Midapack library : Toeplitz Algebra for Cosmic Microwave Background data analysis

Frédéric Dauvergne, Pierre Cargemel, Maude Le Jeune, Radek Stompor

Laboratoire Astroparticule et Cosmologie (APC), rue Alice Domon et Leonie Duquet, 75205 Paris Cedex 13.

Motivations

Provide tools for Cosmic Microwave Background (CMB) data analysis

- High performance ;
- Massively parallel ;
- Portable ;
- Easy to use and flexible set of routines ;
- Including efficient new algorithms and techniques ;
- Capable of dealing with large data volumes.

$\rightarrow$ Solve the maximum likelihood CMB map making equation.

Toeplitz matrix for CMB data analysis

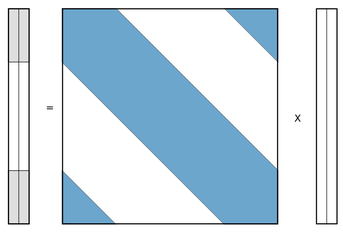
Toeplitz matrices are ubiquitous in the CMB data analysis as they describe correlation properties of stationary time-domain processes (usually instrumental noise). The relevant matrices are therefore symmetric and non-negative definite. They are also band-diagonal as the noise correlation length and the bandwidth is typically much shorter than length of the data.
A piece-wise stationary processes can be then described by a **symmetric Toeplitz block-diagonal matrice** where each of the blocks is a symmetric, band-diagonal Toeplitz matrix, which can be different from the others.
Thus, this give us the possibility to build efficient algorithms related to Fast Fourier Transform (with a complexity of $\mathcal{O}(n \log n)$).

Direct product routine

Any symmetric Toeplitz matrix can be embeded in an appropriately larger symmetric **circulant matrix**. The FFT direct product exploits this property to have an efficient algorithm, thanks to

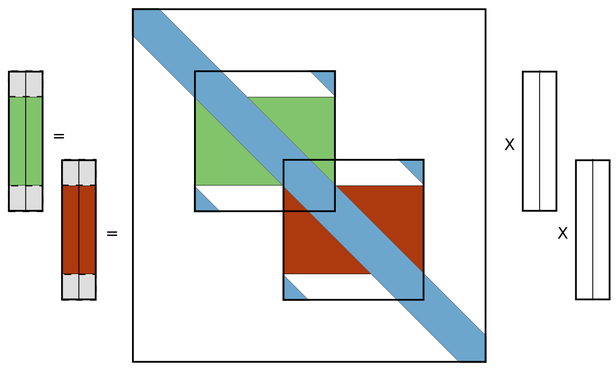
$$\mathcal{F}_n(C_1 \star V) = \mathcal{F}_n(C_1)\,\mathcal{F}_n(V)$$
$$C\,V = \mathcal{F}_n^{-1}\left[\mathcal{F}_n(C_1)\,\mathcal{F}_n(V)\right]$$

where C_1 is the first column of the circulant matrix C and V the data vector.



Core routine – The sliding window algorithm

The product is done as a sequence of submatrix product by overlapping blocks of the arbitrary matrix. Each of these direct product can be perform using FFT on the corresponding circulant matrix of the Toeplitz to cuts the complexity of the operation down to $\mathcal{O}(n \ln bs)$.
You also have the choice to use a classical product without using FFT, which could be efficient for very small bandwith.
The schematic of the algorithm is shown in the figure below for two consecutive submatrix.



The size bs of the subblock can be set appropriately to optimize the calculation and typically is a function of the bandwith.
For optimal performance computation, input data may need to be reshaped considering the input data structure and the number of available OpenMP threads (simultaneous FFTs). This take into account all the possibilities and required at most one copy of the data.

Acknowledgment

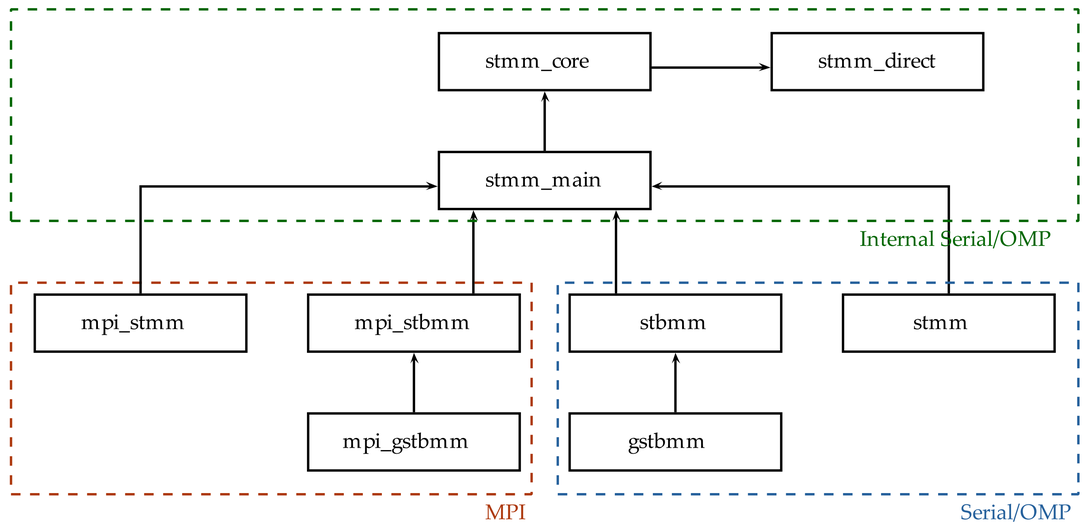
This work has been supported in part by French National Research Agency (ANR) through its COSINUS program (project MIDAS no. ANR-09-COSI-009).

High performance computing resources have been provided :

- in France by CCRT, TGCC, and IDRIS supercomputing centers under the GENCI program through projects: 2011-066647 and 2012-066647 ;
- in the US by the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Toeplitz algebra module structure

API routines dependency diagram :

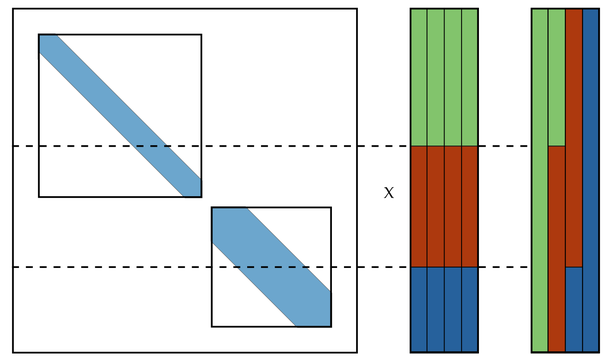


Toeplitz block diagonal routine

This routines multiply a symmetric, block-diagonal matrix, with Toeplitz banded blocks by a data matrix. Each Toeplitz block is defined as any general Toeplitz matrix described here.

The main characteristics are :

- Use Floating blocks position without overlap ;
- Communications between neighbors MPI processes are made only if needed ;
- On each MPI process, one can define the minimum blocks required for the computation or more, even all.



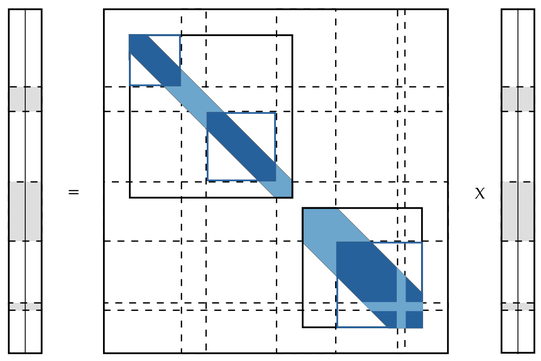
The data input matrix distributed between all the MPI processes can be **row-wise order**, **column-wise like order**, or a **combination of the both**. Each block has the same definition in every processes that use it. If a row of the data matrix do not belong to any block interval, the data remains unchanged.

Gappy Toeplitz block diagonal routine

This routine do the same as above, but include also gap locations which represents data to be neglected. Theses data values are set to zeros for the computation. Using this, a precomputation step is made to build an equivalent set of blocks Toeplitz to optimize the time computation.

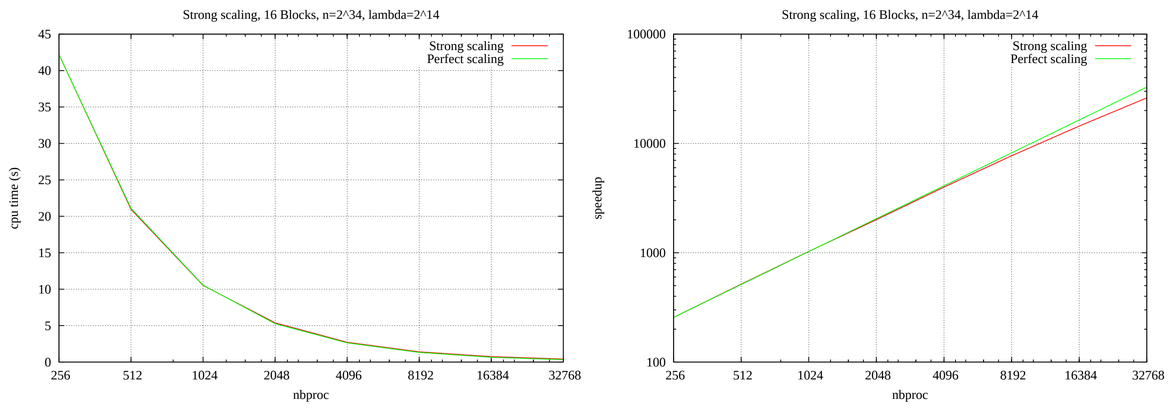
The successive steps of the routine are :

- Build gappy blocks considering gap locations ;
- Reset all the gaps to zeros ;
- Call the Toeplitz block diagonal routine with this freshly build gappy data structure ;
- Reset to zeros all the gaps to clean the wrong results.



The data matrix is assumed to have all the rows on the input including the ones to be neglected.

Scaling performances



This test was made on Hopper (Nersc). We used a 2^{34} raw data vector with 16 Toeplitz blocks. Each of theses blocks are equal and made with a bandwith equal to $2 \times 2^{14} - 1$. Communication between each neighbour process are nonblocking and use at most $2^{14} - 1$ communications.

Conclusion

The Toeplitz algebra module described here provides functionality to compute the products of a generalized piece-wise Toeplitz matrix by a matrix with severals properties :

- Support hybrid MPI/OpenMP high performance computation architecture ;
- This gives an efficient and flexible set of routines ;
- Advances tuning is possible for any expert user.

The Midapack library is available on the official website of the project :
http://www.apc.univ-paris7.fr/APC_CS/Recherche/Adamis/MIDAS09/software.html