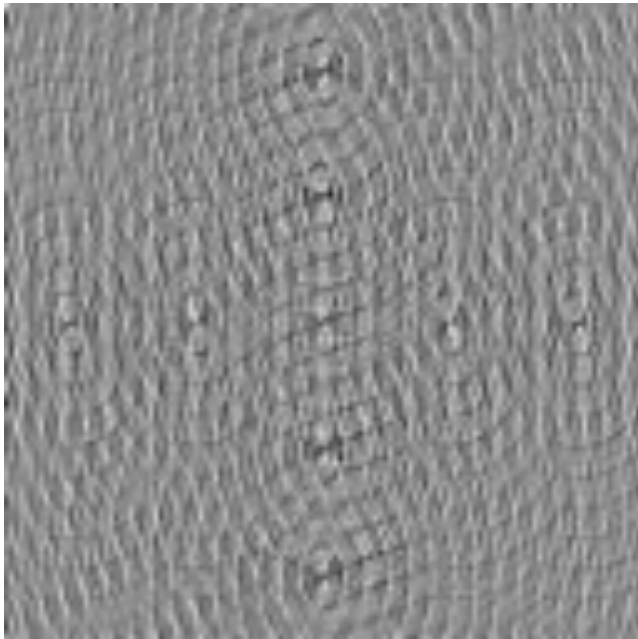
- Sky model: 5x5 cross at 30'
- Enable L Jones
  - Per-source but not per-station
- Open bookmarks to check az/el and L Jones
- Make an IQUV image
  - Note distortions in I map due to time-varying sensitivity of the dipoles
  - Note instrumental QU polarization – direction-dependent!



Stokes  $I$  map.

Note distortions in source shape. These are caused by time-varying sensitivity of the dipoles to *total flux*.

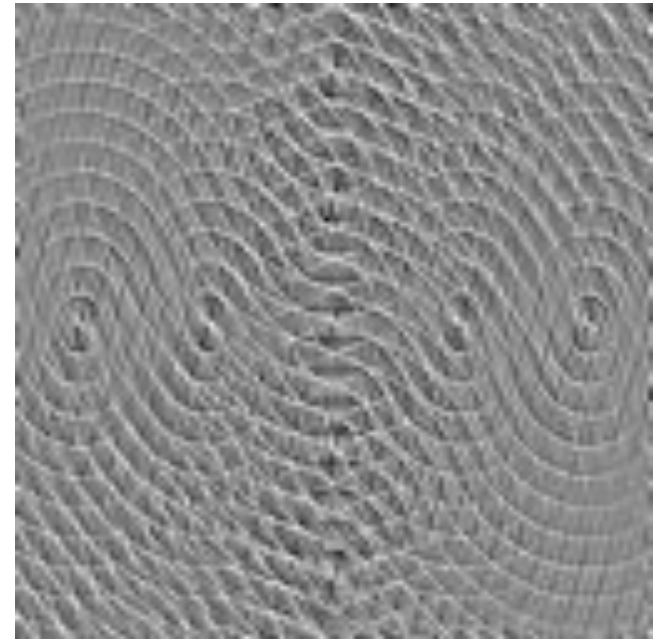
Peak flux is  $\sim .6$  Jy (would be 1 Jy without this effect!)



$Q$  and  $U$  maps.  
Note instrumental polarization

(direction-dependent!)

Peak flux is  $\pm 0.1$  Jy

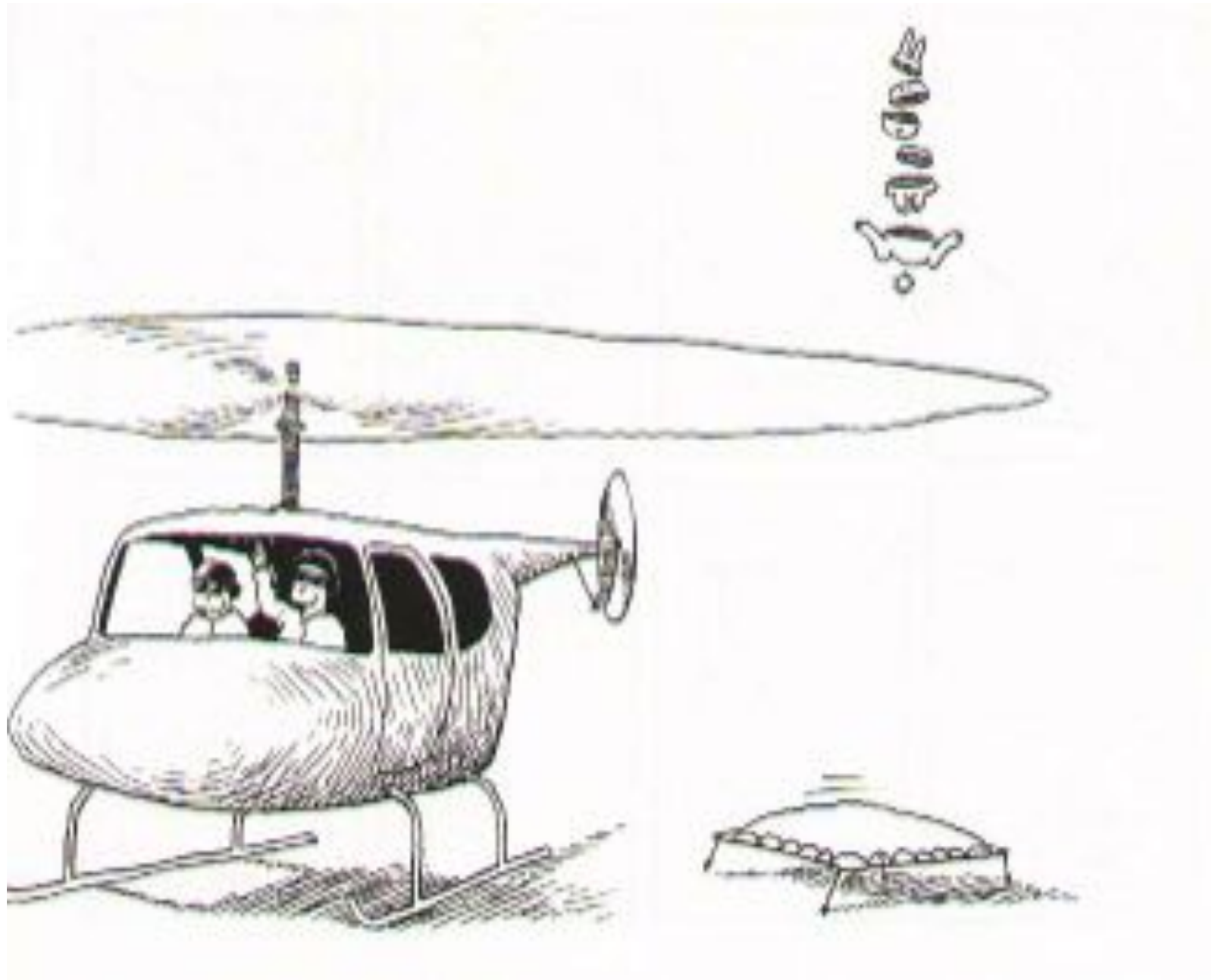




# Optional Exercises

- Instrumental polarization
  - Differences in G-Jones
  - VLA beam squint
- Effects of parallactic angle

# Perils Of The Ionosphere



# The Full-Sky ME

ME of a single point:

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^\dagger$$

The sky has a brightness density:  $\mathbf{B}(\vec{\sigma})$

(where  $\vec{\sigma}$  is a unit direction vector)

So the total visibility is obtained by integrating over a sphere:

$$\mathbf{V}_{pq} = \int_{\text{sky}} \mathbf{J}_p(\vec{\sigma}) \mathbf{B}(\vec{\sigma}) \mathbf{J}_q^\dagger(\vec{\sigma}) d\Omega$$

This is not very useful, so we project  $\mathbf{B}$  onto the  $lm$  plane, tangential at the phase centre...

# The Full-Sky ME

When projecting an integral, we must also project the integration volume:

$$d\Omega = \frac{dl dm}{\sqrt{1-l^2-m^2}} = \frac{dl dm}{n},$$

and in the  $lm$  plane we get:

$$\mathbf{V}_{pq} = \iint_{lm} \mathbf{J}_p(l, m) \frac{\mathbf{B}(l, m)}{n(l, m)} \mathbf{J}_q^{\dagger}(l, m) dl dm$$

# Image-plane vs. *uv*-plane

$\mathbf{J}_p$  is composed of multiple effects:  $\mathbf{J}_p = \mathbf{J}_{pn} \mathbf{J}_{pn-1} \cdots \mathbf{J}_{p1}$

( $\mathbf{J}_{pn}$  is "in the receiver",  $\mathbf{J}_{p1}$  is "in the sky".)

Some  $\mathbf{J}$ 's do not vary with  $l, m$  -- call them *uv-plane effects*.  
e.g. receiver gain, leakage.

Some  $\mathbf{J}$ 's do vary with  $l, m$  -- call them *image-plane effects*.  
e.g.  $\mathbf{K}$ , beam gain, ionosphere

Let's rewrite the  $\mathbf{J}_p$  product as:

$$\mathbf{J}_p = \underbrace{\mathbf{J}_{pn} \cdots \mathbf{J}_{pk+1}}_{\mathbf{G}_p} \mathbf{K}_p \underbrace{\mathbf{J}_{pk-1} \cdots \mathbf{J}_{p1}}_{\mathbf{E}_p(l, m)}$$

uv-plane only                      uv- & image-plane

Or in other words,  $\mathbf{J}_p(l, m) = \mathbf{G}_p \mathbf{K}_p(l, m) \mathbf{E}_p(l, m)$

(and depending on our particular M.E.,  $\mathbf{G}$  or  $\mathbf{E}$  may be  $\equiv \mathbf{1}$ )

# And Back To The ME....

$$\mathbf{V}_{pq} = \iint_{lm} \mathbf{J}_p(l, m) \frac{\mathbf{B}(l, m)}{n(l, m)} \mathbf{J}_q^\dagger(l, m) dl dm$$

then becomes:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{K}_p \mathbf{E}_p \frac{\mathbf{B}}{n} \mathbf{E}_q^\dagger \mathbf{K}_q^\dagger dl dm \right) \mathbf{G}_q^\dagger$$

(with everything under the  $\iint$  being a function of  $l, m$ )

# The Fourier Transform

and now expanding the  $\mathbf{K}$  terms:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \left( \mathbf{E}_p \frac{\mathbf{B}}{n} \mathbf{E}_q^\dagger \right) e^{-2\pi i(u_{pq}l + v_{pq}m + w_{pq}(n-1))} dl dm \right) \mathbf{G}_q^\dagger$$

for narrow fields  $n \rightarrow 1$  (and for coplanar arrays  $w=0$ ), so:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \left( \underbrace{\mathbf{E}_p \mathbf{B} \mathbf{E}_q^\dagger}_{\text{"apparent sky"}} \right) \underbrace{e^{-2\pi i(u_{pq}l + v_{pq}m)}}_{\text{F.T. kernel}} dl dm \right) \mathbf{G}_q^\dagger$$

- The integral then becomes a 2D Fourier transform of the “apparent sky”.

# The Fourier Transform 2

- This is essentially the van Cittert-Zernike theorem
  - ...as a simple consequence of the M.E. and our **K** Jones term.
- Note that the original M.E. (Hamaker-Bregman-Sault, “ME Paper I”) was formulated purely in terms of cohaerency:

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{X}(u_{pq}, v_{pq}) \mathbf{G}_q^\dagger$$

where the cohaerency  $\mathbf{X}(u, v)$  is the F.T. of the sky  $\mathbf{B}(l, m)$

- ...so here we extend the M.E. into the image plane.



# Apparent Skies & Apparent Coherencies

We now have:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{B}_{pq} e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^\dagger = \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^\dagger,$$

$$\text{where } \mathbf{X}_{pq} = \mathcal{F}(\mathbf{B}_{pq}) = \mathcal{F}(\mathbf{E}_p \mathbf{B} \mathbf{E}_q^\dagger)$$

- In other words, each antenna pair p-q measures an *apparent coherency distribution*  $\mathbf{X}_{pq}(u,v)$  that corresponds to a 2D Fourier Transform of its own *apparent projected sky*  $\mathbf{B}_{pq}$ .
- ...at a single point in time!

# Time Is Not On Our Side

- Coherencies are sampled along a “uv track” over some period of time:

$$\mathbf{V}_{pq}(t) = \mathbf{G}_p(t) \mathbf{X}_{pq}(t, u(t), v(t)) \mathbf{G}_q^\dagger(t)$$

- The true sky  $\mathbf{B}$  is probably constant(?) in time
- Image-plane effects (beam shapes, ionosphere) will vary:
  - ...both in time
  - ...and across antennas
- All this is especially relevant with low-frequency and/or wide-field observations.

# The “Classic” Assumptions

The full-sky ME:  $\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^\dagger$ ,  
 where  $\mathbf{X}_{pq} = \mathcal{F}(\mathbf{B}_{pq})$ ,  $\mathbf{B}_{pq} = \mathbf{E}_p \mathbf{B} \mathbf{E}_q^\dagger$

If we assume that  $\mathbf{B}(t) \equiv \mathbf{B}$ , and  $\mathbf{E}_p(t) \equiv \mathbf{E}_p \equiv \mathbf{E}$ ,  
 then all baselines will see the same, constant apparent sky:

$$\mathbf{B}_{pq}(t) = \mathbf{E} \mathbf{B} \mathbf{E}^\dagger \equiv \tilde{\mathbf{B}}$$

and the array will sample one apparent coherency plane:

$$\mathbf{X}_{pq}(t, u, v) \equiv \mathbf{X}(u, v)$$

- Only under these assumptions can we consider a single F.T. of the sky as being an accurate representation of what an interferometer sees.

# Conclusions

- Under the “classic” assumptions, the visibilities measured by an array correspond to ONE coherency distribution  $\mathbf{X}$  that is in an F.T. relationship with ONE apparent sky.
- In the presence of non-trivial image plane effects – such as the ionosphere -- each interferometer  $p$ - $q$  measures its “own” coherency  $\mathbf{X}_{pq}(t)$ , corresponding to its “own” apparent sky  $\mathbf{B}_{pq}(t)$  -- variable in time!

# Ionosphere In The M.E.

The ionosphere introduces two effects:

- \* phase delay (**Z**-Jones)
- \* Faraday rotation (**F**-Jones)

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{Z}_p \mathbf{F}_p \mathbf{B} \mathbf{F}_q^t \mathbf{Z}_q^t e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^t$$

$$\mathbf{Z}(l, m) = e^{-i\zeta(l, m)}$$

$$\mathbf{F} = \text{Rot } \phi = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix}$$

$$\phi = \text{RM} \cdot \lambda^2$$

# Some Ballpark Numbers

$$\mathbf{Z} = e^{-i\zeta}$$

Ionospheric phase delay:  $\zeta \approx 25 \cdot \lambda \cdot \text{TEC}$

e.g. at  $\lambda = 1\text{m}$ ,  $\Delta \text{TEC} = 0.1$  corresponds to  $\zeta = 2.5$  rad

$$\mathbf{F} = \text{Rot}(\text{RM} \cdot \lambda^2)$$

Ionospheric rotation measure is proportional to TEC, but also depends on the Earth's magnetic field.

Typical RM values are  $1 - 10$  rad/m<sup>2</sup>

(and the differential RM is a lot smaller, so we ignore it...)

# The Famous Four Regimes: the trivial...

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{Z}_p \mathbf{B} \mathbf{Z}_q^\dagger e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^\dagger$$

**Small array, narrow field:**  $\mathbf{Z}_p(l, m) \simeq \mathbf{Z}$  for all  $l, m, p$ :

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{B} e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^\dagger \quad (\text{since } \mathbf{Z} \mathbf{Z}^\dagger = \mathbf{1})$$

$\Rightarrow$  we don't see any effect.

**Small array, wide field:**  $\mathbf{Z}_p(l, m) \simeq \mathbf{Z}(l, m)$  for all  $p$ :

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{B} e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^\dagger \quad (\text{since } \mathbf{Z} \mathbf{Z}^\dagger = \mathbf{1})$$

$\Rightarrow$  we don't see any effect.

# The Famous Four Regimes: the simple, and the nasty

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \iint_{lm} \mathbf{Z}_p \mathbf{B} \mathbf{Z}_q^\dagger e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{G}_q^\dagger$$

**Large array, narrow field:**  $\mathbf{Z}_p(l, m) \simeq \mathbf{Z}_p$  for all  $l, m$

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{Z}_p \left( \iint_{lm} \mathbf{B} e^{-2\pi i(u_{pq}l + v_{pq}m)} dl dm \right) \mathbf{Z}_q^\dagger \mathbf{G}_q^\dagger$$

$\Rightarrow$  we absorb it in  $\mathbf{G}$  phases during calibration.

**Large array, wide field:** different  $\mathbf{Z}_p(l, m)$

$\Rightarrow$  this is the general ME above.



# Exercise: Ionospheric Sim

- Need a low-frequency MS:

```
$ cd ~/GLOW2010/MS
$ makems WSRT_lf_makems.cfg
$ mv WSRT_lf.MS_p0 WSRT_lf.MS
$ cd ../Sim
```
- LSM: 5x5 grid at 10'
- Sine-TIDs ~ 100-200km, amplitude 0.01-0.02
- Make per-channel images
  - And then repeat with “correct for center phase”

# Calibration (Can Be Fun)



Personnel stack 4,000 boxes of TNT, at 50 pounds a box, to make a stack of 100 tons of TNT for the May 7, 1945, calibration test.

# Classic (Scalar) Selfcal

- Start with a sky model (point source at center, etc.)
- Solve for complex gains by fitting observed data:

$$V_{xx,pq} = (I + Q) e^{-i\phi_{pq}} g_{x,p} g_{x,q}^* \rightarrow d_{xx,pq}$$

$$V_{yy,pq} = (I - Q) e^{-i\phi_{pq}} g_{y,p} g_{y,q}^* \rightarrow d_{yy,pq}$$

- Iteratively refine sky model, rinse, repeat

# M.E.-based (Matrix) Selfcal

- Start with a sky model (point source at center, etc.)
- Solve for **G** Jones elements by fitting observed data:

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger \mathbf{G}_q^\dagger \rightarrow \mathbf{D}_{pq}$$

- Iteratively refine sky model, rinse, repeat
- Arbitrary Jones terms may be added (and solved for!)

# The M.E. Calibration Loop

Assume this m.e.:  $\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger \mathbf{G}_q^\dagger$

1. Start with a model for the source,  $\mathbf{B}$
2. Derive "model" coherencies:  $\mathbf{X}_{pq} = \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger$
3. Predict "corrupted" model:  $\mathbf{X}'_{pq} = \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^\dagger$
4. Find  $\mathbf{G}_p$ s by fitting  $\mathbf{X}'_{pq}$  to observed  $\mathbf{V}_{pq}$
5. Compute "corrected" visibilities:  $\mathbf{V}'_{pq} = \mathbf{G}_p^{-1} \mathbf{V}_{pq} (\mathbf{G}_q^{-1})^\dagger$   
(note that  $(\mathbf{G}^\dagger)^{-1} = (\mathbf{G}^{-1})^\dagger$ )

The "corrected" visibilities should then correspond to the "true", uncorrupted source.

# Or In Broad Terms...

1. Predict corrupted visibilities
  - we already do this with Siamese
2. Fit to observed visibilities
  - solving for parameters of the sky and/or the instrument
3. (Optional: subtract bright sources)
4. Correct
5. Rinse & repeat
  - aka the “major loop”: source extraction, updating sky model, etc.

# Calibration Of An MS?

- A model tree computes corrupted visibilities  $\mathbf{X}'_{pq}(t, \nu)$ 
  - we've used Siamese for this
- MS DATA column contains observed data  $\mathbf{V}_{pq}(t, \nu)$
- We can take the difference and form up a  $\chi^2$  sum...
- ...and try to minimize it w.r.t. the solvable parameters.
- Which is the same as fitting the model to the data, in a least-squares sense.
- We can thus solve for any (reasonable) subset of parameters of a measurement equation.



# Calico

## (Calibration Components)





# Calico

- Siamese's calibration cousin.
- Same principle: build a measurement equation from a sky model + plug-in Jones modules.
  - Siamese modules are fully compatible
- ...but, modules can also specify their solvable parameters.
- Calico provides a standard solving interface to these.
- **cd Workshop2008/Day2**

# Exercise: Calibration 0

- Load **example-cal.py**
  - Use 2x2 data, diagonal terms only
  - **Enable calibrate & correct**
  - **Use sky model with 1 source at center**
  - **Enable G Jones (FullReallmag)**
- Open bookmarks for inspectors
- Solve for G diagonal terms
  - Subtiling of 1 in time
  - Tile size 20
- Make an image of the corrected data
  - Make a cleaned image, save to purrlog
  - Go back and do corrected residuals....

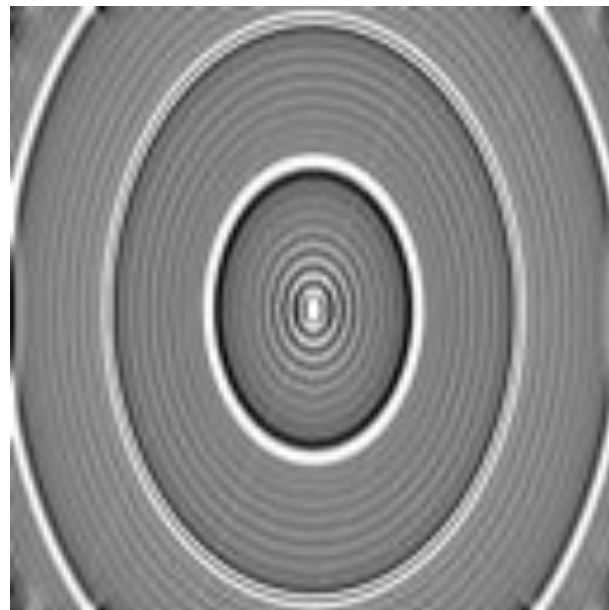
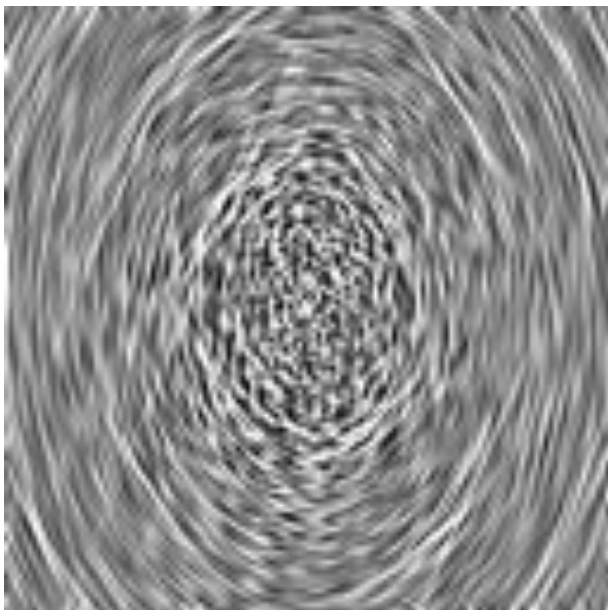
# Calibration is...

- Calibration is **HARD**
  - Lots of details that you need to get right
  - One little mistake, and everything goes south
  - (North if you're in Australia?...)
- Calibration is **EASY**
  - One little mistake, and everything goes south

# M.E. Calibration Terminology

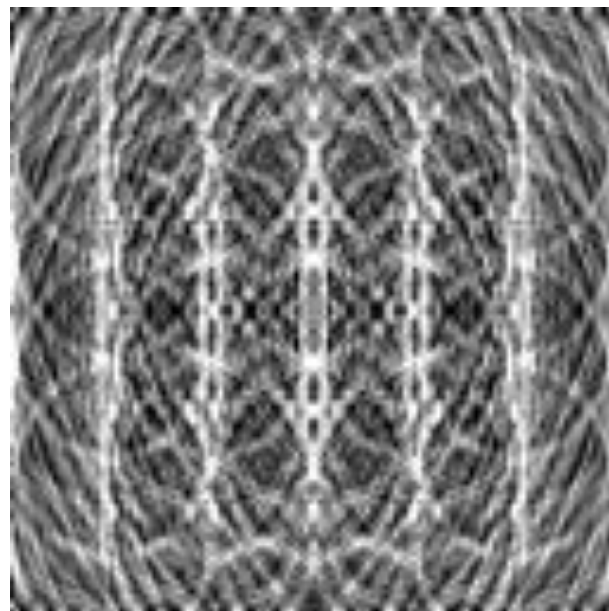
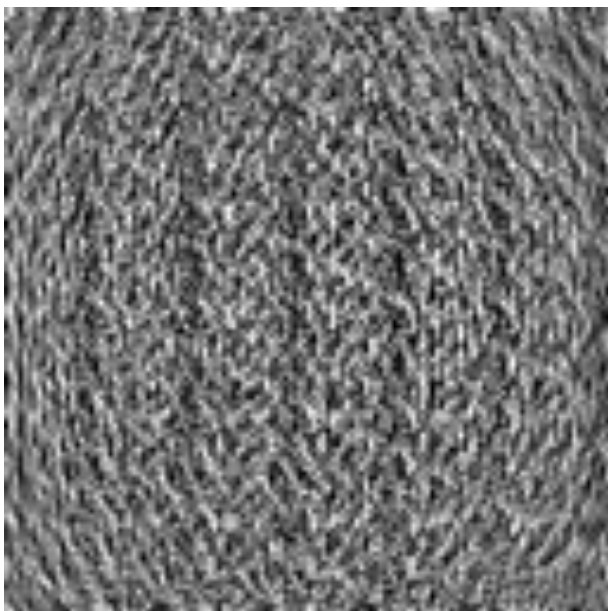
- $\mathbf{D}_{pq}$  : observed visibilities ('data')
- $\mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger$  : sky model (or  $\sum \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)\dagger}$ )
- $\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger \mathbf{G}_q^\dagger$  : corrupted model ('predict')
- $\mathbf{D}_{pq} - \mathbf{V}_{pq} \rightarrow \min$  : calibration
- $\mathbf{D}_{pq} - \mathbf{V}_{pq}$  : corrupted residuals
- $\mathbf{G}_p^{-1} \mathbf{D}_{pq} \mathbf{G}_q^{-1\dagger}$  : corrected data
- $\mathbf{G}_p^{-1} (\mathbf{D}_{pq} - \mathbf{V}_{pq}) \mathbf{G}_q^{-1\dagger}$  : corrected residuals

$\mathbf{D}_{pq}$  (data)



$\mathbf{G}_p^{-1} \mathbf{D}_{pq} \mathbf{G}_q^{-1t}$   
(corrected data)

Jy  
level



mJy  
level

$\mathbf{D}_{pq} - \mathbf{V}_{pq}$   
(corrupted residuals)

$\mathbf{G}_p^{-1} (\mathbf{D}_{pq} - \mathbf{V}_{pq}) \mathbf{G}_q^{-1t}$   
(corrected residuals)

# Major Loop Of Calibration

- Make initial sky model
  - Calibrate, subtract sky model, and generate *corrected residuals*
  - Use corrected residuals (deconvolution, etc.) to improve sky model
  - Repeat until satisfied
- 
- What is satisfaction?

# Calibration (Noordam Definition)

# Real-Life Residuals

- Real-life residuals are always contaminated by imperfect subtraction of sources (due to calibration error)
- Causes of error:
  - Contamination from sources not included in sky model
  - Imperfect instrument models
  - RFI, insufficient flagging
- Can even have ghosts!





# Optional Exercise: Selfcal Ghosts

- For the brave only!
- Simulate 1 Jy source at center →MODEL\_DATA
- Add 1 mJy “contaminator” source,  $10^\circ$  away  
→DATA
- Calibrate, using sky model of 1 Jy source at center, write residuals →CORRECTED\_DATA
  - Make image ( $30^\circ$  across)
  - Dominated by contaminator
- Generate corrected residuals from MODEL\_DATA to CORRECTED\_DATA
  - This is now contaminator-free
  - Observe the ghosts



# Polarisation Selfcal

# Classical Equation For Polarization Selfcal

$$R_j R_k^* = G_{R_j} G_{R_k}^* \left[ E_{R_j} E_{R_k}^* e^{-i(\phi_j - \phi_k)} + D_{R_k}^* E_{R_j} E_{L_k}^* e^{-i(\phi_j + \phi_k)} + \right. \\ \left. D_{R_j} E_{L_j} E_{R_k}^* e^{i(\phi_j + \phi_k)} + D_{R_j} D_{R_k}^* E_{L_j} E_{L_k}^* e^{i(\phi_j - \phi_k)} \right]$$

$$R_j L_k^* = G_{R_j} G_{L_k}^* \left[ E_{R_j} E_{L_k}^* e^{-i(\phi_j + \phi_k)} + D_{L_k}^* E_{R_j} E_{R_k}^* e^{-i(\phi_j - \phi_k)} + \right. \\ \left. D_{R_j} E_{L_j} E_{L_k}^* e^{i(\phi_j - \phi_k)} + D_{R_j} D_{L_k}^* E_{L_j} E_{R_k}^* e^{i(\phi_j + \phi_k)} \right].$$

(With thanks to Huib Jan van Langevelde)

# The Measurement Equation For Polarization Selfcal

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{K}_p \mathbf{B} \mathbf{K}_q^\dagger \mathbf{G}_q^\dagger$$

$$\mathbf{G}_p = \begin{pmatrix} g_{11,p} & g_{12,p} \\ g_{21,p} & g_{22,p} \end{pmatrix}$$

- The only difference w.r.t. the previous m.e. is that the  $\mathbf{G}$  matrix has off-diagonal terms.
- Polarization not so scary after all!

# Thinking About Polarization 1

- Diagonal Jones matrices will intermix  $I \leftrightarrow Q$  and  $U \leftrightarrow V$ :

$$\begin{pmatrix} a_x & 0 \\ 0 & a_y \end{pmatrix} \begin{pmatrix} I+Q & U+iV \\ U-iV & I-Q \end{pmatrix} \begin{pmatrix} b_x & 0 \\ 0 & b_y \end{pmatrix} = \begin{pmatrix} (I+Q)a_x b_x & (U+iV)a_x b_y \\ (U-iV)a_y b_x & (I-Q)a_y b_y \end{pmatrix}$$

- e.g.  $a_x \neq a_y$  causes instrumental polarization:  $I \rightarrow Q$
- e.g. phase difference transfers  $U \rightarrow V$
- Off-diagonal terms will intermix  $IQUV$
- Pure real terms won't touch  $V$ 
  - e.g. a rotation matrix mixes  $QU$ , but not  $V$
- Thinking in terms of matrices will help you understand everything about polarization!

# Faraday Rotation

$$\begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} I+Q & U+iV \\ U-iV & I-Q \end{pmatrix} \begin{pmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix}$$

$$\phi = RM\lambda^2$$

- F.R. rotates the angle of polarization
  - i.e. intermixes  $Q$  and  $U$ .
- Interstellar medium: “intrinsic” F.R.
  - same for all antennas ( $\phi = \psi$ ), varies slowly/not at all
- Ionospheric F.R.
  - effectively “intrinsic” for short wavelengths and small arrays
  - for LOFAR long baselines: differential F.R.

# Exercise: Calibrating 3C286

- 3C286 is a standard WSRT calibrator
  - has significant  $QU$  ( $\sim 10\%$ )
- We'll try to calibrate a short WSRT 1.4GHz observation.
  - $\sim 15$  mins at 10 sec. integration

```
$ cd ~/GLOW2010/MS
$ tar zxvf 3C286.MS.tgz
$ cd ../Cal
```

# Thinking About Polarization 2

- WSRT is an equatorial mount, so (to first order) it only has diagonal Jones terms
  - for off-center sources, we do get significant instrumental  $Q$  (on the diagonal)
  - this occurs ***after*** any F.R.
  - in fact, there *are* small off-diagonal Jones terms: “polarization leakage”
  - but *if the source is intrinsically unpolarized*, we can ignore the off-diagonal correlations
- With an alt-az mount, the sky rotates, so you'll see instrumental  $Q$  and  $U$ .



# What Causes Polarization Leakage?

- Simple errors in dipole geometry:

$$\mathbf{J} = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \psi & \cos \psi \end{pmatrix}$$

- these vary in time (slowly) due to the telescope mechanically deforming as it elevates
- Electromagnetic cross-talk:

$$\mathbf{J} = \begin{pmatrix} g_{xx} & g_{xy} \\ g_{yx} & g_{yy} \end{pmatrix} \quad \begin{matrix} g_{xx}, g_{yy} \rightarrow 1 \\ g_{xy}, g_{yx} \rightarrow 0 \end{matrix}$$

- Both cases correspond to small values off the main diagonal of  $\mathbf{J}$

# Obfuscating The M.E., Part 2: Death by a million Jones matrices

- The M.E. is, at core, very simple
- The literature (M.E. papers, AIPS++ Note 185, etc.) is full of interminable M.E.s of the form:

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{D}_p \mathbf{B}_p \mathbf{C}_p \mathbf{E}_p \mathbf{Z}_p \mathbf{F}_p \mathbf{K}_p \mathbf{B}_q \mathbf{K}_q^\dagger \mathbf{F}_q^\dagger \mathbf{Z}_q^\dagger \mathbf{E}_q^\dagger \mathbf{C}_q^\dagger \mathbf{B}_q^\dagger \mathbf{D}_q^\dagger \mathbf{G}_q^\dagger$$

(or, to add insult to injury, the same in Evil Mueller form)

- Product of late-90s AIPS++ enthusiasm, driven by the discovery of the ME:
  - “We’ll just quickly catalog every Jones matrix there is!”
  - “All we need to do is implement these Jones matrices now, and we’re sitting pretty forever!”
- This is not “**The** Measurement Equation”, so please don’t use it to scare impressionable students

# The Simple View: Build Your Own M.E.s

- Start with your basic equation: 
$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{X}_{pq} \mathbf{J}_q^\dagger$$
and tailor to taste.
- For **simulation**: what physics are we trying to simulate?
  - full-on simulations: insert as many Jones terms as you understand
  - specific simulations: one or two Jones terms can be is sufficient
- For **calibration**:
  - what can we measure? We do, after all, only measure the cumulative effect of all Jones terms.
  - insert the Jones terms you know apriori (beam, parallactic rotation, etc.)
  - Insert generic solvable matrices for the rest

# Phenomenological M.E.s

- A *phenomenological* M.E. expresses the effect of a corruption without regard to the underlying physics
- For example, we can calibrate WSRT using the following

M.E.:

$$\mathbf{V}_{pq} = \mathbf{B}_p \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^\dagger \mathbf{B}_q^\dagger$$

- $\mathbf{G}$  (diagonal): short-term freq-independent variations
- $\mathbf{B}$  (full 4-element): long-term freq-dependent component (bandpass and polarization leakage)
- Unlike simulations, neither  $\mathbf{G}$  nor  $\mathbf{B}$  is all that physical – each combines several physical effects
  - Only distinguishable by their time/freq behaviour
  - We don't care, as long as there's enough degrees of freedom in our model to fit the physics
  - Can't fit more DoF anyway, they're all rolled up in the measurement

# Correcting For Multiple Jones Terms

Given an m.e. of the form:

$$\mathbf{V}_{pq} = \mathbf{J}_{pn} \cdots \mathbf{J}_{p1} \mathbf{X}_{pq} \mathbf{J}_{q1}^\dagger \cdots \mathbf{J}_{qn}^\dagger$$

the corrections need to be applied in reverse order:

$$\begin{aligned} \mathbf{V}'_{pq} &= \mathbf{J}_{p1}^{-1} \cdots \mathbf{J}_{pn}^{-1} \mathbf{V}_{pq} (\mathbf{J}_{qn}^{-1})^\dagger \cdots (\mathbf{J}_{q1}^{-1})^\dagger = \\ &= \mathbf{J}_{p1}^{-1} \cdots \underbrace{\mathbf{J}_{pn}^{-1} \mathbf{J}_{pn}}_{=1} \cdots \mathbf{J}_{p1} \mathbf{X}_{pq} \mathbf{J}_{q1}^\dagger \cdots \underbrace{\mathbf{J}_{qn}^\dagger (\mathbf{J}_{qn}^{-1})^\dagger}_{=1} \cdots (\mathbf{J}_{q1}^{-1})^\dagger = \\ &= \mathbf{X}_{pq} \end{aligned}$$

...and all matrix (non-)commutation rules apply.

# Calibrating 3C286

- Load **calico-286.py** and look at Options
- This is the M.E. we're going to use:

$$\mathbf{V}_{pq} = \mathbf{B}_p \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^\dagger \mathbf{B}_q^\dagger$$

- **G** (diagonal): short-term phase and gain variations
- **B** (full 4-element): bandpass and pol. leakage

# The Local Sky Model

- Instead of pretty grid, we obviously want to calibrate on a realistic sky
- The LSM module will read in a file and create a sky model based on it.
  - various formats supported
- lsm286.txt is our model for this calibrator  
**cat lsm286.txt**

# MS/data selection options

- This menu (in TDL Exec) determines what subset of the data we solve for
  - Input column is always DATA (for observed data)
  - Select channels 8 through 55, step 1
  - No Hanning tapering
  - “Data description ID” determines the spectral window. Pick one...



# Step 1: Solving For G

- **G** represents receiver + troposphere/ionosphere gain/phases
  - diagonal (i.e. no cross-terms)
  - same across all channels
  - would like a separate solution per timeslot.
- Build the tree, and open up the “Calibrate G diagonal terms” option.

# Tiles And Solution Intervals

- We go through the MS in chunks of time called **tiles** (a full MS wouldn't fit in memory...)
  - a tile contains  $N$  consecutive timeslots, and all [selected] channels of a spectral window
- By default, each parameter has one solution per tile (constant or polynomial in time/freq)
  - but you can use a **smaller** solution interval via the “solution subinterval” option. This cannot be bigger than the tile itself.
- Bigger tiles are (to a point) faster, but too big can lead to poor convergence.

# Step 1: Solving For $G$

- Set “solution subinterval (time)” to 1
- Set tile size to e.g. 20
- Load the “inspector:G” and “inspect corrected residuals” bookmarks
- Run “Calibrate G diagonal terms”
- Watch the  $\chi^2$  display at the bottom of the browser window.
- Make a residual IQUV image and save it to the purrlog

# MEP tables

- Parameter solutions are stored in MEP (ME Parameter) tables.
  - these are called “\*.mep” or “\*.fmep”, and are generally kept within the MS directory
- Once a solution is stored in the table, it is reused in all subsequent runs
  - so we can go on and solve for **B**, while including our **G** estimate in the predict
- To clear out solutions and start anew:  
**rm -fr xxxxx.MS/\*.mep**

## Step 2: Solving For $B$ (diagonal terms)

- Open up “Calibrate for B diagonal terms”
- We want separate solutions per each channel, for all timeslots.
  - set “solution subinterval (freq)” to 1
  - set tile size to 100
- Load the “inspect corrected data” bookmarks, and some “B diagonal terms”
- Run “Calibrate B diagonal terms”
- Observe results... check also the  $XY/YX$  residuals
- Make a residual IQUV image, save it to the purrlog

## Step 3: Solving For $B$ (off-diagonal terms)

- Open up “Calibrate for B off-diagonal terms”
- Select a simultaneous solution for B diag.
- Run “Calibrate B off-diagonal terms”
- Observe results... check also the XY/YX residuals
- Make a residual IQUV image and save it to the purrlog

# Exercise: A Wrong LSM

- We'll start with the wrong sky model: lsm286-1.lsm.html (check it with Tigger)
- Clear out previous calibrations:  
`rm -fr ../MS/3C286.MS/*mep`
- Repeat solutions for G-diag, B-offdiag+B-diag
  - May need to repeat G-diag – why?
  - Make intermediate images, and compare to those obtained during the previous exercise
- Think about what we should see at the end. Should we expect to be able to calibrate?

# Limitations Of Selfcal a.k.a. *self-alignment* (Hamaker)

Classic selfcal can't fix the brightness scale:

$$v_{pq} = g_p g_q^* b e^{-i\phi} = (g_p y^{-1}) (g_q y^{-1})^* b y^2 e^{-i\phi} \quad \text{for any real } y.$$

The M.E. analogue is:

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{X}_{pq} \mathbf{J}_q^\dagger = (\mathbf{J}_p \mathbf{Y}^{-1}) (\mathbf{Y} \mathbf{X}_{pq} \mathbf{Y}^\dagger) (\mathbf{J}_q \mathbf{Y}^{-1})^\dagger$$

for any non-singular matrix  $\mathbf{Y}$ .

- So, we can only “know” the sky to within a (non-singular) Jones factor of  $\mathbf{Y}$ .
- What can this do to polarization?
  - anything...



# Thinking About Polarization 3

- Diagonal Jones matrices will intermix  $I \leftrightarrow Q$  and  $U \leftrightarrow V$
- Off-diagonal terms will intermix  $IQUV$

$$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \rightarrow \begin{pmatrix} I+Q & 0 \\ 0 & I-Q \end{pmatrix} \rightarrow \begin{pmatrix} I+Q & U \\ U & I-Q \end{pmatrix} \rightarrow \begin{pmatrix} I+Q & U+iV \\ U-iV & I-Q \end{pmatrix}$$

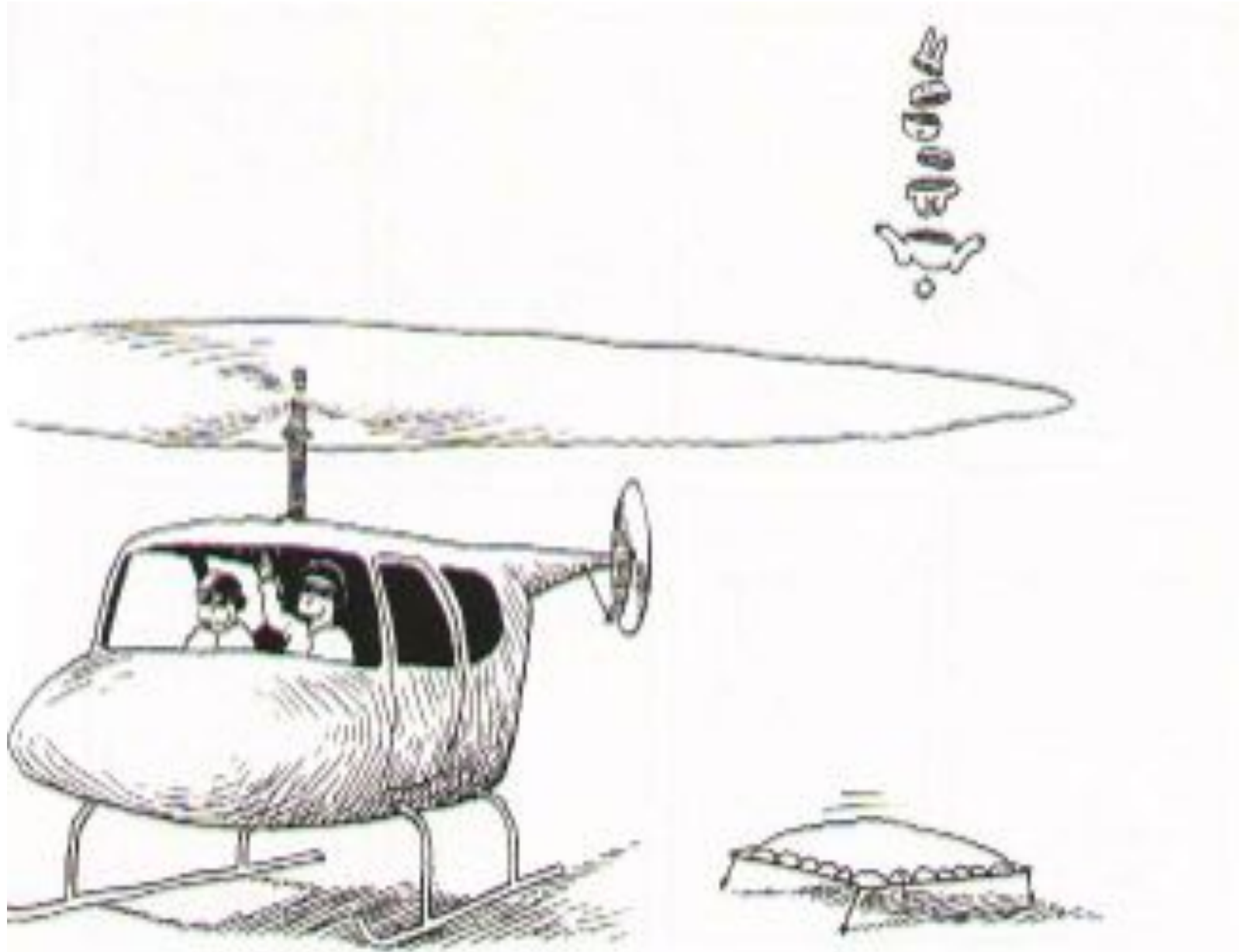
- So, starting with an unpolarized source,  $I$  flux can be moved into  $Q$ ,  $Q$  can be rotated into  $U$ , and  $U$  can be phase-shifted into  $V$ .



# The Evil Leprechaun Principle

- If evil leprechauns were to come and tamper with your telescope in the middle of the night, would you notice?
  - without knowledge of the sky, that is.
- ...not if they changed them all by the same factor of  $Y$ .
  - e.g.: rotate all dipoles by the same angle
  - e.g.: change gain of all  $X$  dipoles
  - etc.
- Moral: we can only get so far without known calibrators.

# Direction-Dependent Effects



# Calibrating For Dipole Projection?

- The ME we are using is:

$$\mathbf{V}_{pq} = \mathbf{G}_p \left( \sum_s \mathbf{L}_p^{(s)} \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)\dagger} \mathbf{L}_q^{(s)\dagger} \right) \mathbf{G}_q^\dagger$$

- For calibration, we can use the same ME and solve for  $\mathbf{G}$  Jones again
- No need to solve for  $\mathbf{L}$  Jones since we know it analytically
  - we simply incorporate it into the ME at the predict stage
- But can we really correct for it?



# Limitations Of Correction

Given an m.e. of the form:

$$\mathbf{V}_{pq} = \mathbf{J}_{pn} \cdots \mathbf{J}_{p1} \mathbf{X}_{pq} \mathbf{J}_{q1}^\dagger \cdots \mathbf{J}_{qn}^\dagger$$

the corrections need to be applied in reverse order:

$$\begin{aligned} \mathbf{V}'_{pq} &= \mathbf{J}_{p1}^{-1} \cdots \mathbf{J}_{pn}^{-1} \mathbf{V}_{pq} (\mathbf{J}_{qn}^{-1})^\dagger \cdots (\mathbf{J}_{q1}^{-1})^\dagger = \\ &= \mathbf{J}_{p1}^{-1} \cdots \underbrace{\mathbf{J}_{pn}^{-1} \mathbf{J}_{pn}}_{=1} \cdots \mathbf{J}_{p1} \mathbf{X}_{pq} \mathbf{J}_{q1}^\dagger \cdots \underbrace{\mathbf{J}_{qn}^\dagger (\mathbf{J}_{qn}^{-1})^\dagger}_{=1} \cdots (\mathbf{J}_{q1}^{-1})^\dagger = \\ &= \mathbf{X}_{pq} \end{aligned}$$

- Matrix inversion has its pitfalls
- What happens if  $\mathbf{J}$  is singular?
  - and what physics does this correspond to?

# Limitations Of Correction 2

- If a Jones matrix is singular, then we don't have enough information to begin with
  - e.g. if a dipole gain is 0, then we haven't measured the EM field in one direction...
- It is also possible for a Jones term to be ill-conditioned
  - numerical inversion of an ill-conditioned matrix breaks down due to precision limitations
  - e.g.: projection matrix of an aperture array (Tobia Carozzi)

# Problem 2: Correcting For Direction-Dependent Effects

ideal sky is  $\mathbf{S}_{pq} = \sum_s \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)\dagger}$

w/o DD effects

with DD effects

observed data is:

$$\mathbf{D}_{pq} = \mathbf{G}_p \left( \sum_s \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)\dagger} \right) \mathbf{G}_q^\dagger$$

(plus noise)

calibration yields  $\tilde{\mathbf{G}}_p \approx \mathbf{G}_p$ ,

corrected data is:

$$\tilde{\mathbf{G}}_p^{-1} \mathbf{D}_{pq} \tilde{\mathbf{G}}_q^{\dagger-1} \approx \mathbf{S}_{pq}$$

observed data is:

$$\mathbf{D}_{pq} = \mathbf{G}_p \left( \sum_s \mathbf{L}_p^{(s)} \mathbf{K}_p^{(s)} \mathbf{B}^{(s)} \mathbf{K}_q^{(s)\dagger} \mathbf{L}_q^{(s)\dagger} \right) \mathbf{G}_q^\dagger$$

(plus noise)

calibration yields  $\tilde{\mathbf{G}}_p \approx \mathbf{G}_p$ ,

corrected data is:

$$\tilde{\mathbf{G}}_p^{-1} \mathbf{D}_{pq} \tilde{\mathbf{G}}_q^{\dagger-1} \neq \mathbf{S}_{pq}$$

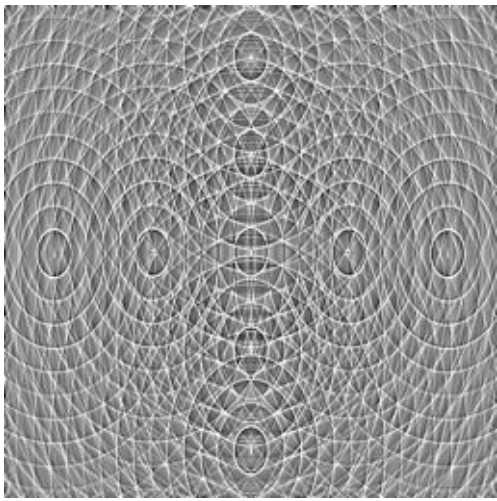
at best we can pick a direction  $s_0$ :

$$\mathbf{L}_p^{(s_0)-1} \tilde{\mathbf{G}}_p^{-1} \mathbf{D}_{pq} \tilde{\mathbf{G}}_q^{\dagger-1} \mathbf{L}_q^{(s_0)\dagger-1}$$



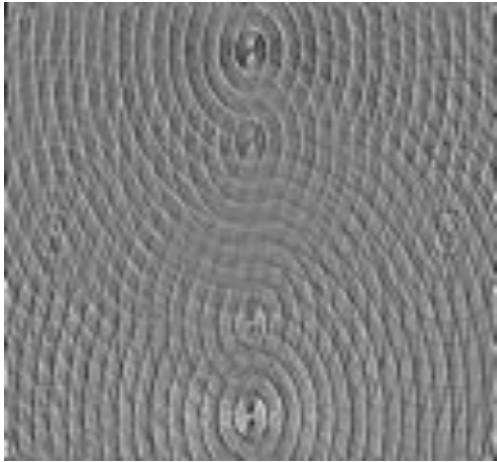
# Exercise: Correcting At Center

- In general, visibility data can only be “corrected” for a single direction on the sky.
- Hence, e.g., facet imaging.
- Bhatnagar (EVLA Memo 100) suggests an approximate method to apply on-the-fly corrections during imaging
- Correction Demo:
  - Repeat L-Jones simulation (to DATA column)
  - Run `example-cal-lj.py`
  - Enable `correct`, disable `calibrate` and `subtract`
  - Apply L Jones correction (for center of field) and make an image



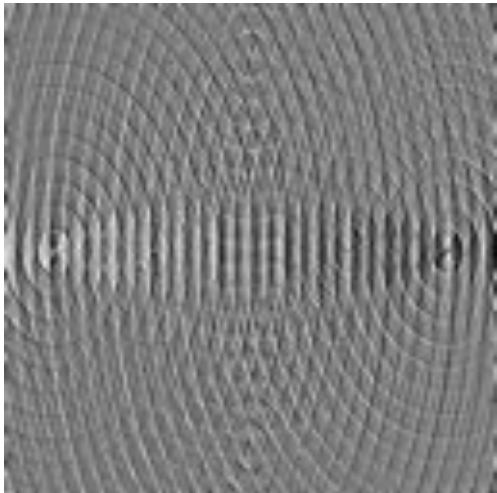
Stokes  $I$  map.

Distortions in source shape no longer visible (though from the math we know they must remain, on a low level.) Peak flux is 1 Jy.



$Q$  and  $U$  maps.

Note how instrumental polarization corrects perfectly at center, but increases towards edge of field.

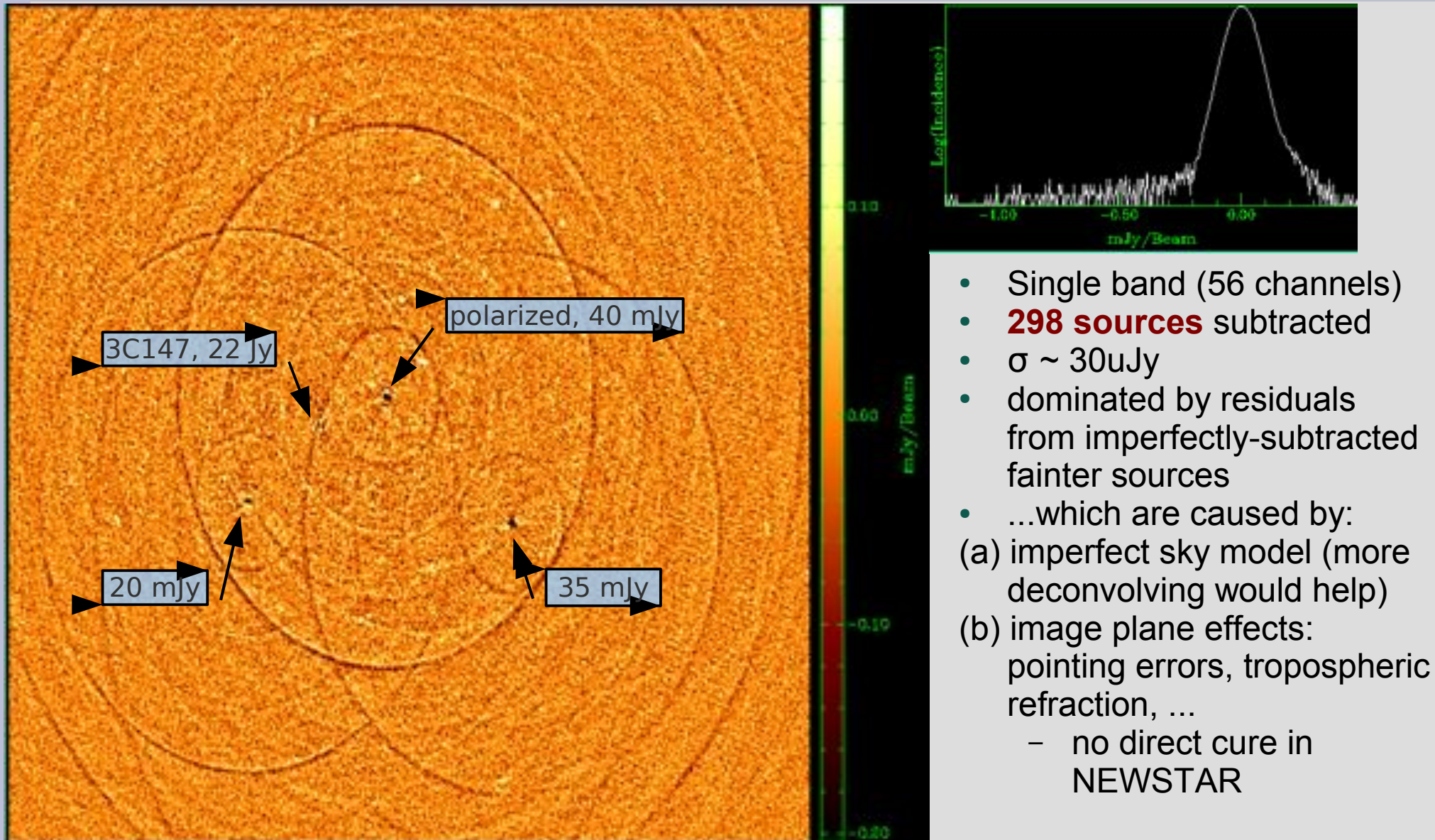


Peak flux is  $\pm 50$  mJy.

# Dealing With DD Effects

- The same issue arises with other DDEs:
  - Ionosphere
  - Beam shapes & pointing errors
- Becoming critical for LOFAR & pother new instruments, and will be even more so for the SKA itself
- Solution: subtract sources bright enough to cause trouble
  - Since we can predict them “perfectly” (within the limits of calibration error)

# Example: WSRT Off-Axis Effects



- Single band (56 channels)
- **298 sources** subtracted
- $\sigma \sim 30 \mu\text{Jy}$
- dominated by residuals from imperfectly-subtracted fainter sources
- ...which are caused by:
  - (a) imperfect sky model (more deconvolving would help)
  - (b) image plane effects: pointing errors, tropospheric refraction, ...
    - no direct cure in NEWSTAR



# Luxury Problems Of Calibration



# More On Phenomenological M.E.s

- Adding a beam to the previous M.E.:

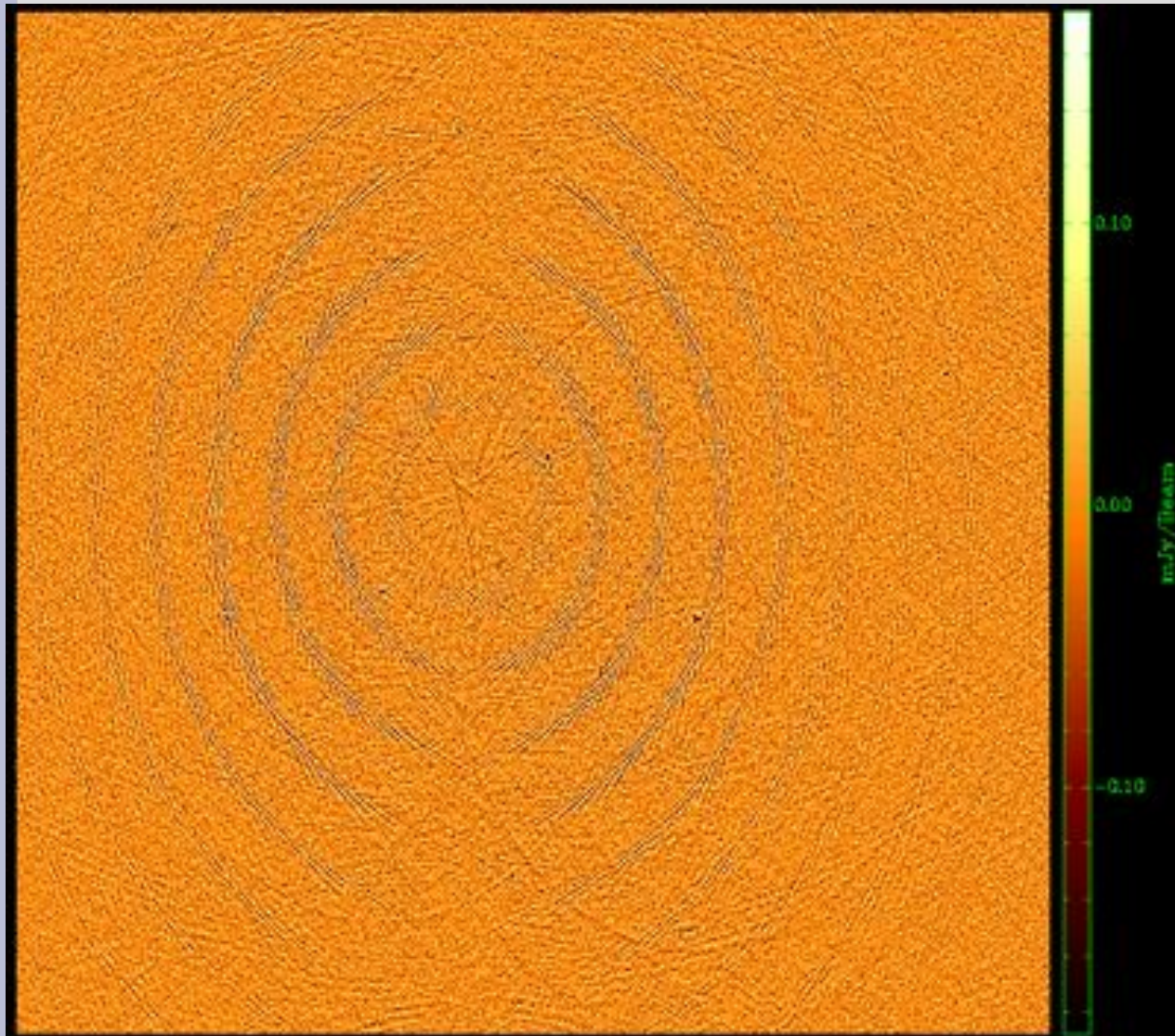
$$\mathbf{V}_{pq} = \overbrace{\mathbf{B}_p}^{\text{bandpass}} \overbrace{\mathbf{G}_p}^{\text{gain}} \underbrace{\left( \sum_s \overbrace{\mathbf{E}_p^{(s)}}^{\text{beam}} \overbrace{\mathbf{X}_{pq}}^{\text{source coherency}} \mathbf{E}_q^{(s)\dagger} \right)}_{\text{sum over sources}} \mathbf{G}_q^\dagger \mathbf{B}_q^\dagger$$

$\mathbf{E}_p^{(s)}$  is an analytic expression,  $E(l, m, \nu) = \cos^3(C\nu\sqrt{l^2 + m^2})$

$\mathbf{G}_p(t)$  is a solvable

$\mathbf{B}_p(\nu)$  is a solvable (with a long-scale time variation)

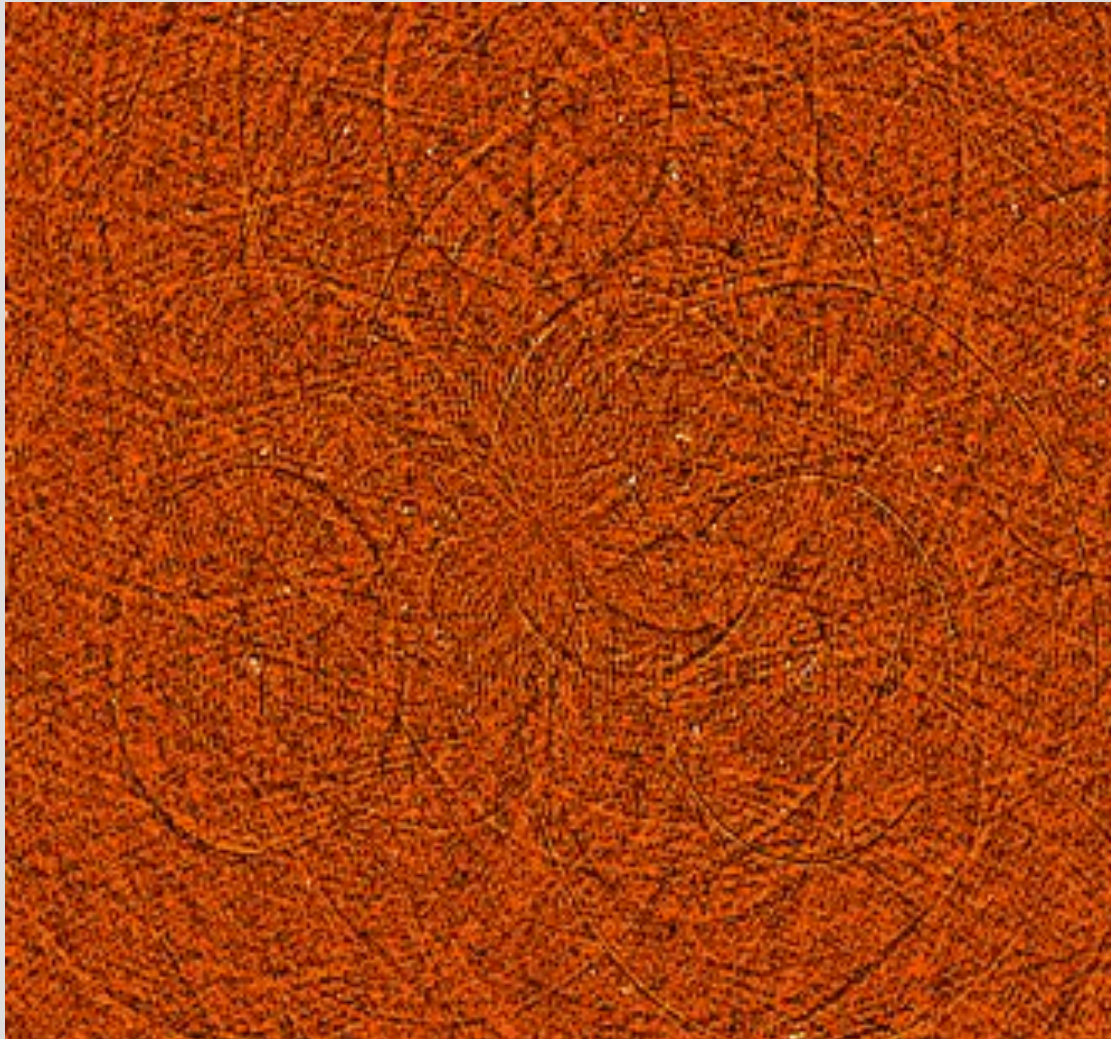
# Bandpass Artifacts



- Residual pattern from 3C147 due to *bandpass instability*.
- We do a separate **B** solution every 30 min.
- Error pattern caused by variations in actual bandpass over the solution interval
  - error  $\sim 1/10,000$
- We can mitigate this by making **B** a 1st-degree polynomial in time
  - error  $\sim 1/200,000$
  - close to noise level but plainly visible
- Further increase polynomial degree?
  - or spline?



# Limits Of Bandpass Stability



- **B** solution every 7.5 minutes
- **G** solution every 30 sec.
- Followed by smoothing of **B** and repeated **G** solution
- We're still left with a DR-limiting error pattern left over from 3C147 itself.
- My tentative conclusion: WSRT bandpass is “jittery” on short timescales.
- ...but you need to get past 100,000:1 DR for this to bite you!
  - A luxury problem



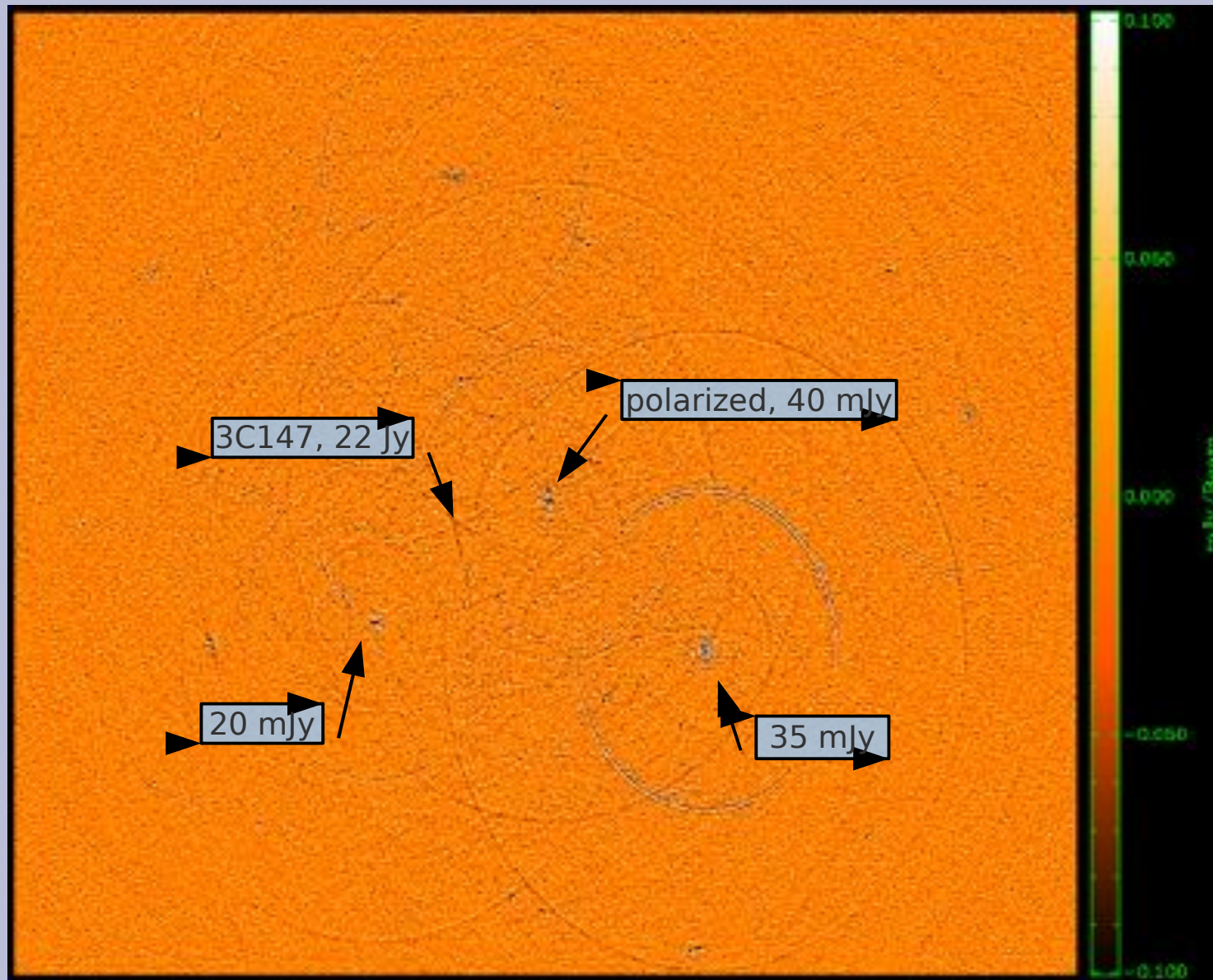
# Dropping The Bandpass

- Do a per-channel selfcal
  - with sufficient S/N, why not?
  - this is what Ger does in NEWSTAR
- In M.E. terms:

$$\mathbf{V}_{pq} = \overbrace{\mathbf{G}_p}^{\text{gain \& bandpass}} \underbrace{\left( \sum_s \overbrace{\mathbf{E}_p^{(s)}}^{\text{beam}} \overbrace{\mathbf{X}_{pq}}^{\text{source coherency}} \mathbf{E}_q^{(s)\dagger} \right)}_{\text{sum over sources}} \mathbf{G}_q^\dagger$$

$\mathbf{G}_p(\nu, t)$  solved separately at each  $\nu, t$  point.

# Seeing The DDE's



- Residual image, 298 sources subtracted
- Per-channel selfcal + closure errors
- Dominant features are residuals from off-axis sources.
- Some of it is due to missing/too much flux in the sky model and can be CLEANed away.
- But not all of it! (and this is what causes artifacts in the final map.)

# Solving For Pointing Errors

- Bhatnagar “Pointing selfcal” approach, in terms of our ME:

$$\mathbf{V}_{pq} = \overbrace{\mathbf{G}_p}^{\text{gain \& bandpass}} \underbrace{\left( \sum_s \overbrace{\mathbf{E}_p^{(s)}}^{\text{beam}} \overbrace{\mathbf{X}_{pq}}^{\text{source coherency}} \mathbf{E}_q^{(s)t} \right)}_{\text{sum over sources}} \mathbf{G}_q^t$$

Instead of using  $\mathbf{E}_p^{(s)} \equiv E(l, m, \nu)$  for all  $p$ ,

offset the beam pattern at each antenna  $p$  by  $\Delta l_p, \Delta m_p$ :

$$\mathbf{E}_p(l, m, \nu) = E(l + \Delta l_p, m + \Delta m_p, \nu)$$

...and solve for the offsets.

# Differential Gains

- Or we can introduce differential gains:

$$\mathbf{V}_{pq} = \overbrace{\mathbf{G}_p}^{\text{gain \& bandpass}} \underbrace{\left( \sum_s \overbrace{\Delta \mathbf{E}_p^{(s)}}^{\text{differential gain}} \overbrace{\mathbf{E}_p^{(s)}}^{\text{beam}} \overbrace{\mathbf{X}_{pq}}^{\text{source coherency}} \mathbf{E}_q^{(s)\dagger} \Delta \mathbf{E}_q^{(s)\dagger} \right)}_{\text{sum over sources}} \mathbf{G}_q^\dagger$$

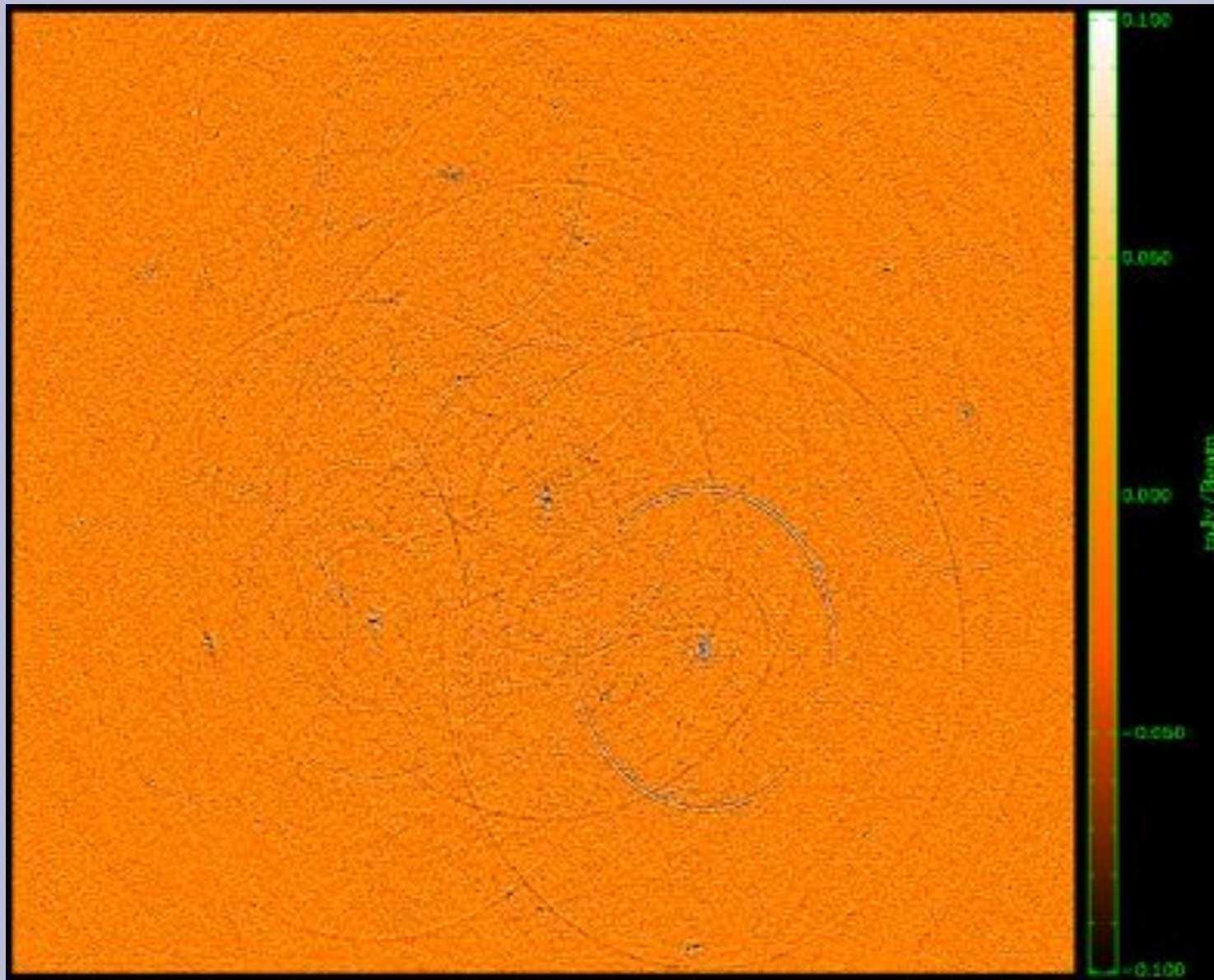
$\Delta \mathbf{E}_p^{(s)}$  is frequency-independent, slowly varying in time.

Solvable for a handful of "troublesome" sources,

and set to unity for the rest.

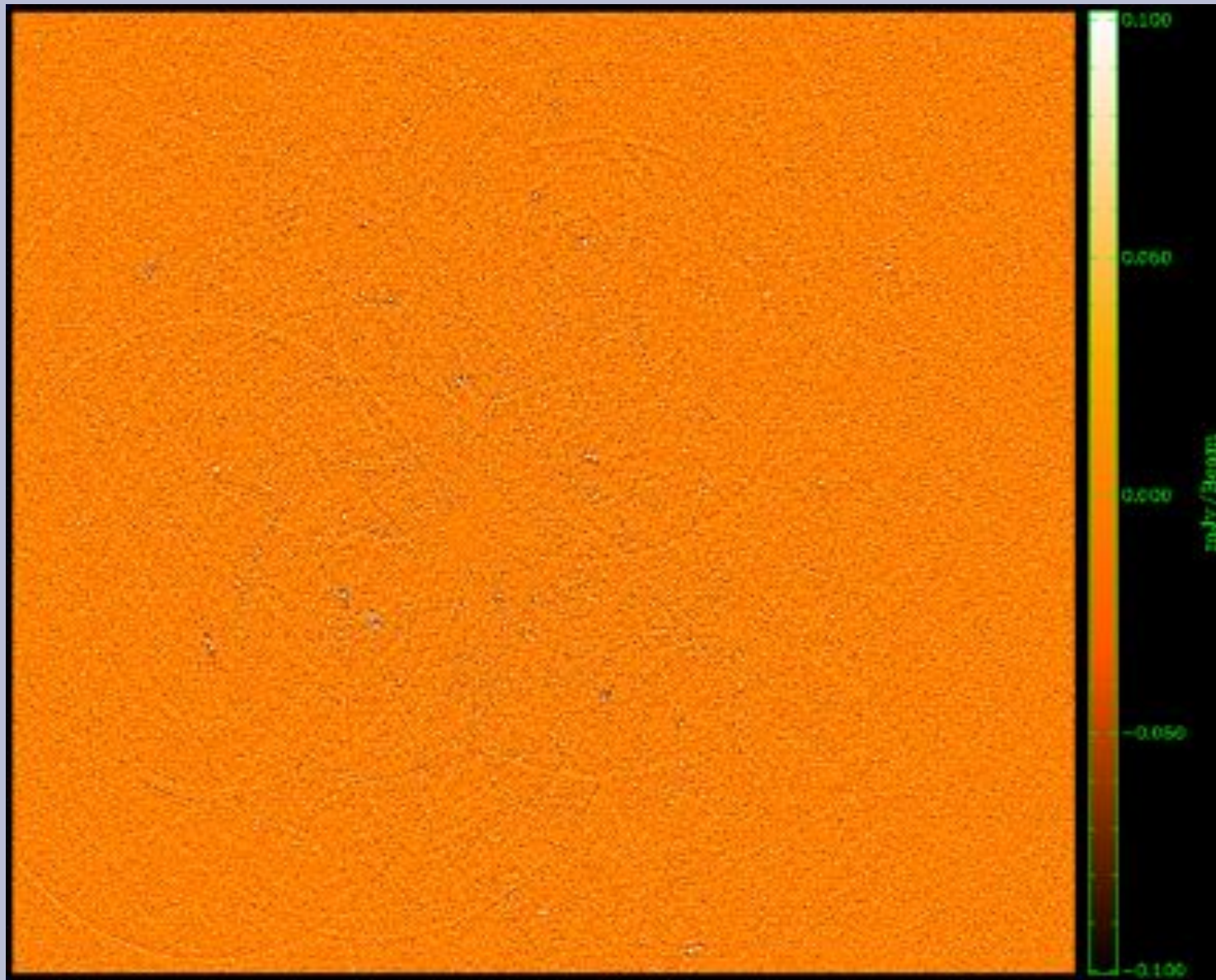


# Flyswatter I



- The “before” image.

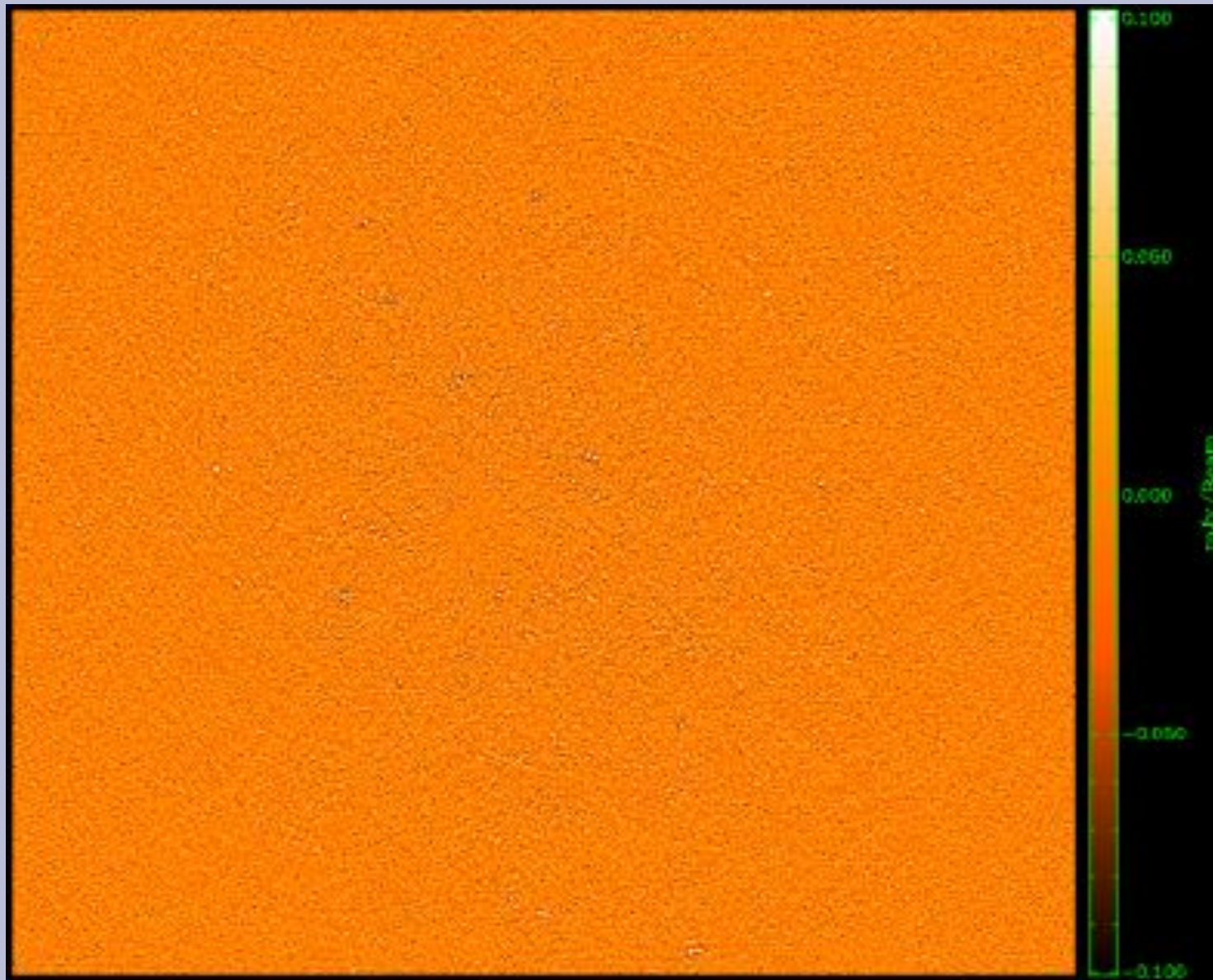
# Flyswatter II



- Solved for  $\Delta E$  for 5 sources.



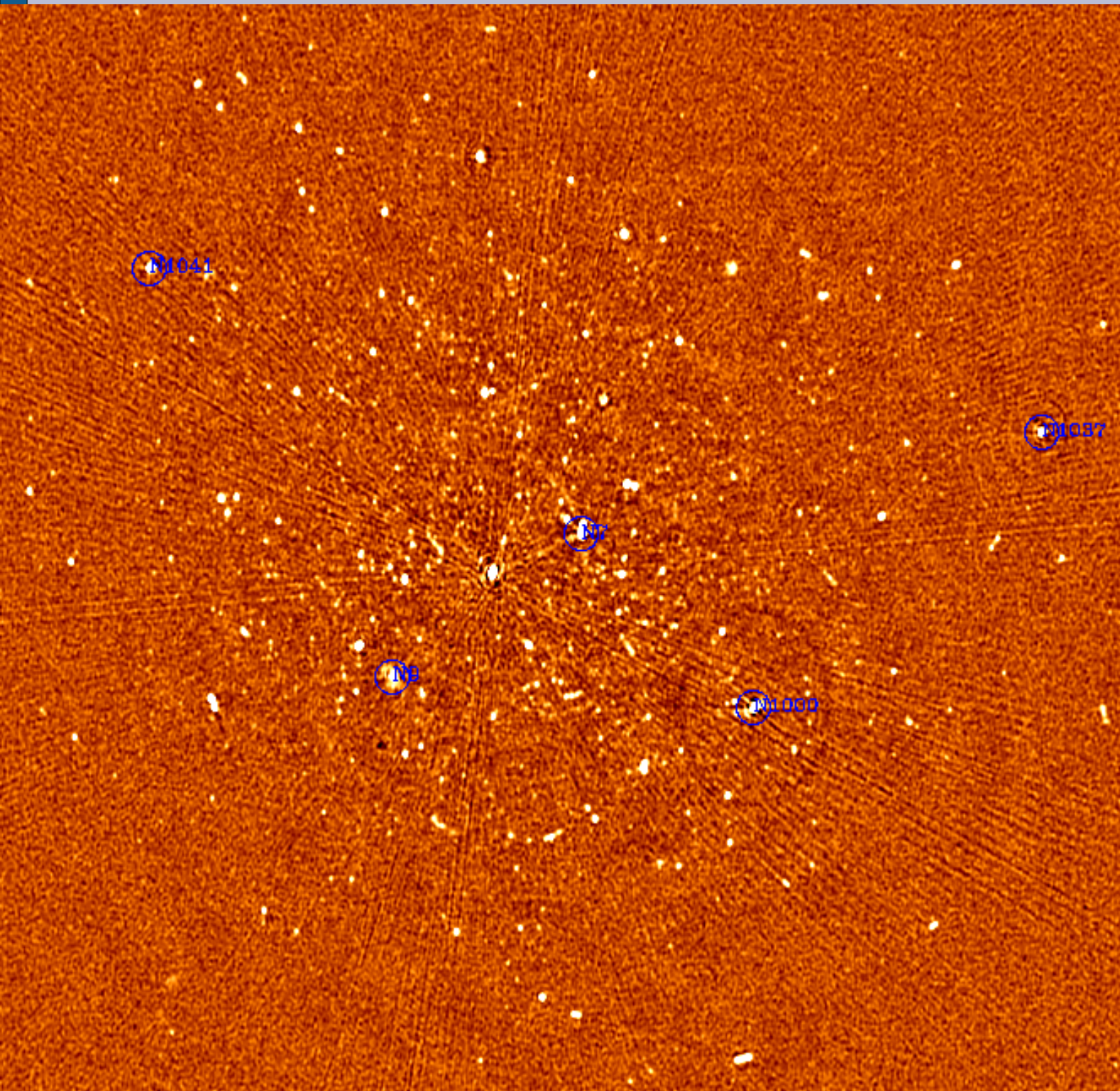
# Flyswatter III



- Solved for  $\Delta E$  for 10 sources.
- In the end 10 sources proved unnecessary (deconvolution helps as well!), so in the final images I only solved for 6 sources.



NEWSTAR image



22Jy @21cm  
12h, 8 bands  
13.5 uJy noise

on-axis DR:  
1500000:1

off-axis DR:  
1000:1

Limited by direction-dependent effects (DDEs) such as pointing errors, tropospheric refraction, etc.  
No direct cure in selfcal.



**3C147**

MeqTrees image

M1041

M1037

M1000

M1008

M1000

22Jy @21cm  
12h, 8 bands  
13.5 uJy noise

Same DR as  
NEWSTAR, but  
no off-axis  
artifacts.

# Exercise: Differential Gains

- We'll repeat this exercise on an even more interesting field.

```
$ cd ~/GLOW2010/dEs
```

```
$ ./reset-ms.sh
```

- (You can always redo this later if you mess up your MS.)

# dE's: Step 1

- Load `example-cal.py`
  - And also `./tigger qmc2-new.lsm.html`
- Load “fill model” options and run “Generate predict”
  - Load an inspector for visibilities
- Load “Calibrate G”
  - Load an inspector for corrected residuals. Run the calibration.
  - Make an image and save it to your purrlog.
  - Right-click on an inspector to save a PNG file to the purrlog

## dE's: Step 2

- Now load “calibrate dEs”
- Run the calibration
  - Look at inspectors, compare to previous purrlog entry
  - Make a residual image, compare to previous entry.



# The End!

