

Not satisfied with your processed data?
Don't be irate, simulate and iterate!

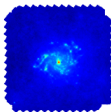
Pierre Chanial, Nicolas Barbey, Marc Sauvage and the
Tamasis collaboration

The Tamasis project

- Tools for Advanced Map-making, Analysis and Simulations of Submillimeter surveys.
- ASTRONET-funded project, P.I: Marc Sauvage
- Tools will be made available to the community (some already are)
- Collaboration between 4 institutes:
 - Garching: cosmological light cones simulations
 - IAS: SPIRE advanced map-making
 - Saclay: PACS advanced map-making
 - Leiden: stacking

An instrument simulator, what for?

- design planning and validation
- pipeline planning and validation
- scientific planning and validation



Telescope



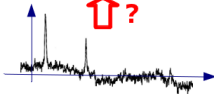
Bolometers



Electronics



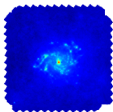
On board compression



PSF
Emissivity
Distortion

Spatial integration
Response
Thermal drift
Influence of cosmic rays

Multiplexing
Cross-talk
Time constant
1/f noise



Telescope

- PSF
- Emissivity
- Distortion



Bolometers

- Spatial integration
- Response
- Thermal drift
- Influence of cosmic rays

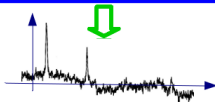


Electronics

- Multiplexing
- Cross-talk
- Time constant
- 1/f noise



On board compression



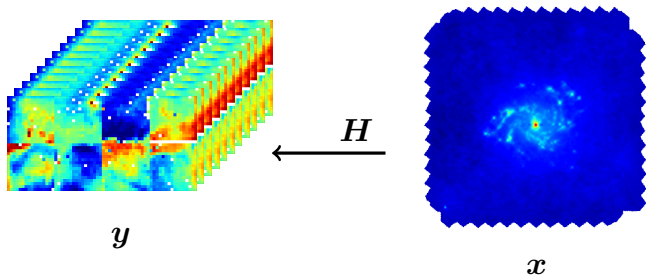
4

An instrument simulator, what for?

- design planning and validation
- pipeline planning and validation
- scientific planning and validation
- improve scientific output from an observation

Doing science with any observation is attempting to solve an inverse problem

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$$



$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \{ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_{N-1}^2 + \Psi(\mathbf{x}) \}$$

- iterative methods (PCG family)
- acquisition model \mathbf{H} must be fast

Doing science with any observation is attempting to solve an inverse problem

Pros:

- take advantage of every bit of information about the instrument, in a natural way
- “deconvolve” instrumental effects

Cons:

- computationally challenging (\Rightarrow supercomputers)
- degeneracy of solution (\Rightarrow regularisation)

Tamasis package

- Easy to use and fast: Python for the abstraction and Fortran/C for the speed
- Scalable: MPI and OpenMP
- Modular: library of linear and non-linear operators (CompressionAverage, Convolution, DistributionGlobal, DistributionLocal, DiscreteDifference, DownSampling, Fft, FftHalfComplex, InvNtt, Masking, Packing, Padding, Projection, RollOperator, ShiftOperator, Unpacking)
- Solvers
 - PCG (MPI)
 - non-linear CG (MPI)
 - `scipy.sparse.linalg` (CG, CGS, BICG, BICGSTAB, GMRES, LGMRES, ...)
 - `scipy.optimize` (`fmin_bfgs`, ...)

A simple model

```
obs = PacsObservation('myfile.fits')  
tod = obs.get_tod
```

```
proj = Projection(obs)  
compression = CompressionAverage(8)  
model = compression * proj
```

```
m = mapper_ls(tod, model)
```

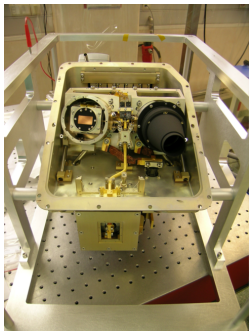
- An acquisition model is built up using bulding blocks

Herschel

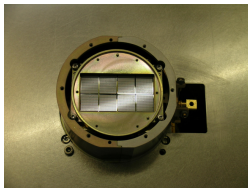


- **Mission:** Galaxy formation, star formation, planets atmosphere, astrochemistry
- **Instruments:** PACS, SPIRE, HIFI
- **Bandwidth:** 60 - 600 μm
- **Diameter:** 3.5 m

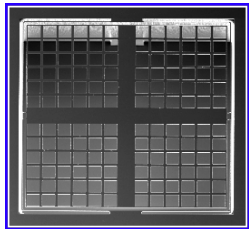
PACS Photometer



PACS $\approx 30\text{cm}$



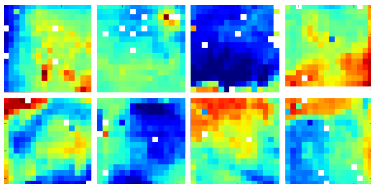
matrix = 1.5cm



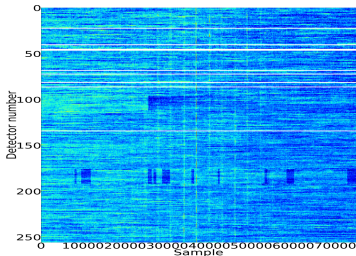
pixel = $750\mu\text{m}$

- **Tech:** array of bolometers
- **Arrays:** 16×32 , 32×64 pixels
- **Sampling:** 40 Hz
- **Modes:** staring, scan and parallel
- **Filters:** (μm) blue: 60–85, green: 85–130, red: 130–210
- **PSF:** blue: $5.5''$, green: $6.7''$, red: $11''$

PACS data artifacts

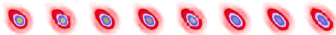


A frame of blue PACS data

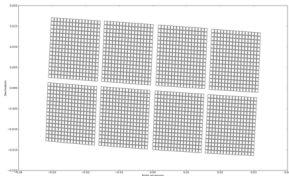


A set of reordered pixel values as a function of time, offset removed, two 25 min scans, 60 000 frames each

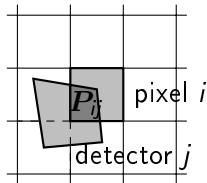
- Offset drift $\Leftrightarrow 1/f$ noise
- On-board compression



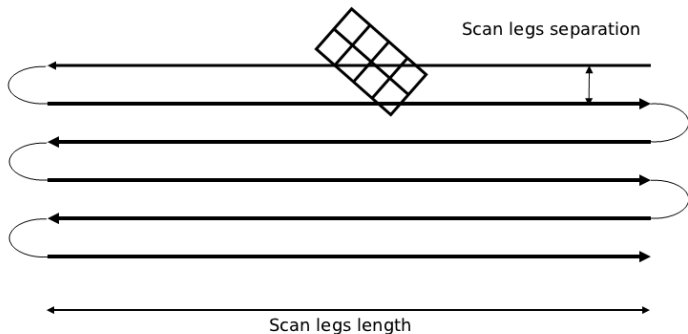
- Distortion



- Projection

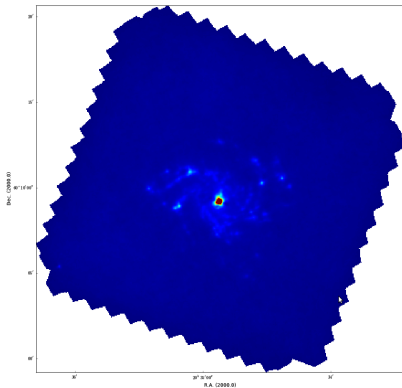


PACS scan mode



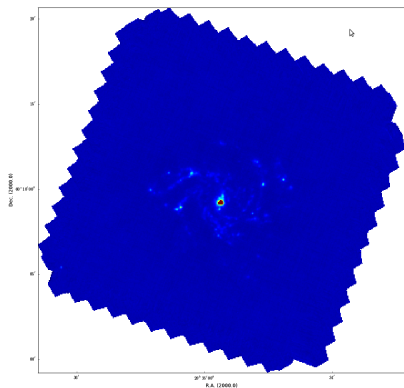
- PACS FOV: 1.8×3.7 arcmin
- Sky maps: $\approx 1 \times 1^\circ$ i.e. 1200×1200 pixels
- Use redundancy to get rid of $1/f$ noise and instrument artifacts

High frequency reconstruction



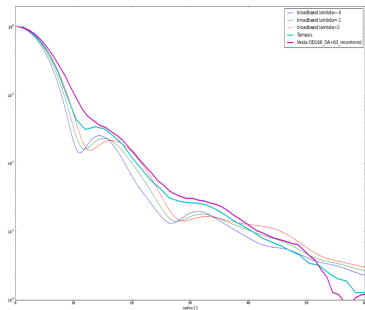
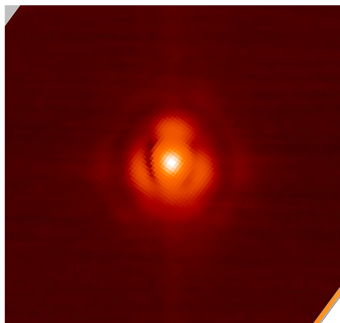
"Pipeline"

Same colour cuts $-\sigma, +10\sigma$



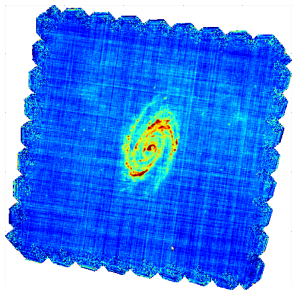
Tamasis RLS

High frequency reconstruction

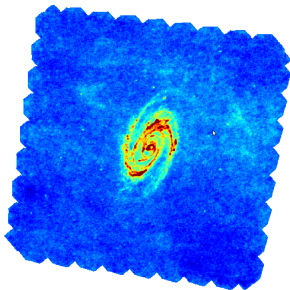


Red channel PACS PSF

Low frequency reconstruction

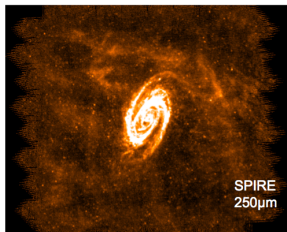


"Pipeline"



Tamasis RLS

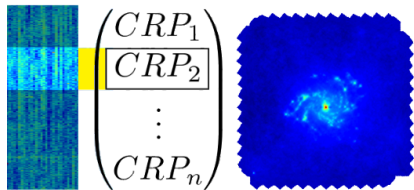
- Artifacts removed
- Structures are real



Optimisations

- operator's rules may simplify an expression
- stack-based memory manager: ensure that no temporary array is created during iterations
- algorithm to minimise the number of temporaries during a composition
- inplace / output operators: whether or not the operator's output can point to the same memory as its input
- inplace reduction
- block partitioning: crucial to take advantage of the cache

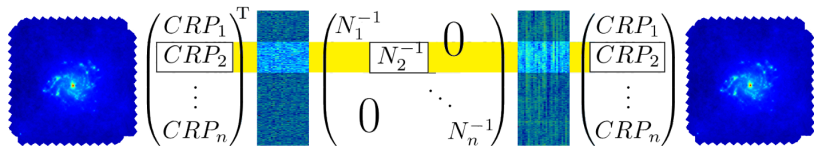
Optimisation: blocks and contiguity



The result of the computation of one block is stored in non-contiguous memory, so Hx operation is not particularly efficient.

But in iterative solvers, what is required is $H^T N^{-1} H$.

Optimisation: blocks and contiguity



Optimisation: blocks and contiguity

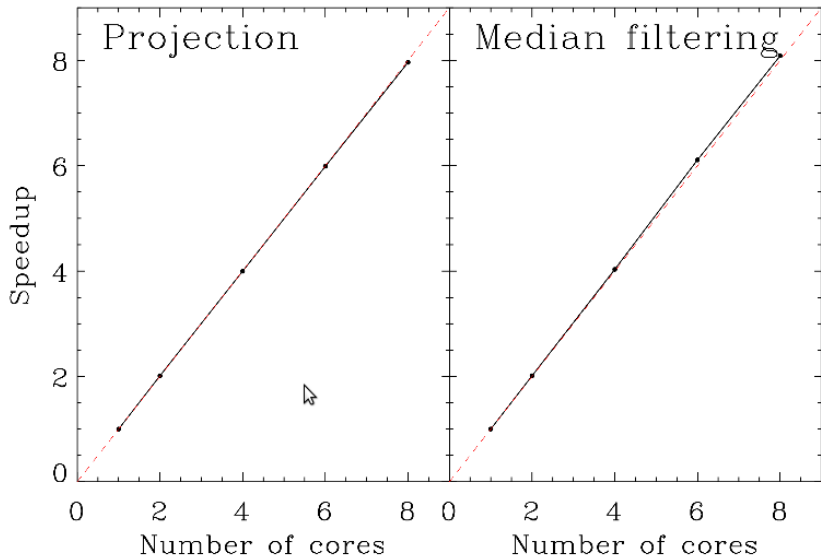
$$\sum_{i=1}^n P_i^T R^T C^T N_i^{-1} C R P_i$$

Perform all computations related to an observation block (increased temporal locality).

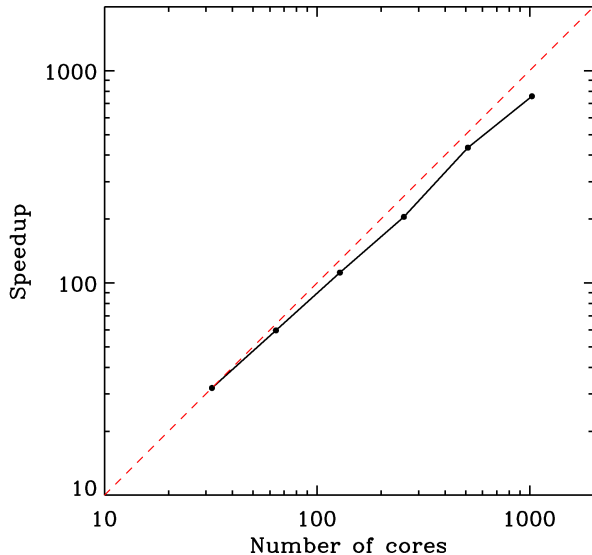
Then, move on to the next block.

All memory accesses are contiguous and may all be in the cache, if the block size is suitably chosen.

Parallelisation: OpenMP scaling



Parallelisation: MPI scaling



A simple model

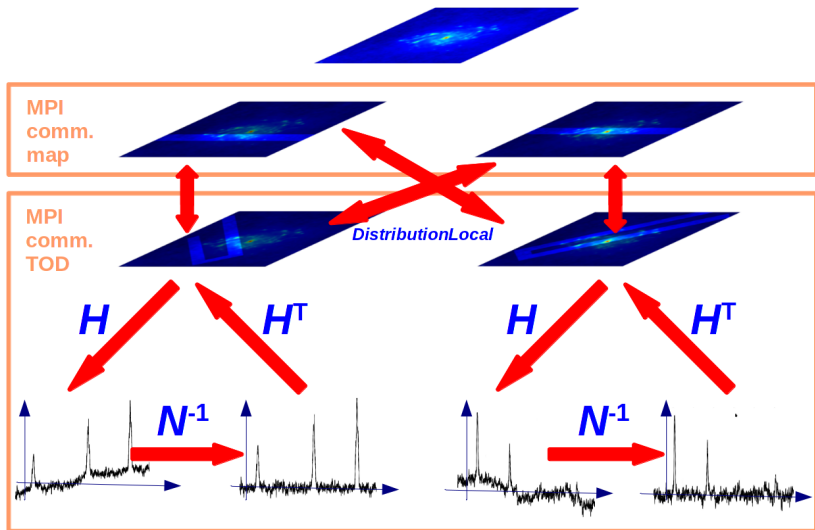
```
obs = PacsObservation('myfile.fits')  
tod = obs.get_tod
```

```
proj = Projection(obs)  
compression = CompressionAverage(8)  
model = compression * proj
```

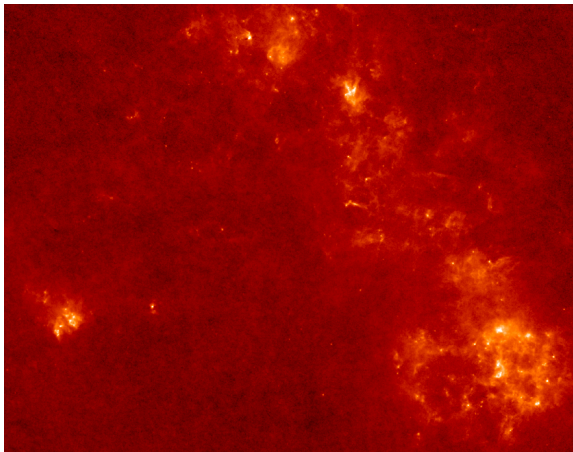
```
m = mapper_ls(tod, model)
```

Bringing MPI / OpenMP to a larger audience

Parallelisation: distributed map



Large dataset



- Map is $9.6^\circ \times 6.5^\circ \approx 0.7$ GB ($8 \cdot 10^7$ pixels)
- TOD is 69h long, 512 detectors $\times 2.5 \cdot 10^6$ frames = $1.3 \cdot 10^9$ samples
- Pointing matrix is 0.85 TB (1 detector intercepts 10 pixels)
- 1h45 on CCRT (282 iterations), using 512 cores (64 MPI processes)

Double loop inference

i.e. : Sparse variational bayesian inference (work by N. Barbey)

- Sparse = better statistical model
- Provably converges to a global solution

Solves :

$$\arg \min_{\mathbf{x}, \gamma} \{ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \mathbf{x}^T \mathbf{B}^T \text{diag}(\gamma) \mathbf{B}\mathbf{x} \}$$

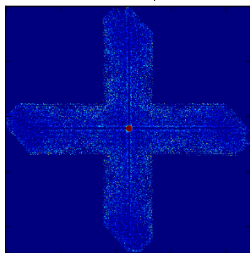
Alternate a few times between :

- Inner loop : find \mathbf{x} with a Quasi-Newton minimization
- Outer loop : find γ using Lanczos eigendecomposition

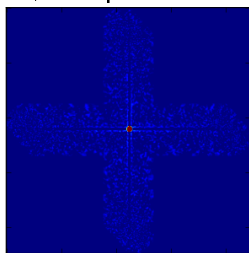
Results on simulation

PACS/Tamasis simulation

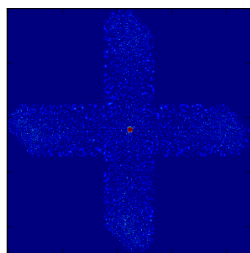
Point source, scan mode, compression $\times 8$



MADmap



Tamasis RLS



Tamasis DLI

Lessons learnt

- Python/numpy is fine with vector operations. For more complex operations, it should be avoided during iterations
- In parts you have to, use a profiler and experiment.
Slowness of methods is unpredictable
- Do not rely on the garbage collector: allocate once and reuse the buffer
- Interfacing Fortran code is incredibly easy thanks to f2py (and paradoxically easier than C code)

Future

- Publish
- Get people interested and involved:
 - ⇒ <https://github.com/pchanial/pyoperators>
(general purpose operators and solvers)
 - ⇒ <http://pchanial.github.com/tamasis-map>
(astronomy)
- Better astrometry reconstruction
- Deconvolution of the telescope PSF
- Better noise characterisation (correlation between detectors)
- GPUs
- Data fusion (combine SPIRE data)