S&C Visibility Processing

Overview

- Visibility processing constructs images from output of correlator
- Very compute intensive
  – Scales as number of samples
- Memory and memory bandwidth intensive
  – Low computational intensity
  – Images can be large e.g. 12k x 12k x 12k
- Algorithms for SKA context still evolving

Measurement equations

- Unlimited field of view

$$V(u,v,w)=\int \frac{A(l,m)I(l,m)}{\sqrt{1-l^2-m^2}}e^{j2\pi\left(ul+vm+w\sqrt{1-l^2-m^2}\right)}dldm$$

- Small field of view

$$V(u,v)=\int I(l,m)e^{j2\pi(ul+vm)}dldm$$

Approaches to visibility processing

1. Offline manual (most common)
2. Offline asynchronous pipeline
3. LOFAR model
   - quasi-synchronous centralised calibration, distributed imaging and science analysis
4. ASKAP
   - quasi-synchronous centralised calibration, imaging and science analysis
5. MeerKAT
   - quasi-synchronous centralised calibration and imaging science

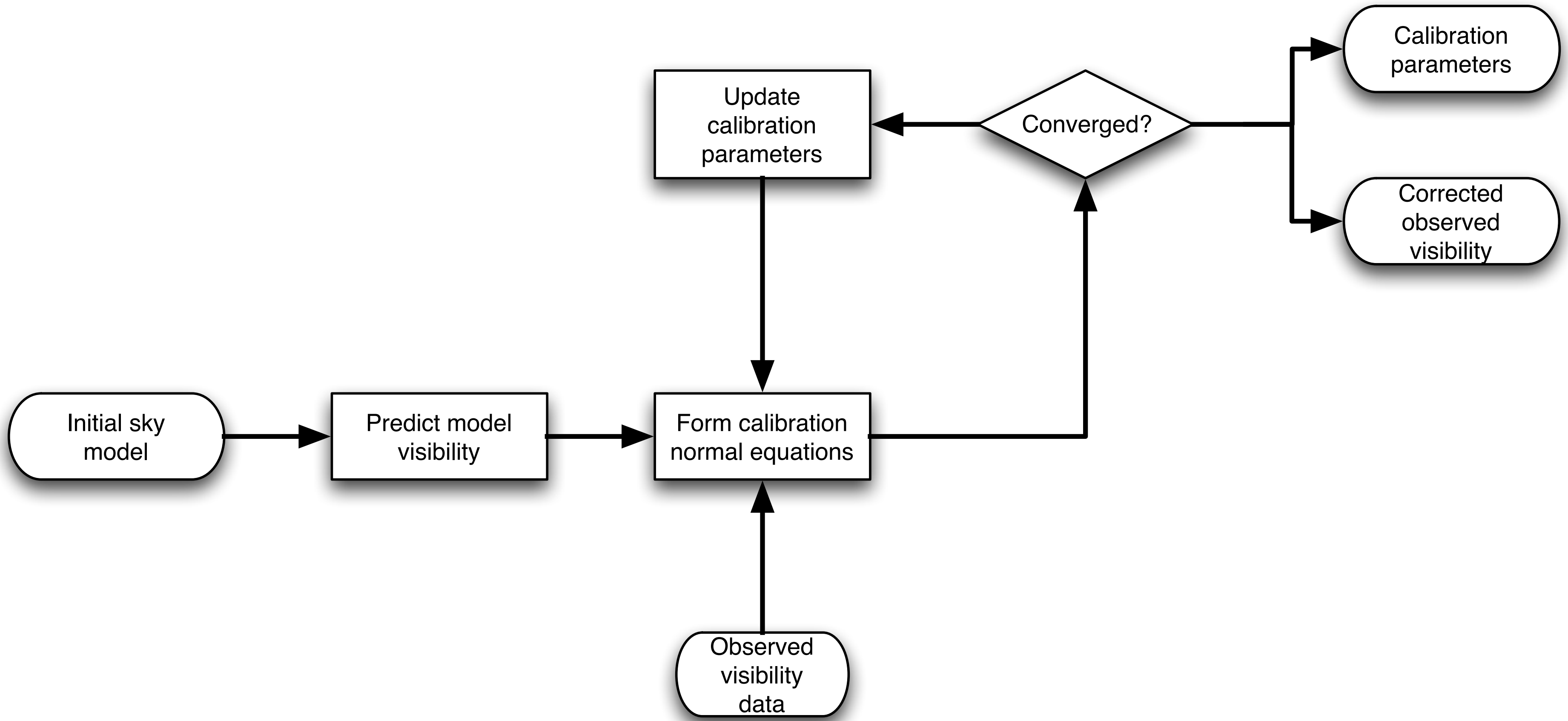
Steps in processing

- Ingest from correlator
- Flag known bad data (e.g. RFI)
- Flag unknown bad data (e.g. new RFI)
- Calibrate
- Flag unknown bad data (e.g. weak RFI)
- Image
- Science processing e.g. find and fit sources

Calibration

- Observe known signal
  - e.g. bright point source
  - e.g. complex field (Local Sky Model)
- Fit calibration parameters
- Correct visibilities for calibration parameters
- Can iterate - self-calibration

Iterative loop



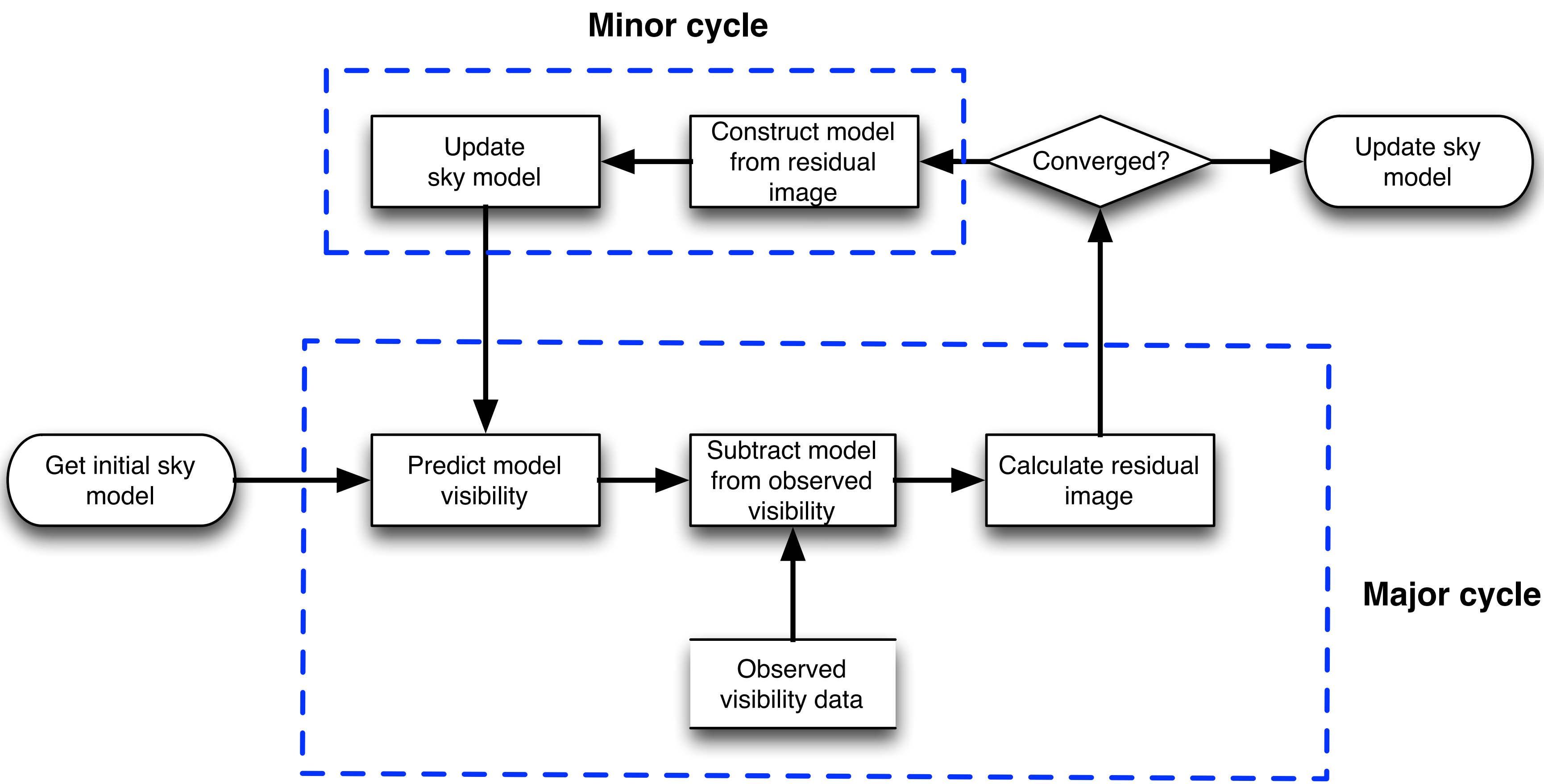
Direction-dependent effects

- Troublesome for wide field of view
- Examples
  - Ionosphere non-isoplanatic at low frequencies
  - Antenna primary beams
  - Antenna pointing
- Needed for high dynamic range imaging i.e. 70dB
- Difficult problem
  - Computationally expensive
  - Software and algorithms hard
  - Still under development
- LOFAR has pioneered this area
  - BBS (Black Board Selfcal)
  - MeqTrees

Imaging

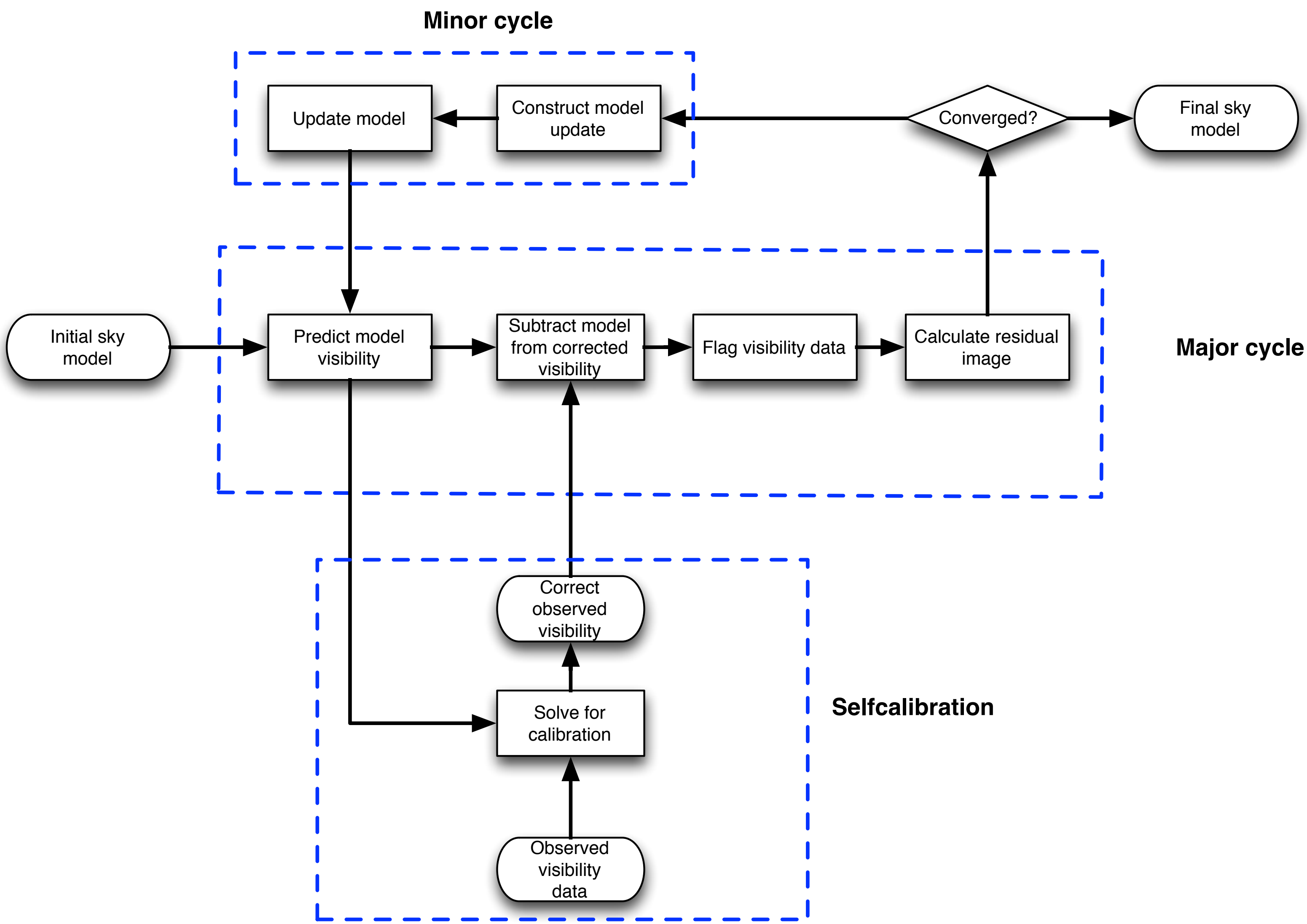
- Small fields of view: ME is invertible
  - Can derive sky brightness from inverse Fourier transform
  - Still need to deconvolve point spread function
- Large fields of view: ME is not invertible
  - Solve by iteration
  - Predict from model exactly
  - Calculate update to model approximately
  - Nearly always needed for SKA

Imaging loop



Exploring the Universe with the world's largest radio telescope

Imaging and calibration loop

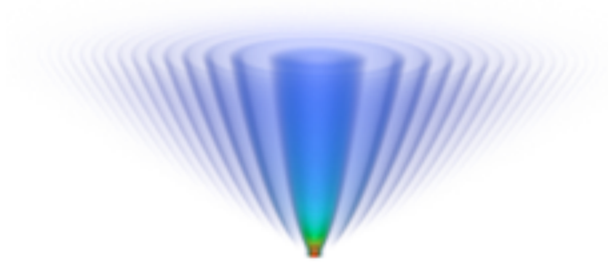


Wide field imaging

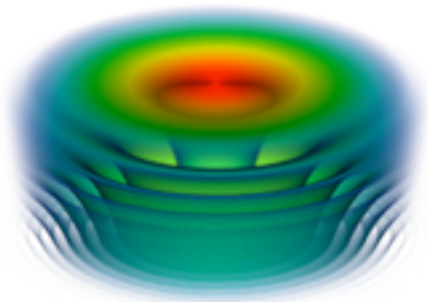
- Numerous algorithms
  - Full three dimensional transform
  - Divide image plane in small facets for which the two dimensional transform is accurate
  - Partition data into parallactic angle bins for which the array can be treated as coplanar
  - Convolution of the visibility samples by the Fresnel term prior to or after the Fourier transform.
- Each approach has merits
- No single approach optimal for all cases

Wide field imaging

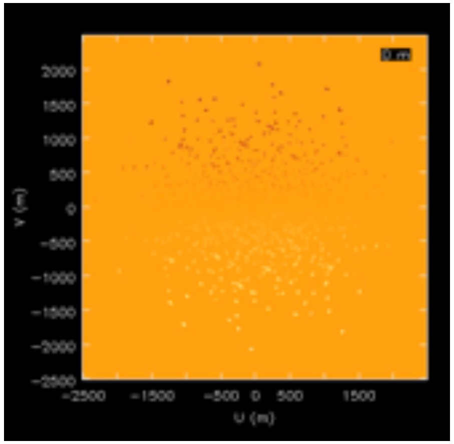
$$V(u,v,w) = \int \left[\frac{I(l,m) e^{j2\pi w\left(\sqrt{1-l^2-m^2}-1\right)}}{\sqrt{1-l^2-m^2}} \right] e^{j2\pi(ul+vm)} dldm$$



Convolution in data space

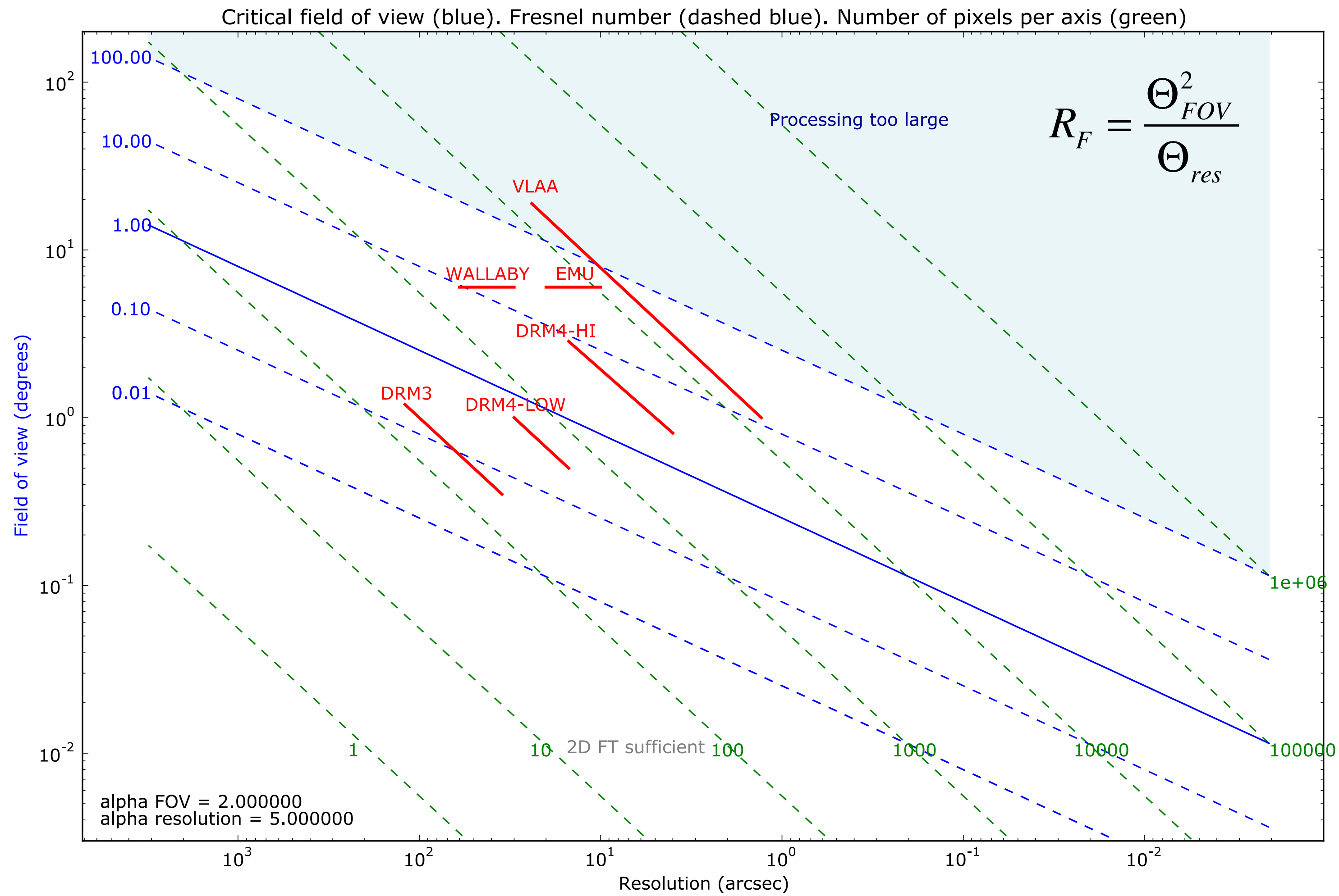


Multiplication in image space



Slices in data space

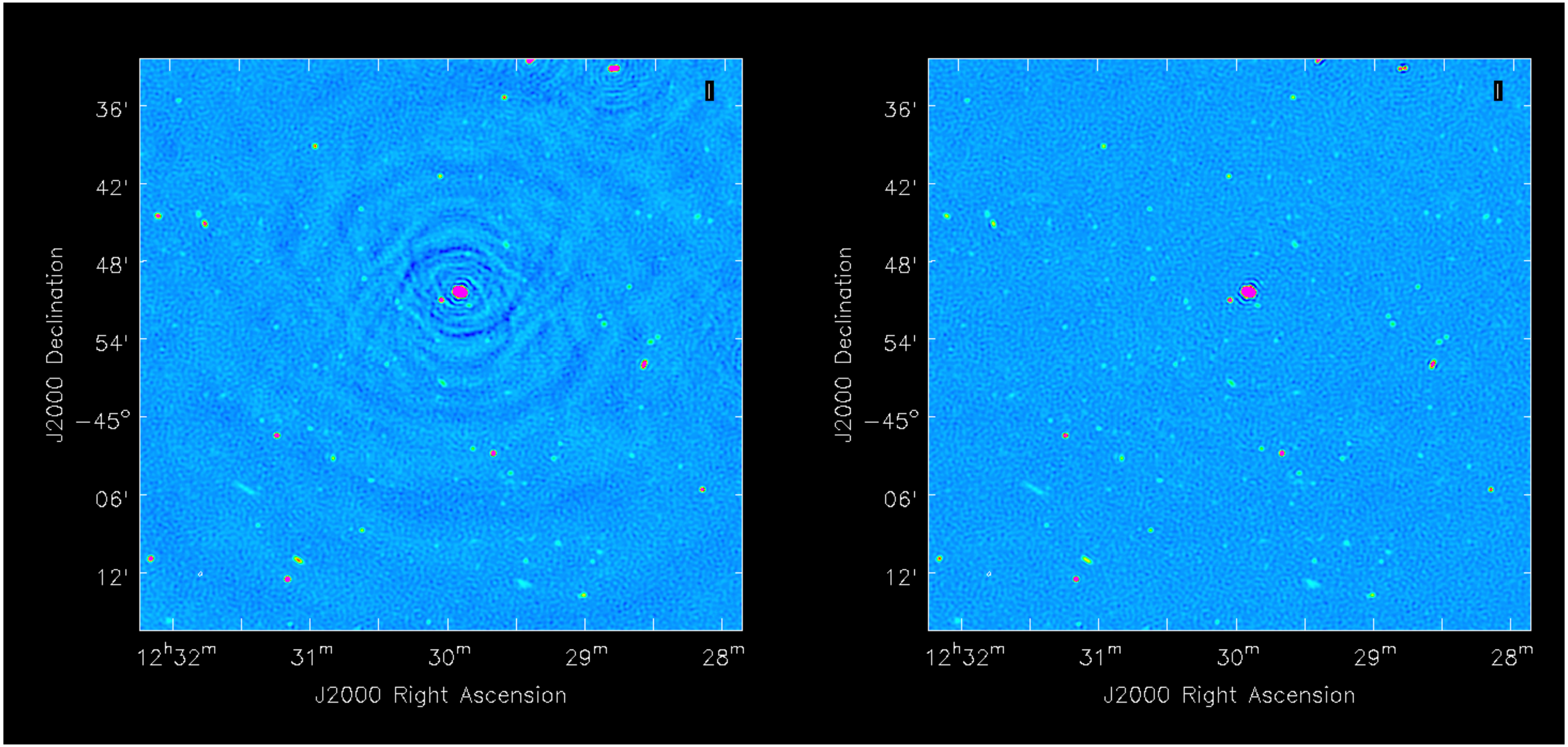
How wide is wide?



Wide band imaging

- Sources change brightness with frequency
- Must correct for high dynamic range imaging
- Best algorithm is Multi-Scale Multi-Frequency Clean
  - Memory intensive
- Compressive sampling also possible
- Likely area of more research

Example of MSMFClean



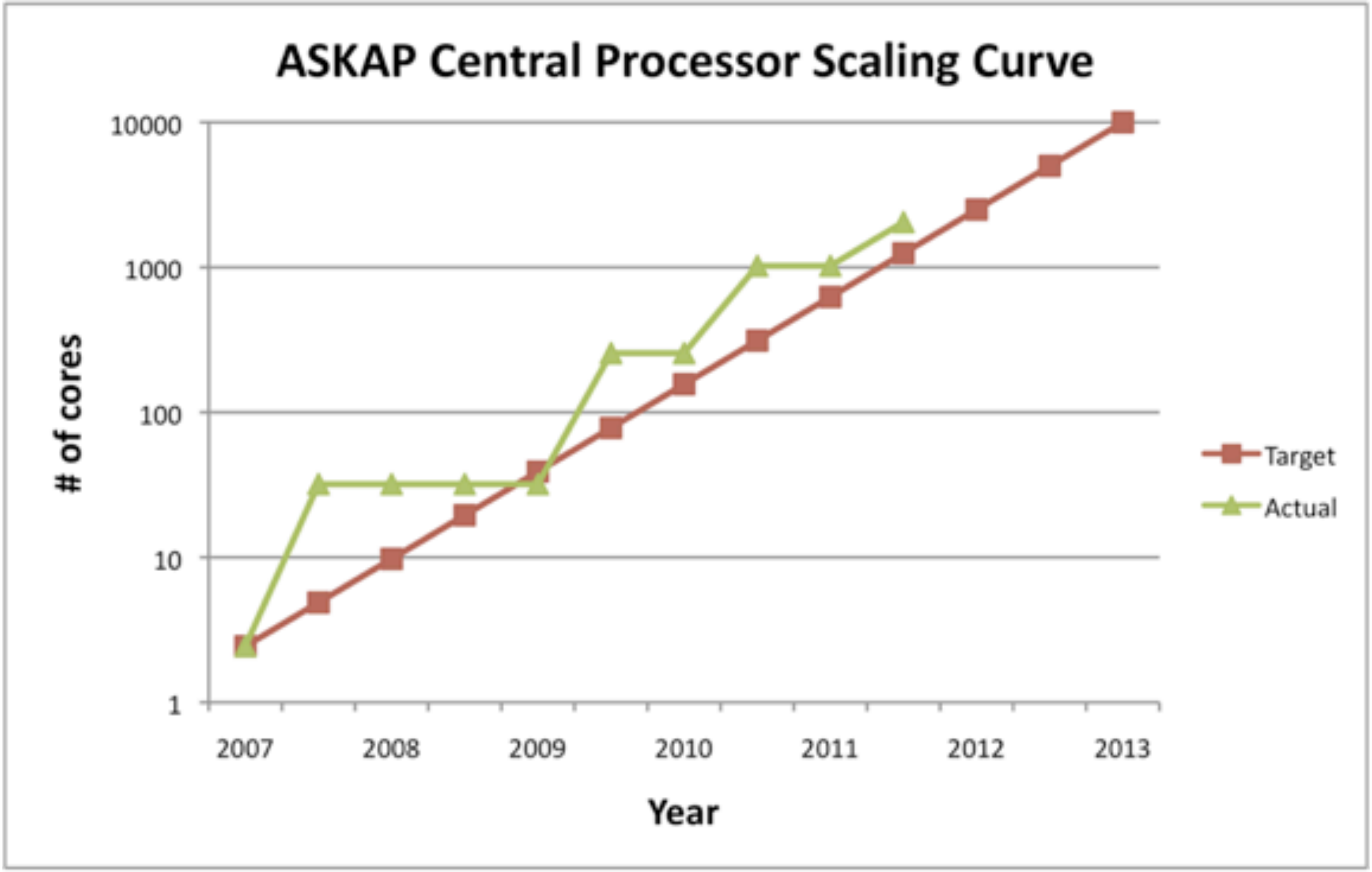
Implementation

- ASKAP/LOFAR
  - C++ with 1 MPI process per core
  - Memory per core ~ 2 - 4 GB is insufficient
  - Shift soon to 1 MPI process per node
  - Eventually 100’s or 1000’s cores per node
- Software platforms
  - 3rd party libraries (e.g. casacore) are OO and not thread safe
  - C++ not optimum for HPC
  - I/O model must be revised for MPI Parallel IO or e.g. Lustre
- Biggest area of change

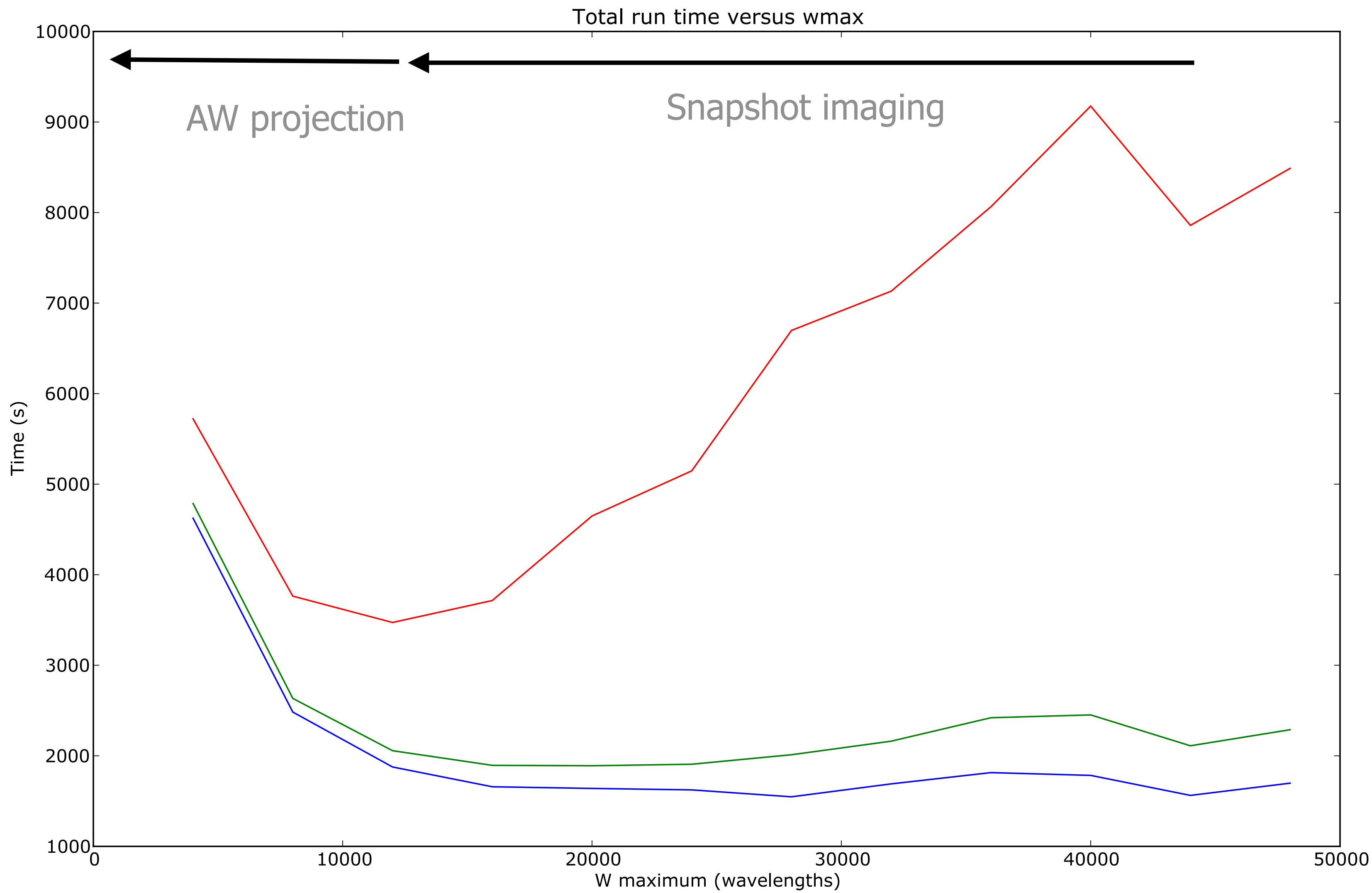
Changes in imaging during scaling work

- **AWProject** (2007)
  - W projection + A projection (for primary beam)
  - Too much CPU
  - Too much memory for convolution function
- **AProjectWStack** (2008)
  - Apply W term in image space
  - Much less CPU
  - Too much memory for w stack
- **AWProject + trimmed convolution function** (2009)
  - Only apply and keep non-zero part of convolution function
  - Still too much memory for convolution function
- **AWProject + trimmed convolution function + multiple snapshot planes** (2011)
  - Fit and remove w=au+bv plane every 30 - 60 min
  - Small memory for convolution function
- **Serialise normal equations piece-by-piece for MPI** (2011)
  - Cuts down short bump in memory use

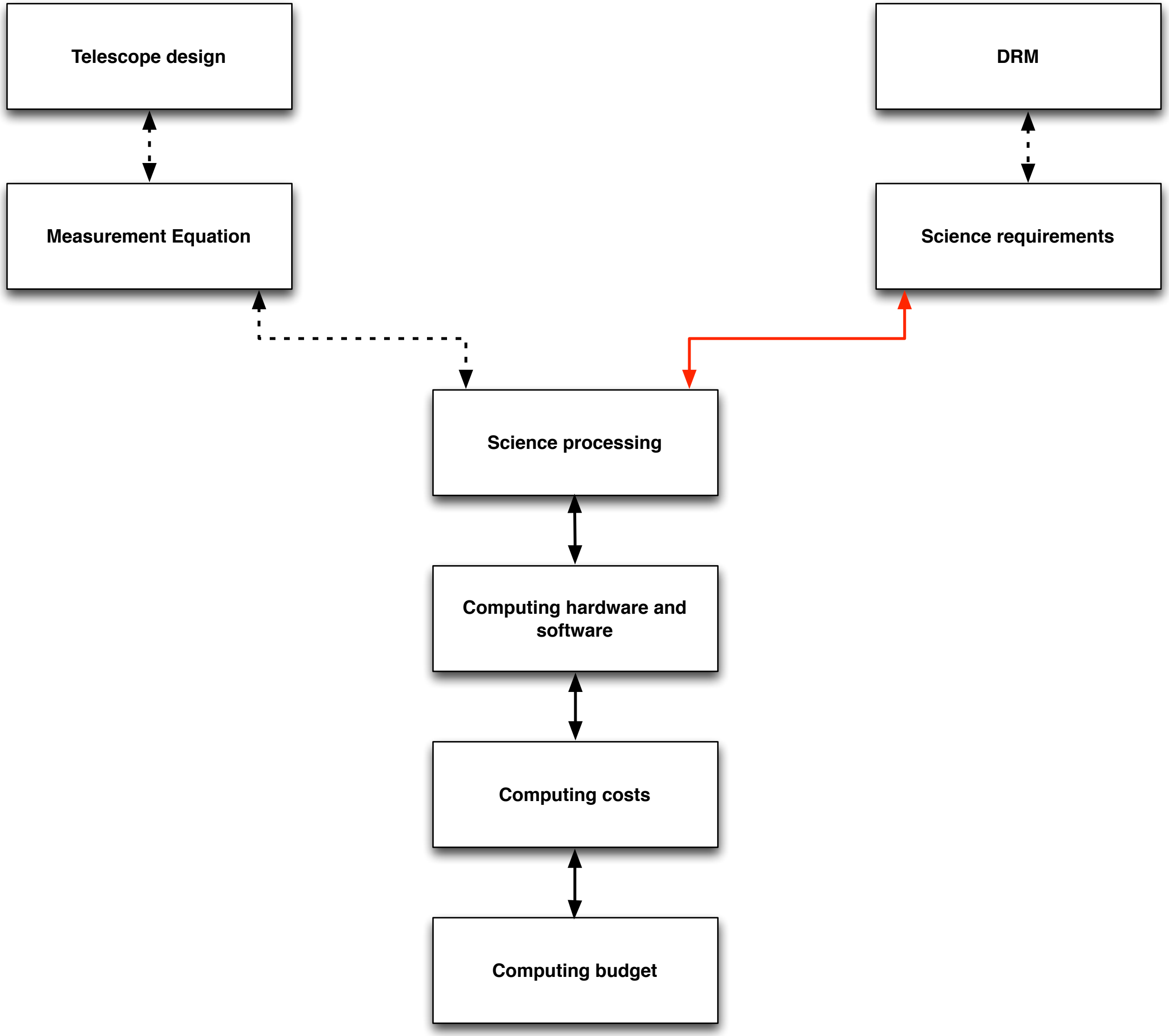
Scaling over time



Tuning of hybrid algorithms



Computing costs



rgest radio telescope

Scaling of data rate

- Data rate very dependent on allowed smearing, antenna diameter, baseline length

What	Symbol	Plausible range	Samples per second	Processing per sample
Amplitude loss due to time averaging	ΔA_{time}	0.001 - 0.05	$\Delta A_{time}^{-1/2}$	-
Amplitude loss due to frequency averaging	ΔA_{freq}	0.001 - 0.05	$\Delta A_{freq}^{-1/2}$	-
Antenna diameter	D	30m - 80m	D^{-2}	D^{-4}
Maximum baseline length	B	2 km - 100km	B^{2}	B^{2}

Example scaling test

- Dish array (with SPF), 8 hour continuum integration.
- The number of channels and the integration time were each chosen to scale as baseline length and to fill Fourier space.
- 48 MPI processes were used in these runs.
- The last column shows the number of MPI processes required to finish a major cycle in one hour.
- Remember that up to 10 major cycles may be required, and another factor of three comes from the necessity to perform multi-frequency synthesis.

Configuration	Data size GB	Major cycle min	Total memory GB	Processes required
1km	88	18	111	7
5km	297	80	115	32
10km	1100	3000	192	240

Computing budget

- Current estimates 10‘s to 100‘s PF
- Usually:
  - Fix science, determine computing cost
- In SKA
  - Fix computing budget, determine science
- How?
  - Consider other costs e.g. power, software development
  - No interest in having fastest machine
  - Spend e.g. $20M - $30M

Algorithm development

- Hire sufficient people for critical path development
  - CALIM shows a new generation of algorithm specialists becoming available
- Data challenges
- Maintain CALIM meetings
  - Focus more on SKA and HPC

Summary

- Scaling to Pflops
  – Algorithms will have to change
  – Software development required
- Scientific performance
  – Reaching very high dynamic range will require experience with telescope
- Coupling science to computational scale is very difficult
  – Unlikely to be able to be definitive