Delivering The Total Sky In TOAST

Julian Borrill

Computational Cosmology Center, Berkeley Lab

Space Sciences Laboratory, UC Berkeley

with

Reijo Keskitalo, Ted Kisner, Jonathan Madsen & Andrea Zonca

and the PySM team

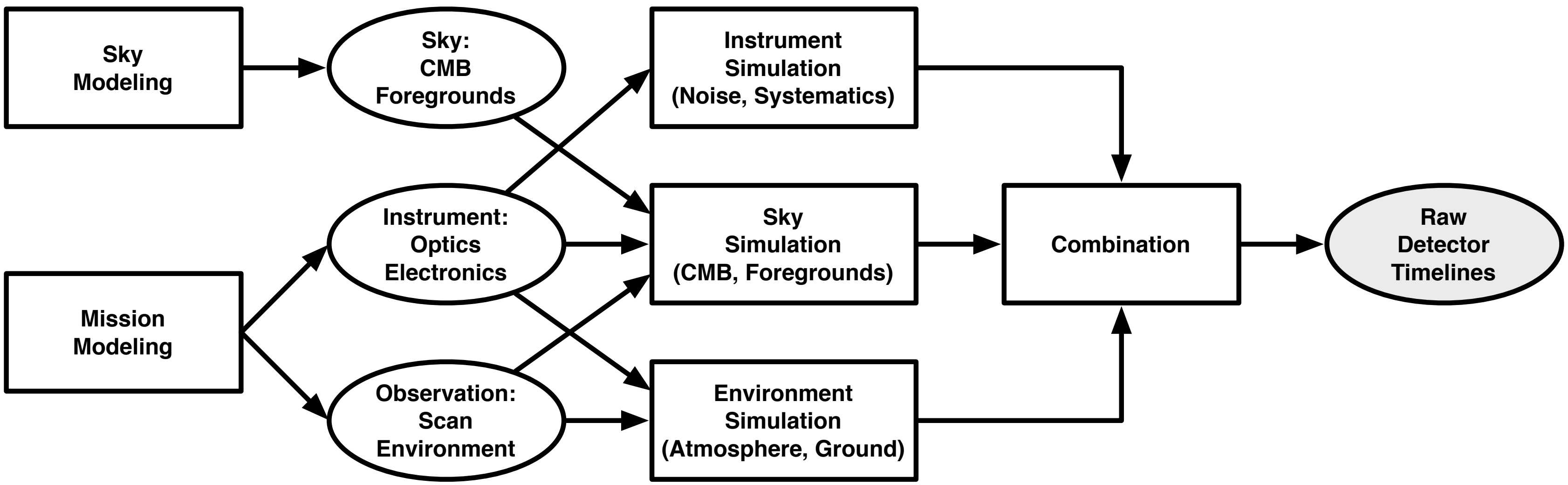
Overview

- CMB missions need synthetic data sets all the time, from their conception to their final data release:
  - Mission design
  - Analysis pipeline validation & verification
  - Uncertainty quantification & debiasing
- Generating a synthetic data set requires:
  - A sky model
    - CMB, Extragalactic & Lensing, Galactic
  - A mission model
    - Instrument (optics, electronics, …)
    - Observation (environment, scanning strategy, …)

Data Domains

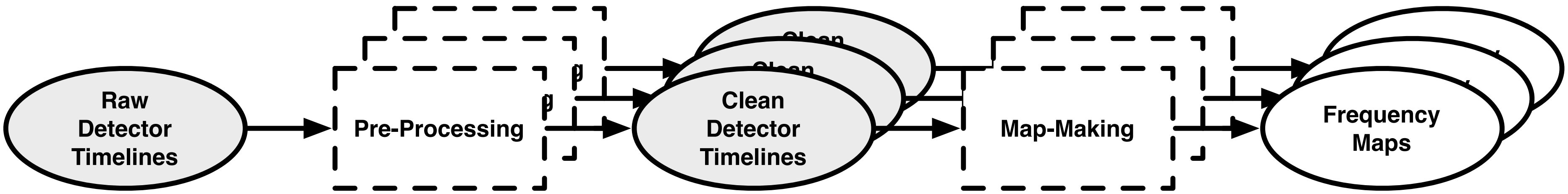
- CMB data can be synthesized in:
  - The time-domain:
    - Provides the most realistic data set,
    - Supports the fullest range of requirements,
    - Comes the highest computational cost
  - The map domain:
    - Provides intermediate realism, use, cost
  - The spectral domain
    - Provides lowest realism, use, cost
- All of these have their place; we are focused on the first here, using TOAST as an example.

Schematic: Data Synthesis



- Given the volume of the time-domain data:
  - Synthesis & reduction are necessarily massively parallel.
  - Time-domain data objects must never touch disk.

Schematic: Data Reduction



- Given the expense of time-domain data synthesis we want to be able to apply multiple reductions to a single data set *in situ*.
  - Different reduction algorithms
  - Data splits

Sky Model Requirements

- We need sky models that can support all analyses
  - Across scales: low- and high-ell science
  - Across missions: comparison & complementarity
  - Across science: CMB x LSS, CMB x 21cm, …
- This means:
  - All sky maps
  - Of all components (including complementary representations)
  - In temperature and polarization
  - At sub-arcminute resolution
  - At 1-1000 GHz
  - *Independent of any single mission*

~~Current~~ Past Status

- Active missions + WMAP & Planck span 91 frequency bands
- Modelers currently generate their sky including bandpass
  - Tophat & delta bandpassed skies delivered to NERSC.
    - /project/projectdirs/{ccat, cmbs4, core, litebird, pico, sobs, … }
    - accounts at http://crd.lbl.gov/cmb (repo mp107, PI Borrill)
  - Mostly PySM to date; others for specific subsets.
- This is not sustainable!
  - Sky modelers should not be concerned with (evolving) instrument specifications.
  - This approach can't support bandpass mismatch over tens to hundreds of thousands of detectors.

~~Future~~ Current Status

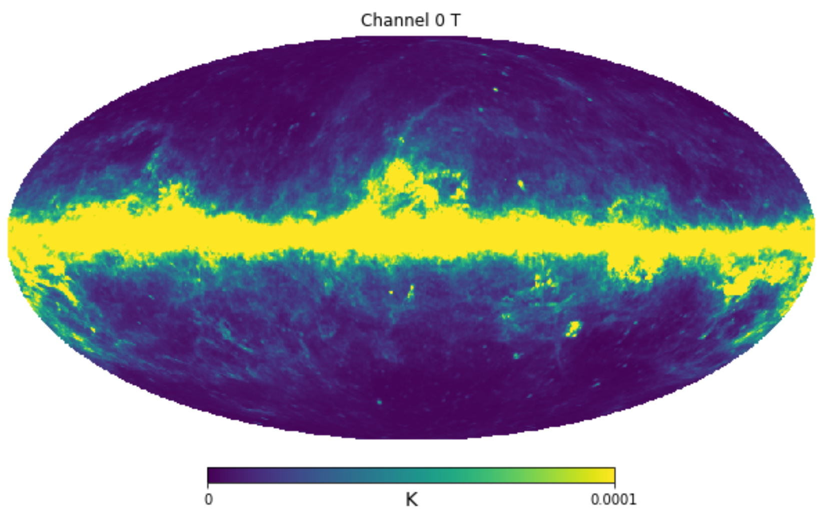
- Any sky model is a parallelizable function that returns a map of its sky components at any given frequency.
- Mission data synthesis then includes on-the-fly, per detector,
  - Arbitrary bandpass integration
    - Numerical integration with bandpass- & model-based sampling
  - Arbitrary beam convolution
    - Circular, elliptical, irregular compact, 4π
    - Possibly different in intensity and polarizations
  - Arbitrary sky sampling
    - Time- or pixel-data redistribution to “locally observed” sky

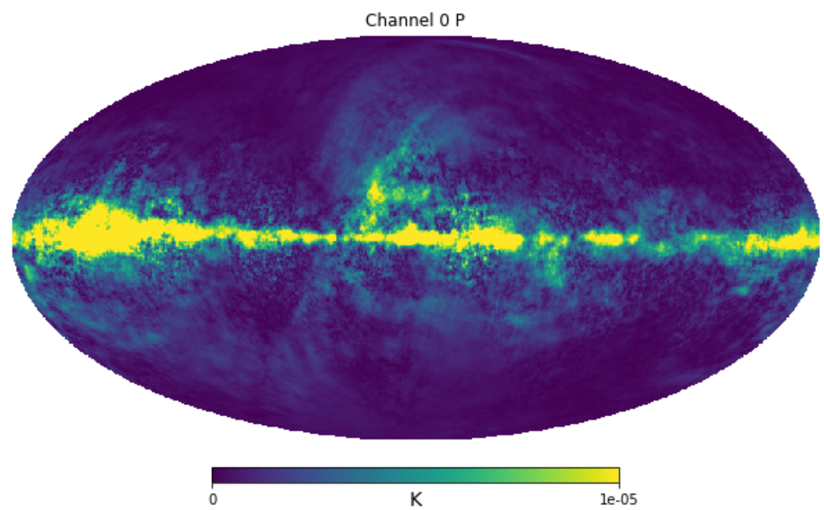
Pseudo-Code

- For a given instrument, over many processes
  - For each detector
    - Numerically integrate over the detector-specific band, using the sky model function(s) to give the sky at each frequency in the integral.
    - Convolve the sky with the detector-specific beam in the pixel or harmonic domain.
    - Apply the detector-specific pointing to generate the sky signal time-stream.
- The sky coverage and distribution of the pixels over the processes depends on the type of beam convolution, so the sky model must be agnostic to this.

PySM Example

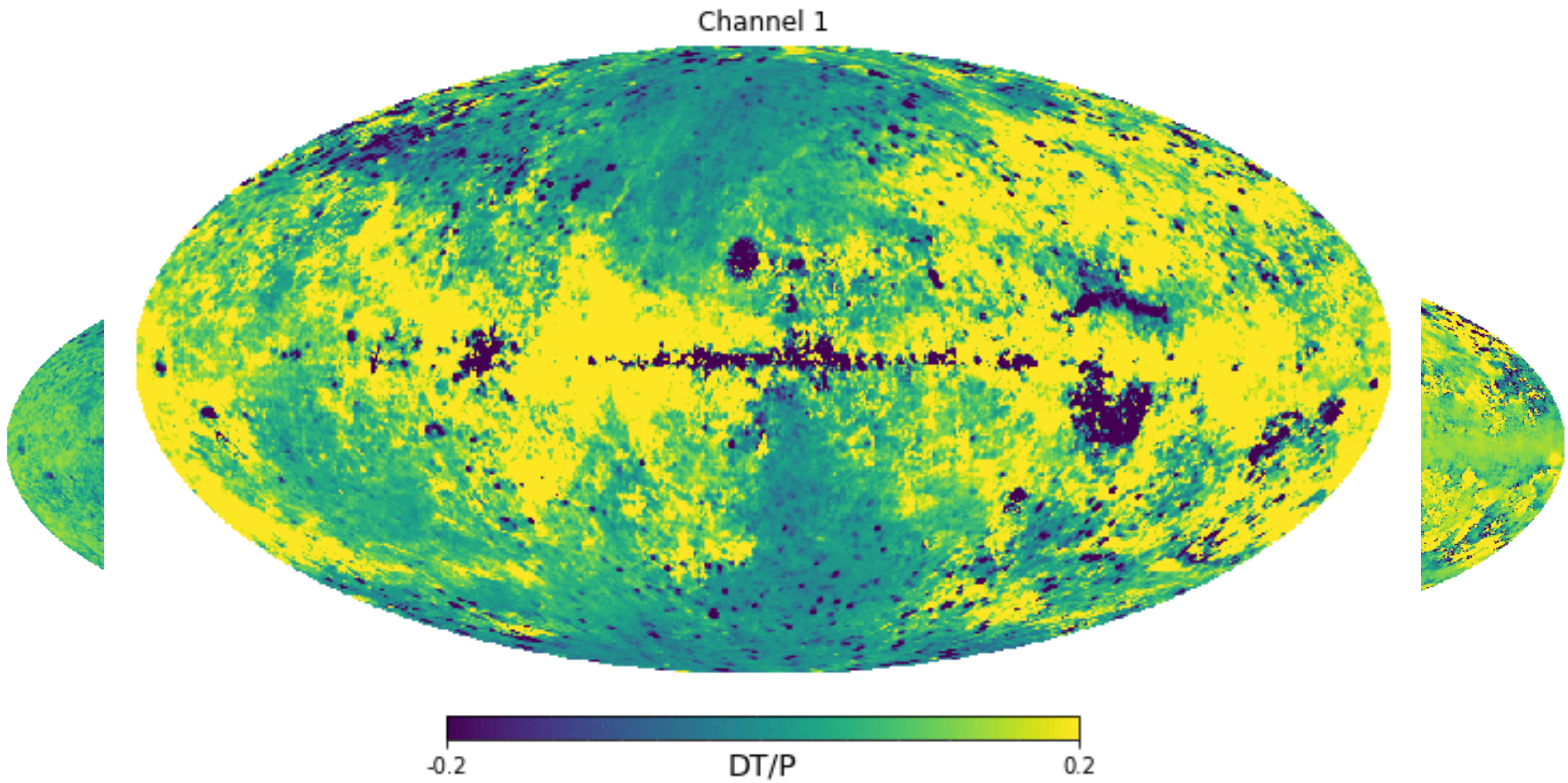
- Added support in PySM for partial skies.
- Distributed full sky over latitudinal rings to apply symmetric Gaussian beam via parallel SHT.
- PySM a2d7f1s3 (CDT model 3):
  - 95GHz band center, 20% bandwidth, 24 arcminute beam





Bandpass/Beam Mismatch

- Now include 1% variation in band center/width & beam FWHM



Conclusions

- A new distribution of labor:
  - Sky modelers provide
    - Parallelizable functions (in unlicensed languages!)
    - Returning high-resolution IQU maps of some or all of the sky
    - For some sky component(s)
    - At any given frequency
  - Mission data synthesizers perform
    - Bandpass integration*
    - Beam convolution
    - Sky sampling to the appropriate domain

Questions

1. Can sky models:
   - Be parallelized?
     - Serial sub-sky or parallel all-sky.
   - Return an arbitrary frequency?
     - At very least generate a sufficiently dense sampling of frequencies and interpolate.
   - Run *in situ* in a very massively parallel context?
     - Language(s), data distribution, MPI/threads …
2. Is there value in a common wrapper (cf. PySM)?
3. For EG models, can we lens on-the-fly or do we need a library of pre-lensed CMB skies?
4. Recipe for populating halos with sources?