MeqTrees, Measurement Equations, And All That

O. Smirnov (ASTRON)

Plans

- MeqTrees
- Measurement Equations
- Live Demos

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

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

$\Rightarrow$ can be fully described by a 2x2 complex matrix:

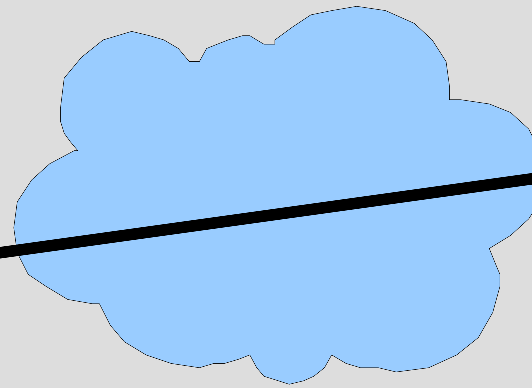
$$\vec{e}' = \boldsymbol{J}\vec{e} \quad \text{i.e.} \quad \begin{pmatrix} e'_x \\ e'_y \end{pmatrix} = \begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix}$$

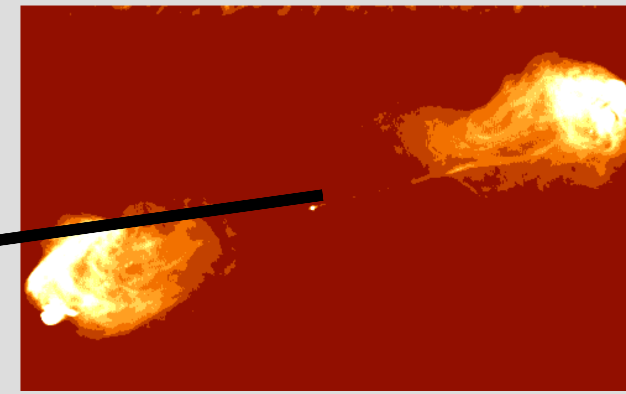
2. Receptor voltages $\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$ are also linear w.r.t. $\vec{e}$

$$\Rightarrow \quad \vec{v} = \boldsymbol{J}\vec{e}$$

Single Dish

$$\vec{v} = \boldsymbol{J}\,\vec{e}$$

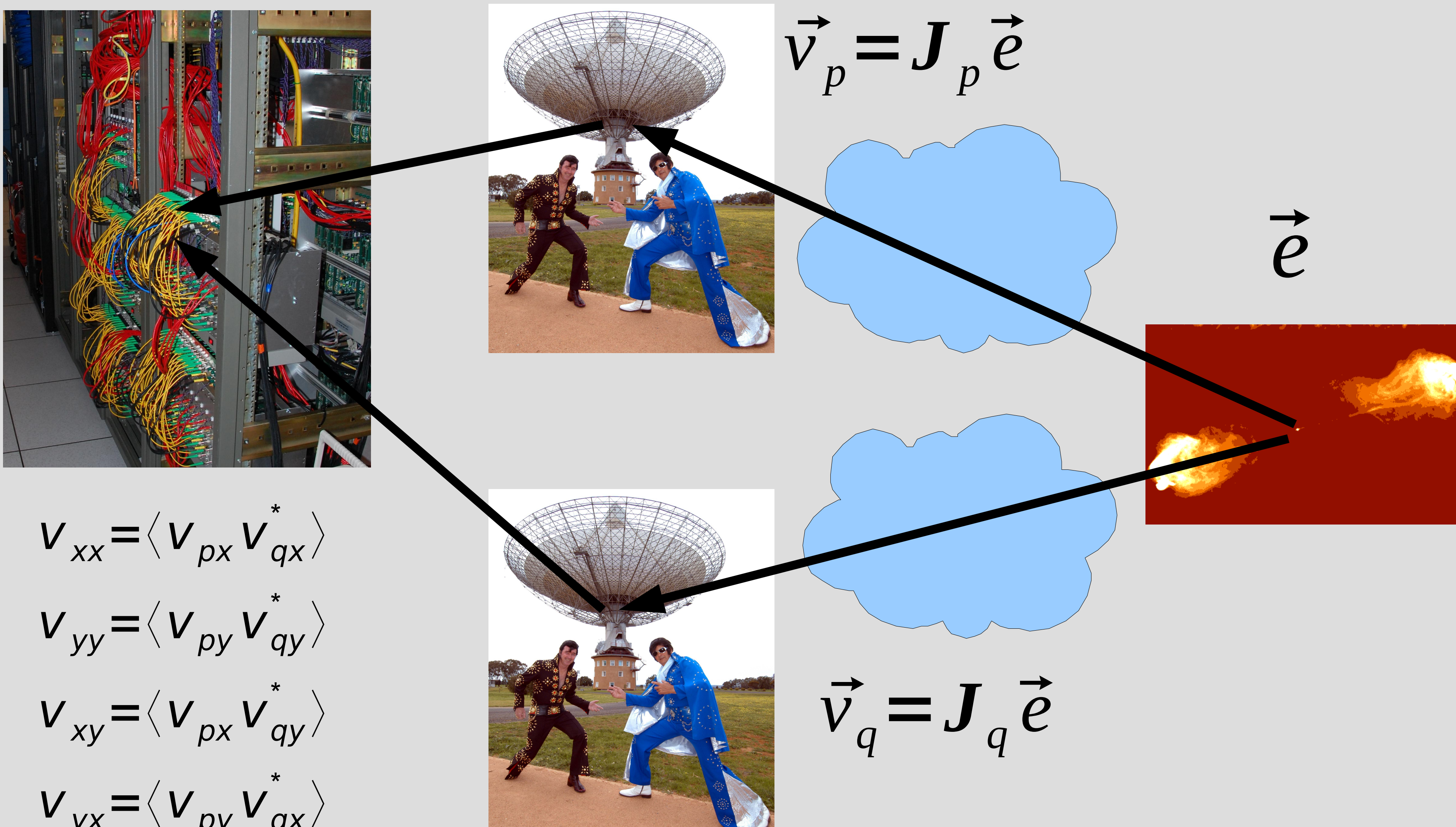




$$\vec{e}$$

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

Interferometry



$$\vec{v}_p = J_p \vec{e}$$

$$\vec{e}$$

$$\vec{v}_q = J_q \vec{e}$$

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

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

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

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

A Wafer-Thin Slice of Physics: Correlations & Visibilities

An interferometer measures *correlations* btw voltages $\vec{v}_p, \vec{v}_q$:

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

It is convenient to represent these as a matrix product:

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

($\langle\,\rangle$: 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: $\vec{v}_p = \boldsymbol{J}_p \vec{e}, \quad \vec{v}_q = \boldsymbol{J}_q \vec{e}$

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

Then $\boldsymbol{V}_{pq} = \langle (\boldsymbol{J}_p \vec{e})(\boldsymbol{J}_q \vec{e})^\dagger \rangle = \langle \boldsymbol{J}_p (\vec{e}\,\vec{e}^\dagger) \boldsymbol{J}_q^\dagger \rangle = \boldsymbol{J}_p \langle \vec{e}\,\vec{e}^\dagger \rangle \boldsymbol{J}_q^\dagger$

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

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

$$\boldsymbol{B} = \langle \vec{e}\,\vec{e}^\dagger \rangle \equiv \frac{1}{2} \begin{pmatrix} I+Q & U \pm iV \\ U \mp iV & I-Q \end{pmatrix} \leftrightarrow (I, Q, U, V)$$

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

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

And That's The Measurement Equation!

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

- Or in more pragmatic terms:

$$\overbrace{\begin{pmatrix} XX & XY \\ YX & YY \end{pmatrix}}^{\text{measured}} = \overbrace{\begin{pmatrix} j_{xx(p)} & j_{xy(p)} \\ j_{yx(p)} & j_{yy(p)} \end{pmatrix}}^{\boldsymbol{J}_p} \overbrace{\frac{1}{2}\begin{pmatrix} I+Q & U+iV \\ U-iV & I-Q \end{pmatrix}}^{\text{source}} \overbrace{\begin{pmatrix} j^*_{xx(q)} & j^*_{yx(q)} \\ j^*_{xy(q)} & j^*_{yy(q)} \end{pmatrix}}^{\boldsymbol{J}_q^\dagger}$$

- 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

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

$$\vec{v} = \boldsymbol{J}_n(\boldsymbol{J}_{n-1}(\ldots \boldsymbol{J}_1 \vec{e})) = (\prod_{i=n}^{1} \boldsymbol{J}_i)\vec{e} = \boldsymbol{J}\vec{e}$$

where $\boldsymbol{J}_1 \ldots \boldsymbol{J}_n$ describes the full signal path.

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

Accumulating Jones Terms

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

$$\boldsymbol{J}_p = \boldsymbol{J}_{pn} \dots \boldsymbol{J}_{p1}, \quad \boldsymbol{J}_q = \boldsymbol{J}_{qm} \dots \boldsymbol{J}_{q1}$$

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

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

or in the "onion form":

$$\boldsymbol{V}_{pq} = \boldsymbol{J}_{pn} (\dots (\boldsymbol{J}_{p2} (\boldsymbol{J}_{p1} \boldsymbol{B} \boldsymbol{J}_{q1}^\dagger) \boldsymbol{J}_{q2}^\dagger) \dots) \boldsymbol{J}_{qm}^\dagger$$

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:

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

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

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

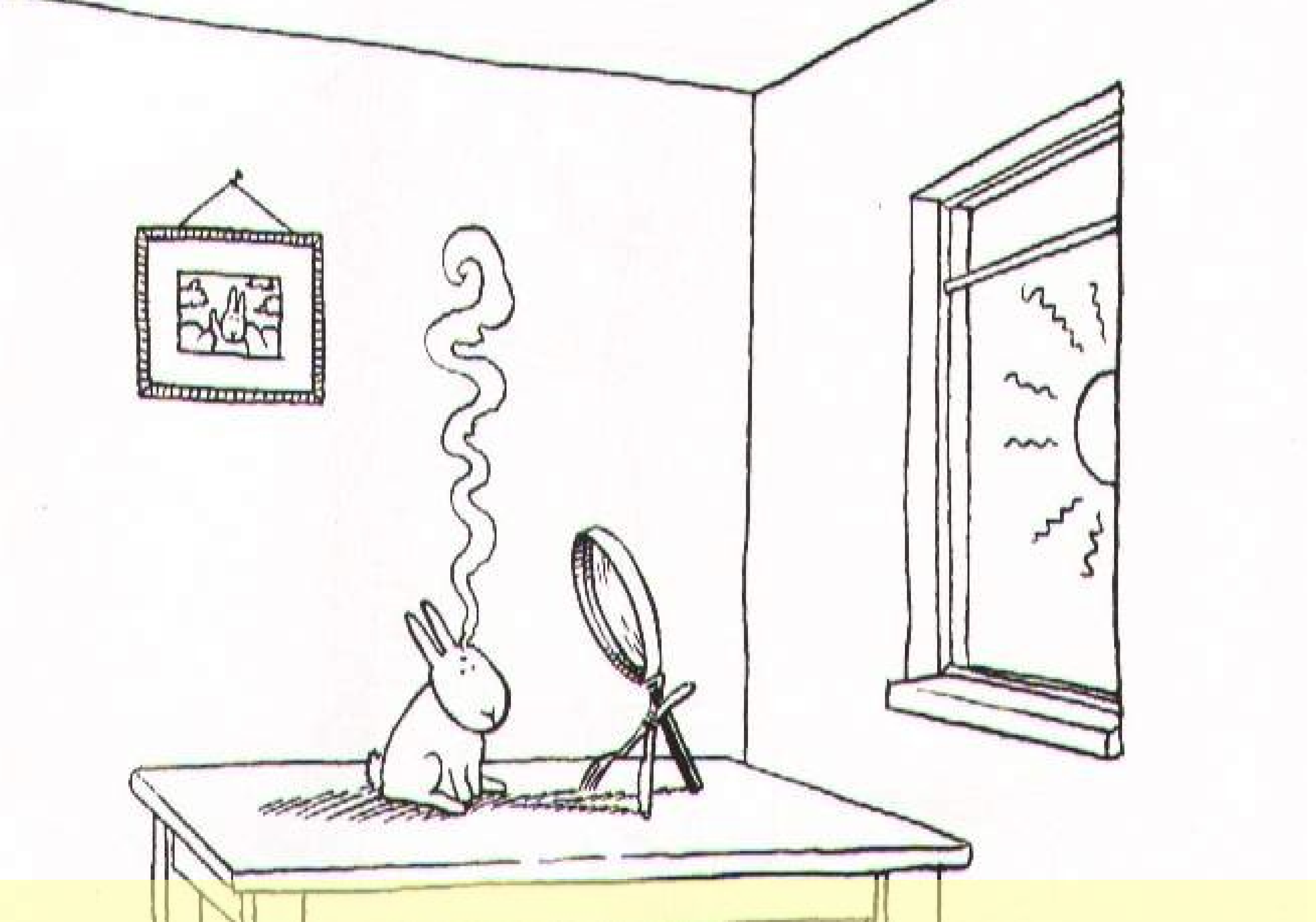
Three Layers Of Intuition

- **Physical**
  - Beam pattern of X and Y dipoles different, causes instrumental polarization of off-center sources
  - Parallactic angle rotates angle of polarization
- **Geometrical**
  - A Jones matrix is also a coordinate tranform
  - gain is stretching => instrumental polarization
  - P.A. is a rotation
  - The two do not commute
- **Mathematical:** matrix properties

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

ME ME ME

- The general formulation above is *"<u>The Measurement Equation</u>"* *(of a generic radio interferometer...)*
- When we want to simulate a specific instrument, we put specific Jones terms into the ME, and derive <u>*a*</u> *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.



Example 1: A Lonely Point Source

Observing a point source with a perfect instrument

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

$$V_{pq} = K_p B K_q^\dagger$$

K_p is the *phase shift* term, a ***scalar*** Jones matrix:

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

- ***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 ϕ_{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:

$$v_{xx,pq} = \frac{1}{2}(I+Q)e^{-i\phi_{pq}} = e^{-i\phi_p}(I+Q)(e^{-i\phi_q})^*$$

$$v_{yy,pq} = \frac{1}{2}(I-Q)e^{-i\phi_{pq}} = e^{-i\phi_p}(I-Q)(e^{-i\phi_q})^*$$

etc...

compare this to:

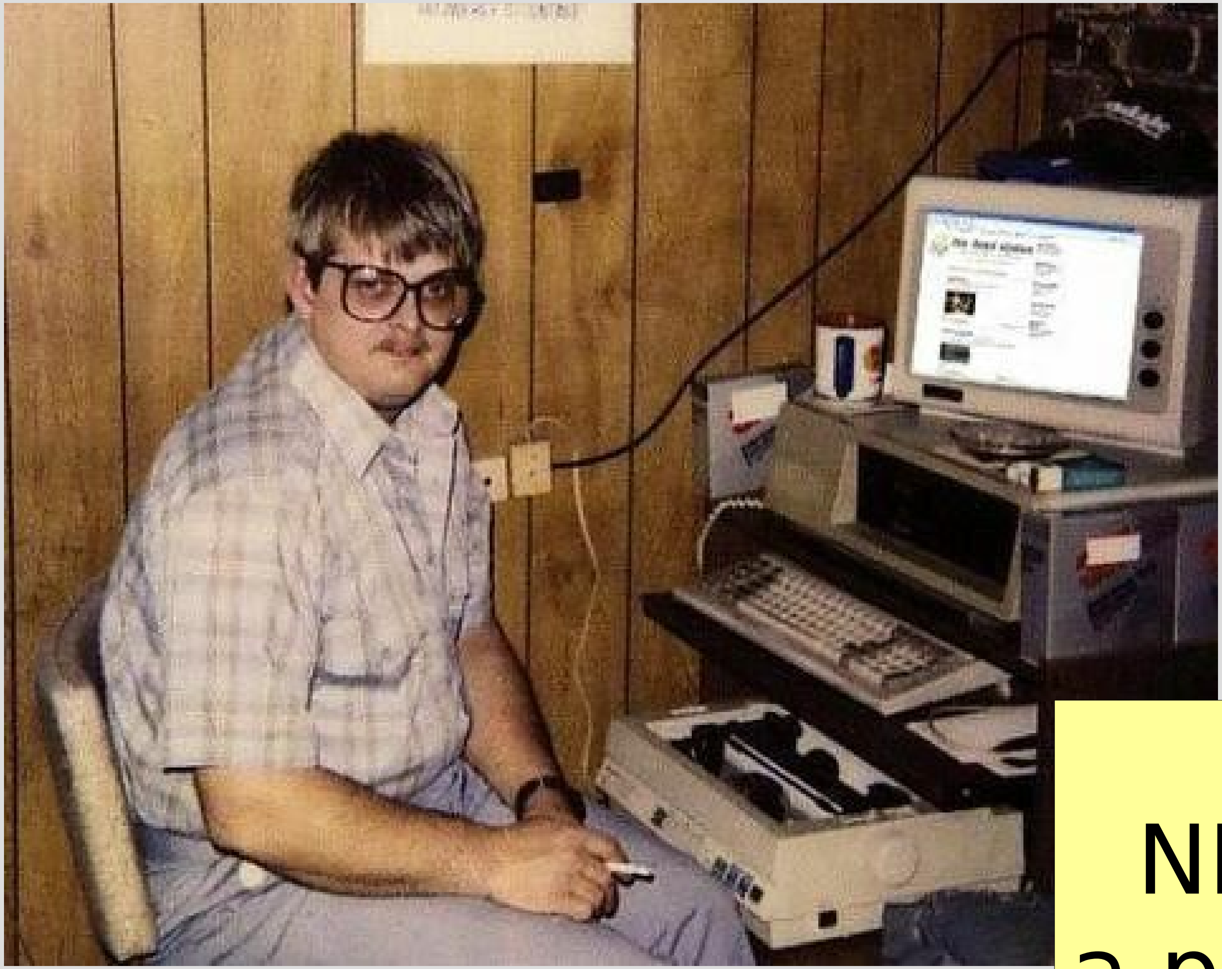
$$\boldsymbol{V}_{pq} = \boldsymbol{K}_p \boldsymbol{B} \boldsymbol{K}_q^{\dagger},$$

with $\boldsymbol{B} = \frac{1}{2}\begin{pmatrix} I+Q & 0 \\ 0 & I-Q \end{pmatrix}$

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.)

Group 1: Developers



Developers:

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

NB: this is not a picture of Oleg

Group 2: Power Users

Power Users:

- have more fun
- steal glory from developers

NB: this is also not a picture of Oleg

Group 3: Button-Pushing Astronomers

The ideal astronomer GUI (Tony Willis):

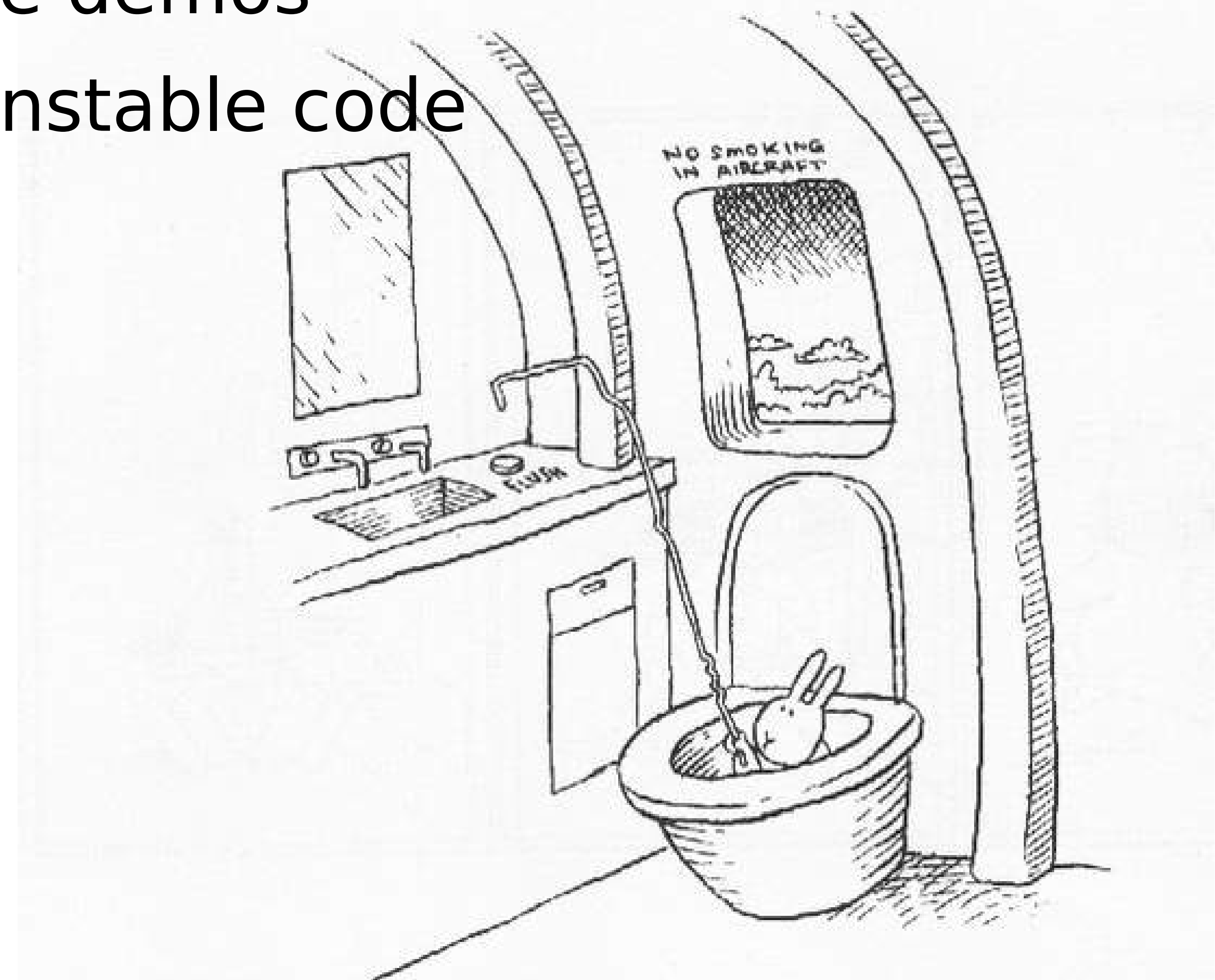
GO

GO FASTER

DO WHAT I MEAN!

The Two Cardinal Rules Of Doing Live Demos

1. Don't do live demos
2. Don't use unstable code



Simulation Demo 1

- Run the browser (meqbrower.py)
- Start a meqserver from the browser
- Load a TDL script (sim.py)
- Setup options
  - MS: WSRT
  - sky model: single point source at center of field
  - no Jones terms
- Compile script, run the tree to fill MS with simulated visibilities
- Run the imager to make a dirty image of the simulation

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
- Purrlogs 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/

Enthroned chicken:

http://www-astro.physics.ox.ac.uk/~ianh/PURRLOGS/enthroned/

Introducing Complex Gains

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

$$v_{xx,pq} = \frac{1}{2}(I+Q)e^{-i\phi_{pq}} g_{x,p} g^*_{x,q}$$

$$v_{yy,pq} = \frac{1}{2}(I-Q)e^{-i\phi_{pq}} g_{y,p} g^*_{y,q}$$

$$v_{xy,pq} = \frac{1}{2}(U+iV)e^{-i\phi_{pq}} g_{x,p} g^*_{y,q}$$

$$v_{yx,pq} = \frac{1}{2}(U-iV)e^{-i\phi_{pq}} g_{y,p} g^*_{x,q}$$

Gains: The ME View

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

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

and with multiple sources:

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

Simulation Demo 2

- 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 sim.py
  - Grid model, 5x5 mJy sources at 5', 1 Jy at center
  - enable G Jones phase error
  - 120 degrees, 2-4 hours
  - Add some 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

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:

$$v_{xx,pq}=\frac{1}{2}(I+Q)e^{-i\phi_{pq}}g_{x,p}g^{*}_{x,q}\rightarrow d_{xx,pq}$$

$$v_{yy,pq}=\frac{1}{2}(I-Q)e^{-i\phi_{pq}}g_{y,p}g^{*}_{y,q}\rightarrow d_{yy,pq}$$

- 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:

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

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

Calibration Demo 1

- Load **cal.py**
  - Use 2x2 data, diagonal terms only
  - **Enable calibrate & correct**
  - **Use sky model with 1 source at center**
  - **Enable G Jones (FullRealImag)**
- Open bookmarks for G and for corrected residuals
- Solve for G diagonal terms
  - Subtiling of 1 in time
  - Tile size 20
- Make an image of the corrected data

M.E. Calibration Terminology

$\boldsymbol{D}_{pq}$: observed visibilities ('data')

$\boldsymbol{K}_p \boldsymbol{B} \boldsymbol{K}_q^{\dagger}$: sky model (or $\sum \boldsymbol{K}_p^{(s)} \boldsymbol{B}^{(s)} \boldsymbol{K}_q^{(s)\dagger}$)

$\boldsymbol{V}_{pq} = \boldsymbol{G}_p \boldsymbol{K}_p \boldsymbol{B} \boldsymbol{K}_q^{\dagger} \boldsymbol{G}_q^{\dagger}$: corrupted model ('predict')

$\boldsymbol{D}_{pq} - \boldsymbol{V}_{pq} \rightarrow \min$: calibration

$\boldsymbol{D}_{pq} - \boldsymbol{V}_{pq}$: corrupted residuals

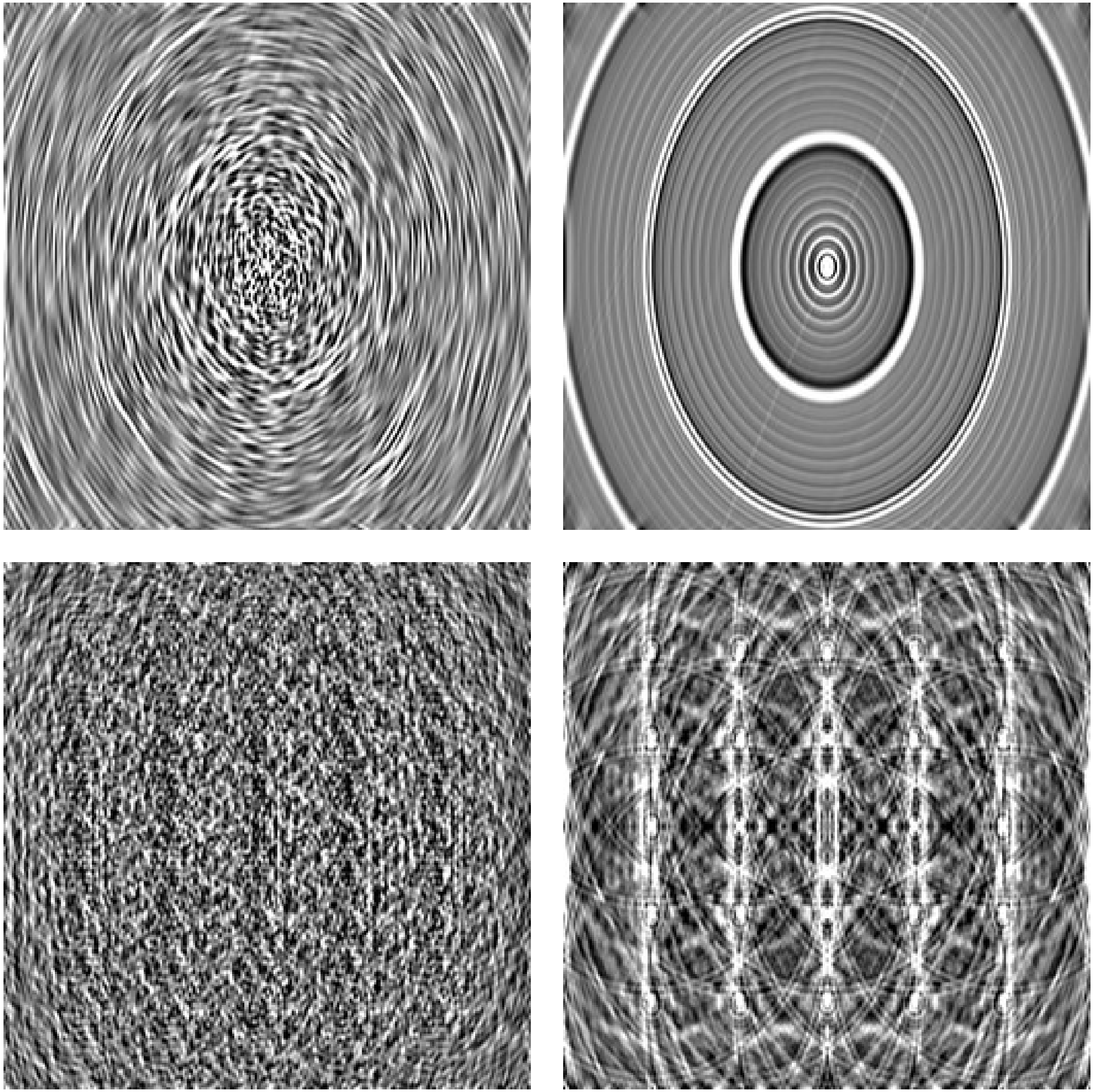
$\boldsymbol{G}_p^{-1} \boldsymbol{D}_{pq} \boldsymbol{G}_q^{-1\dagger}$: corrected data

$\boldsymbol{G}_p^{-1} (\boldsymbol{D}_{pq} - \boldsymbol{V}_{pq}) \boldsymbol{G}_q^{-1\dagger}$: corrected residuals

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

$\boldsymbol{G}_p^{-1}\boldsymbol{D}_{pq}\boldsymbol{G}_q^{-1\dagger}$

(corrected data)



Jy
level

mJy
level

$\boldsymbol{D}_{pq}-\boldsymbol{V}_{pq}$

(corrupted residuals)

$\boldsymbol{G}_p^{-1}(\boldsymbol{D}_{pq}-\boldsymbol{V}_{pq})\boldsymbol{G}_q^{-1\dagger}$

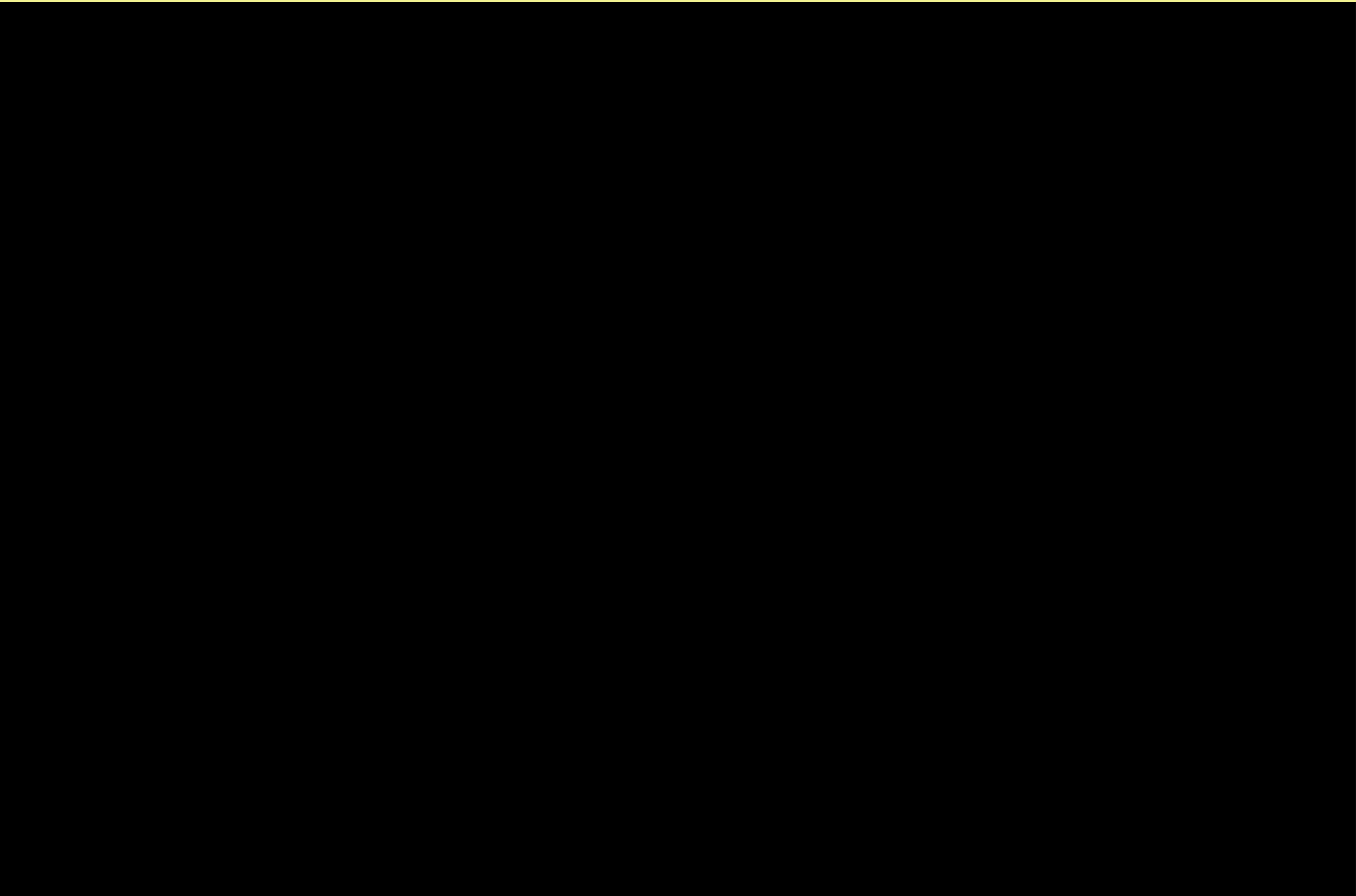
(corrected residuals)

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

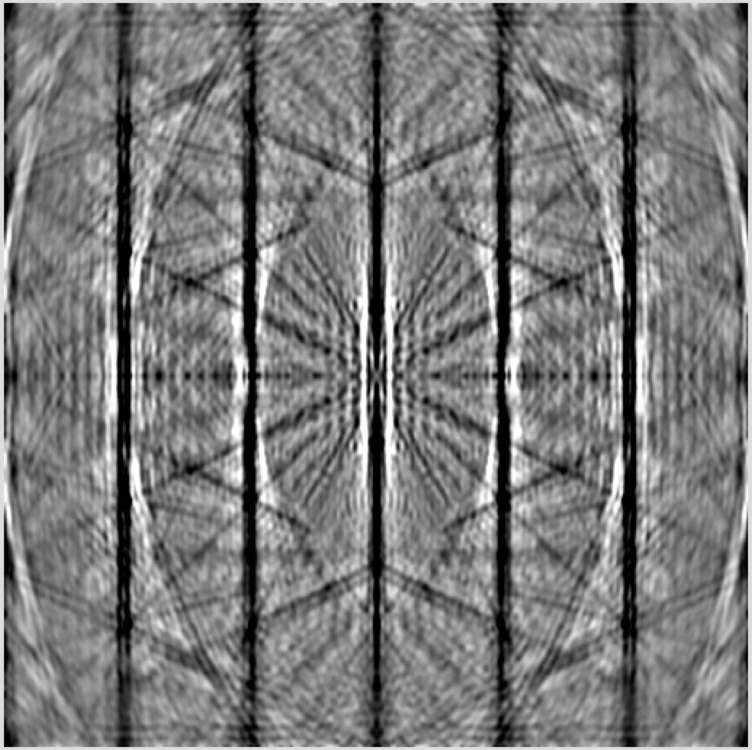
- What is satisfaction?

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
- Error level here ~0.01 mJy (dynamic range: 1:100,000)



The Classical Approach To Polarization

Classical Equation For Polarization Selfcal

$$R_j R_k^* = G_{Rj} G_{Rk}^* \Big[E_{Rj} E_{Rk}^* e^{-i\left(\phi_j - \phi_k\right)} + D_{Rk}^* E_{Rj} E_{Lk}^* e^{-i\left(\phi_j + \phi_k\right)} +$$

$$D_{Rj} E_{Lj} E_{Rk}^* e^{i\left(\phi_j + \phi_k\right)} + D_{Rj} D_{Rk}^* E_{Lj} E_{Lk}^* e^{i\left(\phi_j - \phi_k\right)} \Big]$$

$$R_j L_k^* = G_{Rj} G_{Lk}^* \Big[E_{Rj} E_{Lk}^* e^{-i\left(\phi_j + \phi_k\right)} + D_{Lk}^* E_{Rj} E_{Rk}^* e^{-i\left(\phi_j - \phi_k\right)} +$$

$$D_{Rj} E_{Lj} E_{Lk}^* e^{i\left(\phi_j - \phi_k\right)} + D_{Rj} D_{Lk}^* E_{Lj} E_{Rk}^* e^{i\left(\phi_j + \phi_k\right)} \Big] .$$

(With thanks to Huib Jan van Langevelde)

The Measurement Equation For Polarization Selfcal

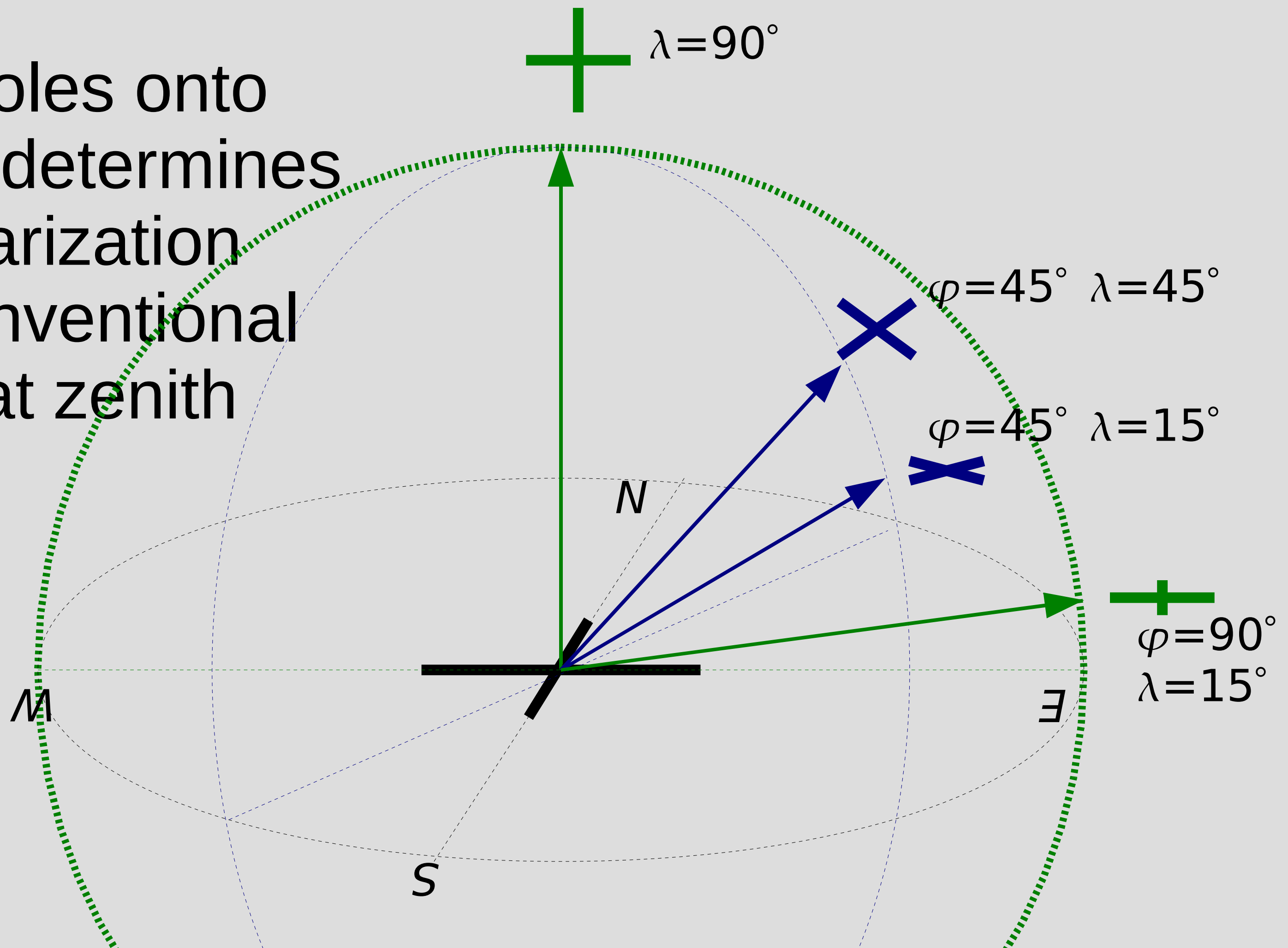
$$\boldsymbol{V}_{pq} = \boldsymbol{G}_p \boldsymbol{K}_p \boldsymbol{B} \boldsymbol{K}_q^\dagger \boldsymbol{G}_q^\dagger$$

$$\boldsymbol{G}_p = \begin{pmatrix} g_{11,p} & g_{\text{١٢},p} \\ g_{21,p} & g_{22,p} \end{pmatrix}$$

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

A Case Study: 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

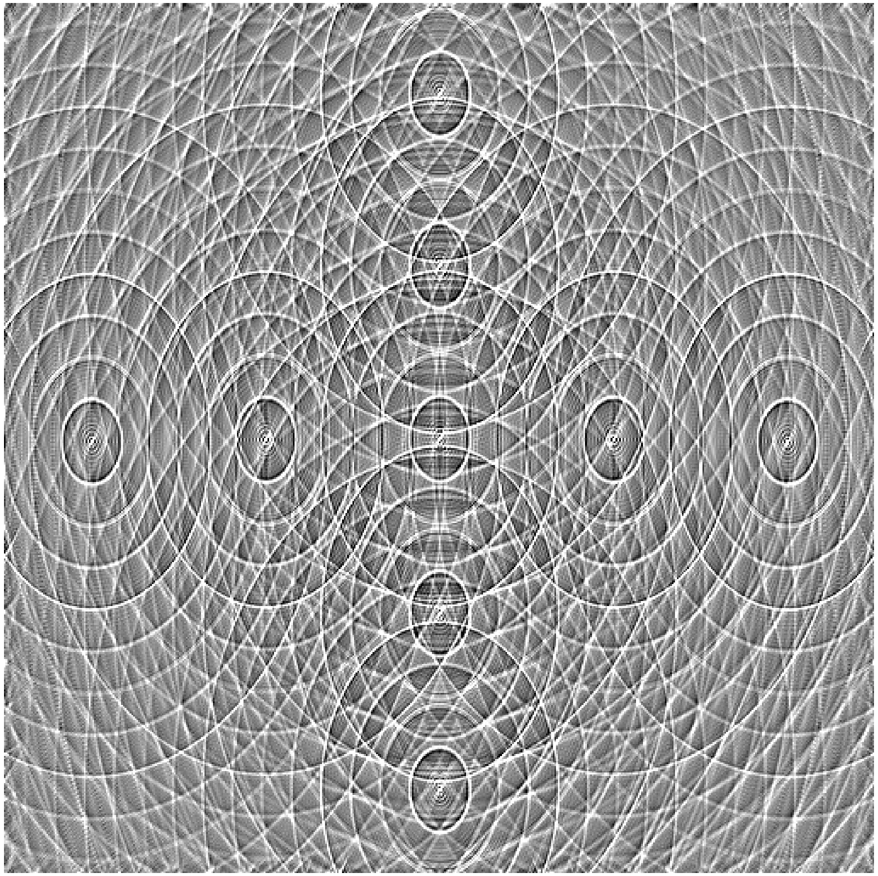
- Projection can be described by a Jones matrix:

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

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

Simulation Demo 3

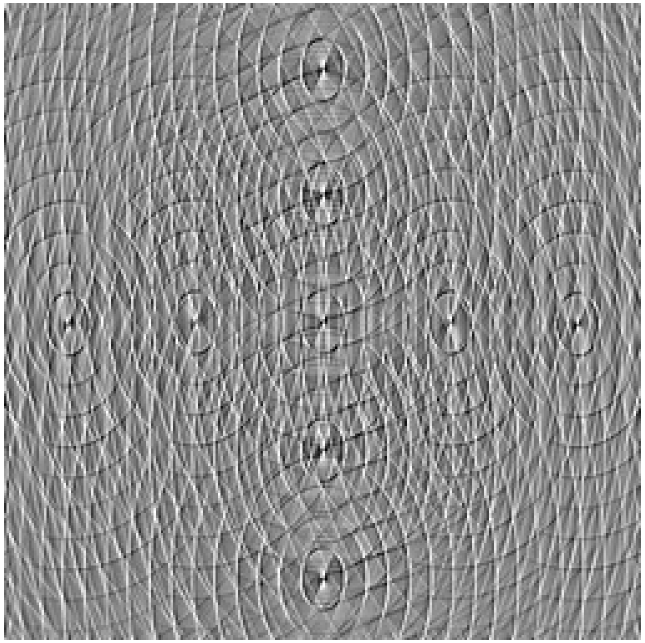
- We'll simulate dipole projection
- Run sim2.py
- Sky model: 5x5 cross at 30'
- Enable L Jones
  - Per-source but not per-station
- Open bookmarks to check az/el and L Jones
- Make an IQUV image
  - Note distortions in I map due to time-varying sensitivity of the dipoles
  - Note instrumental QU polarization – direction-dependent!



Stokes *I* map.

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

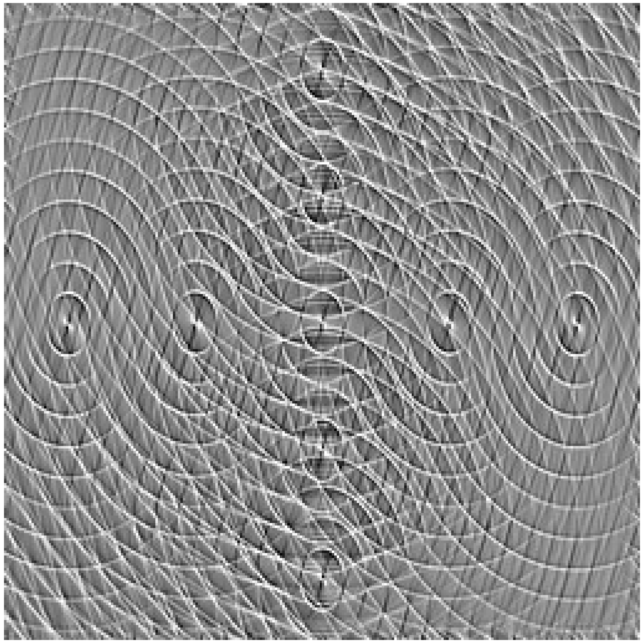
Peak flux is ~.6 Jy (would be 1 Jy without this effect!)



Q and *U* maps. Note instrumental polarization

(direction-dependent!)

Peak flux is ±0.1 Jy



Calibrating For Dipole Projection?

- The ME we are using is:

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

- For calibration, we can use the same ME and solve for ***G*** Jones again
- No need to solve for ***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?

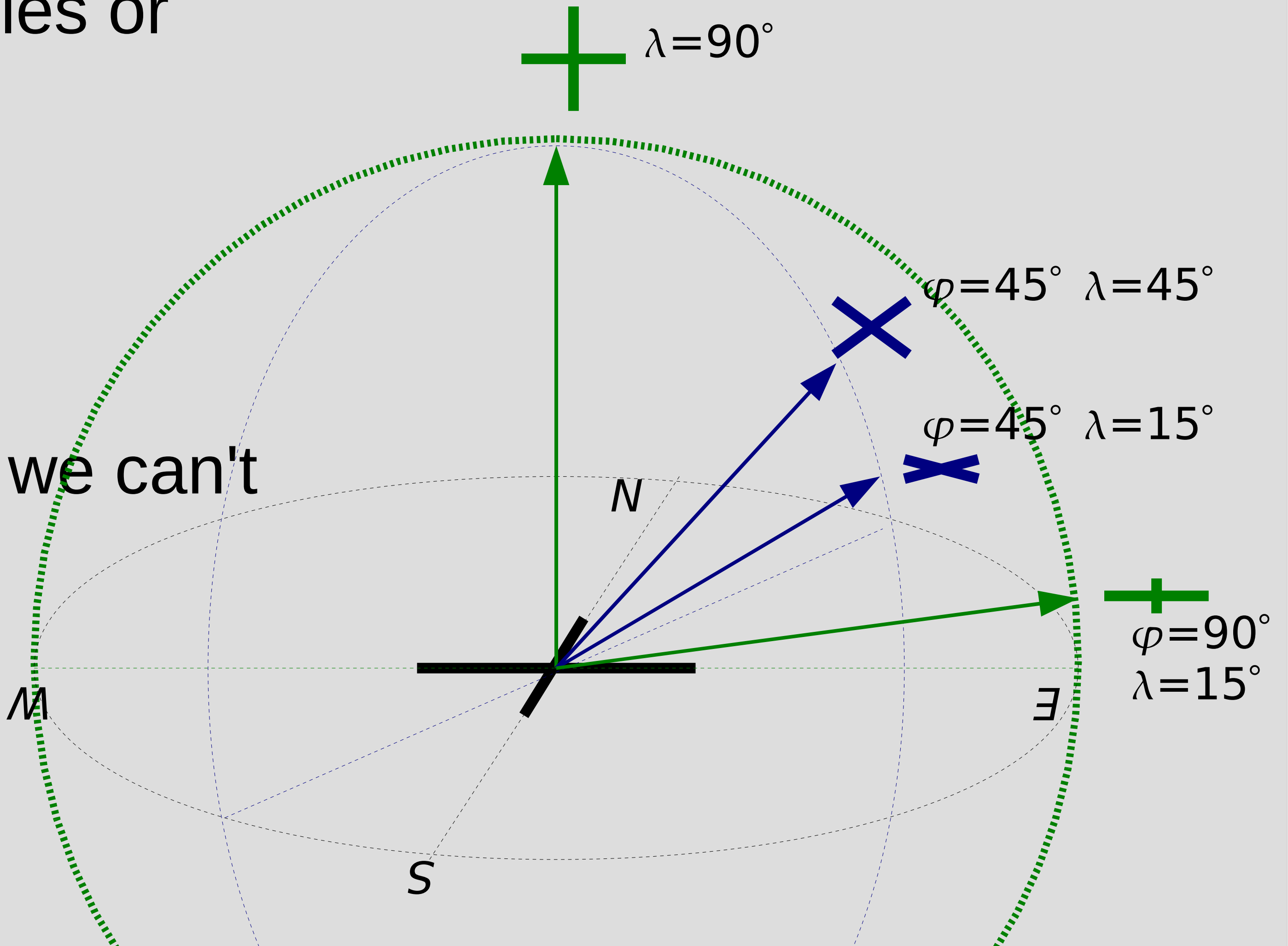
Problem 1: Inverting Jones Terms

- The ME allows us to write out corrected visibilities or residuals:

$$\boldsymbol{L}_p^{-1} \boldsymbol{D}_{pq} \boldsymbol{L}_q^{-1\dagger}$$

$$\boldsymbol{L}_p^{-1} (\boldsymbol{D}_{pq} - \boldsymbol{V}_{pq}) \boldsymbol{L}_q^{-1\dagger}$$

- What happens if we can't invert $\boldsymbol{L}$?



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

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:

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

with DD effects

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:

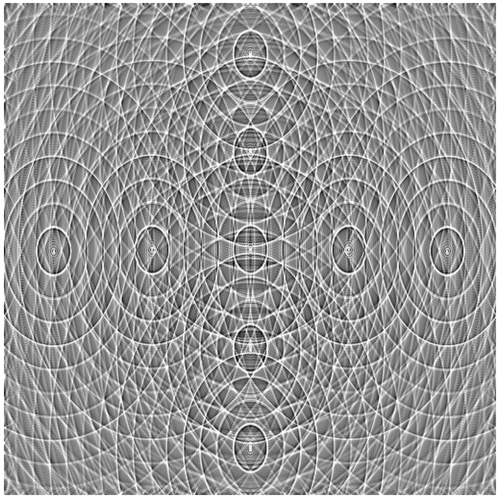
$$\tilde{\mathbf{G}}_p^{-1} \mathbf{D}_{pq} \tilde{\mathbf{G}}_q^{\dagger -1} \neq \mathbf{S}_{pq}$$

at best we can pick a direction s_0:

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

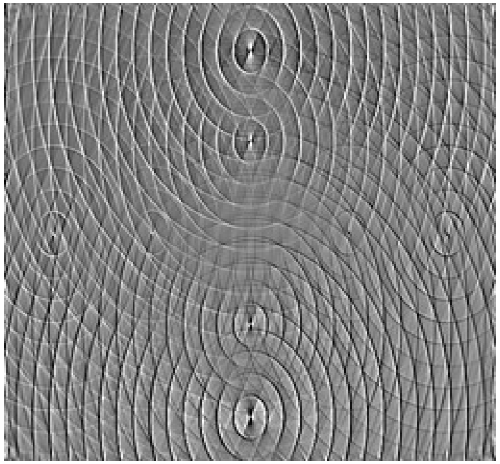
Demo: Correcting For a Single Direction

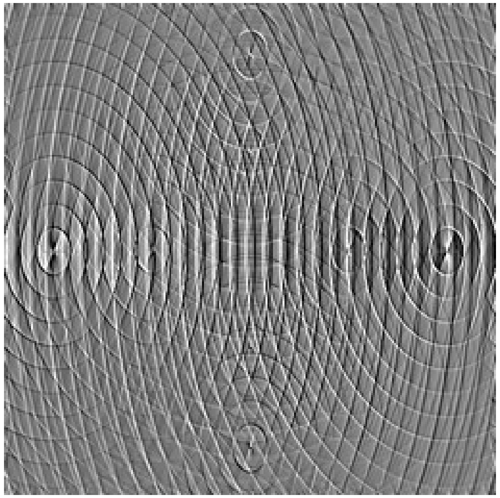
- In general, visibility data can only be “corrected” for a single direction on the sky.
- Hence, e.g., facet imaging.
- Bhatnagar (EVLA Memo 100) suggests an approximate method to apply on-the-fly corrections during imaging)
- Correction Demo:
  - Run cal2.py
  - Enable correct, disable calibrate and 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.

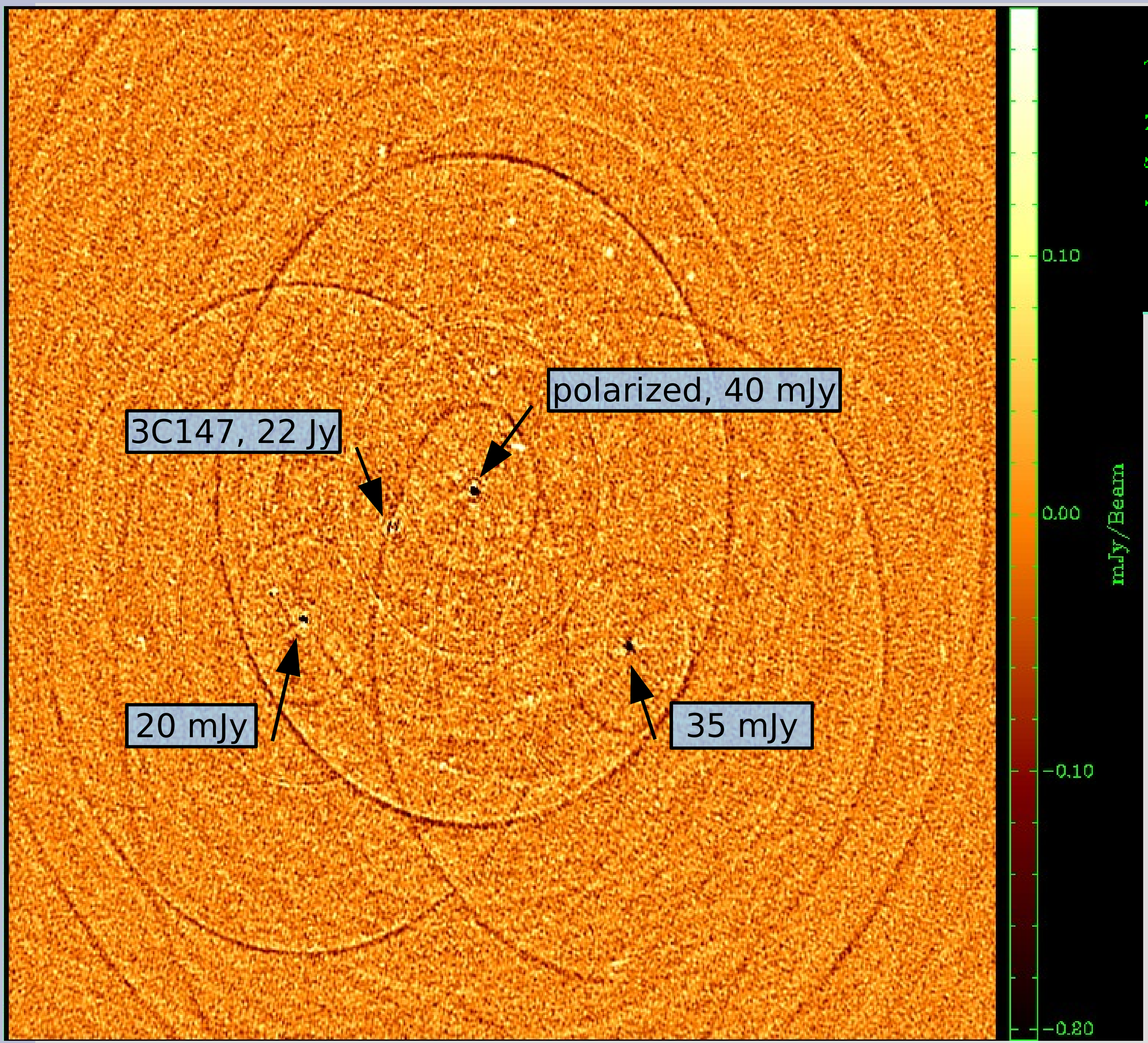
Note how instrumental polarization corrects perfectly at center, but increases towards edge of field.

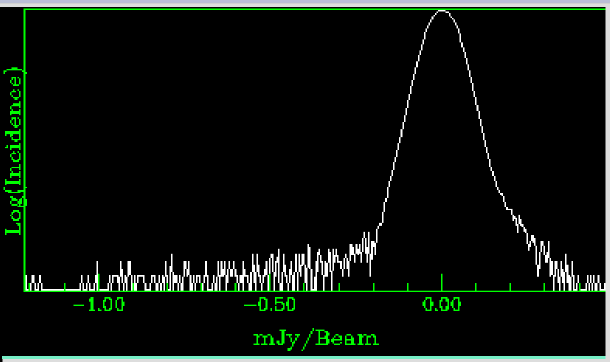
Peak flux is ±50 mJy.

Dealing With D-D Effects

- The same issue arises with other D-D effects:
  - Ionosphere
  - Beam shapes & pointing errors
- Becoming critical for today's SKA pathfinders, 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
- σ ~ 30uJy
- 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

Differential Gains

- We can write an m.e. with differential gains:

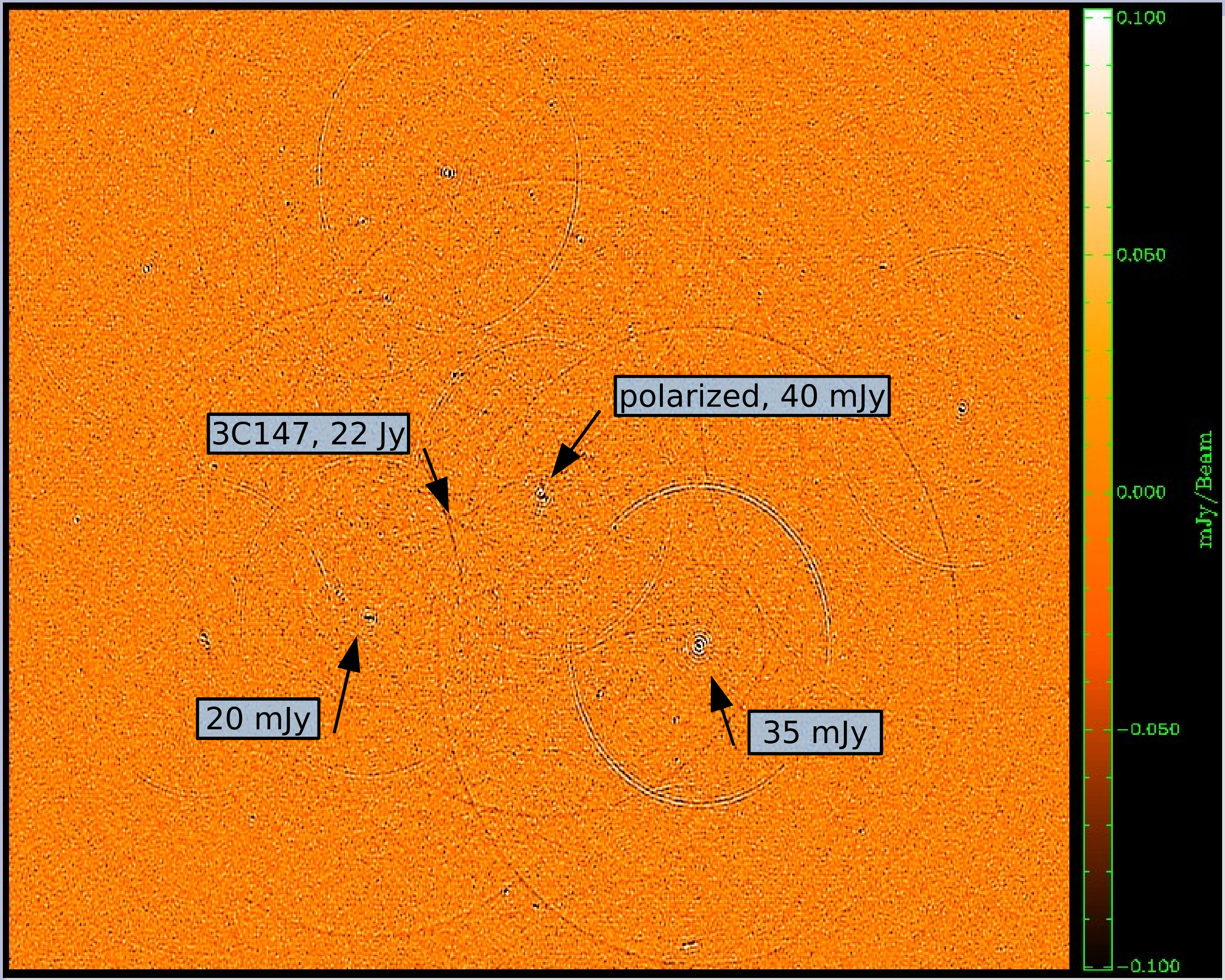
$$\mathbf{V}_{pq} = \underbrace{\mathbf{G}_p}_{\text{gain \& bandpass}} \underbrace{\left(\sum_s \underbrace{\Delta \mathbf{E}_p^{(s)}}_{\text{differential gain}} \underbrace{\mathbf{E}_p^{(s)}}_{\text{beam}} \underbrace{\mathbf{X}_{pq}}_{\text{source coherency}} \mathbf{E}_q^{(s)\dagger} \Delta \mathbf{E}_q^{(s)\dagger}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

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

Solvable for a handful of "troublesome" sources,

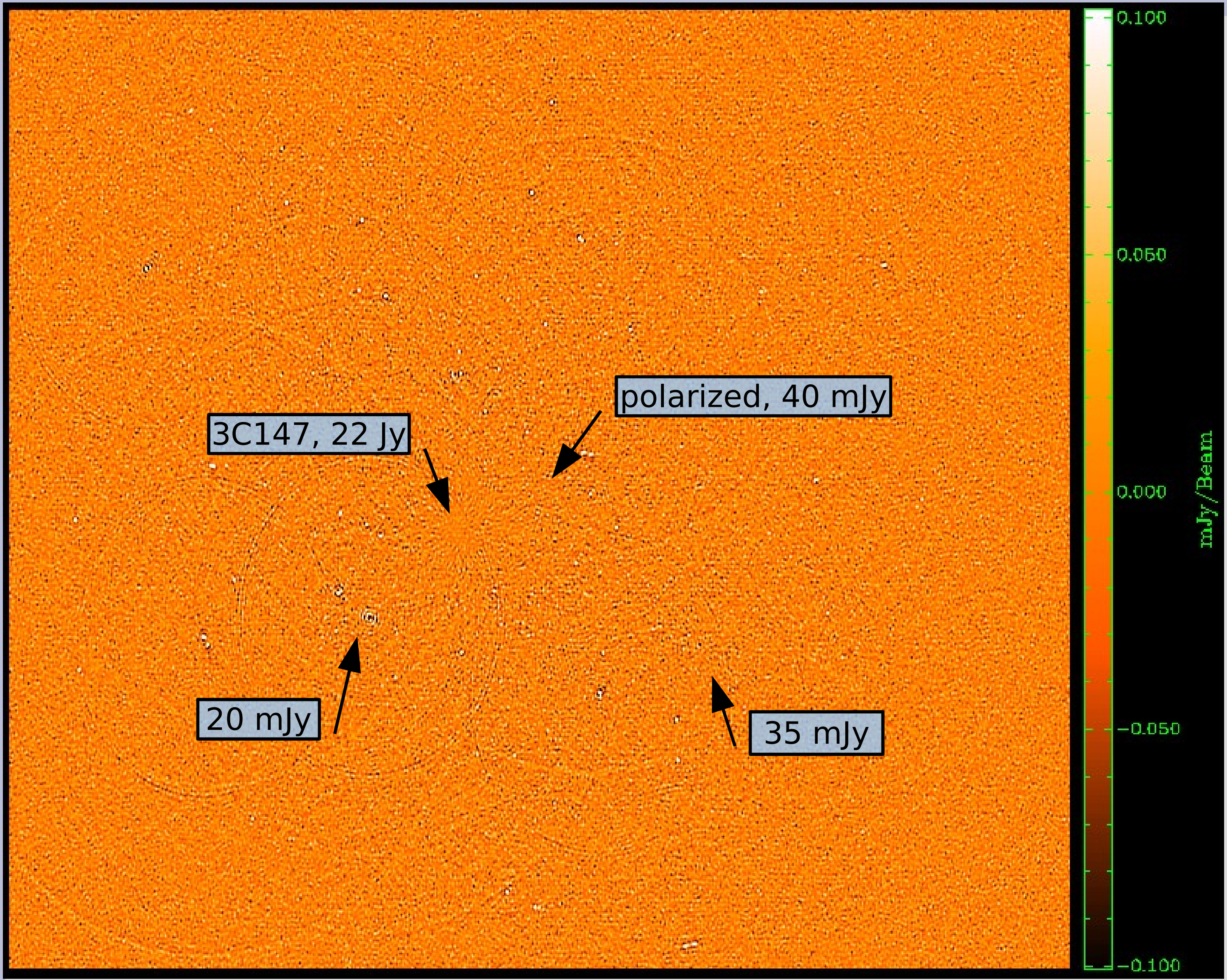
and set to unity for the rest.

Flyswatter I



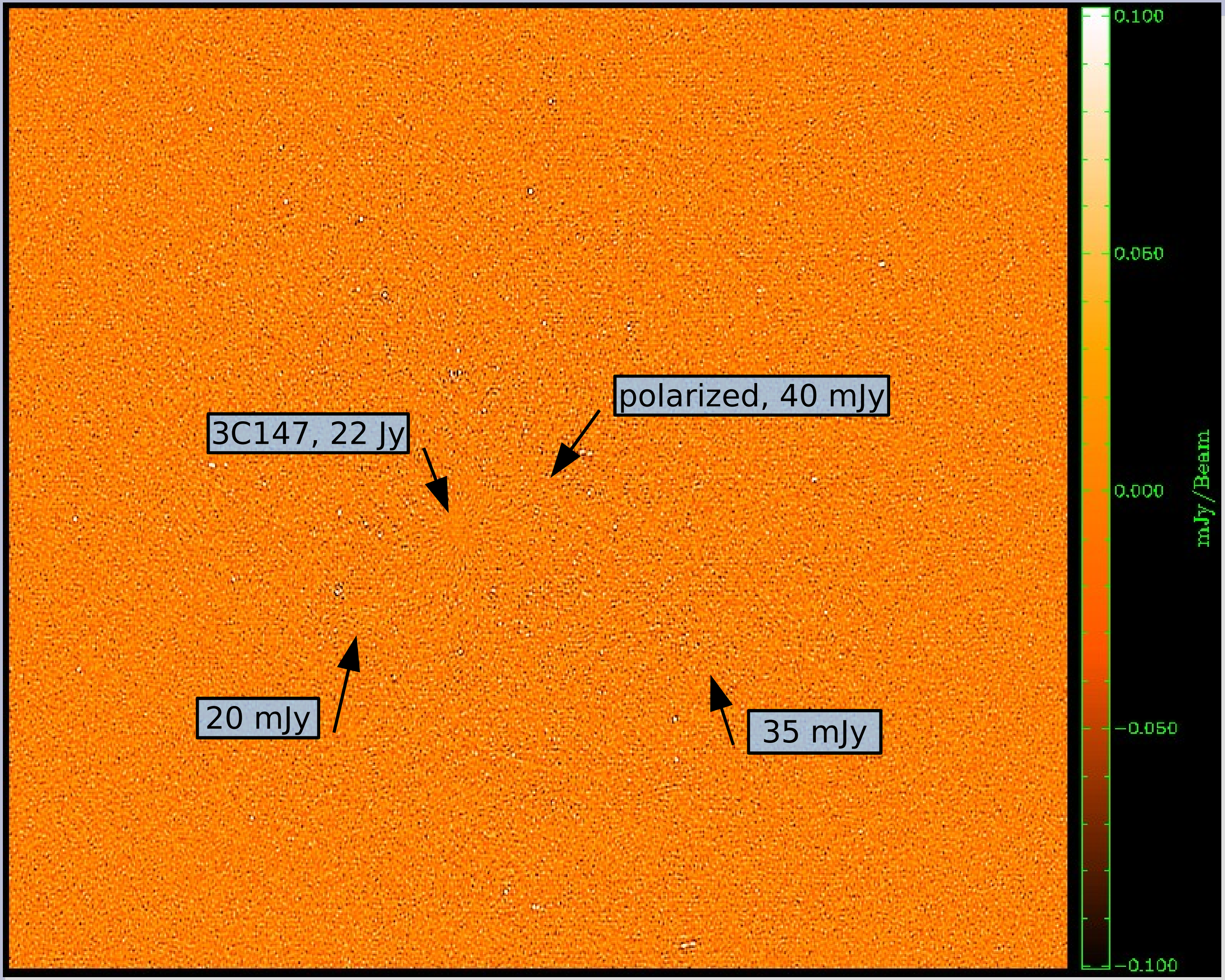
- The “before” image.

Flyswatter II



- Solved for ***ΔE*** for 5 sources.

Flyswatter III



- Solved for ***ΔE*** for 10 sources.

Nancay Workshop, SSSC

- MCCT SKADS Workshop
  "Towards 3rd Generation Calibration In Radio Astronomy"
  Nancay, Sep 27 – Oct 10, 2009
  http://mcct.skads-eu.org/nancay/nancay-mcct.php

- Qualification via the SKADS Set Of Standard Challenges:
  http://www-astro.physics.ox.ac.uk/~ianh/SSSC/index.html

The End!