OpenHmX, an open-source $\mathcal{H}$-Matrix toolbox in Matlab

Matthieu Aussal[1,*]

[1]CMAP - Ecole Polytechnique, Université Paris-Saclay, Route de Saclay, 91128, Palaiseau Cedex, France.

[*]Email: matthieu.aussal@polytechnique.edu

Abstract

For many applications in numerical physics, fast convolutions with a Green kernel on unstructured grids are needed to compute in a reasonable time the matrix-vector products. To this aim, many methods have been developed in the last decades. They are divided in two major classes, those which use analytical approximation of the Green kernel [1,2,4] and those based on algebraic compression [3, 5]. Associated to this paper, we provide a new open-source Matlab toolbox named *OpenHmX* [6] for the second class of compression.

Keywords : $\mathcal{H}$-Matrix, open source, green kernel convolution, hierarchical tree, ACA, SVD

1 Context

As a generic example, the case of boundary element formulations for tri-dimensional acoustics is considered, associated to the single layer potential expression :

$$\mathcal{S}\lambda(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{y})\lambda(\mathbf{y})d\Gamma_y, \quad \forall \mathbf{x} \in \mathbb{R}^3,$$

where $G(\mathbf{x},\mathbf{y})$ is the Green kernel :

$$G(\mathbf{x},\mathbf{y}) = \frac{e^{ik|\mathbf{x}-\mathbf{y}|}}{4\pi|\mathbf{x}-\mathbf{y}|}, \tag{1}$$

and Γ the boundary. Using a discrete quadrature of Γ, this convolution product needs a fast computation of discrete sums as :

$$G \star f(\mathbf{x}) \sim \sum_{n=1}^{N} G(\mathbf{x},\mathbf{y}_n) f_n, \tag{2}$$

where the potential $(f_n)_{1\leq n\leq N}$ is known for all $\mathbf{y}_n$. Since each particle $\mathbf{x}$ interacts with each particle $\mathbf{y}$, numerical implementation of equation (2) naturally leads to the computation of a dense matrix-vector product. However, thanks to local rank defaults, algebraic compression with *divide and conquer* process can be used to approximate accurately many parts of the full matrix with low rank pieces ($\mathcal{H}$-Matrix based methods [3,5]).

2 Overview of *OpenHmX*

OpenHmX [6] is an open-source toolbox for $\mathcal{H}$-Matrix computations, natively written in Matlab language. This library builds the compressed matrix representation of equation (2) in three steps.

Firstly, two independent binary trees are computed for the tri-dimensional clustering of the particles $\mathbf{x}$ and $\mathbf{y}$ respectively. These trees try to keep a well balanced spatial distribution with any spatial configuration. To do so, it uses both geometric and median cutting, dealing with the best way for all particle groups encountered at each depth of the tree. The cutting of the particles is carried on until the number of particles in the leaf reaches the value :

$$N_{leaf} \approx \log(N)^{\frac{3}{2}}. \tag{3}$$

Secondly, from the binary trees associated to particles $\mathbf{x}$ and $\mathbf{y}$, the allowed block interactions for algebraic compression is constructed hierarchically (fig. 2, left). In order to proceed, for each block defined by the sets of particles $(\mathbf{x}_i)_{i\in I}$ indexed by I and $(\mathbf{y}_j)_{j\in J}$ indexed by J, we use the admissibility condition :

$$D(\mathbf{x}_I,\mathbf{y}_J) > \max\left(d(\mathbf{x}_I), d(\mathbf{y}_I)\right), \tag{4}$$

where $D(\mathbf{x}_I,\mathbf{y}_J)$ is the distance between the centres of the two boxes surrounding each set of particles, and $d(\mathbf{x}_I)$ and $d(\mathbf{y}_J)$ are respectively the diameters of theses boxes.

Finally, an Adaptive Cross Approximation [3] is done for admissible interactions, completed by the standard full computation for close interactions, both with Matlab parallel computation. To do, we use a new criterion in order to evaluate the convergence of the ACA algorithm, instead of the classic one. We evaluate the distances between each set of particles $(\mathbf{x}_i)_{i\in I}$ and $(\mathbf{y}_j)_{j\in J}$, from their projections on the axis defined by the two centres of each dataset. We then compute exactly the Green kernel for some representative interactions, indexed by $I_0 \subset I$ and

N	Cores	τ	k	Direct (s)	$\mathcal{H}$-Matrix (s)	Memory (Mo)	L^2 error for 10^3 values
10^3	1	10^{-3}	0	0.52	0.71	6.9	$7.6\ 10^{-5}$
10^4	1	10^{-3}	0	5.4	7.5	130	$1.1\ 10^{-4}$
10^5	1	10^{-3}	0	-	74	$2.2\ 10^3$	$1.2\ 10^{-4}$
10^6	1	10^{-3}	0	-	783	$4.2\ 10^4$	$1.2\ 10^{-4}$
10^3	8	10^{-3}	0	0.18	0.31	7.1	$7.1\ 10^{-5}$
10^4	8	10^{-3}	0	1.1	1.5	130	$1.1\ 10^{-4}$
10^5	8	10^{-3}	0	-	11	$2.2\ 10^3$	$1.0\ 10^{-4}$
10^6	8	10^{-3}	0	-	120	$4.1\ 10^4$	$1.1\ 10^{-4}$
10^4	8	10^{-6}	0	1.1	2.4	243	$7.1\ 10^{-8}$
10^4	8	10^{-9}	0	1.1	3.4	379	$2.7\ 10^{-11}$
10^4	8	10^{-12}	0	1.1	4.4	520	$2.2\ 10^{-14}$
10^3	8	10^{-3}	5	0.19	0.41	8.4	$1.8\ 10^{-4}$
10^4	8	10^{-3}	10	1.6	1.8	183	$5.5\ 10^{-4}$
10^5	8	10^{-3}	20	-	22	$3.3\ 10^3$	$5.1\ 10^{-4}$
10^6	8	10^{-3}	30	-	260	$6.2\ 10^4$	$5.2\ 10^{-4}$

FIGURE 1 – Numerical results for a random spherical distribution

$J_0 \subset J$. With an user fixed threshold τ, the stopping criterion can now be driven numerically by the condition :

$$||G(\mathbf{x}_{I_0}, \mathbf{y}_{J_0}) - A_{I_0} B_{J_0}|| < \tau. \qquad (5)$$

3 Numericals results and conclusion

In this last section, we present few numerical results from *OpenHmX*, computed with Matlab R2013 on a 8 cores CPU at 2.7 GHz with 128 Go memory. To do, we simply consider a random spherical repartition of N particles both for $\mathbf{x}$ and $\mathbf{y}$ (fig. 2, right). We use the standard Helmholtz Green kernel (1), with various wave numbers k and thresholds τ. All results are given in figure 1.

To conclude, we propose in this paper a new open source Matlab library for $\mathcal{H}$-Matrix computation. Some noticeable details enrich standard approaches proposed in the state of the art, and numerical test provide good performances, accuracies and paralelization speed-up factors. In the future, we plan to focus on the memory cost and the high frequency problem.

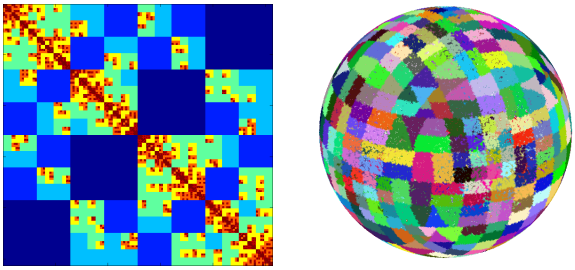

FIGURE 2 – Left : hierarchical interactions from far (blue) to close (red). Right : leaves of the binary hierarchical tree for a spherical distribution of particles.

Références

[1] Alouges, F., & Aussal, M. (2015). The sparse cardinal sine decomposition and its application for fast numerical convolution. *Numerical Algorithms*, 70(2), 427-448.

[2] Aussal, M. (2014). Méthodes numériques pour la spatialisation sonore, de la simulation à la synthèse binaurale. *Doctoral dissertation, Palaiseau, Ecole polytechnique.*

[3] Bebendorf, M., & Rjasanow, S. (2003). Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1), 1-24.

[4] Greengard, L., & Rokhlin, V. (1997). A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta numerica*, 6, 229-269.

[5] Hackbusch, W. (1999). A sparse matrix arithmetic based on cal h-matrices. part i : Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2), 89-108.

[6] *http ://www.cmap.polytechnique.fr/∼aussal/*