European Research Council
Established by the European Commission

Map-based beam treatment

A new approach for Simons Observatory analysis pipeline

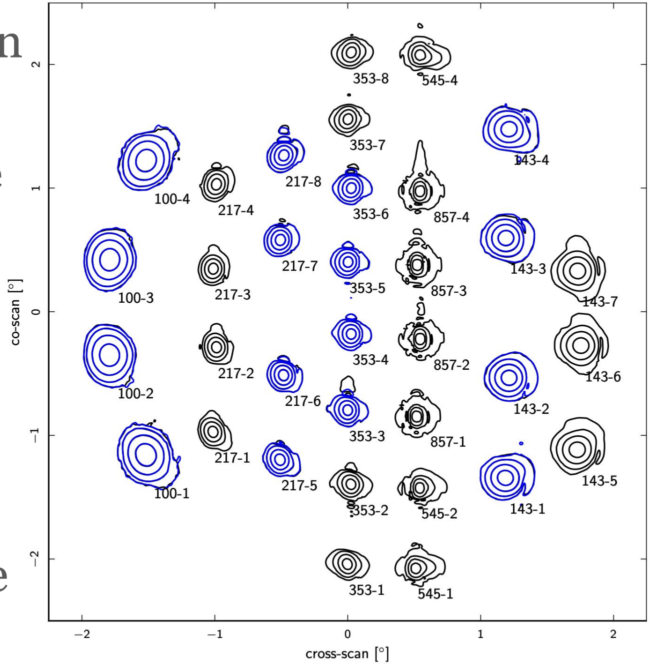
Artem Basyrov, APC, CNRS

Traditional approach

- Beam is a fundamental systematic effect present in any observations
- Traditionally they are treated in a harmonic space

$$c_\ell = b_\ell^2 \cdot c_\ell^0$$

- It creates problems with asymmetrical beams requiring complex ℓ and m coupling
- It is non-trivial to deal with the polarization mixing (beam asymmetry, coordinate dependance of Stokes Q and U parameters, partial sky coverage)
- It is non-trivial to deal with sidelobes effects



Planck Collaboration, 2014

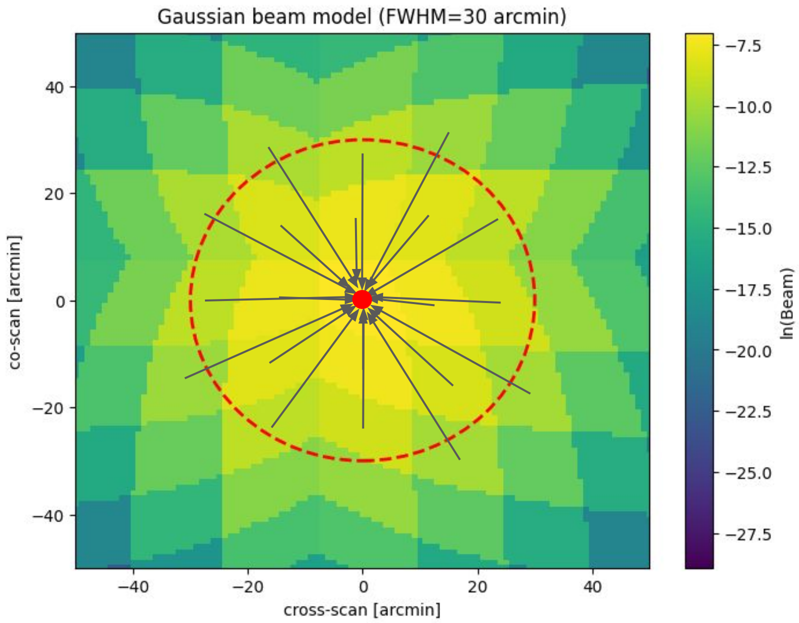
Our solution: think in map space!

Core Idea:
The beam convolution is a local operation. The observed value at a pixel is a weighted sum of the true sky values in nearby pixels

The Operator (see FURAX talk):
We represent the beam as a **sparse matrix** B, with size [**npix, npix**]

The observation equation becomes a simple matrix multiplication:

$$d_{\text{observed}} = B \cdot d_{\text{true}}$$

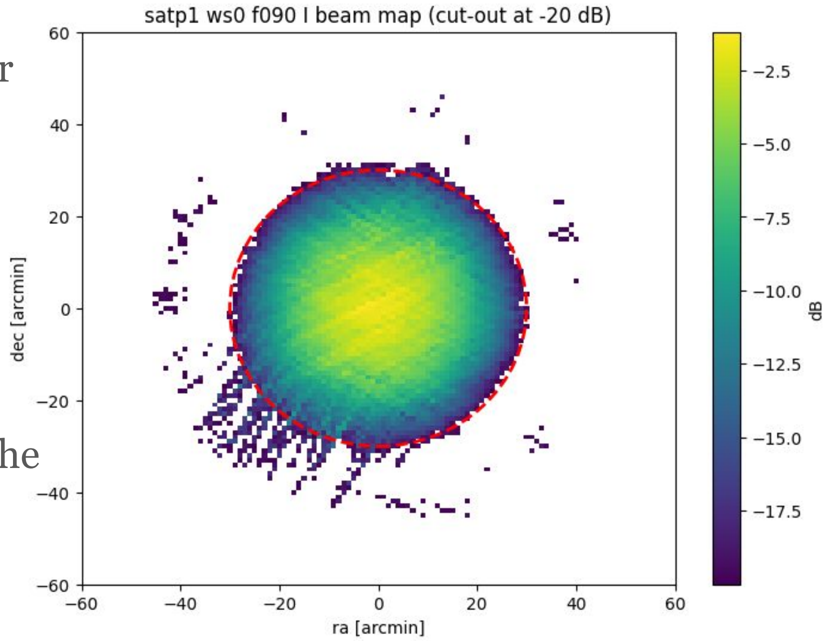


Advantages of the map space approach

- Allows to handle any beam shape. Perfect for non-spherical beams from physical optics models or measured data
- Easy sidelobes treatment. Sidelobes are just non-local connections in the matrix

$$B_{\text{full}} = B_{\text{main}} + B_{\text{sl}}$$

- Straightforward Polarization handling. *I, Q, U* are treated independently or coupled naturally within the matrix structure
- Seamless integration into map-based analysis pipelines



The computational engine: JAX and FURAX

- **JAX:** Provides accelerated linear algebra (CPU/GPU/TPU) and automatic differentiation.
- **FURAX:** A Python library for building and composing linear operators for cosmological data analysis. It allows us to treat *B* not just as a matrix, but as a high-level, composable *linear operator*.

This enables powerful operations like *B.T @ B* or solving *B.I * d* using conjugate gradient, all without manually handling the sparse matrix.

```
from furax.obs import BeamOperatorMapspace, read_beam_matrix
B = read_beam_matrix(in_structure=d.structure,
                     path_to_file='sparse_beam_matrix.npz')

# convolve data with the beam
d_conv = B @ d

# deconvolve data with the beam
d_deconv = B.I @ d_conv
```

https://github.com/CMBSciPol/furax

The basic building block: Beam Operator

$$\begin{pmatrix} 5 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 7 \end{pmatrix}$$

sparse matrix

per frequency
per detector
per Stokes comp
→ **BeamOperator** B

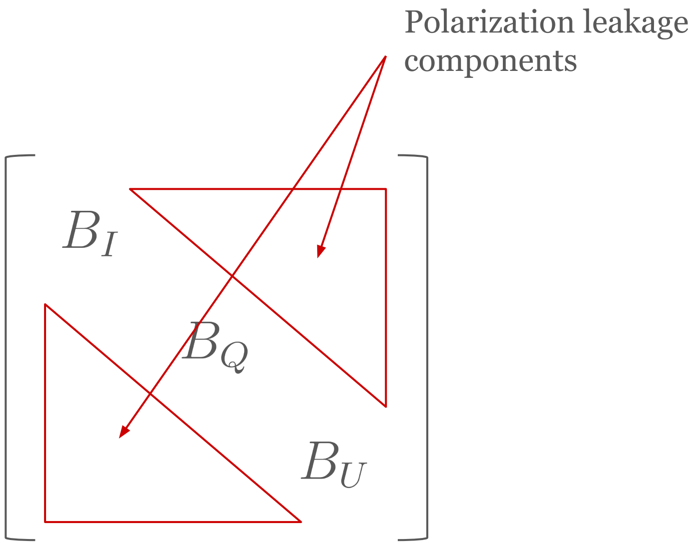
The basic building block: Beam Operator

$$\begin{pmatrix} 5 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 7 \end{pmatrix}$$

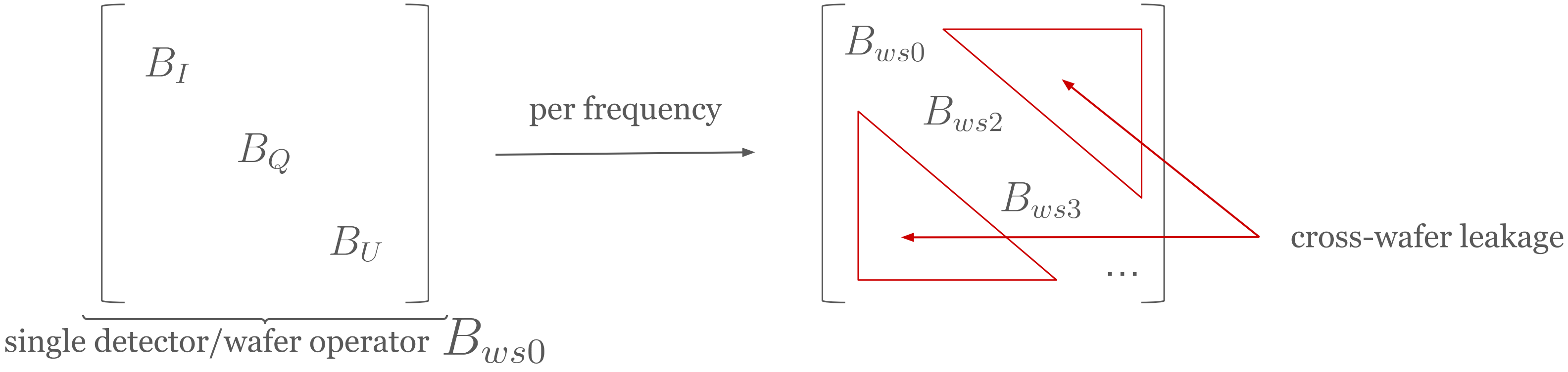
sparse matrix

per frequency
per detector
per Stokes comp
→ **BeamOperator** B

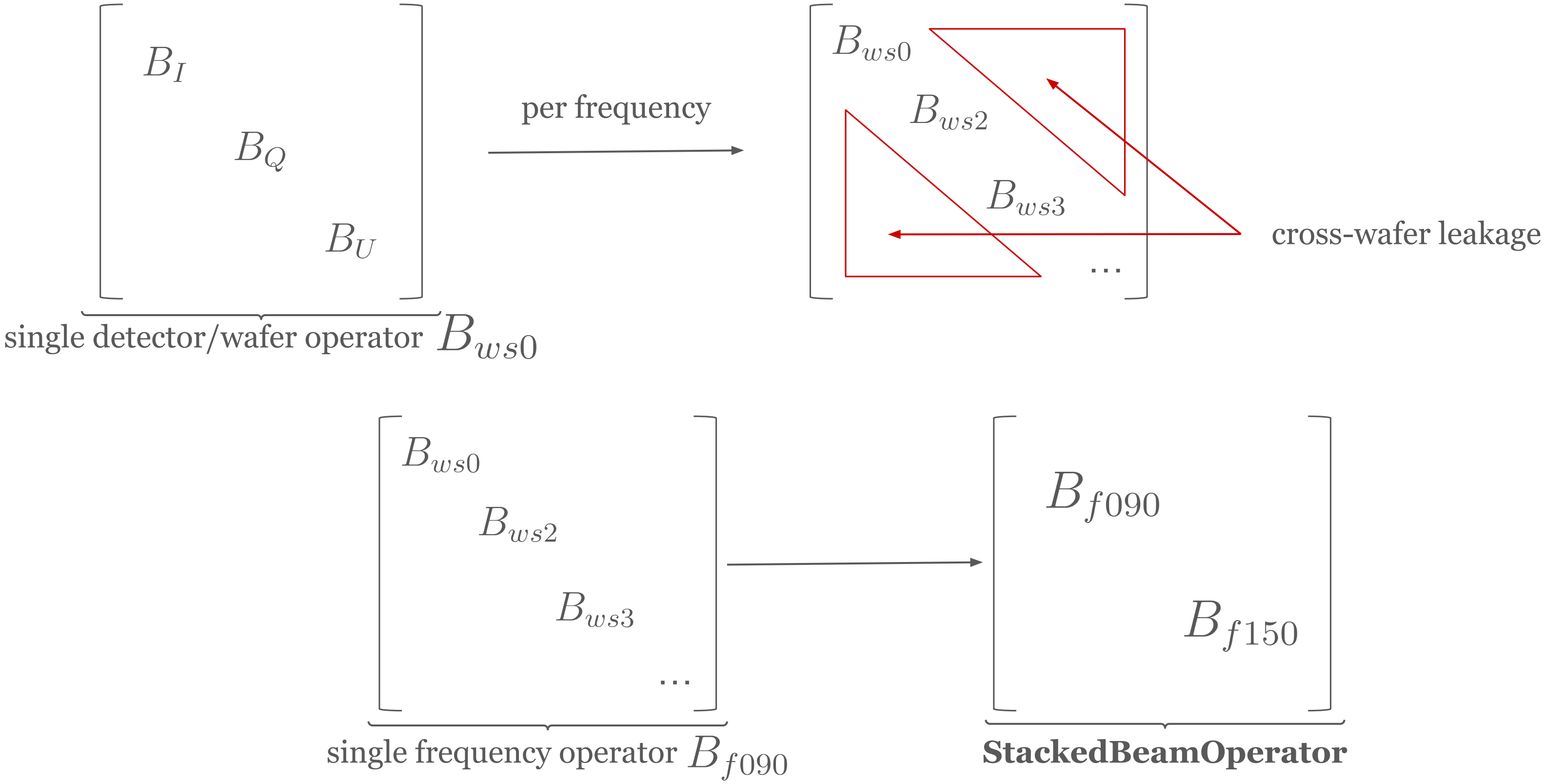
BeamOperator B → per frequency
per detector



Handling Multi-Frequency Data



Handling Multi-Frequency Data



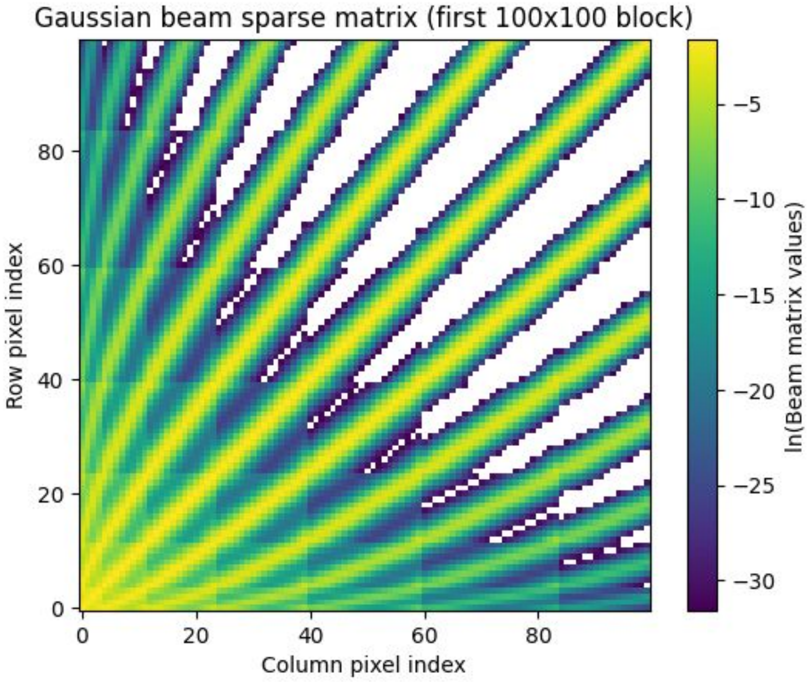
Creating a beam matrix

- Signal conservation:

$$\sum_i B_{ij} = 1,\ \forall j$$

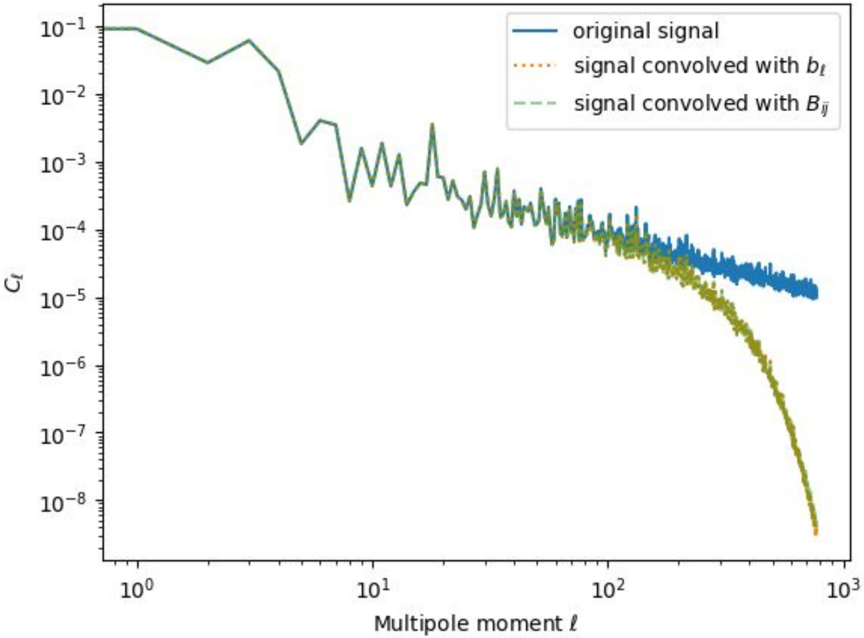
- Ensuring the same number of *active* non-zero elements in each row for JAX acceleration
- Functionality to create a beam from a functional representation:

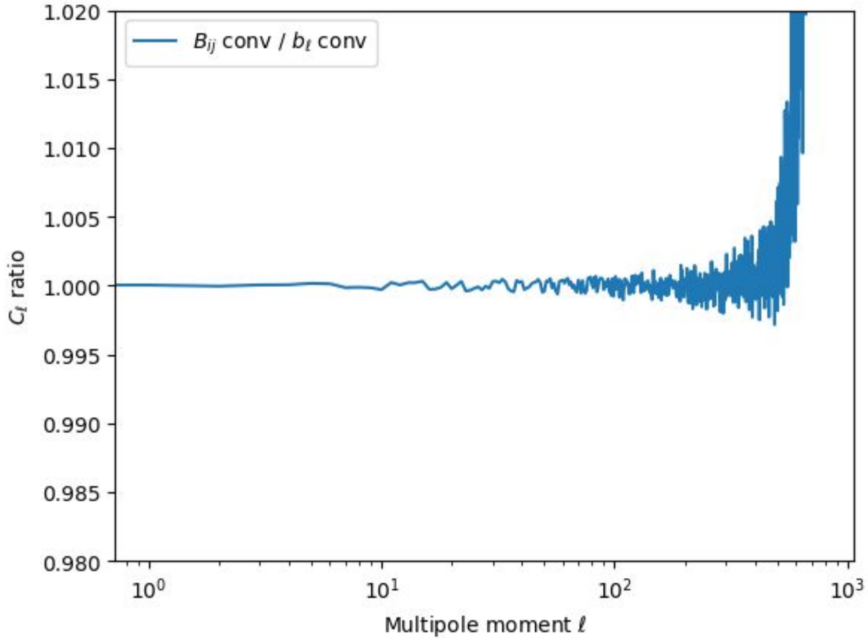
$$B(\theta_{ij}) = \exp\left(-\frac{\theta_{ij}^2}{2\sigma^2}\right)$$



Creating a beam matrix

$C_\ell \sim \ell^{-2}$
$N_{\rm side} = 256$
$\mathrm{FWHM} = 30\,\mathrm{arcmin}$

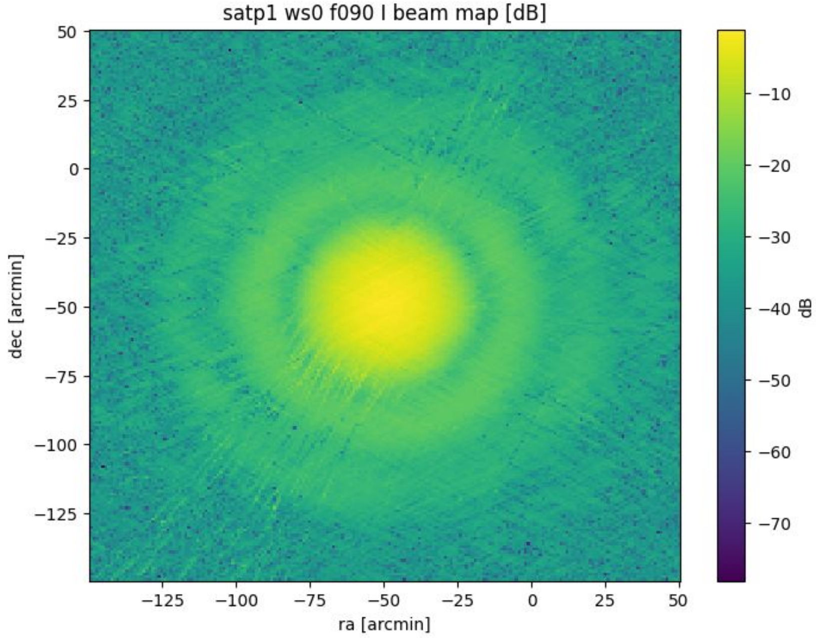




Comparison of a beam matrix approach to spherical harmonics approach for the simple case of a symmetric Gaussian beam

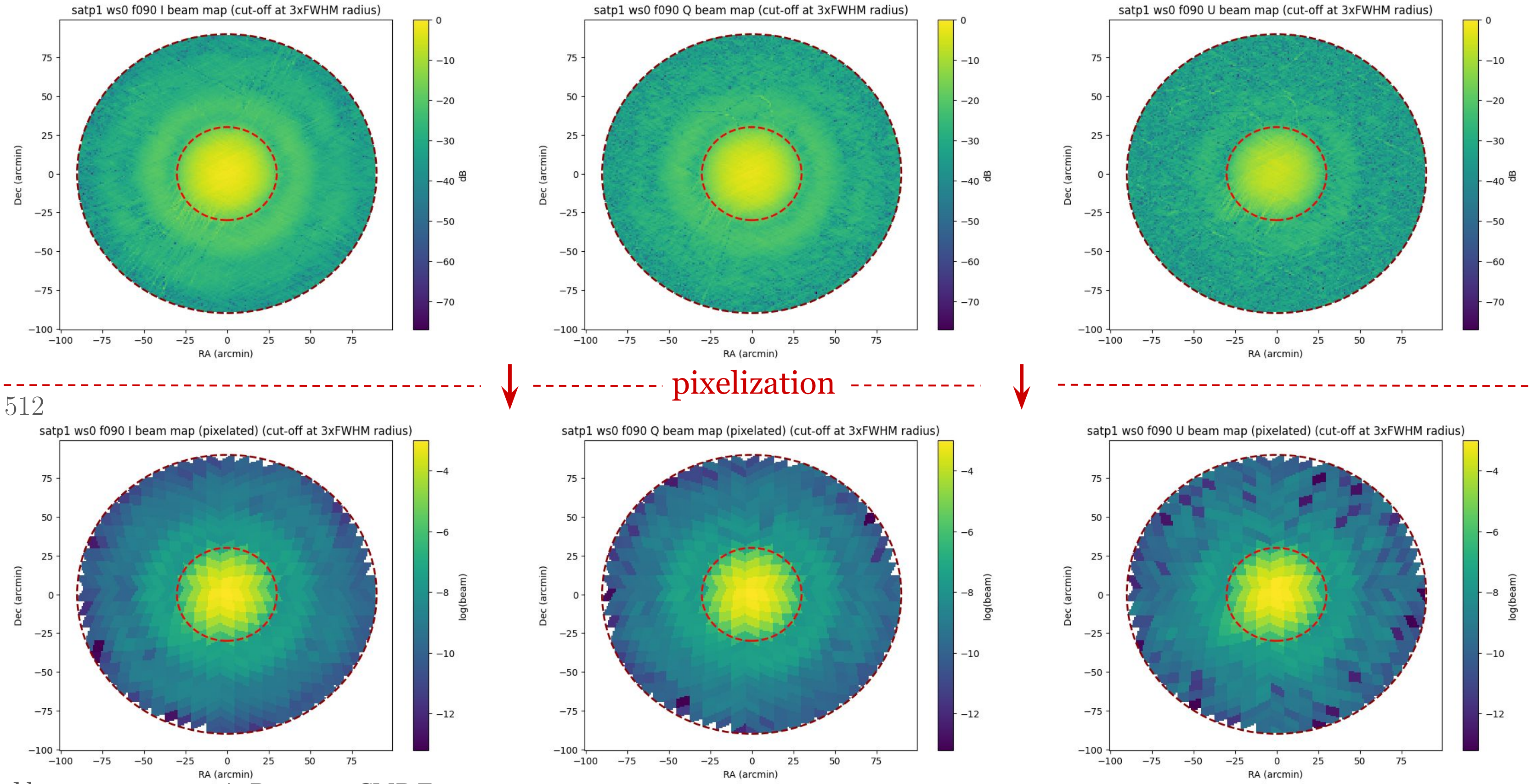
Creating a beam matrix

- We can use a high-fidelity beam pixel map from the instrument calibration (e.g., planets, drone for SO)
- Reading the pixel map, and for each Healpix pixel, we can re-center the high-resolution beam pattern and "stack" it onto the Healpix grid



Credit: Nadia Dachlythra

Creating a beam matrix



$N_{\rm side} = 512$

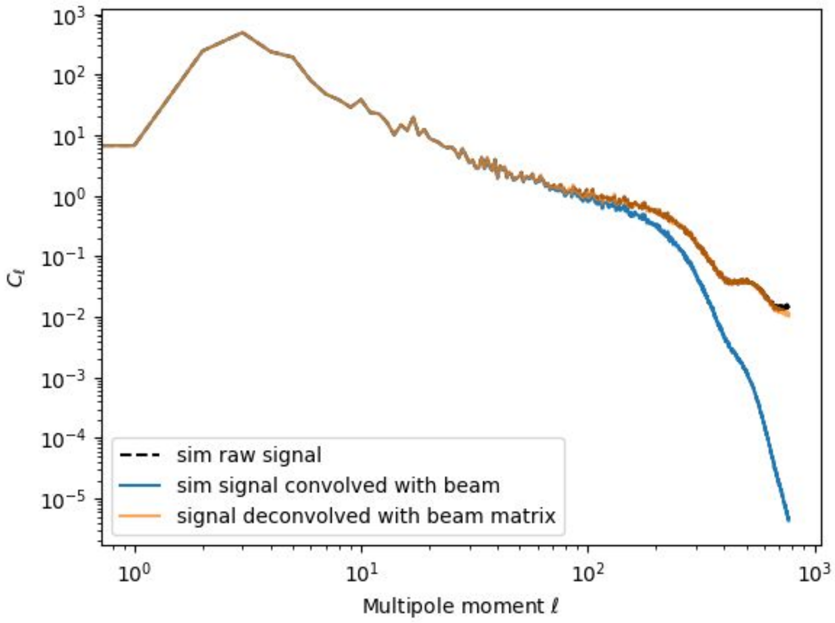
Demonstration of deconvolution

$$C_\ell = C_\ell^{\mathrm{CMB},90\,\mathrm{GHz}} + C_\ell^{d,90\,\mathrm{GHz}} + C_\ell^{s,90\,\mathrm{GHz}}$$

$$N_{\mathrm{side}} = 256$$

$$\mathrm{FWHM} = 30\,\mathrm{arcmin}$$

- Deconvolution is one of the most complex operations related to the beam applications
- Using a matrix for deconvolution requires calculating the inverse of the said matrix
- Calculating an inverse of the matrix usually is done by solving a linear equation using Conjugate Gradient method
- We have developed a method of solving a sparse matrix inverse problem accelerated by JAX



Conclusion and Outlook

- We have developed a map-space beam treatment that is flexible, intuitive, powerful, and public
- It has the potential to overcomes key limitations of harmonic-space methods, especially for non-spherical beams (although, harmonic methods do exist for non-symmetric beams)
- Powered by modern software (JAX/FURAX), it is computationally efficient and integrates well into the broader analysis framework on any platform (CPU/GPU)

Next Steps:

- Add partial map coverage support
- Integrate into the SO map-making and power spectrum pipeline
- Explore more complex beam systematics (e.g., cross-polarization), as well as other systematics that can be represented by a sparse matrices (e.g., detector cross-talk, instrumental polarization)
- *Basyrov et al (in prep)*

Artem Basyrov
basyrov@apc.in2p3.fr

Thank you!

SciPol team: