MADmap (and TOAST) for PACS

Theodore Kisner
Computational Cosmology Center, LBNL
2013-01-28
Historical Perspective

MADCAP v1 (C, MPI)
- MADspec
- MADnes
- MADpre
- MADmap

MADCAP v2 (C, MPI/OpenMP)
- MADspec
- M3
- MADnes
- MADpre
- MADmap

TOAST (C++, MPI/OpenMP)
- Math Lib
- I/O Lib
- Mapmaking Lib
- Apps

MADmap / HIPE (Java)
Generalized Least Squares Map-making

\[ b = C \ x \quad \text{But we have (Gaussian) noise...} \]

\[ d_t = A_{tp} \ s_p + n_t \quad \mathcal{N}_{tt'} = \langle n_t \ n_{t'} \rangle \]

Algebra: plug into maximum likelihood expression...

\[
\left( A_{tp}^T \ \mathcal{N}_{tt'}^{-1} \ A_{t'p'} \right) \ s_p = A_{tp}^T \ \mathcal{N}_{tt'}^{-1} \ d_t
\]

...and we have the noise-weighted least squares equation for the maximum likelihood map
Generalized Least Squares Map-making

\[
(\mathbf{A}_{tp}^T \mathbf{N}_{tt'}^{-1} \mathbf{A}_{t'p'}) \mathbf{s}_p = \mathbf{A}_{tp}^T \mathbf{N}_{tt'}^{-1} \mathbf{d}_t
\]

Inverse noise covariance
(Toeplitz - use FFT for product)
RHS / “noise weighted map”

Solve this iteratively with a PCG...
GLS Map-makers do not Stand Alone...

- (Except for simulations)
- Real data requires some way to estimate the noise covariance: need a "noise dominated" timestream with the same noise properties as your dataset.
  - If S/N is bad, just use the data!
  - Find some other data without signal which has the same noise properties
  - Subtract estimate of the sky signal (e.g. destriped map) from the TOD and use that to build covariance
PACS TOD
PACS Noise Estimation

How to get a noise TOD for covariance estimation?

- How stable is noise? Can “dark” observations be trusted?
- Make a first guess at signal map by destriping or filter-and-binning

Are common-mode drifts “signal” or “noise”?

- If Signal: subtract it from the beginning, BUT this breaks linear data model, so need simulations to verify that the effect is small
- If Noise: need to handle full cross spectra between detectors...
Time Ordered Astrophysics Scalable Tools (TOAST)

- Multiple instruments supported through use of I/O plugins (basically C++ derived classes)
- Instrument-specific data selection (in python) used to build up a “run” file containing which data to process
- Built in data distribution and caching across MPI tasks
- OpenMP and MPI/OpenMP versions of map-making tools: diagonal pixel noise covariance, GLS mapmaker, simple destriper, noise estimation, etc
- Hi-level apps for “usual” map-making tasks
TOAST Code Organization

- BOOST, Intel / AMD / IBM / Apple Vendor Libraries
- Instrument plugins for data formats
- I/O Library
  - pointing, TOD, noise, metadata ops, data distribution, etc
- Math Library
  - FFT, SRNG, PCG, etc
- C, Python, F03 Language Bindings
- "Apps"
- Map-Making Library
## Status of TOAST with PACS

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class for PACS TOD reading</td>
<td>IN PROGRESS</td>
</tr>
<tr>
<td>Class for reading PACS boresight pointing</td>
<td>IN PROGRESS</td>
</tr>
<tr>
<td>WCSLIB integration</td>
<td>IN PROGRESS</td>
</tr>
<tr>
<td>Plan for (spherical) focalplane geometry</td>
<td>DONE</td>
</tr>
<tr>
<td>Format for focalplane geometry</td>
<td>NOT STARTED</td>
</tr>
<tr>
<td>Python class for building runs</td>
<td>NOT STARTED</td>
</tr>
<tr>
<td>PSF model / partial deconvolution?</td>
<td>R &amp; D</td>
</tr>
<tr>
<td>Testing!</td>
<td>NEVER DONE...</td>
</tr>
</tbody>
</table>
Future

- Our group looks forward to more engagement with Herschel people and data!
- Techniques and experience from CMB map-making may be useful for Herschel (and vice-versa)
- We are working on TOAST documentation and trying to release the code “soon”. Code is currently in a private github repo, and internal LBNL paperwork is proceeding...