Pixel space convolution using sparse matrices

Pedro Fluxá

March 8, 2023

No abstract for now.

1 Objective

Propose a novel way of calculating full-sky convolution of a polarized sky with antenna beams in pixel space.

Note

This section uses the same notation as the paper that describes PISCO.

2 Beams of a Polarization Sensitive Bolometer

A Polarization Sensitive Bolometer (PSB) is a detection device that uses a pair of bolometers to sense both the total intensity (Stokes parameter I) and the degree of linear polarization in incoming light (Stokes parameter Q and U). Denoting each bolometer of the PSB as a and b we can write [1]

[1] cite Rossett et al and link the calculations that show this formulation is equivalent to O'Dea et al.

$$\begin{aligned}
\tilde{I}_a &= |\vec{E}_a|^2 && (1)\\
\tilde{I}_b &= |\vec{E}_b|^2 && (2)\\
\tilde{Q}_a &= |\vec{E}_{a\|}|^2 - |\vec{E}_{a\times}|^2 && (3)\\
\tilde{Q}_b &= |\vec{E}_{b\|}|^2 - |\vec{E}_{b\times}|^2 && (4)\\
\tilde{U}_a &= 2\Re\left(\vec{E}_{a\|}\vec{E}_{a\times}\right) && (5)\\
\tilde{U}_b &= 2\Re\left(\vec{E}_{b\|}\vec{E}_{b\times}\right) && (6)\\
\tilde{V}_a &= -2\Im\left(\vec{E}_{a\|}\vec{E}_{a\times}\right) && (7)\\
\tilde{V}_b &= -2\Im\left(\vec{E}_{b\|}\vec{E}_{b\times}\right) \quad , && (8)
\end{aligned}$$

where $\vec{E}$ denotes the time-averaged electric field density at co-latitude ρ and longitude σ (see 1). Subscript $\|$ is used to denote the co-polarized field, while $\times$ is used for the cross-polarized field.

Maybe add a note about polarization efficiency

3 Sky model

The sky can be modeled as a set of 4 scalar fields

$$S^i = S^i(\theta, \phi) = (I, Q, U, V) \quad , \tag{9}$$

where I, Q, U and V are Stokes parameters[2] of light coming from co-latitude θ and longitude ϕ in the sky basis.

4 Coupling of sky and PSB beams in pixel space

Note that if the PSB beams and sky are considered discrete quantities, then each pixel k will have an associated pair of coordinates in its corresponding basis. In pixel space, the power (per unit-wavelength) measured by a PSB pointing at $\bar{q}_0$ becomes

$$d_\alpha = \sum_{k=1}^{N_s} {}_kD_{\alpha,i}(\chi, \epsilon, s_\alpha)\, {}_kS^i \quad , \tag{10}$$

where α denotes a and b.

[2] Note about definition of polarization

The quantity D for detector a is

$$
\begin{aligned}
D_{a,i}(\chi,\epsilon,s_a) = & \frac{s_a}{2}\Big[\\
& \tilde{I}_a + \epsilon\tilde{I}_b, \\
& \left(\tilde{Q}_a - \epsilon\tilde{Q}_b\right)\cos(2\chi) - \left(\tilde{U}_a - \epsilon\tilde{U}_b\right)\sin(2\chi), \\
& \left(\tilde{U}_a - \epsilon\tilde{U}_b\right)\cos(2\chi) + \left(\tilde{Q}_a - \epsilon\tilde{Q}_b\right)\sin(2\chi), \\
& 0\Big] \quad ,
\end{aligned}
\tag{11}
$$

and the quantity D for detector b

$$
\begin{aligned}
D_{b,i}(\chi,\epsilon,s_b) = & \frac{s_b}{2}\Big[\\
& \tilde{I}_b + \epsilon\tilde{I}_a, \\
& -\left(\tilde{Q}_b - \epsilon\tilde{Q}_a\right)\cos(2\chi) + \left(\tilde{U}_b - \epsilon\tilde{U}_a\right)\sin(2\chi), \\
& -\left(\tilde{U}_b - \epsilon\tilde{U}_a\right)\cos(2\chi) - \left(\tilde{Q}_b - \epsilon\tilde{Q}_a\right)\sin(2\chi), \\
& 0\Big] \quad ,
\end{aligned}
\tag{12}
$$

5 Computation of D

In practice, discrete quantities such as sky and PSB beams are represented as vectors. Note that the definition of D is implicitly assuming that the PSB beam is aligned with the sky: more precisely, that pixel k of the beam corresponds to pixel k of the sky. In this sense D hides a *rotation* of the PSB beams from the beam basis to the sky basis.

Calculating the *rotated* PSB beam can be represented as a matrix-vector product. For each component (i) of the beam, its rotated counterpart becomes

$$
\tilde{X}' = \mathbf{R}\tilde{X}^T \quad , \tag{13}
$$

where $\mathbf{R}$ is an $N_s \times N_b$ sparse matrix with exactly 4 non-zero entries per row[3].

In some way, every row of $\mathbf{R}$ represents the dot-product of 4 pixels from the PSB beam component (North, South, East and West pixels around $\bar{q}_0$) and their interpolation weights. Denoting S, N, E and W (all integers from 0 to N_b) to neighbor pixels and z_N, z_S, z_E and z_W to the interpolating weights

$$
{}_k\tilde{X}' = {}_k\mathbf{R}\tilde{X}^T = \left[0 \cdots z_N \cdots z_S \cdots z_E \cdots z_W \cdots 0\right] \cdot \tilde{X}^T \tag{14}
$$

[3]For bi-linear interpolation

6 Remarks of algorithm

- Formalism is designed for full-sky convolution.
- Every row of $\mathbf{R}$ can be calculated in parallel using already existing libraries (like HEALPix)
- Matrix $\mathbf{R}$ is very sparse, and can be represented efficiently using compression techniques (like CSR)
- The (sparse) matrix-vector product can be calculated in accelerators, like a GPU, using existing libraries for Sparse Linear Algebra (like cuSparse from NVIDIA).

References

`http://www.xkcd.com/242/`