Wirtinger calibration and imaging for the extragalactic surveys

Cyril Tasse

In collaboration with **Oleg Smirnov**

With the help of

- **Martin Hardcastle**
- **Tim Shimwell**
- **Wendy Williams**
- **Francesco de Gasperin**

And **Nicolas Vilchez** for the SkyModel module

A brief history of CohJones

- When I heard about StefCal in **an Oleg Smirnov talk** *(Salvini & Wijnholds 2014 and refs therein)*
- I could find **an algorithm** that was **using the same idea**, but generalising the problem to **simultaneous N-Directional solving** (not pealing), using a **special linear operator**
- I could have it working quickly
- Only ~1 year later I realised **this operator was a Wirtinger Jacobian**

A brief history of CohJones

- When I heard about StefCal in **an Oleg Smirnov talk** *(Salvini & Wijnholds 2014 and refs therein)*
- I could find **an algorithm** that was **using the same idea**, but generalising the problem to **simultaneous N-Directional solving** (not pealing), using a **special linear operator**
- I could have it working quickly
- Only ~1 year later I realised **this operator was a Wirtinger Jacobian**
- Together with **Oleg Smirnov**, we understood the missing links to connect **LM** to **CohJones** to **StefCal**
- …. and there are still a few mysterious bits

You can read:

- **Tasse 2014: Arxiv/1410.8706**
- **Smirnov & Tasse 2015 MNRAS (in press)**

- **Paper 2015+:**
  Extending use of Wirtinger Jacobian to Kalman filters

Wilhelm Wirtinger derivatives

- What is the derivative of a complex number ?
- Traditionally one takes the derivative against
  - Real part
  - Imaginary part
- But you can also use Wirtinger's derivative definition:
  - Complex
  - Complex conjugate

$$\frac{\partial \overline{z}}{\partial z} = 0 \text{ and } \frac{\partial z}{\partial \overline{z}} = 0$$

Let's look at NL optimisation algos

- It has been demonstrated that Levenberg-Maquardt still converges with Wirtinger Jacobian (Kreutz-Delgado 2009; Laurent et al. 2012)

$$\widehat{\boldsymbol{g}_{i+1}} = \widehat{\boldsymbol{g}_i} + \mathbf{K}\big|_{\widehat{\boldsymbol{g}_i}}^{-1} \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}^{H} \mathbf{C}^{-1} \left(\mathbf{v_m} - h\left(\widehat{\boldsymbol{g}_i}\right)\right)$$

$$\text{with } \mathbf{K}\big|_{\widehat{\boldsymbol{g}_i}} = \mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}} + \lambda.\text{diag}\left(\mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}}\right)$$

$$\text{and } \mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}} = \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}^{H} \mathbf{C}^{-1} \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}$$

  - BBS is using it with a « traditional Jacobian »
- What happen if we replace the « traditional Jacobian » by the wirtinger Jacobian ?

Let's look at NL optimisation algos

- It has been demonstrated that Levenberg-Maquardt still converges with Wirtinger Jacobian (Kreutz-Delgado 2009; Laurent et al. 2012)

$$\widehat{\boldsymbol{g}_{i+1}} = \widehat{\boldsymbol{g}_i} + \mathbf{K}\big|_{\widehat{\boldsymbol{g}_i}}^{-1} \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}^{H} \mathbf{C}^{-1} \left(\mathbf{v_m} - h\left(\widehat{\boldsymbol{g}_i}\right)\right)$$

$$\text{with } \mathbf{K}\big|_{\widehat{\boldsymbol{g}_i}} = \mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}} + \lambda.\text{diag}\left(\mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}}\right)$$

$$\text{and } \mathbf{H}\big|_{\widehat{\boldsymbol{g}_i}} = \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}^{H} \mathbf{C}^{-1} \boldsymbol{J}\big|_{\widehat{\boldsymbol{g}_i}}$$

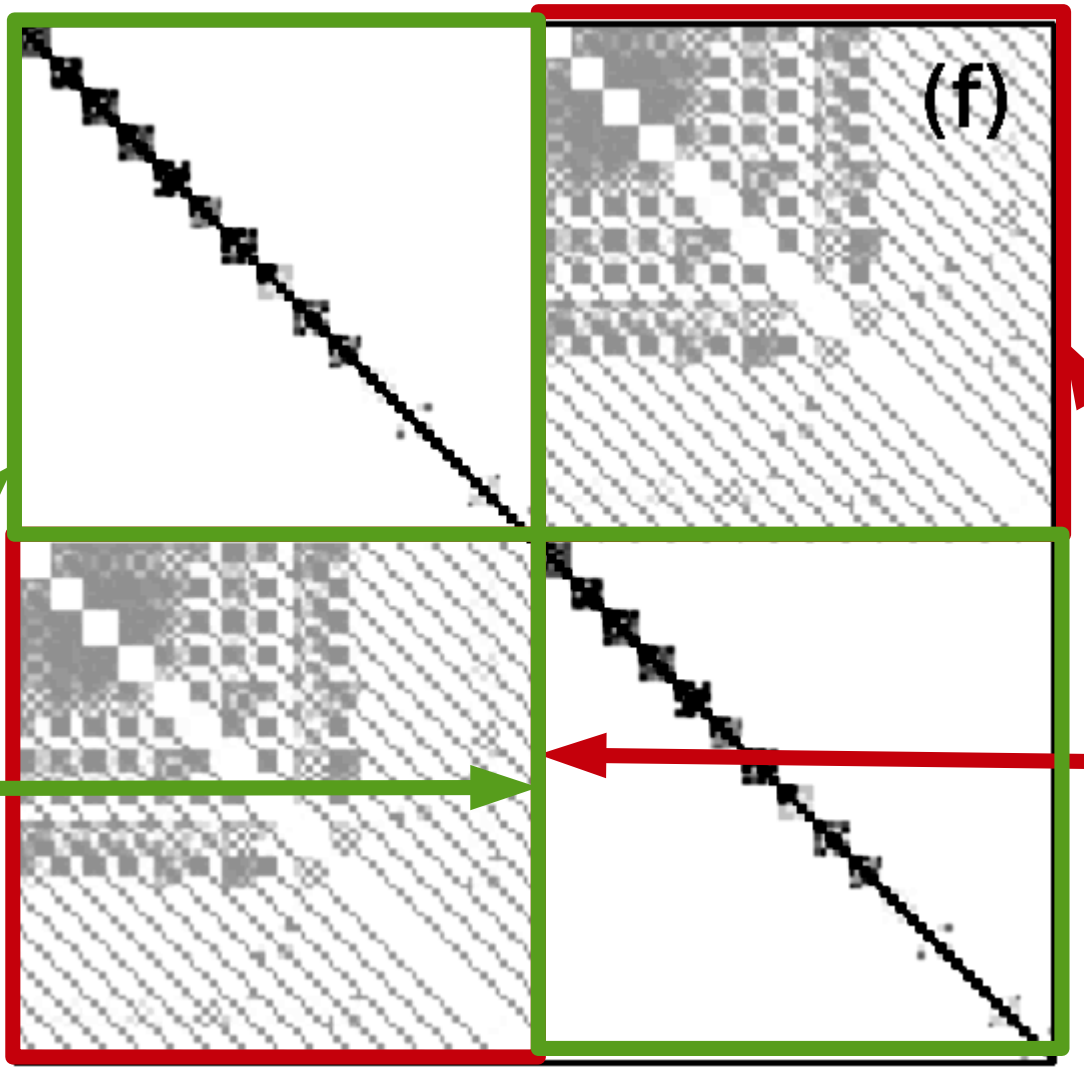
That matrix becomes very interesting

  - BBS is using it with a « traditional Jacobian »
- What happen if we replace the « traditional Jacobian » by the wirtinger Jacobian ?

Let's look at NL optimisation algos

$J^H J$

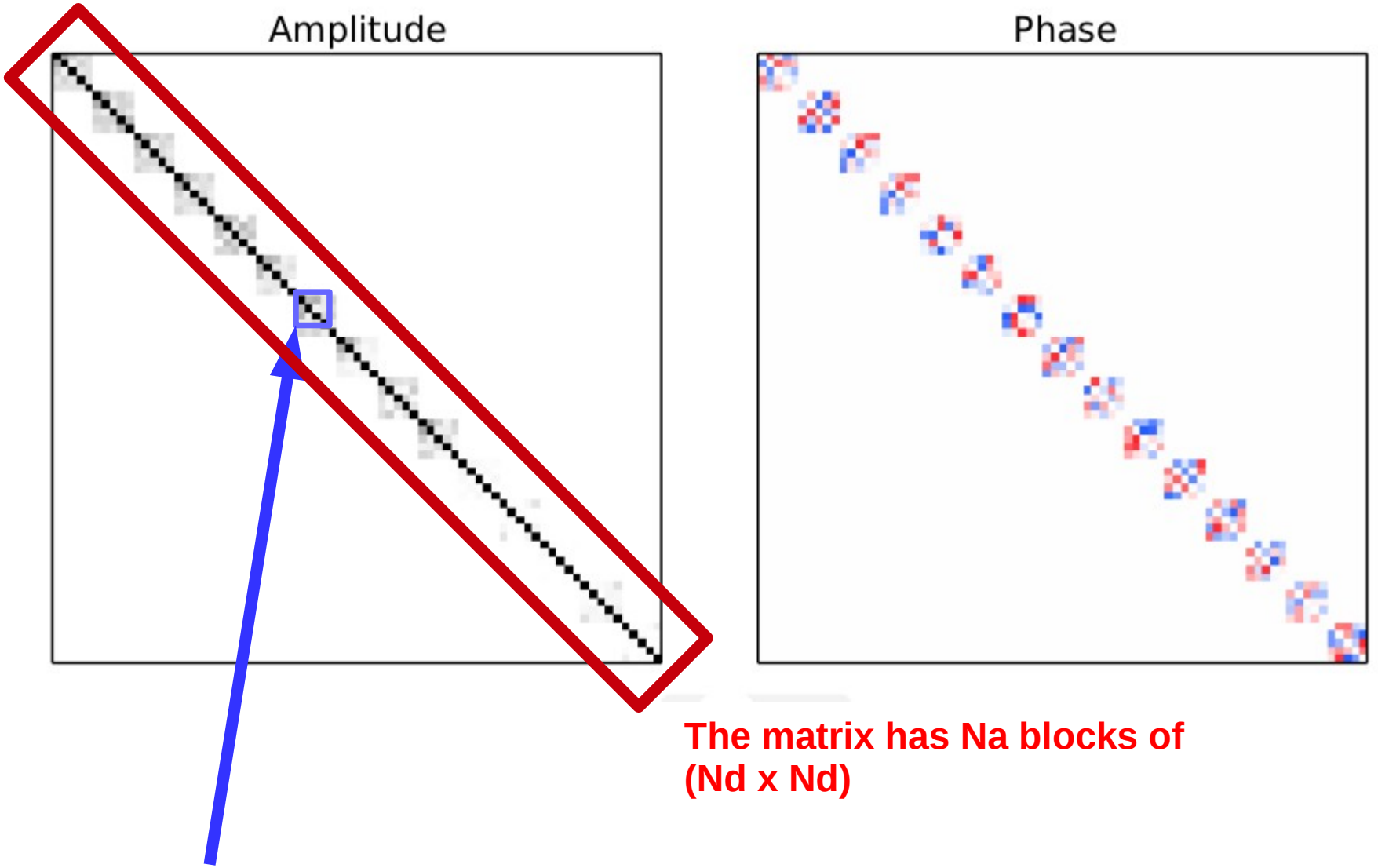
For Classical and Wirtinger, for different variable ordering (simulation with 5 directions, 15 antennas)



Those are equal

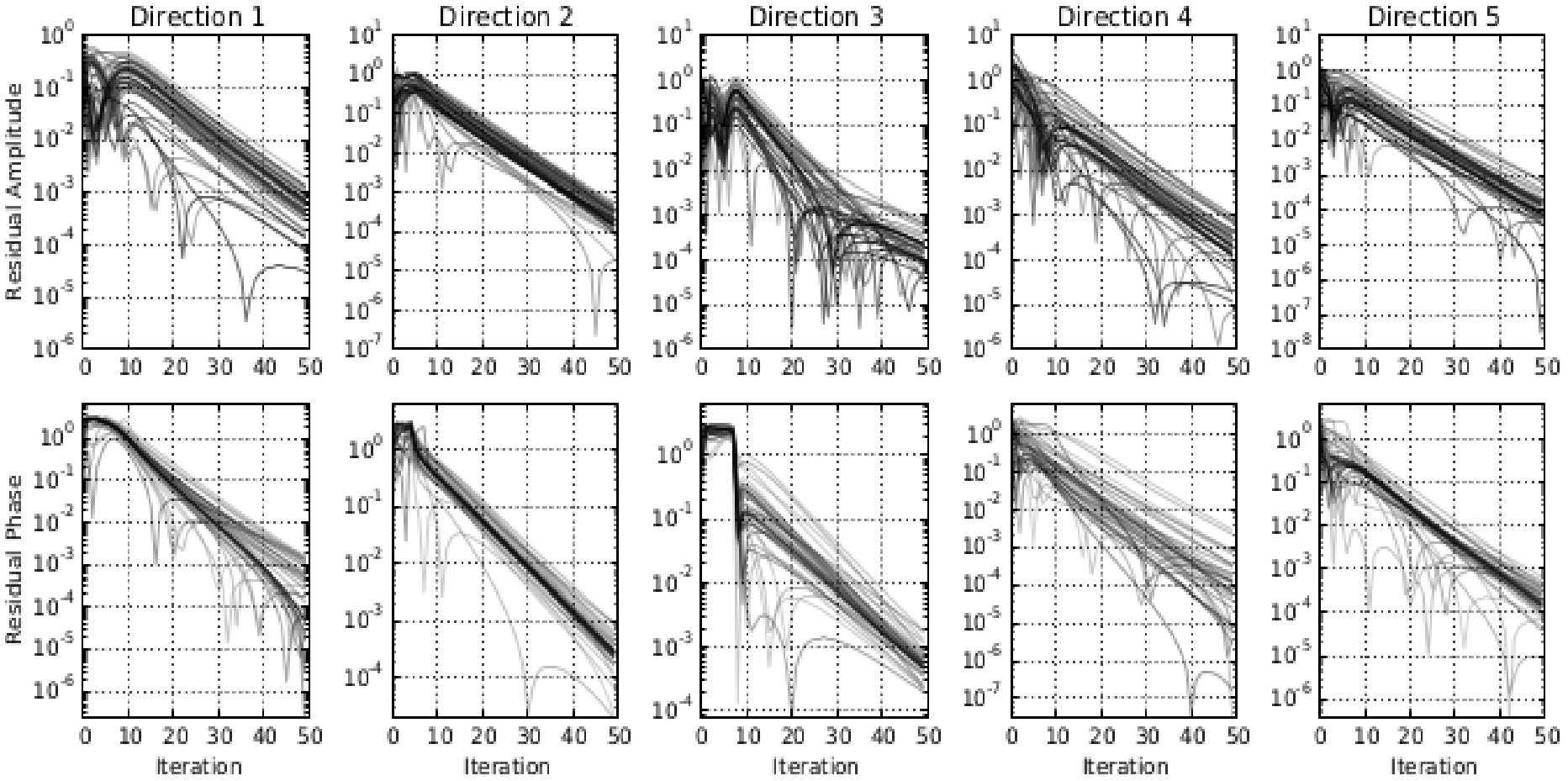
Those blocks have a small amplitude

Let's look at NL optimisation algos



Tests

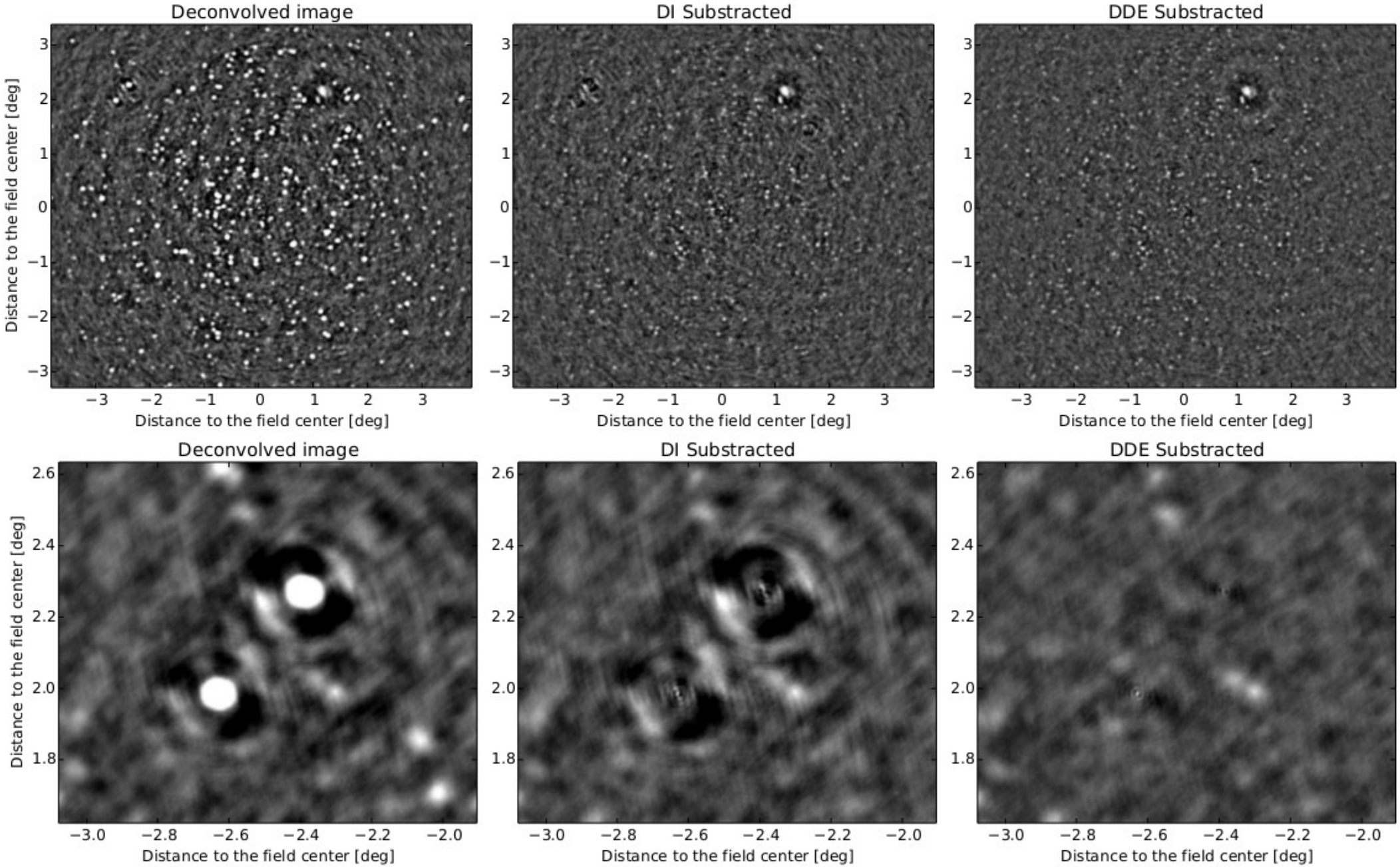
- Simulation



Convergence

Tests

- 3C295 field



Softwares?

[C + Python] KillMS

- a toolbox to do different suff, using the **Wirtinger Jacobian trick**

- KillMSv1 (Public):
  - Scalar CohJones only

- KillMSv2 (Not public – unpublished):
  - fullPol or Scalar
  - **CohJones** (LM) or **KAFCA** (Kalman filter)
  - Better parallelisation (faster, less memory)

Softwares?

[C + Python] KillMS

- a toolbox to do different suff, using the **Wirtinger Jacobian trick**

- KillMSv1 (Public):
  - Scalar CohJones only

- KillMSv2 (Not public – unpublished):
  - fullPol or Scalar
  - **CohJones** (LM) or **KAFCA** (Kalman filter)
  - Better parallelisation (faster, less memory)

DDFacet [C + Python]

- A new born baby imager
- tightly connected top killMS
- Takes **full-pol DD** solutions and apply them while deconvolving

How do you apply DDE

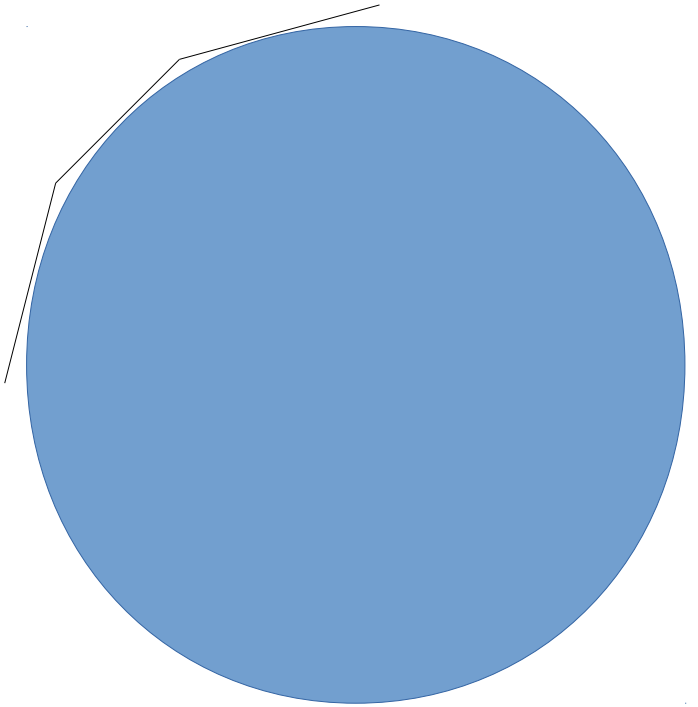
- A-Projection

  Continuous DDE

- Facets

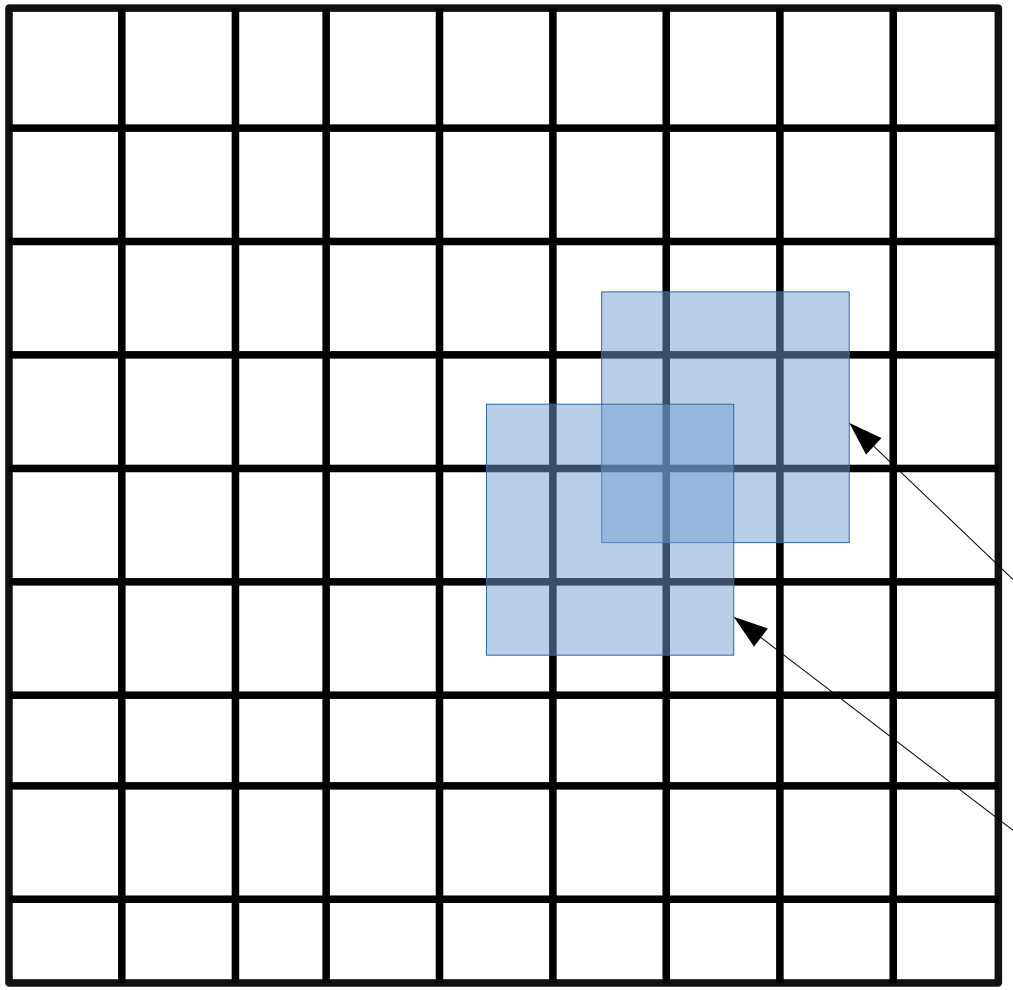
  - Discrete DDE

  - Facets are individually tengenting the sphere

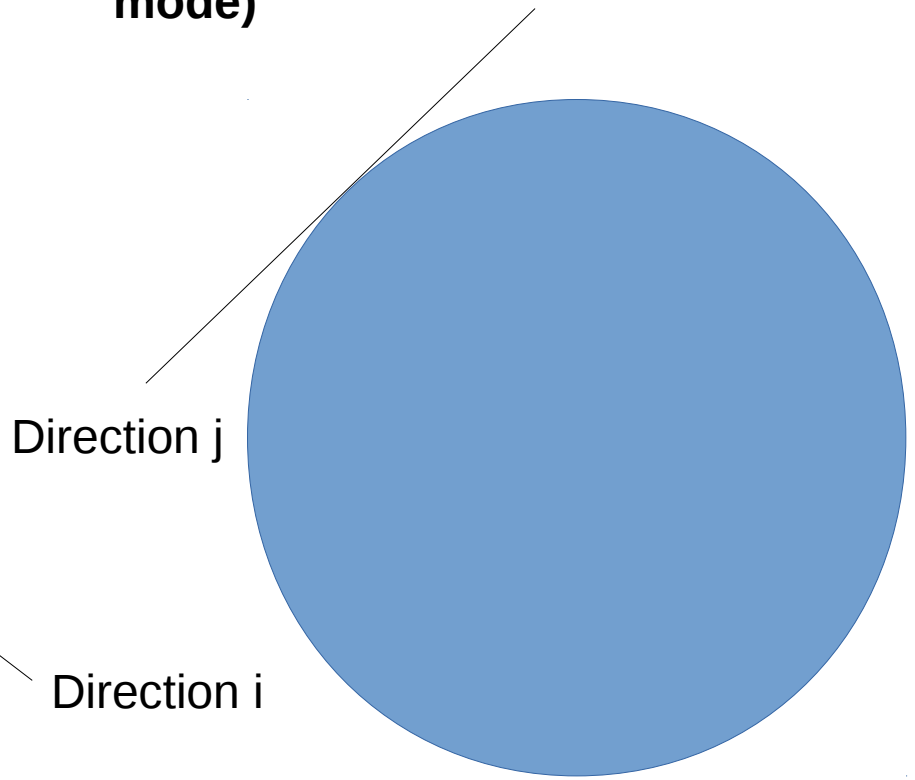


DDFacet (a baby imager)

- DDFacet is the imager associated to killMS

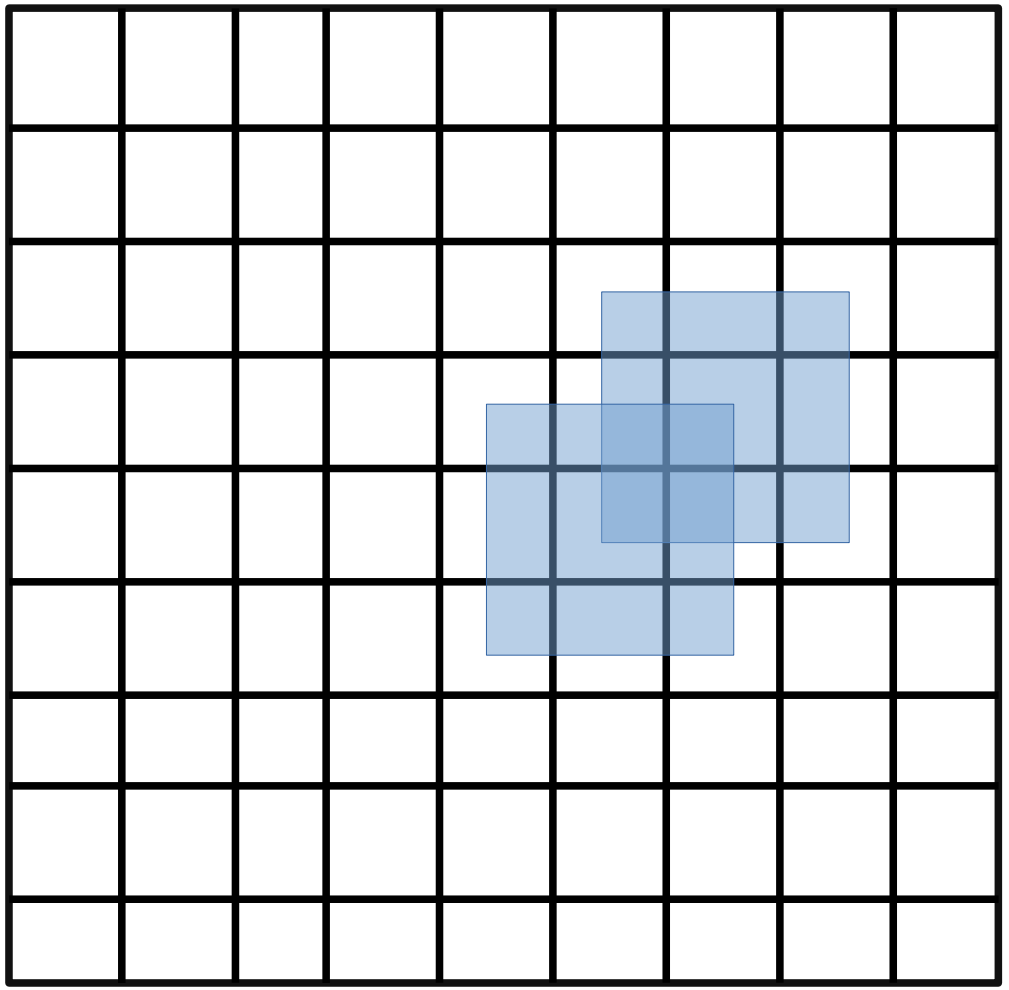


(1) Produces a single tengential plane ! (no « noise jumps » thanks to the kalman filter, and facetting mode)



DDFacet (a baby imager)

- DDFacet is the imager associated to killMS



(1) Produces a single tengential plane ! (no « noise jumps » thanks to the kalman filter, and facetting mode)

(2) Does full polarisation DDE correction

(3) Baseline Dependent Averaging
90 % of the data can be compressed
(collaboration with O.Smirnov and M. Atemkeng)

BUT
(4) Need to interpolate DDE (if drawn from Voronoi tesselation)

Bootes field (not selfcaled)

Precalibrated by Wendy
(when visiting Reinout ?)

Using losoto (Gasperin et al.)
- Clock correction
- offset

KillMS v2 setting :
- dt=1 minute
- 10 subbands (2 chan/SB, 8 sec timebin, 8 hours)
- Scalar cal
- using Kalman filter
- 36 directions
- 5.000 sources in the SkyModel

Computing time :
~40 minutes / 10 subbands (32 threads)

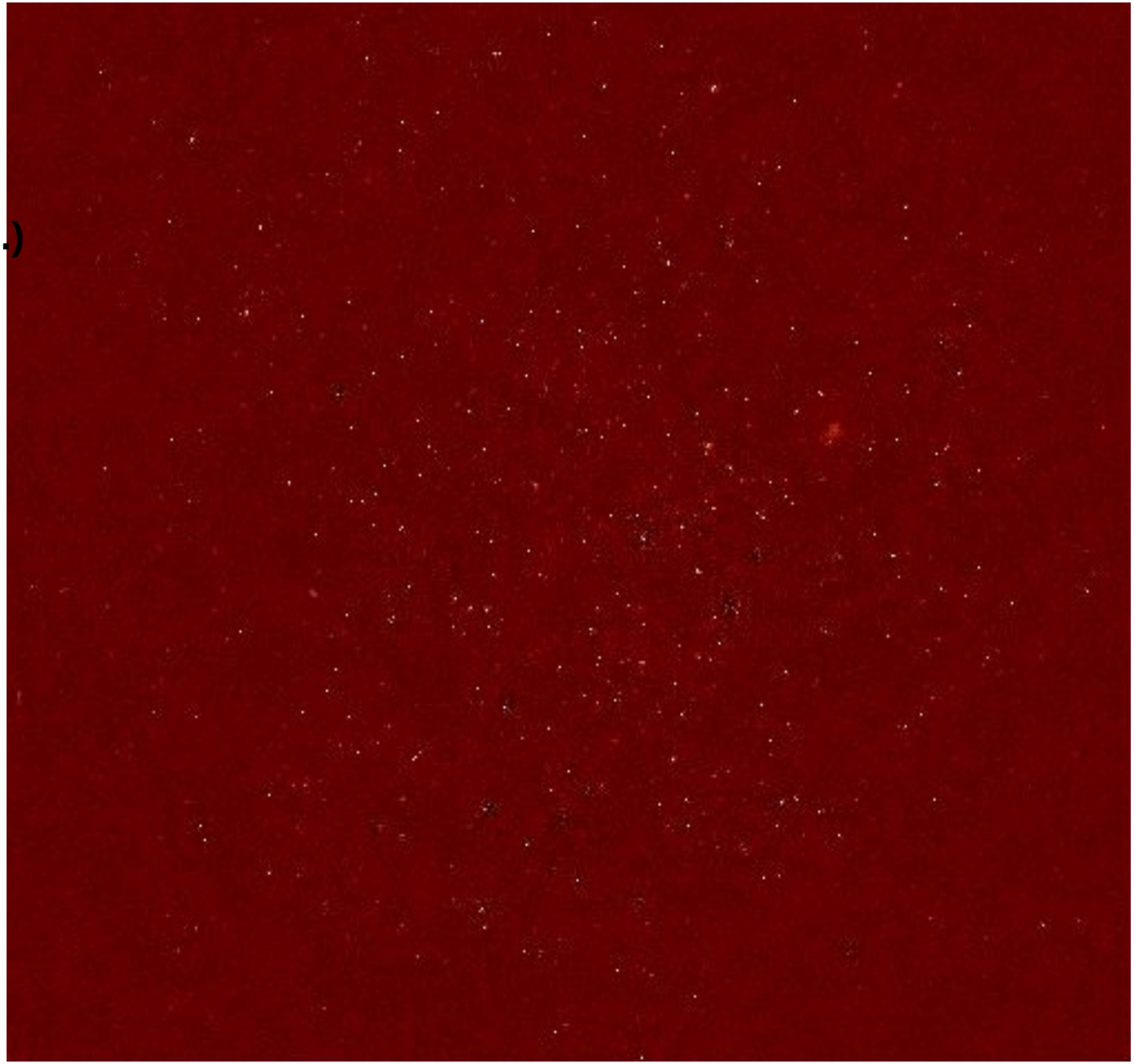
Bootes field (not selfcaled)

Precalibrated by Wendy (when visiting Reinout ?)

Using losoto (Gasperin et al.)
- Clock correction
- offset

DDFacet result :
- 9 degrees
- 121 facets
- 16.000 pixels
- 2''/pixel
- 50 subbands
- ~310 uJy rms

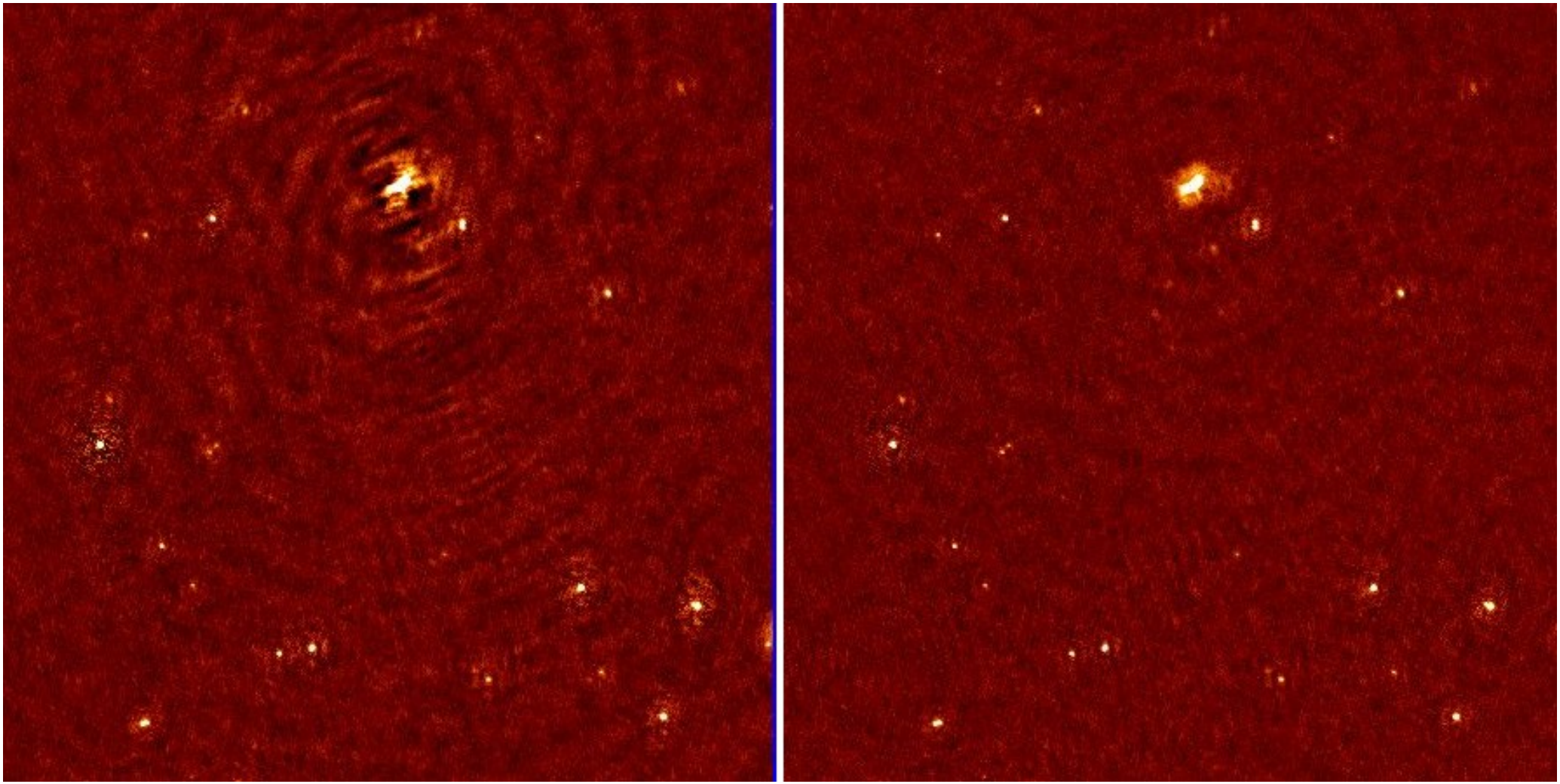
Computing time :
6 hours
(10 major cycles)



Bootes field (not selfcaled)

DDE correction : WITHOUT

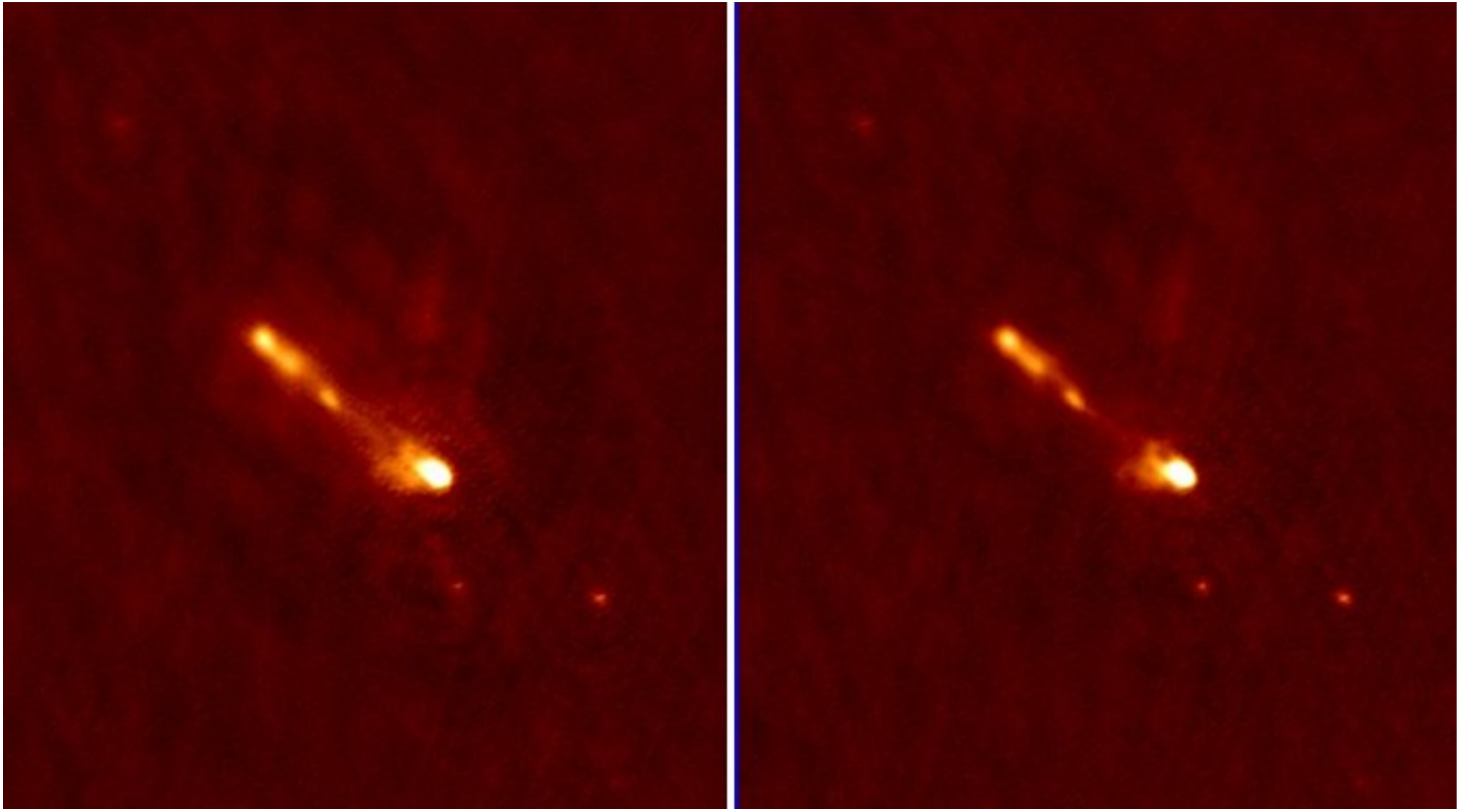
DDE correction : WITH



Bootes field (not selfcaled)

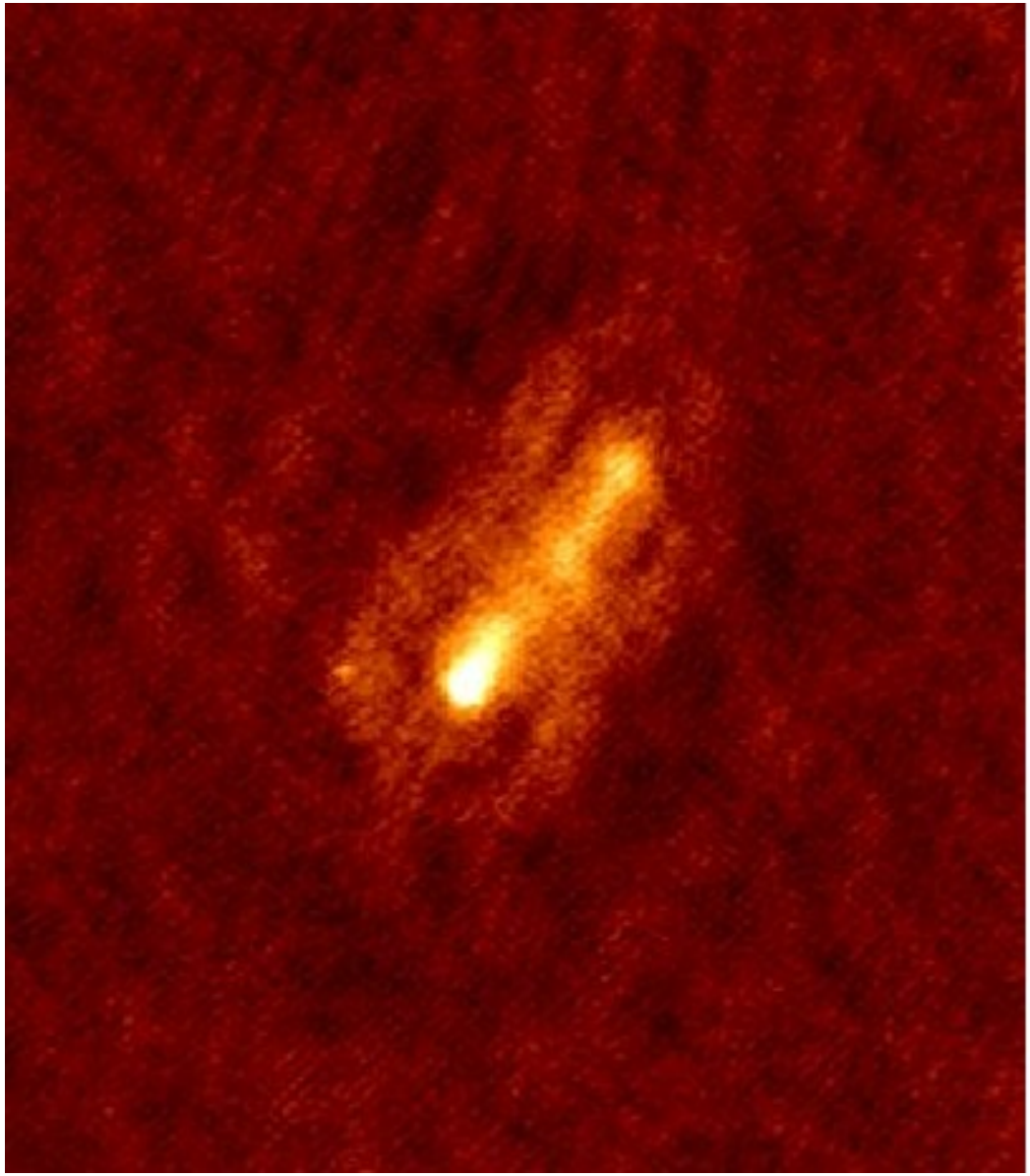
DDE correction : WITHOUT

DDE correction : WITH

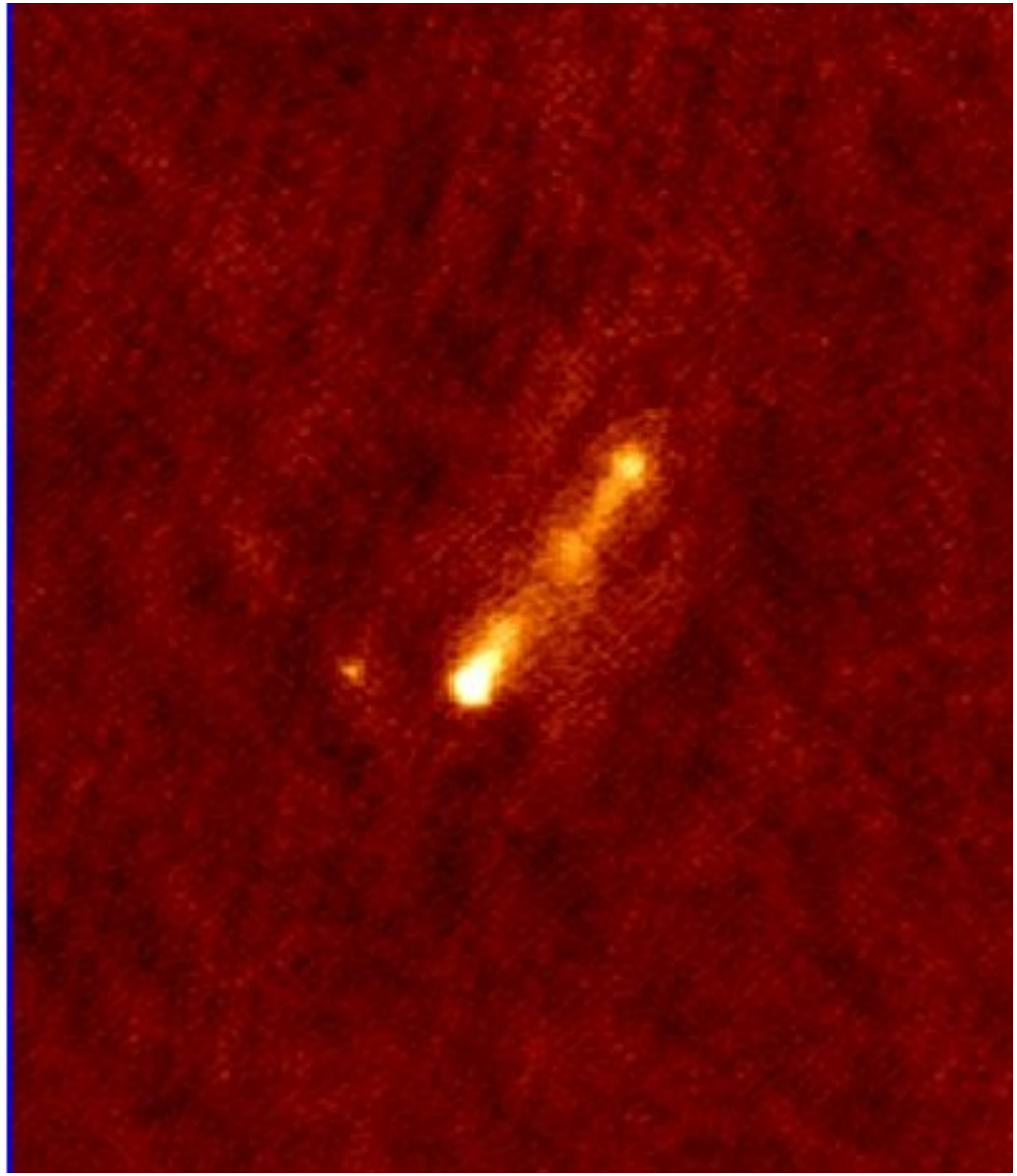


Bootes field (not selfcaled)

DDE correction : WITHOUT

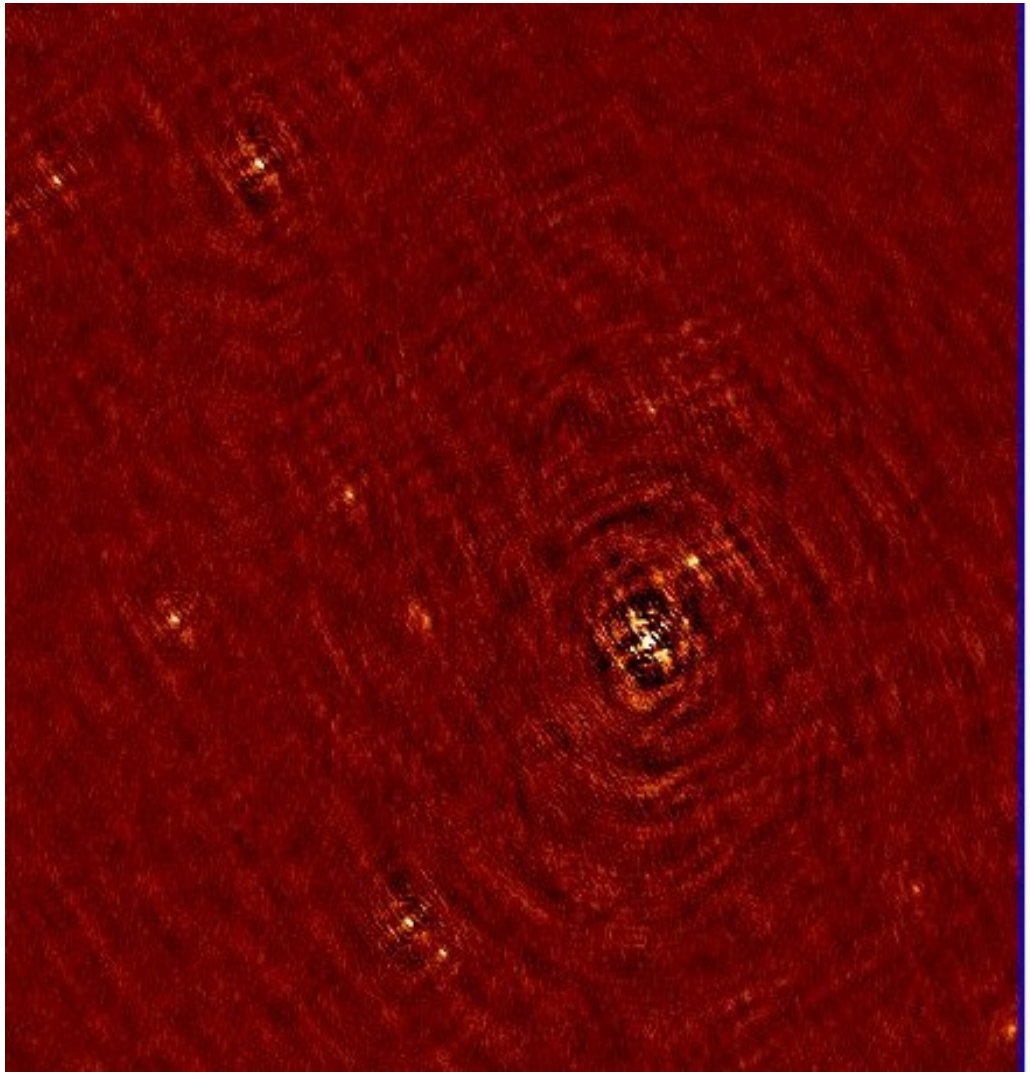


DDE correction : WITH

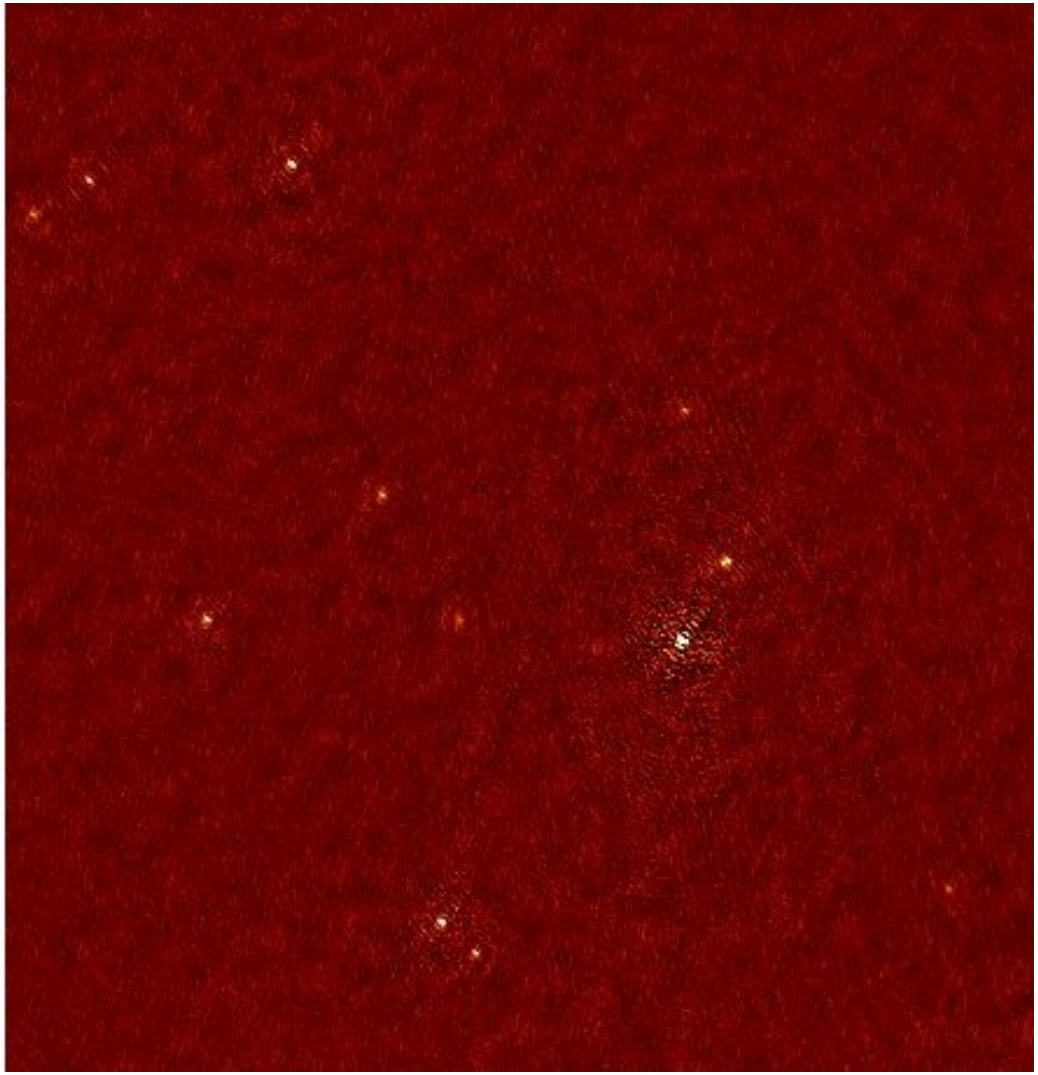


Bootes field (not selfcaled)

DDE correction : WITHOUT

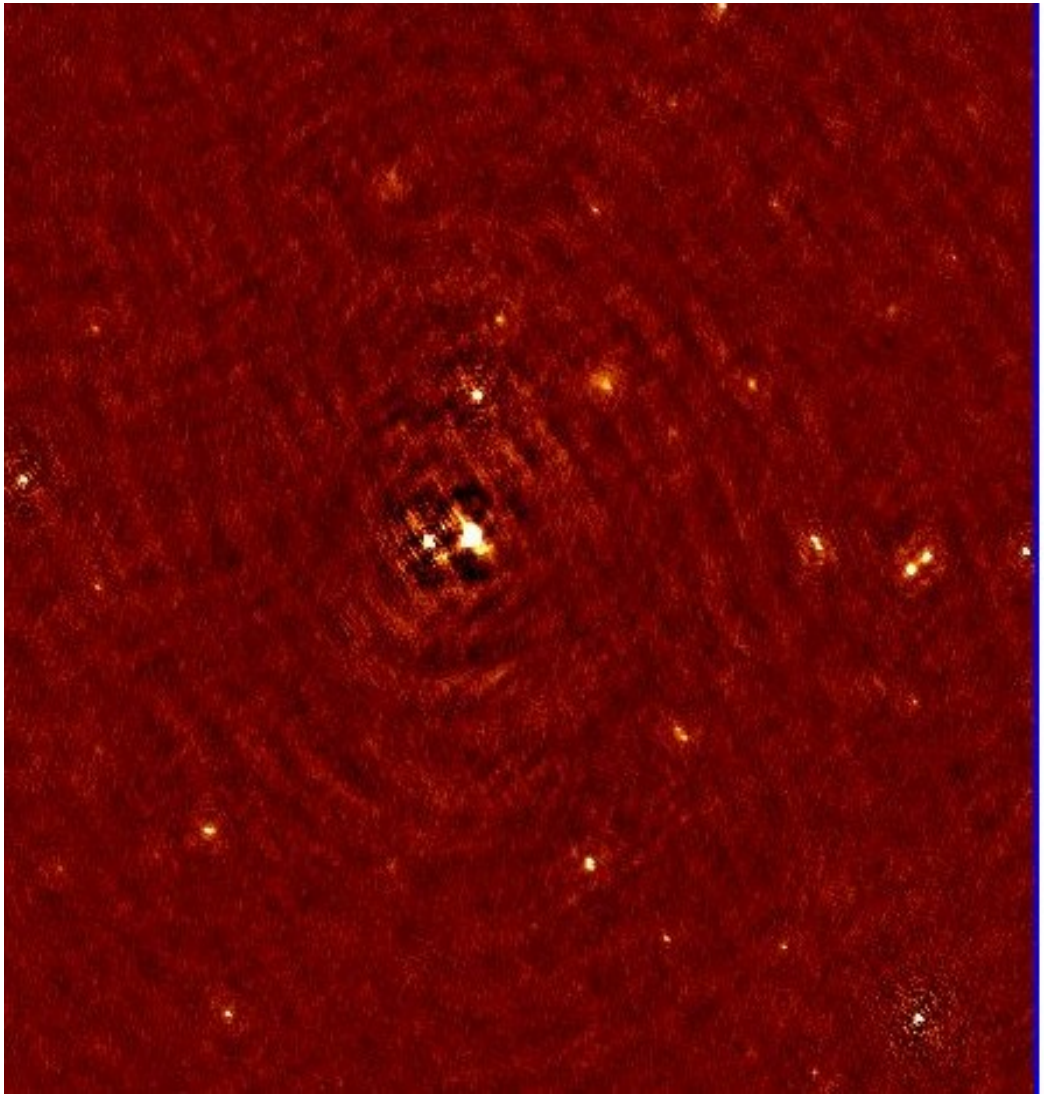


DDE correction : WITH

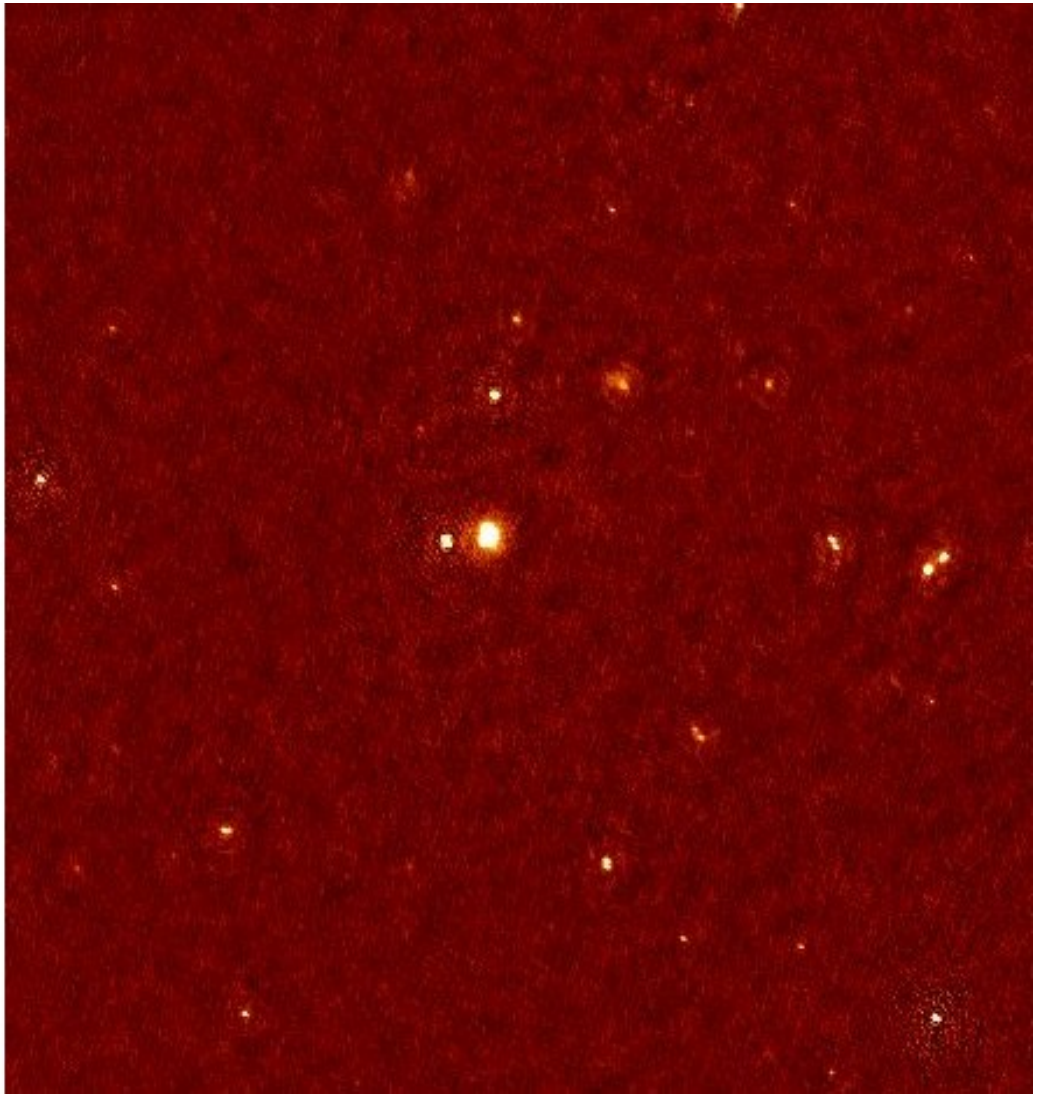


Bootes field (not selfcaled)

DDE correction : WITHOUT



DDE correction : WITH



What's now?

Before self-calibration

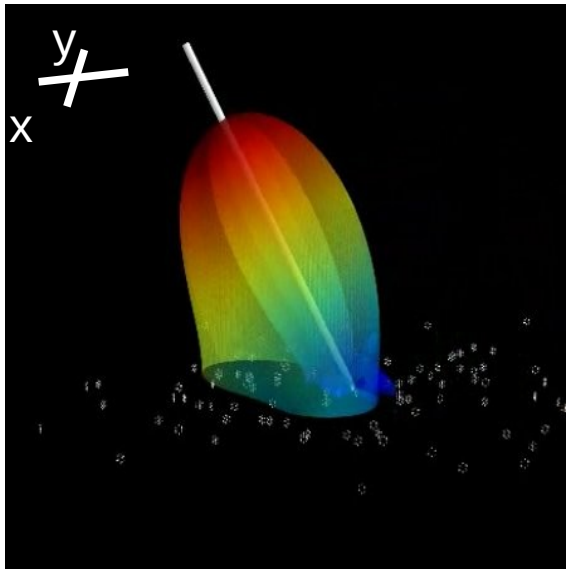
- Wideband clean

- Etimate flux density

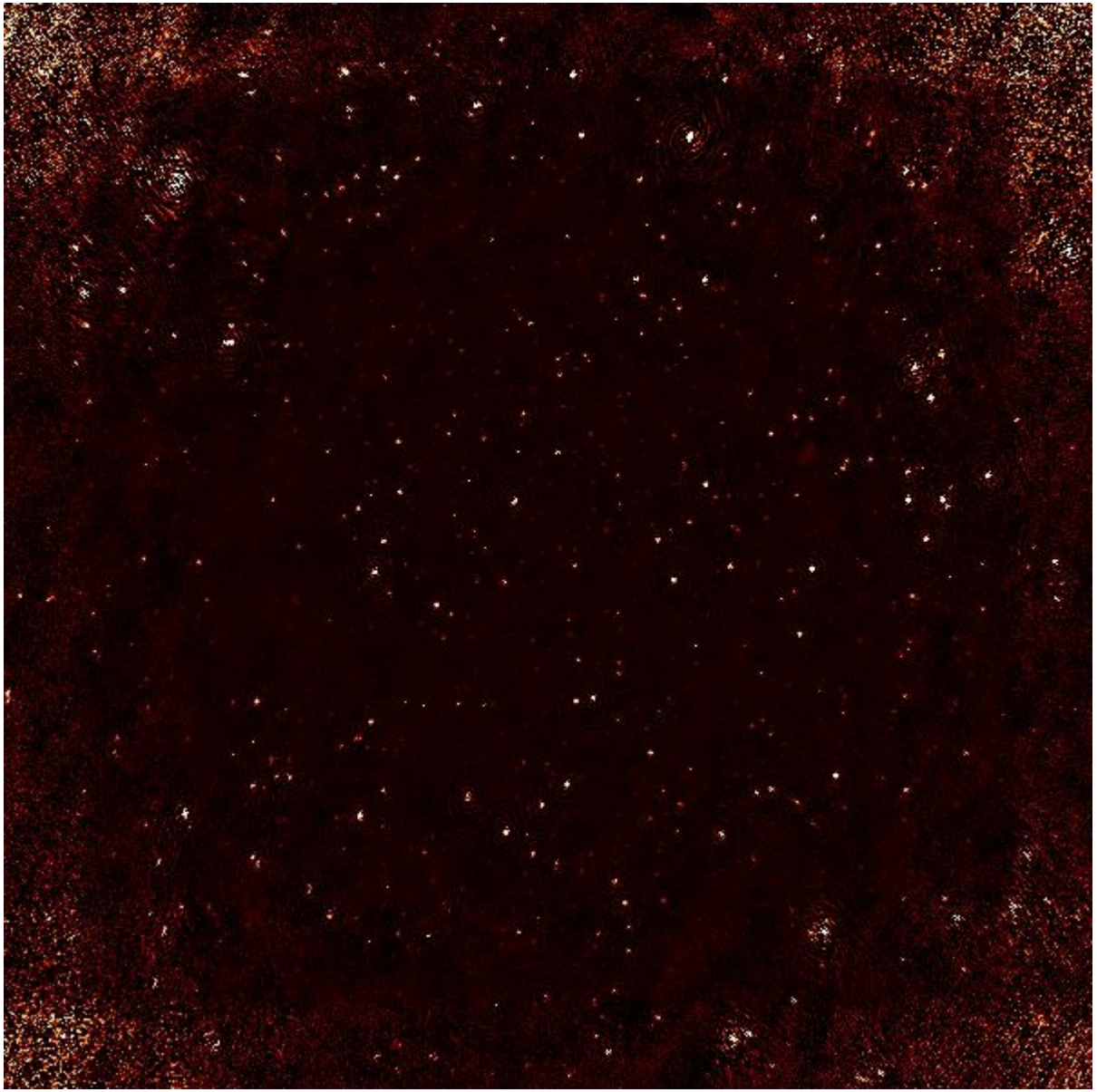
- Spectral indices

- MultiScale

- Beam correction



Here the image is corrected from the LOFAR beam



Conclusion

- Compared to facet based calibration (« extreme pealing »)

  - No pealing here. Simultaneous DDE calibration (again, read aforementioned papers)

  - selfcal has to be done on the whole FOV (ie not per facet)

- The whole approach is quite fast (one selfcal cycle in 10-11 hours for 50 subbands – 32 threads) and ~20.000x20.000 pixels image

- **TODO:**
  - Have wide-band CLEAN working (if not working the whole approcah is useless)
  - Go in selfcal loop with killMS/DDF
  - Try full-pol killMS alternative to demixing

- **@SKA Office**
  - Marcellin Atemkeng adding Pseudo PSF capabilities (smearing)
  - Jon Kenyon working on generalising Wirtinger Jacobian to pointing errors calibration
  - Benna Hugo will add GPU gridder

- To be continued….