Journal of Computational Physics 351 (2017) 165–186

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Theory and implementation of $\mathcal{H}$-matrix based iterative and direct solvers for Helmholtz and elastodynamic oscillatory kernels

Stéphanie Chaillat *, Luca Desiderio, Patrick Ciarlet

Laboratoire POEMS UMR CNRS-INRIA-ENSTA, Université Paris-Saclay ENSTA-UMA, 828 Bd des Maréchaux, 91762 Palaiseau Cedex, France

ARTICLE INFO

Article history:
Received 9 June 2017
Received in revised form 6 September 2017
Accepted 7 September 2017
Available online 30 September 2017

Keywords:
Time-harmonic elastic waves
$\mathcal{H}$-matrices
Low-rank approximations
Estimators
Algorithmic complexity
Fast BEMs

ABSTRACT

In this work, we study the accuracy and efficiency of hierarchical matrix ($\mathcal{H}$-matrix) based fast methods for solving dense linear systems arising from the discretization of the 3D elastodynamic Green's tensors. It is well known in the literature that standard $\mathcal{H}$-matrix based methods, although very efficient tools for asymptotically smooth kernels, are not optimal for oscillatory kernels. $\mathcal{H}^2$-matrix and directional approaches have been proposed to overcome this problem. However the implementation of such methods is much more involved than the standard $\mathcal{H}$-matrix representation. The central questions we address are twofold. (i) What is the frequency-range in which the $\mathcal{H}$-matrix format is an efficient representation for 3D elastodynamic problems? (ii) What can be expected of such an approach to model problems in mechanical engineering? We show that even though the method is not optimal (in the sense that more involved representations can lead to faster algorithms) an efficient solver can be easily developed. The capabilities of the method are illustrated on numerical examples using the Boundary Element Method.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The 3D linear isotropic elastodynamic equation for the displacement field $\boldsymbol{u}$ (also called Navier equation) is given by

$$\operatorname{div}\sigma(\boldsymbol{u}) + \rho\omega^2\boldsymbol{u} = 0 \tag{1}$$

where $\omega > 0$ is the circular frequency. It is supplemented with appropriate boundary conditions which contain the data. The stress and strain tensors are respectively given by $\sigma(\boldsymbol{u}) = \lambda(\operatorname{div}\boldsymbol{u})\mathbf{I}_3 + 2\mu\varepsilon(\boldsymbol{u})$ and $\varepsilon(\boldsymbol{u}) = \frac{1}{2}\left([\nabla\boldsymbol{u}] + [\nabla\boldsymbol{u}]^{\mathsf{T}}\right)$, where $\mathbf{I}_3$ is the 3-by-3 identity matrix and $[\nabla\boldsymbol{u}]$ is the 3-by-3 matrix whose β-th column is the gradient of the β-th component of $\boldsymbol{u}$ for $1 \leq \beta \leq 3$, μ and λ are the Lamé parameters and ρ the density. Denoting $\kappa_p^2 = \rho\omega^2(\lambda + 2\mu)^{-1}$ and $\kappa_s^2 = \rho\omega^2\mu^{-1}$ the so-called P and S wavenumbers, the Green's tensor of the Navier equation is a 3-by-3 matrix-valued function expressed by

$$\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{\rho\omega^2}\left(\boldsymbol{curl}\,\boldsymbol{curl}_{\boldsymbol{x}}\left[\frac{e^{i\kappa_s|\boldsymbol{x}-\boldsymbol{y}|}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|}\mathbf{I}_3\right] - \nabla_{\boldsymbol{x}}\mathbf{div}_{\boldsymbol{x}}\left[\frac{e^{i\kappa_p|\boldsymbol{x}-\boldsymbol{y}|}}{4\pi|\boldsymbol{x}-\boldsymbol{y}|}\mathbf{I}_3\right]\right) \tag{2}$$

* Corresponding author.
E-mail addresses: stephanie.chaillat@ensta-paristech.fr (S. Chaillat), luca.desiderio@yahoo.it (L. Desiderio), patrick.ciarlet@ensta-paristech.fr (P. Ciarlet).

http://dx.doi.org/10.1016/j.jcp.2017.09.013
0021-9991/© 2017 Elsevier Inc. All rights reserved.

where the index $\boldsymbol{x}$ means that differentiation is carried out with respect to $\boldsymbol{x}$ and $\mathbf{div}_{\boldsymbol{x}}\mathbb{A}$ corresponds to the application of the divergence along each row of $\mathbb{A}$. One may use this tensor to represent the solution of (1). Alternately, one may use the tensor $\mathbf{T}_\omega(\boldsymbol{x},\boldsymbol{y})$, which is obtained by applying the traction operator

$$\boldsymbol{T} = 2\mu\frac{\partial}{\partial \boldsymbol{n}} + \lambda \boldsymbol{n}\,\mathrm{div} + \mu\,\boldsymbol{n}\times \boldsymbol{curl} \tag{3}$$

to each column of $\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})$: $\mathbf{T}_\omega(\boldsymbol{x},\boldsymbol{y}) = [\boldsymbol{T}_{\boldsymbol{y}}\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})]$ where the index $\boldsymbol{y}$ means that differentiation is carried out with respect to $\boldsymbol{y}$.

We consider the fast solution of dense linear systems of the form

$$\mathbb{A}\mathbf{p} = \mathbf{b}, \quad \mathbb{A}\in\mathbb{C}^{3N_c\times 3N_c} \tag{4}$$

where $\mathbb{A}$ is the matrix corresponding to the discretization of the 3-by-3 Green's tensors $\mathbf{U}_\omega(\boldsymbol{x}_i,\boldsymbol{y}_j)$ or $\mathbf{T}_\omega(\boldsymbol{x}_i,\boldsymbol{y}_j)$ for two clouds of N_c points $(\boldsymbol{x}_i)_{1\le i\le N_c}$ and $(\boldsymbol{y}_j)_{1\le j\le N_c}$. Here $\mathbf{p}$ is the unknown vector approximating the solution at $(\boldsymbol{x}_i)_{1\le i\le N_c}$ and $\mathbf{b}$ is a given right hand side that depends on the data. Such dense systems are encountered for example in the context of the Boundary Element Method [6,35].

If no compression or acceleration technique is used, the storage of such a system is of the order $O(N_c^2)$, the iterative solution (e.g. with GMRES) is $O(N_{\text{iter}}N_c^2)$ where N_{iter} is the number of iterations, while the direct solution (e.g. via LU factorizations) is $O(N_c^3)$. In the last decades, different approaches have been proposed to speed up the solution of dense systems. The most known method is probably the fast multipole method (FMM) proposed by Greengard and Rokhlin [23] which enables a fast evaluation of the matrix-vector products. We recall that the matrix-vector product is the crucial tool in the context of an iterative solution. Initially developed for N-body simulations, the FMM has then been extended to oscillatory kernels [22,17]. The method is now widely used in many application fields and has shown its capabilities in the context of mechanical engineering problems solved with the BEM [15,37].

An alternative approach designed for dense systems is based on the concept of hierarchical matrices ($\mathcal{H}$-matrices) [3]. The principle of $\mathcal{H}$-matrices is to partition the initial dense linear system, and then approximate it into a data-sparse one, by finding sub-blocks in the matrix that can be accurately estimated by low-rank matrices. In other terms, one further approximates the matrix $\mathbb{A}$ from (4). The efficiency of hierarchical matrices relies on the possibility to approximate, under certain conditions, the underlying kernel function by low-rank matrices. The approach has been shown to be very efficient for asymptotically smooth kernels (e.g. Laplace kernel). On the other hand, oscillatory kernels such as the Helmholtz or elastodynamic kernels, are not asymptotically smooth. In these cases, the method is not optimal [1]. To avoid the increase of the rank for high-frequency problems, directional $\mathcal{H}^2$-methods have been proposed [12,8]. $\mathcal{H}^2$-matrices are a specialization of hierarchical matrices. It is a multigrid-like version of $\mathcal{H}$-matrices that enables more compression, by factorizing some basis functions of the approximate separable expansion [7].

Since the implementation of $\mathcal{H}^2$ or directional methods is much more involved than the one of the standard $\mathcal{H}$-matrix, it is important to determine the frequency-range within which the $\mathcal{H}$-matrices are efficient for elastodynamic problems and what can be expected of such an approach to solve problems encountered in mechanical engineering. Previous works on $\mathcal{H}$-matrices for oscillatory kernels have mainly be devoted to the direct application to derive fast iterative solvers for 3D acoustics [13,36], a direct solver for 3D electromagnetism [28] or iterative solvers for 3D elastodynamics [31,29]. There is no discussion in these references on the capabilities and limits of the method for oscillatory kernels. We show in this work that even though the method is not optimal (in the sense that more efficient approaches can be proposed at the cost of a much more complex implementation effort), an efficient solver is easily developed. The capabilities of the method are illustrated on numerical examples using the Boundary Element Method.

The article is organized as follows. After reviewing general facts about $\mathcal{H}$-matrices in Section 2, $\mathcal{H}$-matrix based solvers and an estimator to certify the results of the direct solver are given in Section 3. Section 4 gives an overview of the Boundary Element Method for 3D elastodynamics. In Section 5, theoretical estimates for the application of $\mathcal{H}$-matrices to elastodynamics are derived. Section 6 presents the implementation issues for elastodynamics. Finally in Sections 7 and 8, we present numerical tests with varying number of points N_c for both representations of the solution ($\mathbf{U}_\omega$ or $\mathbf{T}_\omega$). In Section 7, numerical tests for the low frequency regime i.e. for a fixed frequency are reported and discussed. In Section 8, similar results are given for the high frequency regime i.e. for a fixed density of points per S-wavelength.

2. General description of the $\mathcal{H}$-LU factorization

2.1. $\mathcal{H}$-matrix representation

Hierarchical matrices or $\mathcal{H}$-matrices have been introduced by Hackbusch [24] to compute a data-sparse representation of some special dense matrices (e.g. matrices resulting from the discretization of non-local operators). The principle of $\mathcal{H}$-matrices is (i) to partition the matrix into blocks and (ii) to perform low-rank approximations of the blocks of the matrix which are known *a priori* (by using an admissibility condition) to be accurately approximated by low-rank decompositions. With these two ingredients it is possible to define fast iterative and direct solvers for matrices having a hierarchical representation. Using low-rank representations, the memory requirements and costs of a matrix-vector product are reduced. In addition, using $\mathcal{H}$-matrix arithmetic it is possible to derive fast direct solvers.

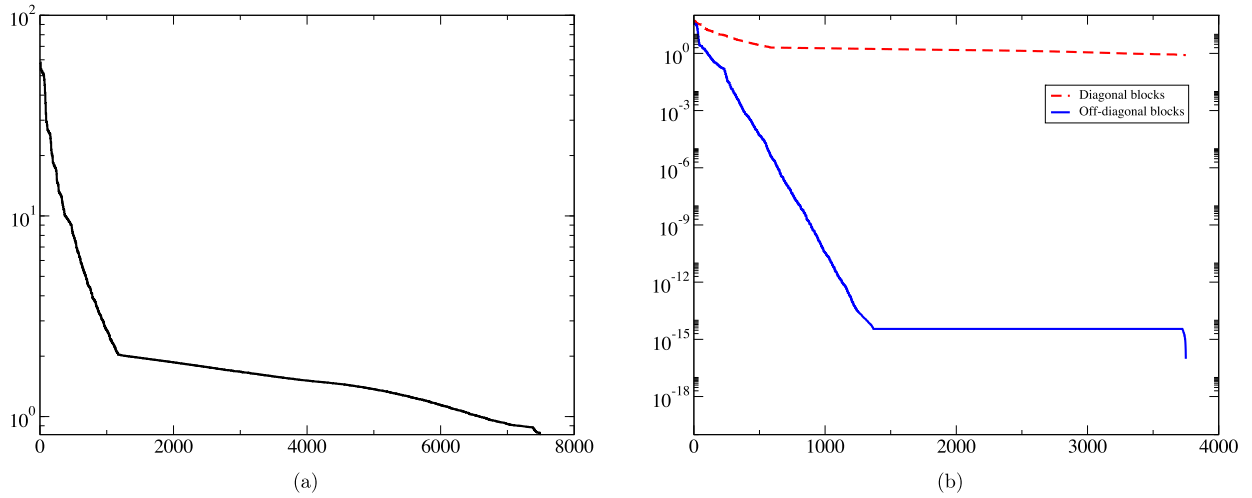


Fig. 1. (a) Decrease of the singular values of the complete matrix $\mathbb{G}_e$ corresponding to the discretization of the elastic Green's tensor for a cloud of 2500 points. (b) Decrease of the singular values of the diagonal and off-diagonal blocks of the same matrix decomposed into a 2-by-2 block matrix.

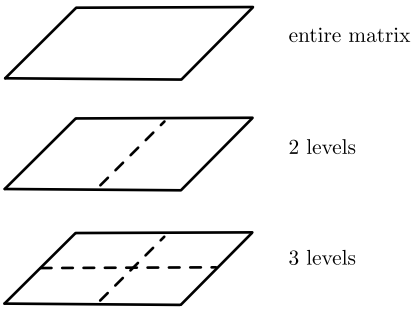


Fig. 2. Illustrative example: partition of the degrees of freedom.

Illustrative example. To illustrate the construction of an $\mathcal{H}$-matrix, we consider the matrix $\mathbb{G}_e$ resulting from the discretization of the 3D elastic Green's tensor $\mathbf{U}_\omega$ for a cloud of points located on the plate $(x_1, x_2, x_3) \in [-a, a] \times [-a, a] \times \{0\}$. The plate is discretized uniformly with a fixed density of 10 points per S-wavelength $\lambda_s = 2\pi/\kappa_s$. We fix the Poisson's ratio ν to $\nu = \frac{\lambda}{2(\lambda+\mu)} = 1/3$ and a non-dimensional frequency $\eta_S = \kappa_s a = 5\pi$ (e.g. $\rho\mu^{-1} = 1$, $\omega = 5\pi$, $\kappa_s = 5\pi$, $\kappa_p = \kappa_s/2$ and $a = 1$). As a result, the discretization consists of $N_d = 50$ points in each direction leading to $N_c = 2500$ points and a matrix of size 7500×7500. We recall that the numerical rank of a matrix $\mathbb{A}$ is

$$r(\varepsilon) := \min\{r \quad | \quad \|\mathbb{A} - \mathbb{A}_r\| \leq \varepsilon \|\mathbb{A}\|\} \tag{5}$$

where $\mathbb{A}_r$ defines the singular value decomposition (SVD) of $\mathbb{A}$ keeping only the r largest singular values and $\varepsilon > 0$ is a given parameter. In Fig. 1a, we report the decrease of the singular values of the matrix $\mathbb{G}_e$. As expected, the decay of the singular values is very slow such that the matrix cannot be approximated by a low-rank decomposition. Now, we partition the plate into two parts of equal area (see Fig. 2) and subdivide the matrix accordingly into 4 subblocks. Fig. 1b gives the decrease of the singular values both for diagonal and off-diagonal blocks. It illustrates the possibility to accurately represent off-diagonal blocks by low-rank matrices while diagonal blocks cannot have low-rank representations.

If we keep subdividing the full rank blocks in a similar manner, we observe in Fig. 3 that diagonal blocks are always full rank numerically, i.e. with respect to (5) where we have chosen $\varepsilon = 10^{-4}$, while off-diagonal blocks become accurately approximated by a low-rank decomposition after some iterations of the subdivision process. This academic example illustrates the concept of data-sparse matrices used to derive fast algorithms. Using the hierarchical structure in addition to low-rank approximations, significant savings are obtained both in terms of computational times and memory requirements.

Clustering of the unknowns. The key ingredient of hierarchical matrices is the recursive block subdivision. On the illustrative example, the subdivision of the matrix is conducted by a recursive subdivision of the geometry, i.e. a recursive subdivision of the plan into partitions of equal areas. The first step prior to the partition of the matrix is thus a partitioning based on the geometry of the set of row and column indices of the matrix $\mathbb{A}$. The purpose is to permute the indices in the matrix to reflect the physical distance and thus interaction between degrees of freedom. Consecutive indices should correspond to DOFs that interact at close range. For the sake of clarity, in this work $\mathbb{A}$ is defined by the same set of indices $I = \{1, \ldots, n\}$

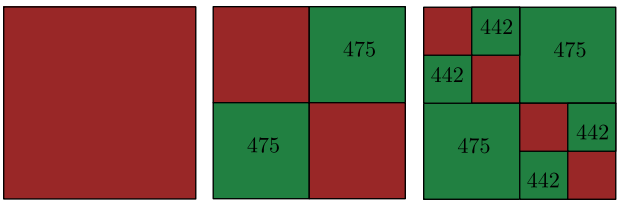


Fig. 3. Computed numerical ranks of each block of the matrix $\mathbb{G}_\epsilon$ to achieve an accuracy of 10^{-4}, i.e. if singular values smaller than 10^{-4} are neglected in the singular value decomposition.

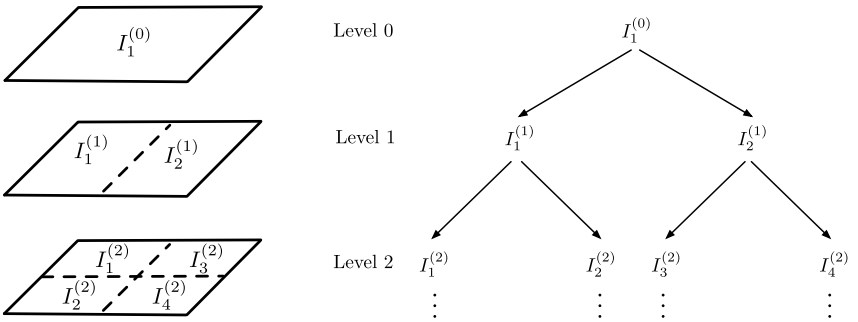


Fig. 4. Illustration of the clustering of the degrees of freedom: (a) partition of the degrees of freedom in the domain and (b) corresponding binary tree.

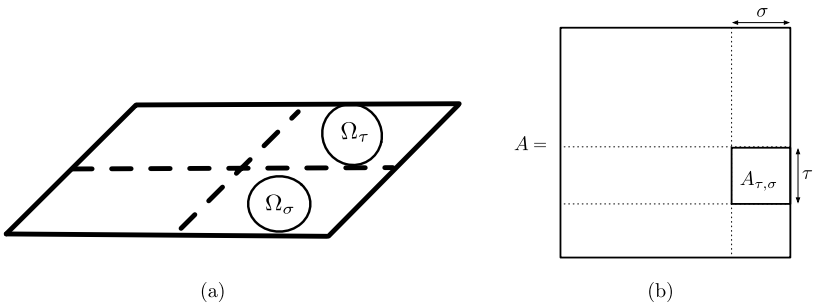


Fig. 5. Illustration of the construction of the block cluster tree: (a) Clustering of the unknowns on the geometry and (b) corresponding block clustering in the matrix.

for rows and columns. A binary tree $\mathcal{T}_I$ is used to drive the clustering. Each node of the tree defines a subset of indices $\sigma \subset I$ and each subset corresponds to a part in the partition of the domain, see Fig. 4 for the case of the illustrative example. There exist different approaches to perform the subdivision [25]. We consider the simplest possible one: based on a geometric argument. For each node in the tree, we determine the box enclosing all the points in the cloud and subdivide it into 2 boxes, along the largest dimension. The subdivision is stopped when a minimum number of DOFs per box is reached ($N_{\text{leaf}} = 100$ in the following). For uniform meshes, this strategy defines a balanced binary tree [3] such that the number of levels in the tree $\mathcal{T}_I$ is given by $L(I) = \lceil \log_2(\frac{|I|}{N_{\text{leaf}}}) \rceil \leq \log_2 |I| - \log_2 N_{\text{leaf}} + 1$. Note that for this step only geometrical information is needed.

Subdivision of the matrix. After the clustering of the unknowns is performed, a block cluster representation $\mathcal{T}_{I\times I}$ of the matrix $\mathbb{A}$ is defined by going through the cluster tree $\mathcal{T}_I$. Each node of $\mathcal{T}_{I\times I}$ contains a pair (σ, τ) of indices of $\mathcal{T}_I$ and defines a block of $\mathbb{A}$ (see Fig. 5). This uniform partition defines a block structure of the matrix with a full pattern of $4^{L(I)-1}$ blocks, in particular every node of the tree at the leaf level is connected with all the other nodes at the leaf level (Fig. 6a). Fig. 1b shows that this partition is not optimal. As a matter of fact, some parts of the matrix $\mathbb{A}$ can accurately be approximated by a low-rank matrix at a higher level (i.e. for larger clusters). Such blocks are said to be *admissible*. A hierarchical representation $\mathcal{P} \subset \mathcal{T}_{I\times I}$ that uses the cluster tree $\mathcal{T}_I$ and the existence of *admissible* blocks is more appropriate (Fig. 6b). Starting from the initial matrix, each block is recursively subdivided until it is either *admissible* or the leaf level is achieved. In the illustrative example, only diagonal blocks are subdivided due to the very simple 2D geometry (see again Fig. 3, red-colored blocks). For more complex 3D geometries, an admissibility condition based on the geometry and the interaction distance between points

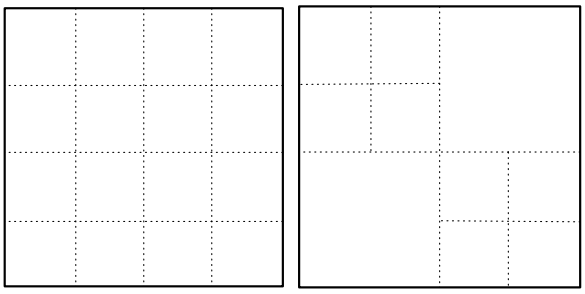


Fig. 6. (a) Block cluster representation $\mathcal{T}_{I\times I}$ for the illustrative example (full structure); (b) Hierarchical partition $\mathcal{P} \subset \mathcal{T}_{I\times I}$ of the same matrix based on the admissibility condition (sparse structure).

is used to determine *a priori* the *admissible* blocks. For more details on the construction of the block cluster tree, we refer the interested reader to [11]. The partition $\mathcal{P}$ is subdivided into two subsets $\mathcal{P}^{\text{ad}}$ and $\mathcal{P}^{\text{non-ad}}$ reflecting the possibility for a block $\tau \times \sigma$ to be either *admissible*, i.e. $\tau \times \sigma \in \mathcal{P}^{\text{ad}}$; or *non-admissible*, i.e. $\tau \times \sigma \in \mathcal{P}^{\text{non-ad}}$. It is clear that $\mathcal{P} = \mathcal{P}^{\text{ad}} \cup \mathcal{P}^{\text{non-ad}}$. To sum up, the blocks of the partition can be of 3 types: at the leaf level a block can be either an *admissible* block or a *non-admissible* block, at a non-leaf level a block can be either an *admissible* block or an $\mathcal{H}$-matrix (i.e a block that will be subsequently hierarchically subdivided).

At this point, we need to define the sparsity pattern introduced by L. Grasedyck [21]. For sparse matrices, the sparsity pattern gives the maximum number of non-zero entries per row. Similarly for a hierarchical matrix defined by the partition $\mathcal{P} \subset \mathcal{T}_{I\times I}$, the sparsity pattern of a row cluster $\tau \in \mathcal{T}_I$, resp. a column cluster $\sigma \in \mathcal{T}_I$, is

$$C_{sp}(\tau) = |\{\sigma \in \mathcal{T}_I : \tau \times \sigma \in \mathcal{P}\}|, \text{ resp. } C_{sp}(\sigma) = |\{\tau \in \mathcal{T}_I : \tau \times \sigma \in \mathcal{P}\}|.$$

It is convenient to define the overall sparsity pattern (for row and column clusters):

$$C_{sp} = \max\{\max_{\tau\in\mathcal{T}_I} C_{sp}(\tau),\ \max_{\sigma\in\mathcal{T}_I} C_{sp}(\sigma)\}.$$

Special case of asymptotically smooth kernels. $\mathcal{H}$-matrix representations have been derived for some specific problems and will not result in efficient algorithms for all equations or matrices. The crucial point is to know *a priori* (i) if savings will be obtained when trying to approximate *admissible* blocks with a sum of separated variable functions and (ii) which blocks are *admissible* since the explicit computation of the rank of all the blocks would be too expensive. In the case of asymptotically smooth kernels $G(\boldsymbol{x}, \boldsymbol{y})$, it is proved that under some *a priori* condition on the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, the kernel is a degenerate function (see Section 5). After discretization, this property is reformulated as the efficient approximation of blocks of the matrix by low-rank matrices. The Laplace Green's function is an example of asymptotically smooth kernel for which $\mathcal{H}$-matrix representations have been shown to be very efficient. Since the 3D elastodynamics Green's tensor, similarly to the Helmholtz Green's function, is a linear combination of derivatives of the 3D Laplace Green's function with coefficients depending on the circular frequency ω, this work is concerned with the determination of the frequency range for which hierarchical representations can be successful for 3D elastodynamics.

2.2. Algorithms to perform low-rank approximations

Once the *admissible* blocks are determined, an accurate rank-revealing algorithm is applied to determine low-rank approximations. Such an algorithm must be accurate (i.e. its result, the computed numerical rank, must be as small as possible) to avoid unnecessary computational costs. The truncated Singular Value Decomposition (SVD) [20] gives the best low-rank approximation (Eckart–Young theorem) for unitary invariant norms (e.g. Frobenius or spectral norm). Thus it produces an approximation with the smallest possible numerical rank for a given prescribed accuracy. But the computation of the SVD is expensive, i.e. in the order of $O(\max(m,n) \times \min(m,n)^2)$ for an $m \times n$ matrix, and in addition it requires the computation of all the entries of $\mathbb{A}$. In the context of the $\mathcal{H}$-matrices, the use of the SVD would induce the undesired need to assemble the complete matrix.

The adaptive cross approximation (ACA) [4,5] offers an interesting alternative to the SVD since it produces a quasi-optimal low-rank approximation without requiring the assembly of the complete matrix. The starting point of the ACA is that every matrix of rank r is the sum of r matrices of rank 1. The ACA is thus a greedy algorithm that improves the accuracy of the approximation by adding iteratively rank-1 matrices. At iteration k, the matrix is split into the rank k approximation $\mathbb{B}_k = \sum_{\ell=1}^{k} \mathbf{u}_\ell \mathbf{v}_\ell^* = \mathbb{U}_k \mathbb{V}_k^*$ with $\mathbb{U}_k \in \mathbb{C}^{m\times k}$, $\mathbb{V}_k \in \mathbb{C}^{n\times k}$, and the residual $\mathbb{R}_k = \mathbb{A} - \sum_{\ell=1}^{k} \mathbf{u}_\ell \mathbf{v}_\ell^*$; $\mathbb{A} = \mathbb{B}_k + \mathbb{R}_k$. The information is shifted iteratively from the residual to the approximant. A stopping criterion is used to determine the appropriate rank to achieve the required accuracy. A straightforward choice of stopping criterion is

$$\text{until} \quad \|\mathbb{A} - \mathbb{B}_k\|_F \leq \varepsilon_{\text{ACA}} \|\mathbb{A}\|_F \tag{6}$$

where $\varepsilon_{\text{ACA}} > 0$ is a given parameter, and $\|.\|_F$ denotes the Frobenius norm. In the following, we denote r_{ACA} the numerical rank obtained by the ACA for a required accuracy ε_{ACA}. The complexity of this algorithm to generate an approximation of rank r_{ACA} is $O(r_{\text{ACA}}mn)$.

There are various ACAs that differ by the choice of the best pivot at each iteration. The simplest approach is the so-called fully-pivoted ACA and it consists in choosing the pivot as the largest entry in the residual. But similarly to the SVD, it requires the computation of all the entries of $\mathbb{A}$ to compute the pivot indices. It is not an interesting option for the construction of $\mathcal{H}$-matrices.

The partially-pivoted ACA proposes an alternative approach to choose the pivot avoiding the assembly of the complete matrix. The idea is to maximize alternately the residual for only one of the two indices and to keep the other one fixed. With this strategy, only one row and one column is assembled at each iteration. More precisely, at iteration k, given $\mathbb{B}_k$ and assuming the row index $\hat{i}$ is known the algorithm is given by the following six steps:

1. Generation of the rows $\mathbf{a} := \mathbb{A}^*\mathbf{e}_{\hat{i}}$ and $\mathbb{R}_k^*\mathbf{e}_{\hat{i}} = \mathbf{a} - \sum_{\ell=1}^{k}(u_\ell)_{\hat{i}}\mathbf{v}_\ell$
2. Find the column index $\hat{j} := \text{argmax}_j|(\mathbb{R}_k)_{\hat{i}j}|$ and compute $\gamma_{k+1} = (\mathbb{R}_k)_{\hat{i},\hat{j}}^{-1}$
3. Generation of the columns: $\mathbf{a} := \mathbb{A}\mathbf{e}_{\hat{j}}$ and $\mathbb{R}_k\mathbf{e}_{\hat{j}} = \mathbf{a} - \sum_{\ell=1}^{k}\mathbf{u}_\ell(v_\ell)_{\hat{j}}$
4. Find the next row index $\hat{i} := \text{argmax}_i|(\mathbb{R}_k)_{i\hat{j}}|$
5. Compute vectors $\mathbf{u}_{k+1} := \gamma_{k+1}\mathbb{R}_k\mathbf{e}_{\hat{j}}$, $\mathbf{v}_{k+1} := \mathbb{R}_k^*\mathbf{e}_{\hat{i}}$
6. Update the approximation $\mathbb{B}_{k+1} = \mathbb{B}_k + \mathbf{u}_{k+1}\mathbf{v}_{k+1}^*$

With this approach however, approximates and residuals are not computed explicitly nor stored, so the stopping criteria (6) needs to be adapted. The common choice is a stagnation-based error estimate which is computationally inexpensive. The algorithm stops when the new rank-1 approximation does not improve the accuracy of the approximation. Since at each iteration k, $\mathbb{B}_k - \mathbb{B}_{k-1} = \mathbf{u}_k\mathbf{v}_k^*$, the stopping criteria now reads:

$$\text{until } \|\mathbf{u}_k\|_2\|\mathbf{v}_k\|_2 \leq \varepsilon_{\text{ACA}}\|\mathbb{B}_k\|_F. \tag{7}$$

The complexity of the partially-pivoted ACA is reduced to $O(r_{\text{ACA}}^2(m+n))$. Since the partially-pivoted ACA is a heuristic method, there exist counter-examples where ACA fails [14] and variants have been designed to improve the robustness of the method. Nevertheless, in all the numerical examples we have performed, we do not need them. It is worth mentioning that other approaches exist such as fast multipole expansions [33,23], panel clustering [34,26], quadrature formulas [9] or interpolations [30]. These approaches combine approximation techniques and linear algebra. The advantages of the ACA are to be purely algebraic and easy to implement.

From now on, we shall write the result $\mathbb{B}_{r_{\text{ACA}}}$ as $\mathbb{B}$ for short, and $\mathbb{B} \approx \mathbb{A}$ or $\mathbb{A} \approx \mathbb{B}$.

2.3. Extension to problems with vector unknowns

One specificity of this work is to consider $\mathcal{H}$-matrices and ACA in the context of systems of partial differential equations with vector unknowns. There exist a lot of works both theoretical and numerical on the ACA for scalar problems, in particular on the selection of non-zero pivots (since they are used to normalize the new rank-1 approximation). Indeed for scalar problems, it is straightforward to find the largest non-zero entry in a given column. For problems with vector unknowns in $\mathbb{R}^d$, the system has a block structure, i.e. each pair of nodes on the mesh does not define a single entry but rather a $d \times d$ subblock in the complete matrix. This happens for example for 3D elastodynamics where the Green's tensor is a 3×3 subblock.

Different strategies can be applied to perform the ACA on matrices with a block structure. The first strategy consists in ordering the system matrix such that it is composed of $9 = 3^2$ subblocks of size $N_c \times N_c$ where N_c is the number of points in the cloud. In 3D elastodynamics, this corresponds to the following partitioning of the matrix (below the solution is represented with $\mathbf{U}_\omega$):

$$\begin{bmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} & \mathbb{A}_{13} \\ \mathbb{A}_{21} & \mathbb{A}_{22} & \mathbb{A}_{23} \\ \mathbb{A}_{31} & \mathbb{A}_{32} & \mathbb{A}_{33} \end{bmatrix}, \quad (\mathbb{A}_{\alpha\beta})_{ij} = (\mathbf{U}_\omega)_{\alpha\beta}(\boldsymbol{x}_i, \boldsymbol{y}_j) \;\; 1 \leq \alpha, \beta \leq 3, \;\; 1 \leq i, j \leq N_c.$$

Then each submatrix is approximated independently with the conventional scalar ACA. This strategy is used in [29]. It is well suited for iterative solvers. But it cannot be adapted straightforwardly in the context of direct solvers since the recursive 2×2 block structure inherited from the binary tree is used. Indeed even though it is possible to determine the LU decomposition of each block of $\mathbb{A}$ (see Section 3.2), such that

$$\begin{bmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} & \mathbb{A}_{13} \\ \mathbb{A}_{21} & \mathbb{A}_{22} & \mathbb{A}_{23} \\ \mathbb{A}_{31} & \mathbb{A}_{32} & \mathbb{A}_{33} \end{bmatrix} \approx \begin{bmatrix} \mathbb{L}_{11}\mathbb{U}_{11} & \mathbb{L}_{12}\mathbb{U}_{12} & \mathbb{L}_{13}\mathbb{U}_{13} \\ \mathbb{L}_{21}\mathbb{U}_{21} & \mathbb{L}_{22}\mathbb{U}_{22} & \mathbb{L}_{23}\mathbb{U}_{23} \\ \mathbb{L}_{31}\mathbb{U}_{31} & \mathbb{L}_{32}\mathbb{U}_{32} & \mathbb{L}_{33}\mathbb{U}_{33} \end{bmatrix},$$

on the other hand it would be very expensive to deduce the LU decomposition of $\mathbb{A}$ from these LU decompositions since what we are looking for is a decomposition of the kind

$$\begin{bmatrix} \mathbb{L}_1 & \mathbb{O} & \mathbb{O} \\ \mathbb{L}_2 & \mathbb{L}_3 & \mathbb{O} \\ \mathbb{L}_4 & \mathbb{L}_5 & \mathbb{L}_6 \end{bmatrix} \begin{bmatrix} \mathbb{U}_1 & \mathbb{U}_2 & \mathbb{U}_3 \\ \mathbb{O} & \mathbb{U}_4 & \mathbb{U}_5 \\ \mathbb{O} & \mathbb{O} & \mathbb{U}_6 \end{bmatrix},$$

i.e. 12 factors instead of 18. A solution would be to replace the binary by a ternary tree but in that case the clustering process would be more complex.

The second naive approach consists in considering the complete matrix as a scalar matrix, i.e. to forget about the block structure. While appealing this approach fails in practice for 3D elastodynamics, whatever ordering is used, due to the particular structure of the matrix. Rewriting the Green's tensors (2)–(3) component-wise [6] and using the Einstein summation convention, it reads

$$(\mathbf{U}_\omega)_{\alpha\beta}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi\varrho}(a_1\delta_{\alpha\beta} + a_2\varrho_{,\alpha}\varrho_{,\beta}),$$
$$(\mathbf{T}_\omega)_{\alpha\beta}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi\varrho}\Big[2a_3\varrho_{,\alpha}\varrho_{,\gamma}\varrho_{,\beta} + a_4(\delta_{\alpha\beta}\varrho_{,\gamma} + \delta_{\gamma\beta}\varrho_{,\alpha}) + a_5\delta_{\alpha\gamma}\varrho_{,\beta}\Big]n_\gamma(\boldsymbol{y})$$

where the constants $a_1, \ldots, a_5$ depend only on the mechanical properties, $\varrho = \|\boldsymbol{x}-\boldsymbol{y}\|$ and $\varrho_{,\alpha} = \frac{\partial}{\partial y_\alpha}\varrho(\boldsymbol{x},\boldsymbol{y})$. As a result, as soon as $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to the same plane (let's say $x_3 = y_3$ for simplicity) the Green's tensors simplify to

$$\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi\varrho}\begin{bmatrix} a_1+a_2\varrho_{,1}^2 & a_2\varrho_{,1}\varrho_{,2} & 0 \\ a_2\varrho_{,1}\varrho_{,2} & a_1+a_2\varrho_{,1}^2 & 0 \\ 0 & 0 & a_1 \end{bmatrix}, \ \mathbf{T}_\omega(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi\varrho}\begin{bmatrix} 0 & 0 & a_4\varrho_{,1} \\ 0 & 0 & a_4\varrho_{,2} \\ a_4\varrho_{,1} & a_4\varrho_{,2} & 0 \end{bmatrix}. \tag{8}$$

It is then clear that the complete matrix $\mathbb{A}$ composed of such subblocks is a reducible matrix (there are decoupled groups of unknowns after discretization). While the fully-pivoted ACA will succeed in finding the best pivot to perform low-rank approximations, the partially pivoted ACA will only cover parts of the matrix resulting in a non-accurate approximation of the initial matrix. As a result, this approach cannot be applied for 3D elastodynamics. Another strategy would be to adapt the choice of the pivot to reducible matrices, i.e. to cover all the decoupled groups of unknowns. This is possible with the use of the ACA+ algorithm that defines a reference column and a reference row along whose the pivots are looked for [11]. Even though this approach could fix the problem it does not seem to be the best suited one since it does not use the vector structure of 3D elastodynamic problems.

To sum up, it is now important to use an algorithm that takes into account the particular structure of our matrix such that the vector ACA does not rely anymore on a rank-1 update but instead on a rank-3 update. The central question is then how to find the pivot used for this rank-3 update instead. There are 3 possible strategies:

1. To look for the largest scalar pivot, determine the corresponding point in the cloud and update all 3 DOFs linked to this point simultaneously. This approach is not stable since the 3×3 subblock pivot may not be invertible (cf. a discretized version of (8)).
2. The second strategy is to look for the 3×3 subblock with the largest norm. Again this approach fails in practice since the 3×3 subblock pivot may not be invertible.
3. The third strategy used in this work is to compute the singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3$ of every candidate subblock. In order to achieve convergence and to avoid singular pivots, the safest approach consists in choosing the pivot corresponding to the subblock with the largest σ_3. A similar approach is used for electromagnetism in [32].

Remark 2.1. It is worth noting that for some specific configurations the 3D elastodynamic double layer potential $\mathbf{T}_\omega$ may lead to a matrix with only singular subblocks. In such cases, the randomized SVD [27] is preferred to the vector ACA.

3. $\mathcal{H}$-matrix based iterative and direct solvers

3.1. $\mathcal{H}$-matrix based iterative solver

Once the $\mathcal{H}$-matrix representation of a matrix is computed, it is easy to derive an $\mathcal{H}$-matrix based iterative solver. The only operation required is an efficient matrix-vector product. It is performed hierarchically by going through the block cluster tree $\mathcal{P}$. At the leaf level, there are two possibilities. If the block of size $m\times n$ does not admit a low-rank approximation (*non-admissible* block), then the standard matrix-vector product is used with cost $O(mn)$. Otherwise, the block is marked as *admissible* such that a low-rank approximation has been computed: $\mathbb{A}_{\tau\times\sigma} \approx \mathbb{B}_{\tau\times\sigma}$. The cost of this part of the matrix-vector product is then reduced from $O(mn)$ to $O(r_{\text{ACA}}(m+n))$ where r_{ACA} is the numerical rank of the block $\mathbb{B}_{\tau\times\sigma}$ computed with the ACA.

3.2. $\mathcal{H}$-LU factorization and direct solver

One of the advantages of the $\mathcal{H}$-matrix representation is the possibility to derive a fast direct solver. Due to the hierarchical block structure of a $\mathcal{H}$-matrix, the LU factorization is performed recursively on 2×2 block matrices of the form

$$\begin{pmatrix} \mathbb{A}_{\tau_1\times\sigma_1} & \mathbb{A}_{\tau_1\times\sigma_2} \\ \mathbb{A}_{\tau_2\times\sigma_1} & \mathbb{A}_{\tau_2\times\sigma_2} \end{pmatrix} = \begin{pmatrix} \mathbb{L}_{\tau_1\times\sigma_1} & \mathbb{O} \\ \mathbb{L}_{\tau_2\times\sigma_1} & \mathbb{L}_{\tau_2\times\sigma_2} \end{pmatrix} \begin{pmatrix} \mathbb{U}_{\tau_1\times\sigma_1} & \mathbb{U}_{\tau_1\times\sigma_2} \\ \mathbb{O} & \mathbb{U}_{\tau_2\times\sigma_2} \end{pmatrix}$$

For such block matrices, we recall that the LU factorization is classically decomposed into 4 steps:

1. LU decomposition to compute $\mathbb{L}_{\tau_1\times\sigma_1}$ and $\mathbb{U}_{\tau_1\times\sigma_1}$: $\mathbb{A}_{\tau_1\times\sigma_1} = \mathbb{L}_{\tau_1\times\sigma_1}\mathbb{U}_{\tau_1\times\sigma_1}$;
2. Compute $\mathbb{U}_{\tau_1\times\sigma_2}$ from $\mathbb{A}_{\tau_1\times\sigma_2} = \mathbb{L}_{\tau_1\times\sigma_1}\mathbb{U}_{\tau_1\times\sigma_2}$;
3. Compute $\mathbb{L}_{\tau_2\times\sigma_1}$ from $\mathbb{A}_{\tau_2\times\sigma_1} = \mathbb{L}_{\tau_2\times\sigma_1}\mathbb{U}_{\tau_1\times\sigma_1}$;
4. LU decomposition to compute $\mathbb{L}_{\tau_2\times\sigma_2}$ and $\mathbb{U}_{\tau_2\times\sigma_2}$: $\mathbb{A}_{\tau_2\times\sigma_2} - \mathbb{L}_{\tau_2\times\sigma_1}\mathbb{U}_{\tau_1\times\sigma_2} = \mathbb{L}_{\tau_2\times\sigma_2}\mathbb{U}_{\tau_2\times\sigma_2}$.

The obvious difficulty is the fact that the process is recursive (cf steps 1 and 4). In addition, Step 4 requires the addition and multiplication of blocks which may come in different formats. In more general terms, the difficulty is to evaluate

$$\mathbb{A}'''_{\tau\times\sigma} \leftarrow \mathbb{A}_{\tau\times\sigma} + \mathbb{A}'_{\tau\times\sigma'}\mathbb{A}''_{\tau'\times\sigma}.$$

Since each block may belong to three different classes this results into 27 cases to take into account. For example, if $\mathbb{A}_{\tau\times\sigma}$ is a full matrix, $\mathbb{A}'_{\tau\times\sigma'}$ an $\mathcal{H}$-matrix, i.e. a *non-admissible* block at a non-leaf level and $\mathbb{A}''_{\tau'\times\sigma}$ a matrix having a low-rank representation, then the computation is efficiently decomposed into the following steps. First, $\mathbb{A}''_{\tau'\times\sigma}$ being a low-rank matrix it is decomposed into $\mathbb{A}''_{\tau'\times\sigma} \approx \mathbb{B}''_{\tau'\times\sigma} = \mathbb{U}\mathbb{V}^*$ and using the fast $\mathcal{H}$-matrix/vector product the product between $\mathbb{A}'_{\tau\times\sigma'}$ and each column of $\mathbb{U}$ is evaluated. In a second step, the result is multiplied to $\mathbb{V}^*$ and the final matrix is directly a full matrix. A more detailed review of the 27 cases is presented in [19].

It is known that the numerical rank obtained by the ACA is not optimal. Similarly the numerical rank for blocks obtained after an addition and a multiplication and stored in a low-rank or $\mathcal{H}$-matrix format is not optimal. All these blocks are further recompressed to optimize the numerical ranks. Starting from the low-rank approximation $\mathbb{B}_{\tau\times\sigma} = \mathbb{U}\mathbb{V}^*$, the reduced QR-decompositions of $\mathbb{U}$ and $\mathbb{V}$ are computed, i.e.

$$\mathbb{B}_{\tau\times\sigma} = \mathbb{U}\mathbb{V}^* = \mathbb{Q}_U\mathbb{R}_U\mathbb{R}_V^*\mathbb{Q}_V^*.$$

Recall that the only decomposition that yields the optimal numerical rank is the truncated singular value decomposition. It is thus applied to the reduced matrix $\mathbb{R}_U\mathbb{R}_V^* \approx \mathbb{P}\mathbb{S}\mathbb{L}^*$ such that the cost of this step is reasonable.

$$\mathbb{B}_{\tau\times\sigma} = \mathbb{Q}_U\mathbb{R}_U\mathbb{R}_V^*\mathbb{Q}_V^* = (\mathbb{Q}_U\mathbb{P}\mathbb{S}^{1/2})(\mathbb{S}^{1/2}\mathbb{L}^*\mathbb{Q}_V^*) = \mathbb{U}'\mathbb{V}'^*.$$

During the construction of the $\mathcal{H}$-matrix representation, the parameter ε_{ACA} is also used to determine the number of singular values selected during the recompression process. Moreover, in the case of a recompression after an addition and a multiplication, i.e. when the format of a block is modified, we introduce the parameter ε_{LU} to determine the level of accuracy required.

3.3. Proposition of an estimator to certify the results of the $\mathcal{H}$-LU direct solver

Since the $\mathcal{H}$-LU direct solver is based on a heuristic method (the ACA) to perform the low-rank approximations and due to the various modifications on the $\mathcal{H}$-matrix performed during the LU factorization, it is relevant to propose a simple and efficient way to certify the obtained results. To this aim, we propose an estimator. We consider the initial system

$$\mathbb{A}\mathbf{x} = \mathbf{b}.$$

We denote $\mathbb{A}_{\mathcal{H}}$ the $\mathcal{H}$-matrix representation of $\mathbb{A}$ (in which low-rank approximations have been performed) and $\mathbb{L}_{\mathcal{H}}\mathbb{U}_{\mathcal{H}} \approx \mathbb{A}_{\mathcal{H}}$ its $\mathcal{H}$-LU factorization. The solution of the approximated system is $\mathbf{x}_0$: $\mathbb{L}_{\mathcal{H}}\mathbb{U}_{\mathcal{H}}\mathbf{x}_0 = \mathbf{b}$. Our aim is to give an upper bound of $\|\mathbf{b} - \mathbb{A}\mathbf{x}_0\|_2$. Since

$$\mathbf{b} - \mathbb{A}\mathbf{x}_0 = \mathbf{b} - \mathbb{A}_{\mathcal{H}}\mathbf{x}_0 + \mathbb{A}_{\mathcal{H}}\mathbf{x}_0 - \mathbb{A}\mathbf{x}_0$$

this yields for $\alpha \in \{2, F\}$[1]

$$\frac{\|\mathbf{b} - \mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2} \leq \frac{1}{\|\mathbf{b}\|_2}(\|\mathbf{b} - \mathbb{A}_{\mathcal{H}}\mathbf{x}_0\|_2 + \|\mathbb{A}_{\mathcal{H}} - \mathbb{A}\|_\alpha\|\mathbf{x}_0\|_2) = \frac{1}{\|\mathbf{b}\|_2}(\delta + \delta_{\mathcal{H},\alpha}\|\mathbf{x}_0\|_2) \quad \text{where } \delta = \|\mathbf{b} - \mathbb{A}_{\mathcal{H}}\mathbf{x}_0\|_2 \text{ and } \delta_{\mathcal{H},\alpha} = \|\mathbb{A}_{\mathcal{H}} - \mathbb{A}\|_\alpha. \tag{9}$$

[1] Recall that $\|\mathbb{B}\|_2 \leq \|\mathbb{B}\|_F$ for all square matrices.

It is worth noting that $\delta_{\mathcal{H},\alpha} = \|\mathbb{A}_{\mathcal{H}} - \mathbb{A}\|_\alpha$ estimates the accuracy of the $\mathcal{H}$-matrix representation (i.e. the influence of the parameter ε_{ACA} in the computation of the low-rank approximations) while $\delta = \|\mathbf{b} - \mathbb{A}_{\mathcal{H}}\mathbf{x}_0\|_2$ accounts for the stability of the $\mathcal{H}$-LU factorization (i.e. the influence of the parameter ε_{LU} in the LU factorization). The evaluations of δ, $\|\mathbf{b}\|_2$ and $\|\mathbf{x}_0\|_2$ reduce to the computation of the norm of a vector. The most expensive part is $\delta_{\mathcal{H},\alpha}$ but its evaluation is performed with the Frobenius norm to reduce the cost. Also this term does not depend on the right hand side meaning that for multiple right hand sides this error estimate is not expensive.

4. 3D elastodynamics boundary element method

Let's consider a bounded domain Ω^- in $\mathbb{R}^3$ representing an obstacle with a closed Lipschitz boundary $\Gamma := \partial\Omega^-$. Let Ω^+ denote the associated exterior domain $\mathbb{R}^3\backslash\overline{\Omega^-}$ and $\boldsymbol{n}$ its outward unit normal vector field on its boundary Γ. We consider the propagation of time-harmonic waves in a three-dimensional isotropic and homogeneous elastic medium modeled by the Navier equation (1). The following results about traces of vector fields and integral representations of time-harmonic elastic fields can be found in [16].

4.1. Traces, integral representation formula and integral equations

We denote by $H^s_{loc}(\Omega^\pm)$ and $H^s(\Gamma)$ the standard (local in the case of the exterior domain) complex valued Hilbertian Sobolev spaces of order $s \in \mathbb{R}$ ($|s| \leq 1$ for $H^s(\Gamma)$) defined on $\Omega^\pm$ and Γ respectively (with the convention $H^0 = L^2$). Spaces of vector functions will be denoted by boldface letters, thus $\boldsymbol{H}^s = (H^s)^3$. We set $\boldsymbol{\Delta}^*\boldsymbol{u} := \operatorname{div}\sigma(\boldsymbol{u}) = (\lambda + 2\mu)\nabla\operatorname{div}\boldsymbol{u} - \mu\boldsymbol{curl}\,\boldsymbol{curl}\boldsymbol{u}$ and introduce the energy spaces $\boldsymbol{H}^1_+(\boldsymbol{\Delta}^*) := \left\{\boldsymbol{u} \in \boldsymbol{H}^1_{loc}(\Omega^+) :\ \boldsymbol{\Delta}^*\boldsymbol{u} \in \boldsymbol{L}^2_{loc}(\Omega^+)\right\}$ and $\boldsymbol{H}^1_-(\boldsymbol{\Delta}^*) := \left\{\boldsymbol{u} \in \boldsymbol{H}^1(\Omega^-) : \boldsymbol{\Delta}^*\boldsymbol{u} \in \boldsymbol{L}^2(\Omega^-)\right\}$. The traction trace for elastodynamic problems is defined by $\boldsymbol{t}_{|\Gamma} := \mathbf{T}\boldsymbol{u}$ where $\mathbf{T}$ is the traction operator defined in (3). We recall that we have $\boldsymbol{u}_{|\Gamma} \in \boldsymbol{H}^{\frac{1}{2}}(\Gamma)$ and $\boldsymbol{t}_{|\Gamma} \in \boldsymbol{H}^{-\frac{1}{2}}(\Gamma)$ for all $\boldsymbol{u} \in \boldsymbol{H}^1_\pm(\boldsymbol{\Delta}^*)$.

For a solution $\boldsymbol{u} \in \boldsymbol{H}^1_+(\boldsymbol{\Delta}^*)$ to the Navier equation (1) in Ω^+, that satisfies the Kupradze radiation conditions, the Somigliana integral representation of the field is given by

$$\boldsymbol{u}(\boldsymbol{x}) = \mathcal{D}\boldsymbol{u}_{|\Gamma}(\boldsymbol{x}) - \mathcal{S}\boldsymbol{t}_{|\Gamma}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega^+, \tag{10}$$

where the single- and double-layer potential operators are respectively defined by

$$\mathcal{S}\boldsymbol{\varphi}(\boldsymbol{x}) = \int_\Gamma \mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\varphi}(\boldsymbol{y})ds(\boldsymbol{y}) \quad \text{and} \quad \mathcal{D}\boldsymbol{\psi}(\boldsymbol{x}) = \int_\Gamma \left[\mathbf{T}_{\boldsymbol{y}}\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})\right]^{\mathsf{T}}\boldsymbol{\psi}(\boldsymbol{y})ds(\boldsymbol{y}) \quad \boldsymbol{x} \in \mathbb{R}^3\backslash\Gamma. \tag{11}$$

The potentials $\mathcal{S}$ (resp. $\mathcal{D}$) are continuous from $\boldsymbol{H}^{-1/2}(\Gamma)$ to $\boldsymbol{H}^1_-(\boldsymbol{\Delta}^*) \cup \boldsymbol{H}^1_+(\boldsymbol{\Delta}^*)$ (resp. from $\boldsymbol{H}^{1/2}(\Gamma)$ to $\boldsymbol{H}^1_-(\boldsymbol{\Delta}^*) \cup \boldsymbol{H}^1_+(\boldsymbol{\Delta}^*)$). For any $\boldsymbol{\varphi} \in \boldsymbol{H}^{-1/2}(\Gamma)$ and $\boldsymbol{\psi} \in \boldsymbol{H}^{1/2}(\Gamma)$, the potentials $\mathcal{S}\boldsymbol{\varphi}$ and $\mathcal{D}\boldsymbol{\psi}$ solve the Navier equation in Ω^+ and Ω^-, and satisfy the Kupradze radiation condition. The exterior and interior Dirichlet ($\gamma_0^\pm$) and traction ($\gamma_1^\pm$) traces of $\mathcal{S}$ and $\mathcal{D}$ are given by

$$\gamma_0^\pm\mathcal{S} = S, \quad \gamma_1^\pm\mathcal{S} = \mp\frac{1}{2}I + D', \quad \gamma_0^\pm\mathcal{D} = \pm\frac{1}{2}I + D$$

where the operators S (resp. D) are continuous from $\boldsymbol{H}^{-1/2}(\Gamma)$ to $\boldsymbol{H}^{1/2}(\Gamma)$ (resp. continuous from $\boldsymbol{H}^{1/2}(\Gamma)$ to $\boldsymbol{H}^{-1/2}(\Gamma)$) and are given by

$$S\boldsymbol{\varphi}(\boldsymbol{x}) = \int_\Gamma \mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\varphi}(\boldsymbol{y})ds(\boldsymbol{y}) \quad \text{and} \quad D\boldsymbol{\psi}(\boldsymbol{x}) = \int_\Gamma \left[\mathbf{T}_{\boldsymbol{y}}\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y})\right]^{\mathsf{T}}\boldsymbol{\psi}(\boldsymbol{y})ds(\boldsymbol{y}), \ \boldsymbol{x} \in \Gamma. \tag{12}$$

The scattering problem is formulated as follows: Given an incident wave field $\boldsymbol{u}^{\text{inc}}$ which is assumed to solve the Navier equation in the absence of any scatterer, find the displacement $\boldsymbol{u}$ solution to the Navier equation (1) in Ω^+ which satisfies the Dirichlet boundary condition on Γ

$$\boldsymbol{u}_{|\Gamma} + \boldsymbol{u}^{\text{inc}}_{|\Gamma} = 0. \tag{13}$$

Applying the potential theory, the elastic scattering problem reduces to a boundary integral equation

$$S(\boldsymbol{t}_{|\Gamma} + \boldsymbol{t}^{inc}_{|\Gamma})(\boldsymbol{x}) = \boldsymbol{u}^{inc}_{|\Gamma}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma \tag{14}$$

$$\text{or} \quad (\frac{I}{2} + D')(\boldsymbol{t}_{|\Gamma} + \boldsymbol{t}^{inc}_{|\Gamma})(\boldsymbol{x}) = \boldsymbol{t}^{inc}_{|\Gamma}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{15}$$

In the following, $\mathcal{H}$-matrix based solvers are applied and studied in the special case of elastodynamic scattering problems but the method can be applied to the solution of any boundary integral equation defined in terms of the single and double layer potential operators.

4.2. Classical concepts of the boundary element method

The main ingredients of the Boundary Element Method are a transposition of the concepts developed for the Finite Element Method [6]. First, the numerical solution of the boundary integral equations (14) or (15) is based on a discretization of the surface Γ into N_E isoparametric boundary elements of order one, i.e. three-node triangular elements. Each physical element E_e on the approximate boundary is mapped onto a reference element Δ_e via an affine mapping

$$\boldsymbol{\xi} \in \Delta_e \rightarrow \boldsymbol{y}(\boldsymbol{\xi}) \in E_e, \quad 1 \leq e \leq N_e.$$

Δ_e is the reference triangle in the (ξ_1, ξ_2)-plane. The N_c interpolation points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_c}$ are chosen as the vertices of the mesh. Each component of the total traction field is approximated with globally continuous, piecewise-linear shape functions $(v_i(\boldsymbol{y}))_{1\leq i\leq N_c}$: $v_i(\boldsymbol{y}_j) = \delta_{ij}$ for $1 \leq i, j \leq N_c$. A boundary element E_e contains exactly 3 interpolation nodes $(\boldsymbol{y}_k^e)_{1\leq k\leq 3}$ associated with 3 basis functions $(v_k^e)_{1\leq k\leq 3}$. These basis functions are related to the canonical basis $(\hat{v}_k)_{1\leq k\leq 3}$ defined on the reference element Δ_e by $v_k^e(\boldsymbol{y}(\boldsymbol{\xi})) = \hat{v}_k(\boldsymbol{\xi})$. Each component of the total traction field $\boldsymbol{p}(\boldsymbol{y}) = (\boldsymbol{t}_{|\Gamma} + \boldsymbol{t}_{|\Gamma}^{inc})(\boldsymbol{y})$ is approximated on the element E_e by

$$\boldsymbol{p}_\alpha(\boldsymbol{y}) \approx \sum_{k=1}^{3} p_\alpha^k v_k^e(\boldsymbol{y}) \quad (1 \leq \alpha \leq 3),$$

where p_α^k denotes the approximation of the nodal value of the component α of the vector $\boldsymbol{p}(\boldsymbol{y}_k)$. To discretize the boundary integral equations (14) or (15) we consider the collocation approach. It consists in enforcing the equation at a finite number of collocation points $\boldsymbol{x}$. To have a solvable discrete problem, one has to choose N_c collocation points. The N_c traction approximation nodes thus defined also serve as collocation points, i.e. $(\boldsymbol{x}_i)_i = (\boldsymbol{y}_j)_j$. This discretization process transforms (14) or (15) into a square complex-valued linear system of size $3N_c$ of the form

$$\mathbb{A}\mathbf{p} = \mathbf{b}, \tag{16}$$

where the $(3N_c)$-vector $\mathbf{p}$ collects the degrees of freedom (DOFs), namely the nodal traction components, while the $(3N_c)$-vector $\mathbf{b}$ arises from the imposed incident wave field. Assembling the matrix $\mathbb{A}$ classically [6] requires the computation of all element integrals for each collocation point, thus requiring a computational time of order $O(N_c^2)$.

It is interesting to note that a data-sparse representation of the matrix $\mathbb{A}$ is a direct consequence of the discretization of the Green's tensor. For example with the defined process, the discretization of equation (14) leads to the system

$$\mathbb{U}_\omega \mathbb{W}_y \mathbb{V}_y \mathbf{p} = \mathbf{b}$$

where the *diagonal* matrix $\mathbb{W}_y$ corresponds to the weights used to evaluate numerically the integrals, the matrix $\mathbb{V}_y$ corresponds to shape functions evaluated at the quadrature points of the reference element and the matrix $\mathbb{U}_\omega$ corresponds to the evaluation of the Green's tensor at the collocation points $(\boldsymbol{x}_i)_{i=1,\ldots,N_c}$ and interpolation points $(\boldsymbol{y}_j)_{j=1,\ldots,N_c}$. The relevant question is then how the $\mathcal{H}$-matrix representation of $\mathbb{U}_\omega$ can be transmitted to $\mathbb{A}$. The key point is to remark that the matrix $\mathbb{V}_y$ is a sparse matrix whose connectivity can be described as follows: it has a non-zero entry if there exists one triangle that has the two corresponding nodes as vertices. From that argument, it appears that if no efficient $\mathcal{H}$-matrix representation of $\mathbb{U}_\omega$ is available, we cannot expect to find an efficient $\mathcal{H}$-matrix representation of $\mathbb{A}$. On the other hand, since the structure of the matrix $\mathbb{V}_y$ is based on a notion of distance similarly to the construction of the binary tree $\mathcal{T}_I$, we can expect that if an efficient $\mathcal{H}$-matrix representation of $\mathbb{U}_\omega$ is available, an efficient $\mathcal{H}$-matrix representation of $\mathbb{A}$ will be found. However, the interface separating two subdomains used to compute the binary tree can also separate nodes belonging to the same elements. It is thus very likely that the ranks of the subblocks of $\mathbb{A}$ will be larger than the ranks of the same blocks in $\mathbb{U}_\omega$. Importantly, this property is independent of the order of approximation used in the BEM.

As a consequence, although the behavior of the $\mathcal{H}$-matrix based solvers for 3D elastodynamics are presented in the context of the BEM, the observations can nevertheless be extended to other configurations where a Green's tensor is discretized over a cloud of points.

5. Application of $\mathcal{H}$-matrices to oscillatory kernels: theoretical estimates

The efficiency of $\mathcal{H}$-matrix based solvers depends on the possible storage reduction obtained by low-rank approximations. It is thus important to estimate the memory requirements of the method in the context of 3D elastodynamics. We follow the proof of [25] proposed in the context of asymptotically smooth kernels.

5.1. Storage estimate

We use the notations introduced in Section 2.1. Since we are using a balanced binary tree $\mathcal{T}_I$, the number of levels is $L(I) \leq \log_2 |I| - \log_2 N_{\text{leaf}} + 1$. We denote by $N_c = |I|$ the number of points in the cloud. For a *non-admissible* block $\tau \times \sigma \in \mathcal{P}^{\text{non-ad}}$, the memory requirements are

$$N_{st} = |\tau|\,|\sigma| = \min\{|\tau|, |\sigma|\}\,\max\{|\tau|, |\sigma|\} \text{ for } \tau \times \sigma \in \mathcal{P}^{\text{non-ad}}.$$

Since *non-admissible* blocks can only appear at the leaf level, the memory requirements are bounded by

$$N_{st} \leq N_{\text{leaf}}\ \max\{|\tau|, |\sigma|\} \leq N_{\text{leaf}}(|\tau| + |\sigma|) \text{ for } \tau \times \sigma \in \mathcal{P}^{\text{non-ad}}.$$

For *admissible* blocks, the memory requirements are

$$N_{st} = r_{\tau\times\sigma}(|\tau| + |\sigma|) \leq r_{\text{ACA}}^{\max}(|\tau| + |\sigma|) \text{ for } \tau \times \sigma \in \mathcal{P}^{\text{ad}},$$

where $r_{\tau\times\sigma}$ is the rank obtained by the ACA, resp. $r_{\text{ACA}}^{\max}$ denotes the maximum rank obtained by the ACA among all the *admissible* blocks. The total memory requirement is given by

$$N_{st}(\mathbb{A}) = \sum_{\tau\times\sigma\in\mathcal{P}^{\text{ad}}} r_{\tau\times\sigma}(|\tau| + |\sigma|) + \sum_{\tau\times\sigma\in\mathcal{P}^{\text{non-ad}}} |\tau|\,|\sigma| \leq \max\{N_{leaf}, r_{ACA}^{\max}\} \sum_{\tau\times\sigma\in\mathcal{P}} (|\tau| + |\sigma|).$$

It is clear that for a binary tree

$$\sum_{\tau\in\mathcal{T}_i} |\tau| = \sum_{\ell=0}^{L(I)-1} |I^{(\ell)}| = L(I)|I|.$$

Then, using the definition of the sparsity pattern we obtain

$$\sum_{\tau\times\sigma\in\mathcal{P}} |\tau| = \sum_{\tau\in\mathcal{T}_I} \Big[|\tau| \sum_{\sigma\,|\,\tau\times\sigma\in\mathcal{P}} 1\Big] \leq C_{sp} \sum_{\tau\in\mathcal{T}_I} |\tau| \leq C_{sp}\ L(I)\ |I|.$$

A similar bound is found for $\sum_{\tau\times\sigma\in\mathcal{P}} |\sigma|$ such that the storage requirement is bounded by

$$N_{st}(\mathbb{A}) \leq 2C_{sp}\max\{r_{\text{ACA}}^{\max}, N_{\text{leaf}}\}N_c(\log_2 N_c - \log_2 N_{\text{leaf}} + 1).$$

In this storage estimate, the parameters N_c and N_{leaf} are known. The sparsity pattern C_{sp} depends on the partition $\mathcal{P}$ and will be discussed later (Section 6). In the context of oscillatory kernels, $r_{\text{ACA}}^{\max}$ is known to depend on the frequency. The aim of the rest of this section is to characterize this dependence. We consider first the 3D Helmholtz Green's function.

5.2. Convergence of the Taylor expansion of the 3D Helmholtz Green's function

It is well-known that $\mathcal{H}$-matrices are efficient representations for matrices coming from the discretization of asymptotically smooth kernels, i.e. kernels satisfying Definition 5.1 below. For such matrices the method is well-documented and estimates are provided for the Taylor expansion and interpolation errors.

Definition 5.1. A kernel $s(.,.) : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is asymptotically smooth if there exist two constants c_1, c_2 and a singularity degree $\sigma \in \mathbb{N}_0$ such that $\forall z \in \{x_\alpha, y_\alpha\}$, $\forall n \in \mathbb{N}_0$, $\forall \boldsymbol{x} \neq \boldsymbol{y}$

$$|\partial_z^n s(\boldsymbol{x}, \boldsymbol{y})| \leq n! c_1 (c_2 \|\boldsymbol{x} - \boldsymbol{y}\|)^{-n-\sigma}.$$

The capability to produce low rank approximations is closely related to the concept of degenerate functions [3]. A kernel function is said to be degenerate if it can be well approximated (under some assumptions) by a sum of functions with separated variables. In other words, noting X and Y two domains of $\mathbb{R}^3$, we are looking for an approximation s^r of s on $X \times Y$ with r terms such that it writes

$$s^r(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\ell=1}^{r} u_\ell^{(r)}(\boldsymbol{x}) v_\ell^{(r)}(\boldsymbol{y}) \quad \boldsymbol{x} \in X, \quad \boldsymbol{y} \in Y.$$

r is called the *separation rank* of s^r and $s(\boldsymbol{x}, \boldsymbol{y}) = s^r(\boldsymbol{x}, \boldsymbol{y}) + R_s^r(\boldsymbol{x}, \boldsymbol{y})$ with R_s^r the remainder. Such an approximation is obtained for instance with a Taylor expansion with $r := |\{\boldsymbol{\alpha} \in \mathbb{N}_0^3 : |\boldsymbol{\alpha}| \leq m\}|$ terms, i.e.

$$s^r(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^3, |\boldsymbol{\alpha}|\leq m} (\boldsymbol{x} - \boldsymbol{x}_0)^{\boldsymbol{\alpha}} \frac{1}{\boldsymbol{\alpha}!} \partial_x^{\boldsymbol{\alpha}} s(\boldsymbol{x}_0, \boldsymbol{y}) + R_s^r \quad \boldsymbol{x} \in X, \quad \boldsymbol{y} \in Y$$

where $\boldsymbol{x}_0 \in X$ is the centre of the expansion.

A kernel function is said to be a separable expression in $X \times Y$ if the remainder converges quickly to zero. The main result for asymptotically smooth kernels is presented in [3, Lemma 3.15] and [25, Theorem 4.17]. For such kernels, it can be easily shown that

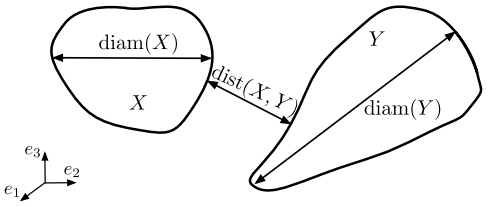


Fig. 7. Definition of the distance and diameter.

$$|R_s^r(\boldsymbol{x},\boldsymbol{y})| \leq C' \sum_{\ell=m}^{\infty} \left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_0\|}{c_2\|\boldsymbol{x}_0-\boldsymbol{y}\|} \right)^{\ell}$$

where c_2 is the constant appearing in Definition 5.1. The convergence of R_s^r is thus ensured for all $\boldsymbol{y} \in Y$ such that

$$\gamma_x := \frac{\max_{\boldsymbol{x}\in X} \|\boldsymbol{x}-\boldsymbol{x}_0\|}{c_2\|\boldsymbol{x}_0-\boldsymbol{y}\|} < 1. \tag{17}$$

Provided that the condition (17) holds, the remainder is bounded by

$$|R_s^r(\boldsymbol{x},\boldsymbol{y})| \leq C' \frac{\gamma_x^m}{1-\gamma_x} \xrightarrow[m\to\infty]{} 0. \tag{18}$$

In this configuration, the Taylor expansion of the asymptotically smooth kernel converges exponentially with convergence rate γ_x. Since r is the cardinal of the set $\{\boldsymbol{\alpha} \in \mathbb{N}_0^3 : |\boldsymbol{\alpha}| \leq m\}$, it holds $m \sim r^{1/3}$, and then $r \approx |\log \varepsilon|^3$ is expected to achieve an approximation with accuracy $\varepsilon > 0$.

In other words, the exponential convergence of the Taylor expansion s^r over X and Y, with center $\boldsymbol{x}_0 \in X$ is constrained by a condition on $\boldsymbol{x}_0$, X and Y. One can derive a sufficient condition, independent of $\boldsymbol{x}_0$, be observing that

$$\max_{\boldsymbol{x}\in X} \|\boldsymbol{x}-\boldsymbol{x}_0\| \leq \operatorname{diam} X \quad \text{and} \quad \operatorname{dist}(\boldsymbol{x}_0, Y) \geq \operatorname{dist}(X,Y)$$

where dist denotes the Euclidean distance, i.e.

$$\operatorname{dist}(X,Y) = \inf\{\|\boldsymbol{x}-\boldsymbol{y}\|,\ \boldsymbol{x}\in X,\ \boldsymbol{y}\in Y\} \tag{19}$$

and diam denotes the diameter of a domain (Fig. 7), i.e.

$$\operatorname{diam}(X) = \sup\{\|\boldsymbol{x}_1-\boldsymbol{x}_2\|,\ \boldsymbol{x}_1,\ \boldsymbol{x}_2 \in X\}. \tag{20}$$

Let us introduce the notion of η-admissibility. X and Y are said to be *η-admissible* if $\operatorname{diam} X \leq \eta \operatorname{dist}(X,Y)$ where $\eta > 0$ is a parameter of the method. In this configuration, it follows that

$$\max_{\boldsymbol{x}\in X} \|\boldsymbol{x}-\boldsymbol{x}_0\| \leq \operatorname{diam} X \leq \eta \operatorname{dist}(X,Y) \leq \eta \operatorname{dist}(\boldsymbol{x}_0, Y) \quad \text{and so} \quad \gamma_x \leq \frac{\eta}{c_2}. \tag{21}$$

For asymptotically smooth kernels, we thus conclude that the exponential convergence is ensured if X and Y are *η-admissible* and if η is chosen such that $\eta < c_2$. We observe that in [3], the condition is more restrictive on η since it reads $\eta\sqrt{3} < c_2$.

The Helmholtz Green's function $G_\kappa(\boldsymbol{x},\boldsymbol{y}) = \frac{\exp(i\kappa\|\boldsymbol{x}-\boldsymbol{y}\|)}{4\pi\|\boldsymbol{x}-\boldsymbol{y}\|}$ is not asymptotically smooth [1]. On the other hand, noting that it holds $G_\kappa(\boldsymbol{x},\boldsymbol{y}) = \exp(i\kappa\|\boldsymbol{x}-\boldsymbol{y}\|)G_0(\boldsymbol{x},\boldsymbol{y})$ and since $G_0(\boldsymbol{x},\boldsymbol{y})$ is asymptotically smooth [3,25], the Helmholtz Green's function satisfies the following estimate [1]: $\exists c_1, c_2$ and $\sigma \in \mathbb{N}_0$ (singularity degree of the kernel $G_0(.,.)$) such that $\forall z \in \{x_\alpha, y_\alpha\}$, $\forall n \in \mathbb{N}_0$, $\forall \boldsymbol{x} \neq \boldsymbol{y}$

$$|\partial_z^n G_\kappa(\boldsymbol{x},\boldsymbol{y})| \leq n! c_1 (1+\kappa\|\boldsymbol{x}-\boldsymbol{y}\|)^n (c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-n-\sigma}. \tag{22}$$

From inequality (22), it is clear that in the so-called *low-frequency regime*, i.e. when $\max_{\boldsymbol{x},\boldsymbol{y}\in X\times Y}(\kappa\|\boldsymbol{x}-\boldsymbol{y}\|)$ is small, the Helmholtz Green's function behaves similarly to an asymptotically smooth kernel because $\kappa\|\boldsymbol{x}-\boldsymbol{y}\|$ is uniformly small for all $\boldsymbol{x},\boldsymbol{y} \in X \times Y$. There exists a lot of works on the $\mathcal{H}$-matrix representation [2,10] of asymptotically smooth kernels such as G_0 but much less attention has been devoted to the case of the 3D Helmholtz equation. We can mention the work [36] where a collocation approach is considered for problems up to 20 480 DOFs and where encouraging numerical results are presented.

We denote G_κ^r the Taylor expansion with $r := |\{\boldsymbol{\alpha} \in \mathbb{N}_0^3 : |\boldsymbol{\alpha}| \leq m\}|$ terms of the Helmholtz Green's function G_κ, i.e.

$$G_\kappa^r(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^3, |\boldsymbol{\alpha}|\leq m} (\boldsymbol{x}-\boldsymbol{x}_0)^{\boldsymbol{\alpha}} \frac{1}{\boldsymbol{\alpha}!} \partial_x^{\boldsymbol{\alpha}} G_\kappa(\boldsymbol{x}_0,\boldsymbol{y}) + R_\kappa^r \quad \boldsymbol{x}\in X, \quad \boldsymbol{y}\in Y \tag{23}$$

where $\boldsymbol{x}_0 \in X$ is the center of the expansion. According to the previous discussion, we know that, for the asymptotically smooth kernel G_0, the asymptotic smoothness allows to prove the exponential convergence of the Taylor series if X and Y are η-*admissible*. On the other hand for the Helmholtz kernel G_κ, the estimate (22) now leads to

$$|R^r_\kappa(\boldsymbol{x},\boldsymbol{y})| \leq C' \sum_{\ell=m}^{\infty} \left((1+\kappa\|\boldsymbol{x}_0-\boldsymbol{y}\|) \frac{\|\boldsymbol{x}-\boldsymbol{x}_0\|}{c_2\|\boldsymbol{x}_0-\boldsymbol{y}\|} \right)^{\ell}.$$

Thus to assure convergence of the series, X and Y must now be chosen in such a way that

$$\gamma_{\kappa,x} := (1+\kappa\|\boldsymbol{x}_0-\boldsymbol{y}\|) \frac{\max_{\boldsymbol{x}\in X}\|\boldsymbol{x}-\boldsymbol{x}_0\|}{c_2\|\boldsymbol{x}_0-\boldsymbol{y}\|} < 1. \tag{24}$$

From now on, we say that X and Y are η_κ-*admissible* if (24) is satisfied. Provided that the condition (24) holds, the remainder is bounded

$$|R^r_\kappa(\boldsymbol{x},\boldsymbol{y})| \leq C' \frac{\gamma^m_{\kappa,x}}{1-\gamma_{\kappa,x}}. \tag{25}$$

Remark that in the low-frequency regime, one has the estimate $\gamma_{\kappa,x} \sim \frac{\max_{\boldsymbol{x}\in X}\|\boldsymbol{x}-\boldsymbol{x}_0\|}{c_2\|\boldsymbol{x}_0-\boldsymbol{y}\|}$ such that the η_κ-admissibility condition is similar to the case of the asymptotically smooth kernel $s = G_0$. Within this framework, for higher frequencies, $\gamma_{\kappa,x}$ depends *linearly* on the wavenumber κ and thus on the circular frequency ω.

Remark 5.2. We do not discuss in this work on the choice of the optimal $\boldsymbol{x}_0$. It is very likely that the notions of η- and η_κ-admissibility can be improved by determining the optimal $\boldsymbol{x}_0$.

It is worth noting that a Taylor expansion with center $\boldsymbol{y}_0 \in Y$ instead of $\boldsymbol{x}_0 \in X$ can be performed. In that case, a similar error estimate is obtained:

$$|R^r_\kappa(\boldsymbol{x},\boldsymbol{y})| \leq C' \frac{\gamma^m_{\kappa,y}}{1-\gamma_{\kappa,y}}, \quad \text{with} \quad \gamma_{\kappa,y} := (1+\kappa\|\boldsymbol{y}_0-\boldsymbol{x}\|) \frac{\max_{\boldsymbol{y}\in Y}\|\boldsymbol{y}-\boldsymbol{y}_0\|}{c_2\|\boldsymbol{y}_0-\boldsymbol{x}\|}. \tag{26}$$

5.3. Theoretical estimate of the rank of 3D elastodynamics single-layer and double-layer potentials

The results on the convergence of the Taylor expansion of the 3D Helmholtz Green's function are easily extended to 3D elastodynamics. From (2) it is clear that the 3D elastodynamic Green's tensor is a combination of derivatives of the 3D Helmholtz Green's tensor. In particular, rewriting (2) component by component, we have

$$(\mathbf{U}_\omega)_{\alpha\beta}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{\kappa_s^2\mu}\left((\delta_{\tau\sigma}\delta_{\alpha\beta} - \delta_{\beta\tau}\delta_{\alpha\sigma}) \frac{\partial}{\partial x_\tau}\frac{\partial}{\partial y_\sigma} G_{\kappa_s}(\boldsymbol{x},\boldsymbol{y}) + \frac{\partial}{\partial x_\alpha}\frac{\partial}{\partial y_\beta} G_{\kappa_p}(\boldsymbol{x},\boldsymbol{y}) \right)$$

$$(\mathbf{T}_\omega)_{\alpha\beta}(\boldsymbol{x},\boldsymbol{y}) = C_{\alpha\gamma\tau\sigma} \frac{\partial}{\partial y_\sigma} \Big(\mathbf{U}_\omega(\boldsymbol{x},\boldsymbol{y}) \Big)_{\beta\tau} n_\gamma(\boldsymbol{y})$$

where the Einstein summation convention is used. Since $\kappa_p \leq \kappa_s$, it follows that each component of the 3D elastodynamics Green's tensor satisfies an inequality similar to (22) with $\kappa = \kappa_s$, i.e.

$$|\partial^n_z(\mathbf{P})_{\alpha\beta}(\boldsymbol{x},\boldsymbol{y})| \leq n! c_1 (1+\kappa_s\|\boldsymbol{x}-\boldsymbol{y}\|)^n (c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-n-\sigma} \tag{27}$$

where $\mathbf{P} = \mathbf{U}_\omega$ or $\mathbf{P} = \mathbf{T}_\omega$. As a result, provided that (25) or (26) holds with $\kappa = \kappa_s$ the Taylor expansion of the 3D elastodynamic Green's tensor converges exponentially with convergence rate $\gamma_{\kappa_s,x}$ or $\gamma_{\kappa_s,y}$. However again two regimes can be distinguished. In the *low-frequency* regime, the Taylor expansion behaves similarly to the case of the asymptotically smooth kernel G_0. The exponential convergence is ensured if X and Y are η-*admissible* and if η is chosen such that $\eta < c_2$. For higher frequencies, $\gamma_{\kappa_s,x}$ and $\gamma_{\kappa_s,y}$ defined in (25) and (26) depend linearly on the circular frequency ω.

The definition of the η_κ-admissibility for oscillatory kernels could be modified in order to keep the separation rank constant while the frequency increases. It is the option followed for example in directional approaches [30,18]. Another approach to avoid the rank increase is the concept of $\mathcal{H}^2$-matrices [1]. The option we consider is to work with the η-admissibility condition defined for the asymptotically smooth kernels G_0. In other words, we choose to keep η constant for all frequencies and use the admissibility condition defined by:

$$\operatorname{diam} X \leq \eta \operatorname{dist}(X,Y) \quad \text{with } \eta < c_2. \tag{28}$$

Our goal is to determine what is the actual behavior of the algorithm in that configuration. Following the study of the remainder of the Taylor expansion, it is clear that the method will not be optimal since the separation rank to achieve a given accuracy will increase with the frequency. Nevertheless, we are interested in determining the limit until which

the low-frequency approximation is viable for oscillatory kernels such as the 3D Helmholtz Green's function or the 3D elastodynamic Green's tensors.

The relevant question to address is then to determine the law followed by the growth of the numerical rank if the η-admissibility condition (28) is used in the context of oscillatory kernels. Importantly, the framework of Taylor expansions is only used to illustrate the degenerate kernel in an abstract way and to determine the optimal admissibility conditions in the low and high-frequency regimes within this framework. In practice the ACA which is a purely algebraic method is used after discretization. As a result, we will use linear algebra arguments to study the growth of the numerical rank. As a matter of fact, one expects a small numerical rank provided that the entries of the matrix are approximated by a short sum of products. From the Taylor expansion (23) this can be achieved if the discretization is accurate enough. Consider two clusters of points $X = (\boldsymbol{x}_i)_{i\in\tau}$ and $Y = (\boldsymbol{y}_j)_{j\in\sigma}$ and the corresponding subblock $\mathbb{A}_{\tau\times\sigma}$. Then using (23) we can build an approximation of $\mathbb{A}_{\tau\times\sigma}$ of the form $\mathbb{A}_{\tau\times\sigma} \approx \mathbb{B}_{\tau\times\sigma} = \mathbb{U}_\tau \mathbb{V}_\sigma^*$ where $\mathbb{U}_\tau$ corresponds to the part that depends on $\boldsymbol{x}$ and $\mathbb{V}_\sigma$ to the part that depends on $\boldsymbol{y}$.

In the following, κ denotes either the wavenumber in the context of the 3D Helmholtz equation, or also the S-wavenumber κ_s in the context of 3D elastodynamics.

To start, let us have a look at two clusters of points X and Y which are *η-admissible* under (28). At the wavenumber κ two cases may follow, whether condition (24) is also satisfied or not.

Let's consider first that the condition (24) is not satisfied at the wavenumber κ, i.e. if the frequency is too large, with respect to the diameters of the clusters, to consider that this block is *η_κ-admissible*. From the study of the Taylor expansion, it follows that the numerical rank $r(\kappa)$ of the corresponding block in the matrix at wavenumber κ will be large and the algorithm will not be efficient.

Let's consider now the case where X and Y are such that the condition (24) is also satisfied at the wavenumber κ. In that case, the numerical rank $r(\kappa)$ will be small.

These two cases illustrate the so-called *low-frequency* regime (latter case), i.e. when $(1 + \kappa\|\boldsymbol{x}_0 - \boldsymbol{y}\|) \sim 1$ and *high-frequency* regime (former case), i.e. when $(1 + \kappa\|\boldsymbol{x}_0 - \boldsymbol{y}\|) \sim \kappa\|\boldsymbol{x}_0 - \boldsymbol{y}\|$.

In this work however, we are not interested by only one frequency but rather by a frequency range. The key point is to remark that in the η_κ-admissibility condition, the important quantity is $\kappa\|\boldsymbol{x}_0 - \boldsymbol{y}\|$, i.e. the product between the wavenumber and the diameter of the blocks (we recall that the η-admissibility (28) relates the distance between blocks and the diameters of the blocks). Let's now consider the following configuration: at the wavenumber κ all the blocks of a matrix are such that the η-admissibility includes the η_κ-admissibility while at the wavenumber 2κ, it is not the case. This configuration represents the transition regime after the *low-frequency* regime. From (25) and (26) it follows that if the frequency doubles, and as a result the wavenumber doubles from κ to $\kappa' = 2\kappa$, an *η_κ-admissible* block $\mathbb{A}_{\tau\times\sigma}$ should be decomposed into a 2×2 block matrix at wavenumber κ' to reflect the reduction by a factor 2 of the leading diameter. Thus the subblocks $\mathbb{A}_{\tau_i\times\sigma_j}$ are shown by the Taylor expansion to be of the order of the small rank $r(\kappa)$ at the wavenumber κ', i.e.

$$\mathbb{A}_{\tau\times\sigma} = \begin{bmatrix} \mathbb{A}_{\tau_1\times\sigma_1} & \mathbb{A}_{\tau_1\times\sigma_2} \\ \mathbb{A}_{\tau_2\times\sigma_1} & \mathbb{A}_{\tau_2\times\sigma_2} \end{bmatrix} \approx \begin{bmatrix} \mathbb{B}_{\tau_1\times\sigma_1} & \mathbb{B}_{\tau_1\times\sigma_2} \\ \mathbb{B}_{\tau_2\times\sigma_1} & \mathbb{B}_{\tau_2\times\sigma_2} \end{bmatrix}$$

where each $\mathbb{B}_{\tau_i\times\sigma_j}$ $(1 \le i, j \le 2)$ is a low-rank approximation of rank $r(\kappa)$ that can be written as $\mathbb{B}_{\tau_i\times\sigma_j} = \mathbb{U}_{\tau_i\times\sigma_j}\mathbb{S}_{\tau_i\times\sigma_j}\mathbb{V}^*_{\tau_i\times\sigma_j}$ and each $\mathbb{S}_{\tau_i\times\sigma_j}$ is a $r(\kappa)\times r(\kappa)$ matrix. What is more, the Taylor expansion (23) says that $\mathbb{U}_{\tau_i\times\sigma_j}$ is independent of σ_j such that $\mathbb{U}_{\tau_i\times\sigma_j} = \mathbb{U}_{\tau_i}$:

$$\mathbb{A}_{\tau\times\sigma} \approx \begin{bmatrix} \mathbb{U}_{\tau_1} & \mathbb{O} \\ \mathbb{O} & \mathbb{U}_{\tau_2} \end{bmatrix} \begin{bmatrix} \mathbb{S}_{\tau_1\times\sigma_1}\mathbb{V}^*_{\tau_1\times\sigma_1} & \mathbb{S}_{\tau_1\times\sigma_2}\mathbb{V}^*_{\tau_1\times\sigma_2} \\ \mathbb{S}_{\tau_2\times\sigma_1}\mathbb{V}^*_{\tau_2\times\sigma_1} & \mathbb{S}_{\tau_2\times\sigma_2}\mathbb{V}^*_{\tau_2\times\sigma_2} \end{bmatrix}.$$

Since one can perform the Taylor expansion also with respect to $\boldsymbol{y}$, $\mathbb{A}_{\tau\times\sigma}$ is in fact of the form

$$\mathbb{A}_{\tau\times\sigma} \approx \begin{bmatrix} \mathbb{U}_{\tau_1} & \mathbb{O} \\ \mathbb{O} & \mathbb{U}_{\tau_2} \end{bmatrix} \begin{bmatrix} \mathbb{S}_{\tau_1\times\sigma_1} & \mathbb{S}_{\tau_1\times\sigma_2} \\ \mathbb{S}_{\tau_2\times\sigma_1} & \mathbb{S}_{\tau_2\times\sigma_2} \end{bmatrix} \begin{bmatrix} \mathbb{V}^*_{\sigma_1} & \mathbb{O} \\ \mathbb{O} & \mathbb{V}^*_{\sigma_2} \end{bmatrix}$$

As a result, $\mathbb{A}_{\tau\times\sigma}$ is numerically at most of rank $2r(\kappa)$ at the wavenumber κ'. The same reasoning can be extended to higher frequencies such that the maximum rank of all the blocks of a given matrix is at most increasing linearly with the frequency.

So far, we have assumed that all the blocks of a matrix satisfy the same conditions. However due to the hierarchical structure of $\mathcal{H}$-matrix representations (i.e. subblocks in the matrix at different levels in the partition $\mathcal{P}$ corresponding to clusters with different diameters), it is very likely that the blocks of a matrix at a given wavenumber will not be all in the *low-frequency* regime nor all in the *high-frequency* regime. In particular assuming the frequency is not *too high* and provided the hierarchical subdivision is deep enough, there will always be some blocks in the *low-frequency* regime, i.e. blocks for which the rank is low numerically since the *η_κ-admissibility* condition is satisfied. Then for *η-admissible* blocks at higher levels, the previous discussion on the transition regime can be applied by induction such that the numerical rank of these blocks is expected to grow linearly through the various levels. Note that this linear increase depends on the choice of the number of points at the leaf level. The crux is to initialize the induction process: for a given choice, one has a certain limit

for the transition regime. If one changes the number of points at the leaf level, the limit changes too. As a result, one cannot continue the process to infinity. To a given choice of number of points per leaf, there is a limit frequency.

From all these arguments, we conclude that if we consider a fixed geometry at a fixed frequency and we increase the density of discretization points per wavelength, and as a consequence the total number of discretization points, the maximum rank of the blocks can be either small or large depending on the value of the product between the frequency and the diameter of the blocks (fixed by the geometry). However we expect that this rank will remain fixed if the density of discretization points per wavelength increases: as a matter of fact, the Taylor expansion with r given, is better and better approximated, so the rank should converge to r. This conclusion will be confirmed by numerical evidences in Section 7. On the other hand, if we consider a fixed geometry with a fixed density of points per S-wavelength, the maximum rank of the blocks is expected to grow linearly with the frequency until the *high-frequency regime* is achieved. This transition regime is in some sense a *pre-asymptotic regime* and we demonstrate numerically in Section 8 its existence. In the *high-frequency* regime, the $\mathcal{H}$-matrix representation is expected to be suboptimal with a rank rapidly increasing.

This paper presents an original algebraic argument to intuit the existence of this pre-asymptotic regime. It would be interesting in the future to compare this pre-asymptotic regime with other more involved hierarchical matrix based approaches proposed for oscillatory kernels, e.g. directional $\mathcal{H}^2$-matrices and with the multipole expansions used in the Fast Multipole Method. Another interesting but difficult question is to determine theoretically the limits of the pre-asymptotic regime.

6. Implementation issues

6.1. Efficient implementation of the η-admissibility condition

A key tool in hierarchical matrix based methods is the efficient determination of *admissible* blocks in the matrix. The first remark from an implementation point of view comes from the possibility to perform the Taylor expansion of 3D elastodynamic Green's tensors either with respect to $\boldsymbol{x} \in X = (\boldsymbol{x}_i)_{i\in\tau}$ or with respect to $\boldsymbol{y} \in Y = (\boldsymbol{y}_j)_{j\in\sigma}$. When a Taylor expansion with respect to $\boldsymbol{x}$ is performed, the clusters X and Y are said to be η-*admissible* if $\operatorname{diam} X \leq \eta \operatorname{dist}(X,Y)$. On the other hand, when a Taylor expansion with respect to $\boldsymbol{y}$ is performed, the clusters X and Y are said to be η-*admissible* if $\operatorname{diam} Y \leq \eta \operatorname{dist}(X,Y)$. Thus the admissibility condition at the discrete level becomes

$$\text{The block } \tau\times\sigma \text{ is } \textit{admissible} \quad \text{if } \min\Big(\operatorname{diam}(X),\operatorname{diam}(Y)\Big) < \hat{\eta}\times\operatorname{dist}(X,Y) \tag{29}$$

where $\hat{\eta}$ is a numerical parameter of the method discussed in the next section but subjected to some constraints highlighted in Section 5.3.

Since the computation of the diameter of a set X with (20) is an expensive operation [25], it is replaced by computing the diameter of the bounding box with respect to the Cartesian coordinates (Fig. 8b)

$$\operatorname{diam}(X) \leq \Big(\sum_{\alpha=1}^{3}(\max_{\boldsymbol{x}\in X}\boldsymbol{x}_\alpha - \min_{\boldsymbol{x}\in X}\boldsymbol{x}_\alpha)^2\Big)^{1/2} =: \operatorname{diam}_{\text{box}}(X)$$

which is an upper bound. Similarly the distance between two sets is replaced by the distance between the closest faces of the bounding boxes

$$\operatorname{dist}(X,Y) \geq \Big(\sum_{\alpha=1}^{3}(\min_{\boldsymbol{x}\in X}\boldsymbol{x}_\alpha - \max_{\boldsymbol{y}\in Y}\boldsymbol{y}_\alpha)^2 + (\min_{\boldsymbol{y}\in Y}\boldsymbol{y}_\alpha - \max_{\boldsymbol{x}\in X}\boldsymbol{x}_\alpha)^2\Big)^{1/2} =: \operatorname{dist}_{\text{box}}(X,Y)$$

which is a lower bound. The practical admissibility condition writes

$$\text{The block } \tau\times\sigma \text{ is } \textit{admissible} \quad \text{if } \min\Big(\operatorname{diam}_{\text{box}}(X),\operatorname{diam}_{\text{box}}(Y)\Big) < \hat{\eta}\times\operatorname{dist}_{\text{box}}(X,Y). \tag{30}$$

The admissibility conditions (29) or (30) imply that X and Y must be disjoint and that their distance is related to their diameters. Also, one can check [25, Th. 6.16] that the sparsity pattern C_{sp} is in fact bounded by a constant number that depends only on the geometry.

In addition, the choice of the parameter $\hat{\eta}$ influences the shape of the *admissible* blocks and the number of *admissible* blocks. On the one hand, from the study of the Taylor expansion of the Green's function, it is clear that the smaller the parameter $\hat{\eta}$, the faster the remainder tends to zero. On the other hand, a parameter $\hat{\eta}$ that tends to zero means that only blocks of very small diameter (with respect to the distance) are *admissible*. The choice of a large $\hat{\eta}$ means in practice that very elongated blocks are allowed. From an algorithmic point of view, our goal is to find a good compromise: i.e. not only to have a large number of *admissible* blocks but also to ensure the convergence of the Taylor expansion. As a result, we want to choose $\hat{\eta}$ as large as possible.

Since an upper bound of the diameter and a lower bound of the distance are used, it is clear that even though a theoretical bound on η is found from the study of (28), larger values of $\hat{\eta}$ can be used and still lead to accurate low

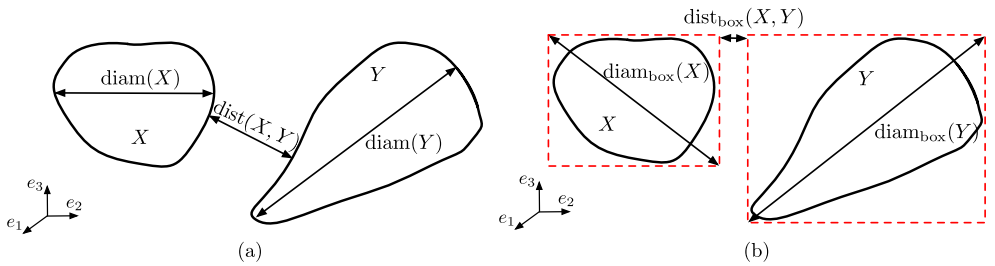


Fig. 8. Efficient implementation of the admissibility condition: (a) true condition and (b) more efficient approach using upper and lower bounds.

Table 1
Values of c_1 and c_2 obtained after a power regression for values of m between 2 and 10.

m	2	3	4	5	6	7	8	9	10
c_1	0.250	0.250	0.238	0.226	0.216	0.206	0.198	0.191	0.184
c_2	0.500	0.500	0.490	0.483	0.476	0.470	0.465	0.461	0.458

rank block approximations in practice. Then, the distances and diameters are defined according to the Cartesian coordinates cf. (30). But depending on the choice of clustering method adopted, it is possible that the distance between some blocks will be very small even though the blocks are disjoints. This remark weighs in favor of the use of a numerical parameter $\hat{\eta}$ larger than the one predicted by the theory.

6.2. Choice of an acceptable parameter $\hat{\eta}$

To determine the adequate parameter $\hat{\eta}$, it is necessary to determine the constant c_2 in (24) that gives the upper bound on η. This constant comes exclusively from the asymptotically smooth kernel G_0. It is easy to determine that

$$\forall z \in \{x_\alpha, y_\alpha\} \quad \partial_z^1 G_0(\boldsymbol{x}, \boldsymbol{y}) \leq \frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|^2} \quad \text{such that} \quad 1 \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-2}.$$

Using Maple, it follows similarly that

$$2 \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-3},\ 4 \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-4},\ \frac{17}{2} \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-5},\ \frac{37}{2} \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-6},$$

$$41 \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-7},\ 92 \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-8},\ \frac{5001}{24} \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-9},$$

$$\frac{3803}{8} \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-10},\ \frac{4363}{4} \leq c_1(c_2\|\boldsymbol{x}-\boldsymbol{y}\|)^{-11}.$$

In Table 1, we report the values of c_1 and c_2 obtained if a power regression up to the order m is performed with $2 \leq m \leq 10$. We observe that c_2 is smaller than 0.5 and decreases if m is increased. As a result, in theory we should use $\eta < c_2 < 0.5$. Note that we have limited our study up to $m = 10$ because $m \sim r^{1/3}$, where we recall that r is the number of terms in the Taylor expansion (23). We expect in practice to consider cases with a maximum numerical rank lower than 1000 such that the current study is representative of the encountered configurations.

In Section 6.1, we have explained why the use of an upper bound of the diameter and a lower bound of the distance in the admissibility condition permits to expect in practice good compression rates even though a larger value $\hat{\eta} > \eta$ is used. Indeed in [28], good numerical results for various geometries are obtained with $\hat{\eta} = 3$. This is the parameter we use in all our numerical examples for 3D elastodynamics.

7. Behavior for low frequency elastodynamics

7.1. Definition of the test problem

In Sections 7 and 8, we consider the diffraction of vertical incident plane P waves by a spherical cavity. The material properties are fixed to $\mu = \rho = 1$ and $\nu = 1/3$. We recall that in all the numerical examples, the binary tree $\mathcal{T}_I$ is built with a stopping criteria $N_{\text{leaf}} = 100$ and the constant in the admissibility condition (29) is set to $\hat{\eta} = 3$. Unless otherwise indicated, the required accuracy is fixed to $\varepsilon_{\text{ACA}} = 10^{-4}$. In the following, we denote by N the number of degrees of freedom such that $N = 3N_c$.

The admissibility condition (29) depends only on the geometry of the domain. In Fig. 9, we illustrate the repartition of the expected low-rank (green blocks) and full blocks (red blocks) for the sphere geometry with $N = 30\,726$.

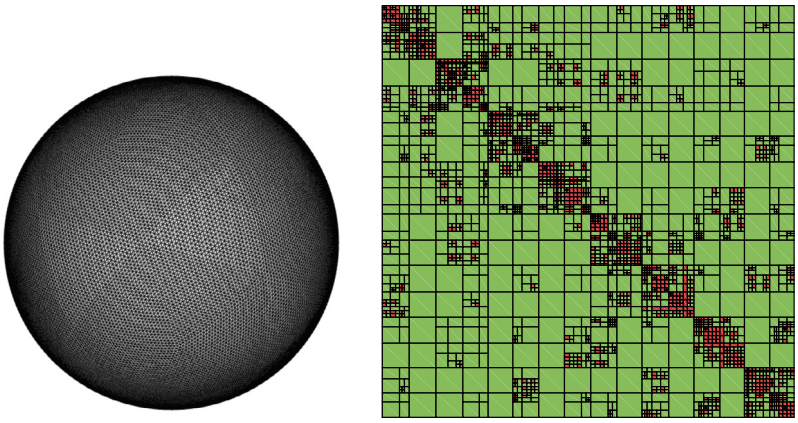


Fig. 9. Illustration of the expected repartition of the full and low-rank blocks for the sphere with $N = 30\ 726$.

Table 2
Maximum numerical rank observed (before/after the recompression step) for a fixed frequency, i.e. while increasing the density of points per S-wavelength, for $\varepsilon_{\text{ACA}} = 10^{-4}$ and the single-layer potential (14).

N	7686	30 726	122 886	183 099	490 629	763 638	985 818
$\omega = 3$	63/39	72/39	75/39	69/37	75/39	69/40	78/39
$\omega = 14$	99/73	105/75	108/76	99/67	114/76	111/76	111/76

Table 3
Maximum numerical rank observed (before/after the recompression step) for a fixed frequency, i.e. while increasing the density of points per S-wavelength, for $\varepsilon_{\text{ACA}} = 10^{-4}$ and the double-layer potential (15).

N	7686	30 726	122 886	183 099	490 629	763 638	985 818
$\omega = 3$	66/37	66/37	72/38	66/36	69/38	72/39	75/39
$\omega = 14$	102/70	117/74	117/74	102/66	114/75	114/75	114/76

The goal of Section 7 is to determine the behavior of the algorithm for a fixed frequency while the number of discretization points is increased (i.e. in the *low-frequency* regime). In the remainder of this section, the circular frequency ω is thus fixed. As a result, the density of points per S-wavelength increases as the number of discretization points increases.

7.2. Storage requirements for the single and double-layer potentials for a fixed frequency

In Section 5, we have seen that the memory requirements are expected to be of the order of $O(\max(r_{\text{ACA}}^{\max}, N_{\text{leaf}}) N \log_2 N)$. In addition, for a fixed frequency, $r_{\text{ACA}}^{\max}$ is expected to be constant. In Tables 2 and 3, we report the maximum numerical rank observed among all the *admissible* blocks for the case of a sphere for a fixed circular frequency ($\omega = 3$ or $\omega = 14$) and for various mesh sizes, for the single-layer and double-layer operators. The first rank corresponds to the numerical rank obtained by the partially pivoted ACA. The second rank corresponds to the numerical rank obtained after the recompression step (presented in Section 3.1). The two ranks are seen to be almost constant while the density of points is increasing, as intuited by the study of the Taylor expansion. The numerical rank without the recompression step is much larger than after the optimization step. The explanation is that the partially pivoted ACA is a heuristic method, the rank is thus not the optimal one. The recompression step permits to recover the intrinsic numerical rank of the blocks. In addition, as expected the maximum numerical rank increases with the frequency. We will study in more details this dependence on the frequency in Section 8. Finally also as expected, since the two Green's tensors satisfy the inequality (27), similar numerical ranks are observed for the single and double layer potentials.

Ultimately we are not interested in the maximum numerical rank but rather on the memory requirements of the algorithm. In Figs. 10a–b, we report the memory requirements N_s and the compression rate $\tau(\mathcal{H}) := N_s/N^2$ with respect to the number of degrees of freedom N for $\varepsilon_{\text{ACA}} = 10^{-4}$ and the single-layer operator. Since the rank is constant, N_s is expected to be of the order of $N \log_2 N$ and $\tau(\mathcal{H})$ of the order of $\log_2 N/N$. We observe for the two frequencies a very good agreement between the expected and observed complexities of the compression rate and memory requirements. Note that the storage is reduced by more than 95% as soon as $N \geq 10^5$.

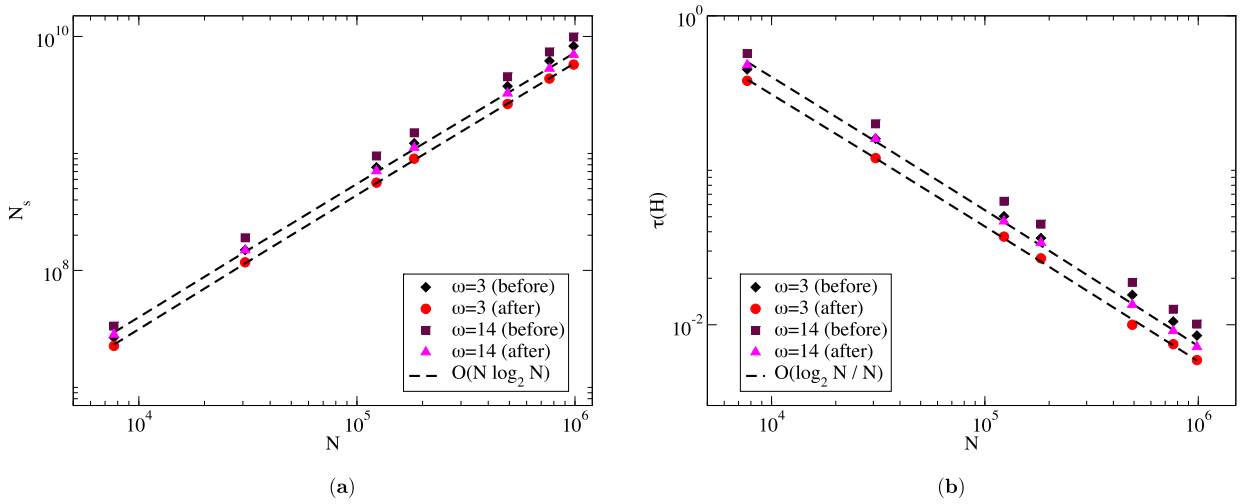


Fig. 10. Observed and estimated (a) memory requirements N_s and (b) compression rate $\tau(\mathcal{H})$ with respect to the number of degrees of freedom N for a fixed frequency ($\omega = 3$ or $\omega = 14$) for the single layer-operator and $\varepsilon_{\text{ACA}} = 10^{-4}$.

Table 4
Accuracy of the direct solver for a fixed frequency ($\omega = 3$ or $\omega = 14$) for the single-layer potential.

# DOFs	ω	$\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}}$	$\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$	$\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\delta_{\mathcal{H},F}$	$\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\frac{\delta}{\|\mathbf{b}\|_2}$
1926	3	10^{-4}	$3.25\ 10^{-15}$	$3.64\ 10^{-15}$	$9.16\ 10^{-17}$	6.40	$2.66\ 10^{-15}$
7686	3	10^{-4}	$4.64\ 10^{-4}$	$9.94\ 10^{-6}$	$7.21\ 10^{-5}$	6.37	$4.62\ 10^{-6}$
30 726	3	10^{-4}	$6.37\ 10^{-4}$	$1.18\ 10^{-5}$	$9.93\ 10^{-5}$	6.36	$5.53\ 10^{-6}$
122 886	3	10^{-4}	$8.14\ 10^{-4}$	$1.47\ 10^{-5}$	$1.27\ 10^{-4}$	6.36	$5.21\ 10^{-6}$
1926	3	10^{-6}	$3.25\ 10^{-15}$	$3.64\ 10^{-15}$	$9.16\ 10^{-17}$	6.40	$2.66\ 10^{-15}$
7686	3	10^{-6}	$5.07\ 10^{-6}$	$7.35\ 10^{-8}$	$7.92\ 10^{-7}$	6.37	$2.48\ 10^{-8}$
30 726	3	10^{-6}	$7.44\ 10^{-6}$	$9.38\ 10^{-8}$	$1.16\ 10^{-6}$	6.36	$2.93\ 10^{-8}$
122 886	3	10^{-6}	$9.25\ 10^{-6}$	$9.82\ 10^{-8}$	$1.45\ 10^{-6}$	6.36	$2.76\ 10^{-8}$
1926	14	10^{-4}	$5.69\ 10^{-15}$	$4.69\ 10^{-15}$	$7.16\ 10^{-17}$	29.29	$3.59\ 10^{-15}$
7686	14	10^{-4}	$3.89\ 10^{-3}$	$1.04\ 10^{-4}$	$1.36\ 10^{-4}$	28.15	$4.73\ 10^{-5}$
30 726	14	10^{-4}	$5.49\ 10^{-3}$	$1.29\ 10^{-4}$	$1.95\ 10^{-4}$	27.88	$5.48\ 10^{-5}$
122 886	14	10^{-4}	$6.24\ 10^{-3}$	$1.49\ 10^{-4}$	$2.22\ 10^{-4}$	27.82	$5.86\ 10^{-5}$
1926	14	10^{-6}	$5.69\ 10^{-15}$	$4.69\ 10^{-15}$	$7.16\ 10^{-17}$	29.29	$3.59\ 10^{-15}$
7686	14	10^{-6}	$4.59\ 10^{-5}$	$9.15\ 10^{-7}$	$1.62\ 10^{-6}$	28.15	$2.03\ 10^{-7}$
30 726	14	10^{-6}	$5.95\ 10^{-5}$	$1.15\ 10^{-6}$	$2.12\ 10^{-6}$	27.88	$2.44\ 10^{-7}$
122 886	14	10^{-6}	$6.42\ 10^{-5}$	$1.26\ 10^{-6}$	$2.30\ 10^{-6}$	27.82	$2.52\ 10^{-7}$

7.3. Certification of the results of the direct solver

While the previous subsection was dedicated to the study of the numerical efficiency of the method with respect to memory requirements, this subsection is more focused on the accuracy of the direct solver presented in Section 3.2. Since the $\mathcal{H}$-LU factorization is based on various levels of approximations, it is important to check the accuracy of the final solution. In Section 3.3, we have derived a simple estimator to certify the results of the $\mathcal{H}$-LU direct solver.

Before we study the accuracy, we recall that there are two important parameters in the method that correspond to two sources of error: ε_{ACA} and ε_{LU}. ε_{ACA} is tuned to the desired accuracy of the low-rank approximations of *admissible* blocks, performed with the ACA described in Section 2.3. ε_{LU} is the parameter used to adapt the accuracy when recompressions are performed to modify the format of a block (Section 3.2). It is clear that the best accuracy that can be achieved by the direct solver is driven by ε_{ACA}. For this reason, the two parameters are set to the same value in the following numerical results.

In Table 4, we report the value of the estimator $\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$ and the observed normalized residual $\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$ if (14) is solved with the $\mathcal{H}$-LU based direct solver. Two frequencies ($\omega = 3$ and $\omega = 14$) and four meshes are considered consistently with the previous subsection. When no compression is possible ($\tau(\mathcal{H}) = 1$ for $N = 1926$) the estimated and observed residuals are, as expected, of the order of the machine accuracy. For larger problems, the difference between the two residuals is not negligible. But importantly, when the required accuracy is reduced by two orders of magnitude, the two residuals follow the same decrease. In addition, for a fixed value of ε_{ACA}, the accuracy of the solver and of the estimator does not deteriorate much while the number of degrees of freedom increases.

To understand the lack of accuracy of our estimator, we add in Table 4 the three components of the estimator: $\delta_{\mathcal{H},F}$ which corresponds to the level of error introduced by the low-rank approximation in the hierarchical representation of

Table 5
Accuracy of the direct solver for a fixed frequency ($\omega = 3$ or $\omega = 14$) for the single-layer potential when the 2-norm is used instead of the Frobenius norm.

# DOFs	ω	$\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}}$	$\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$	$\mathcal{I}(\delta_{\mathcal{H},2}, \delta)$	$\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\delta_{\mathcal{H},F}$	$\delta_{\mathcal{H},2}$
7686	3	10^{-4}	$4.64\ 10^{-4}$	$7.91\ 10^{-5}$	$9.94\ 10^{-6}$	$7.21\ 10^{-5}$	$1.17\ 10^{-5}$
30 726	3	10^{-4}	$6.37\ 10^{-4}$	$5.46\ 10^{-5}$	$1.18\ 10^{-5}$	$9.93\ 10^{-5}$	$7.72\ 10^{-6}$
7686	3	10^{-6}	$5.07\ 10^{-6}$	$7.32\ 10^{-7}$	$7.35\ 10^{-8}$	$7.92\ 10^{-7}$	$1.11\ 10^{-7}$
30 726	3	10^{-6}	$7.44\ 10^{-6}$	$8.50\ 10^{-7}$	$9.38\ 10^{-8}$	$1.16\ 10^{-6}$	$1.29\ 10^{-7}$
7686	14	10^{-4}	$3.89\ 10^{-3}$	$6.32\ 10^{-4}$	$1.04\ 10^{-4}$	$1.36\ 10^{-4}$	$2.08\ 10^{-5}$
30 726	14	10^{-4}	$5.49\ 10^{-3}$	$9.50\ 10^{-4}$	$1.29\ 10^{-4}$	$1.95\ 10^{-4}$	$3.21\ 10^{-5}$
7686	14	10^{-6}	$4.59\ 10^{-5}$	$1.07\ 10^{-5}$	$9.15\ 10^{-7}$	$1.62\ 10^{-6}$	$3.72\ 10^{-7}$
30 726	14	10^{-6}	$5.95\ 10^{-5}$	$1.20\ 10^{-5}$	$1.15\ 10^{-6}$	$2.12\ 10^{-6}$	$4.20\ 10^{-7}$

Table 6
Accuracy of the direct solver for a fixed frequency ($\omega = 14$) for the double-layer potential.

# DOFs	ω	$\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}}$	$\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$	$\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\delta_{\mathcal{H},F}$	$\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\frac{\delta}{\|\mathbf{b}\|_2}$
1926	14	10^{-4}	$3.35\ 10^{-15}$	$3.28\ 10^{-15}$	$5.22\ 10^{-16}$	1.26	$2.69\ 10^{-15}$
7686	14	10^{-4}	$2.49\ 10^{-3}$	$7.16\ 10^{-5}$	$1.94\ 10^{-3}$	1.27	$2.58\ 10^{-5}$
30 726	14	10^{-4}	$3.18\ 10^{-3}$	$6.92\ 10^{-5}$	$2.43\ 10^{-3}$	1.30	$3.05\ 10^{-5}$
122 886	14	10^{-4}	$9.23\ 10^{-3}$	$5.70\ 10^{-3}$	$2.66\ 10^{-3}$	1.33	$5.70\ 10^{-3}$
1926	14	10^{-6}	$3.35\ 10^{-15}$	$3.28\ 10^{-15}$	$5.22\ 10^{-16}$	1.26	$2.69\ 10^{-15}$
7686	14	10^{-6}	$2.43\ 10^{-5}$	$4.77\ 10^{-7}$	$1.90\ 10^{-5}$	1.27	$9.85\ 10^{-8}$
30 726	14	10^{-6}	$3.31\ 10^{-5}$	$5.25\ 10^{-7}$	$2.54\ 10^{-5}$	1.30	$1.15\ 10^{-7}$
122 886	14	10^{-6}	$5.74\ 10^{-3}$	$5.70\ 10^{-3}$	$3.17\ 10^{-5}$	1.33	$5.70\ 10^{-3}$

Table 7
Number of degrees of freedom, corresponding number of wavenumbers spanned in the sphere diameter D ($D = n_{\lambda_s}\lambda_s$) on the different meshes of spheres and corresponding circular frequencies used in the examples of Section 8 (i.e for a fixed density of points per S-wavelength).

ω	8.25	16.5	33	40	66	84	92
N	7686	30 726	122 886	183 099	490 629	763 638	985 818
n_{λ_s}	2.6	5.2	10.5	12.6	21	26.6	29.1

the system matrix (computed with the Frobenius norm), the ratio $\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$ and the normalized ratio $\frac{\delta}{\|\mathbf{b}\|_2}$ that accounts for the stability of the $\mathcal{H}$-LU factorization. On the one hand the contribution from $\frac{\delta}{\|\mathbf{b}\|_2}$ is lower that the normalized residual that we try to estimate. On the other hand, $\delta_{\mathcal{H},F}$ is of the order of the required accuracy ε_{ACA}. But since this term is multiplied by the ratio $\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$ the part in the estimator accounting for the error introduced by the low-rank approximations is overestimated.

To accelerate the computation of the estimator, we have used the Frobenius norm. To check the effect of the use of this norm, we report in Table 5 the values of the estimator $\mathcal{I}(\delta_{\mathcal{H},2}, \delta)$ i.e. when the 2-norm is used. We consider only moderate size problems due to the need of the computation of the singular value decomposition of the complete matrix. As expected, the estimator $\mathcal{I}(\delta_{\mathcal{H},2}, \delta)$ is much more accurate than $\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$ but its cost is prohibitive. A good compromise consists in checking only the two main components of the estimator: $\delta_{\mathcal{H},F}$ and $\frac{\delta}{\|\mathbf{b}\|_2}$ to check that the level of error introduced in the different parts of the solver is consistent with the user parameter $\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}}$.

In Table 6, we consider now the case where the double-layer potential based on equation (15) is used and solved with the $\mathcal{H}$-LU direct solver. It is clear that the behaviors of the single and double layer potentials are similar also in terms of numerical accuracy.

All these numerical experiments confirm the theoretical study of the Taylor expansion, which yields that $\mathcal{H}$-matrix based solvers are very efficient tools for the *low-frequency* elastodynamics.

8. Towards highly oscillatory elastodynamics

The goal of Section 8 is to determine the behavior of the algorithm for a given fixed density of points per S-wavelength while the number of discretization points is increased (i.e. in the *higher frequency* regime). The circular frequency ω is thus fitted to the mesh length to keep the density of points per S-wavelength fixed for the different meshes. In this case, an increase of the number of degrees of freedom N leads to an increase of the size of the body to be approximated. In Table 7 we report the number of degrees of freedom, corresponding circular frequencies and number of S-wavelengths spanned in the sphere diameter used in the examples.

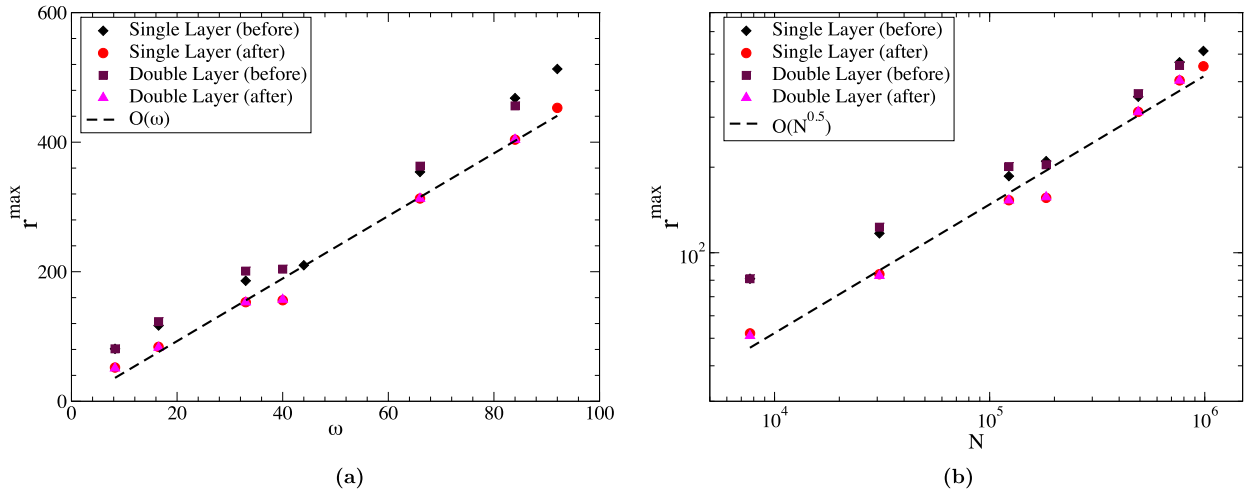


Fig. 11. Maximum numerical rank observed (before/after the recompression step) for a fixed density of points per wavelength, i.e. while increasing the circular frequency with the number of discretization points, for $\varepsilon_{\text{ACA}} = 10^{-4}$: (a) with respect to the circular frequency and (b) with respect to the number of degrees of freedom.

8.1. Storage requirements for the single and double-layer potentials for a fixed density of points per S-wavelength

In Section 5.3, it has been shown that for a fixed geometry with a fixed density of points per S-wavelength, the maximum rank of the *η-admissible* blocks is expected to grow linearly with the frequency until the *high-frequency regime* is achieved. The goal of this section is to validate the existence of this pre-asymptotic regime. In Fig. 11a, we report the maximum numerical rank observed among all the $\hat{\eta}$*-admissible* blocks before and after the recompression step with respect to the circular frequency ω, for both the single and double layer potentials. In accordance to the study of the Taylor expansion, the growth of the numerical rank after the recompression step (i.e. the intrinsic rank of the blocks) is seen to be linear with respect to the circular frequency. The behavior of the single layer and double layer operators are again very similar.

In practice however, we are interested in the complexity with respect to the number of degrees of freedom. In this section, the number of discretization points is fitted to the circular frequency in order to keep 10 discretization points per S-wavelength. Since the density of points is fixed, the mesh size is $h = O(\lambda_S) = O(\omega^{-1})$. In addition since the mesh for the BEM is a surface mesh, the mesh size is $h = O(N^{-2})$. Hence, the maximum numerical rank is expected to be of the order of $O(N^{1/2})$. In Fig. 11b, we report the maximum numerical rank observed among all the $\hat{\eta}$*-admissible* blocks before and after the recompression step with respect to the number of DOFs N, for both the single and double layer potentials. The growth of the numerical rank after the recompression step (i.e. the intrinsic rank of the blocks) is indeed seen to be of the order of $O(N^{1/2})$.

Ultimately we are not interested in the maximum numerical rank but rather on the memory requirements of the algorithm. In Figs. 12a–b, we report the memory requirements N_s and compression rate $\tau(\mathcal{H})$, with respect to the number of degrees of freedom N for $\varepsilon_{\text{ACA}} = 10^{-4}$. Since the rank is of the order of $O(N^{1/2})$, N_s is expected to be of the order of $N^{3/2}\log_2 N$ and $\tau(\mathcal{H})$ of the order of $\log_2 N/N^{1/2}$. In practice, observed complexities are lower than the expected one. The reason is that the estimation of the memory requirements gives only an upper bound based on the estimation of the maximum numerical rank. But, all the *admissible* blocks do not have the same numerical rank such that the complexity is lower than the estimated one.

8.2. Certification of the results of the direct solver for higher frequencies

Once the existence of the *pre-asymptotic regime* is established, and thus the efficiency of the hierarchical representation of the matrix to reduce computational times and memory requirements, is demonstrated, this subsection is devoted to the study of the accuracy of the method in this regime.

In Table 8, we report the values of the estimator $\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$ and the observed normalized residual $\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$ if equation (14) is solved with the $\mathcal{H}$-LU based direct solver. Four meshes are considered and the density of points per S-wavelength is fixed consistently with the previous subsection. When no compression is possible ($\tau(\mathcal{H}) = 1$ for $N = 1\,926$) the estimated and observed residuals are, as expected, of the order of the machine accuracy. When the required accuracy is reduced by two orders of magnitude, the two residuals follow the same decrease. The estimator computed in Frobenius norm is again not very sharp and the solver could appear to be less accurate when the frequency increases. But the discrepancy between the true system matrix and its $\mathcal{H}$-matrix representation is seen to be of the order of ε_{ACA} even though computed in Frobenius norm. The increase of the estimator is due to the increase of the ratio $\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$. In addition, the ratio $\frac{\delta}{\|\mathbf{b}\|_2}$ indicates the very good stability of the $\mathcal{H}$-LU factorization.

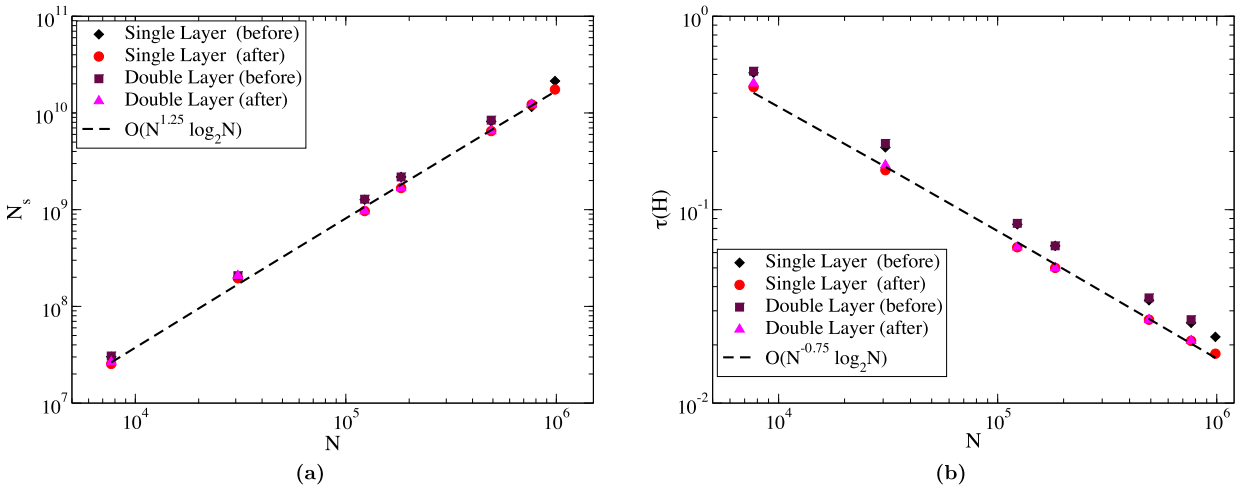


Fig. 12. Observed and estimated (a) memory requirements N_s and (b) compression rate $\tau(\mathcal{H})$ with respect to the number of degrees of freedom N for a fixed density of points per S-wavelength for the single and double layer-operators and $\varepsilon_{\text{ACA}} = 10^{-4}$.

Table 8
Comparison of the estimation of the error and the actual error of the direct solver for a fixed density of ten points per S-wavelength for the single-layer potential.

# DOFs	ω	$\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}}$	$\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$	$\frac{\|\mathbf{b}-\mathbb{A}\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\delta_{\mathcal{H},F}$	$\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{b}\|_2}$	$\frac{\delta}{\|\mathbf{b}\|_2}$
1926	4	10^{-4}	$3.12\ 10^{-15}$	$3.42\ 10^{-15}$	$9.07\ 10^{-17}$	8.37	$2.36\ 10^{-15}$
7686	8.25	10^{-4}	$1.73\ 10^{-3}$	$4.28\ 10^{-5}$	$1.02\ 10^{-4}$	16.83	$2.23\ 10^{-5}$
30 726	16.5	10^{-4}	$6.42\ 10^{-3}$	$1.68\ 10^{-4}$	$1.94\ 10^{-4}$	32.66	$7.04\ 10^{-5}$
1926	4	10^{-6}	$3.12\ 10^{-15}$	$3.42\ 10^{-15}$	$9.07\ 10^{-17}$	8.37	$2.36\ 10^{-15}$
7686	8.25	10^{-6}	$1.79\ 10^{-5}$	$2.61\ 10^{-7}$	$1.06\ 10^{-6}$	16.83	$9.63\ 10^{-8}$
30 726	16.5	10^{-6}	$7.14\ 10^{-5}$	$1.36\ 10^{-6}$	$2.18\ 10^{-6}$	32.66	$2.93\ 10^{-7}$

These results confirm the numerical efficiency and accuracy of $\mathcal{H}$-matrix based solvers even for higher frequencies. The estimator $\mathcal{I}(\delta_{\mathcal{H},F}, \delta)$ gives a rough idea of the accuracy of the solver. But it is better to check separately the two components of the error.

Remark 8.1. The computational times of the iterative and direct solvers have not been compared due to the difficulty to perform a fair and exhaustive comparison. On the one hand, the iterative solver can easily be optimized and parallelized but its performances depend largely on the quality of the preconditioner used to reduce the number of iterations. On the other hand, the optimization of the direct solver is a more involved operation that requires a task-based parallelization. Our experience with the two solvers is that the computational time of the direct solver is increasing very rapidly as a function of the number of DOFs if no parallelization is performed. However, the preconditioning of iterative BEM solvers for oscillatory problems is still an open question.

9. Conclusion and future work

In this work, $\mathcal{H}$-matrix based iterative and direct solvers have been successfully extended to 3D elastodynamics in the frequency domain. To perform the low-rank approximations in an efficient way, the Adaptive Cross Approximation has been modified to consider problems with vector unknowns. In addition, the behavior of the hierarchical algorithm in the context of oscillatory kernels has been carefully studied. It has been shown theoretically and then numerically confirmed that the maximum numerical rank among all the *η-admissible* blocks is constant if the frequency is fixed and the density of discretization points per wavelength is increased (i.e. for the low-frequency regime). On the other hand, if the density of points per S-wavelength is fixed, as it is usually the case for higher frequency problems, the growth of the maximum numerical rank among all the *η-admissible* blocks is limited to $O(N^{1/2})$. Then, since the $\mathcal{H}$-LU factorization is based on various levels of approximations, a simple estimator to certify the results of the $\mathcal{H}$-LU direct solver has been derived. It has been shown that not only the accuracy of the $\mathcal{H}$-matrix representation can be monitored by a simple parameter but also the additional error introduced by the $\mathcal{H}$-LU direct solver. Globally, when combined with the Boundary Element Method formulation, the $\mathcal{H}$-matrix representation of the system matrix permits to drastically reduce computational costs in terms of memory requirements and computational times for the frequency range that can be considered with the current available computational resources.

Applications of the present $\mathcal{H}$-matrix accelerated BEM to 3D realistic cases in seismology are underway. Moreover, a natural extension of this work will be to determine the efficiency of this approach for isotropic and anisotropic visco-elastic problems.

Acknowledgements

The authors would like to thank Shell for funding the PhD of Luca Desiderio.

References

[1] L. Banjai, W. Hackbusch, Hierarchical matrix techniques for low-and high-frequency Helmholtz problems, IMA J. Numer. Anal. 28 (1) (2008) 46–79.
[2] M. Bebendorf, Approximation of boundary element matrices, Numer. Math. 86 (4) (2000) 565–589.
[3] M. Bebendorf, Hierarchical Matrices, Springer, 2008.
[4] M. Bebendorf, C. Kuske, R. Venn, Wideband nested cross approximation for Helmholtz problems, Numer. Math. 130 (1) (2015) 1–34.
[5] M. Bebendorf, S. Rjasanow, Adaptive low-rank approximation of collocation matrices, Computing 70 (1) (2003) 1–24.
[6] M. Bonnet, Boundary Integral Equation Methods for Solids and Fluids, Wiley, 1999.
[7] S. Börm, $\mathcal{H}^2$-matrix arithmetics in linear complexity, Computing 77 (1) (2006) 1–28.
[8] S. Börm, Directional $\mathcal{H}^2$-matrix compression for high-frequency problems, preprint, arXiv:1510.07087, 2015.
[9] S. Börm, S. Christophersen, Approximation of integral operators by Green quadrature and nested cross approximation, Numer. Math. 133 (3) (2016) 409–442.
[10] S. Börm, L. Grasedyck, Hybrid cross approximation of integral operators, Numer. Math. 101 (2) (2005) 221–249.
[11] S. Börm, L. Grasedyck, W. Hackbusch, Introduction to hierarchical matrices with applications, Eng. Anal. Bound. Elem. 27 (5) (2003) 405–422.
[12] S. Börm, J.M. Melenk, Approximation of the high-frequency Helmholtz kernel by nested directional interpolation, preprint, arXiv:1510.07189, 2015.
[13] D. Brunner, M. Junge, P. Rapp, M. Bebendorf, L. Gaul, Comparison of the fast multipole method with hierarchical matrices for the Helmholtz–BEM, Comput. Model. Eng. Sci. 58 (2) (2010) 131–160.
[14] S. Börm, L. Grasedyck, W. Hackbusch, Hierarchical Matrices, Technical Report, Max Planck Institute for Mathematics in the Sciences, 2003.
[15] S. Chaillat, M. Bonnet, Recent advances on the fast multipole accelerated boundary element method for 3d time-harmonic elastodynamics, Wave Motion 50 (7) (2013) 1090–1104.
[16] M. Darbas, F. Le Louër, Well-conditioned boundary integral formulations for high-frequency elastic scattering problems in three dimensions, Math. Methods Appl. Sci. 38 (9) (2015) 1705–1733.
[17] E. Darve, The fast multipole method: numerical implementation, J. Comput. Phys. 160 (1) (2000) 195–240.
[18] K. Delamotte, Une étude du rang du noyau de l'équation de Helmholtz: application des $\mathcal{H}$-matrices à l'EFIE, 2016.
[19] L. Desiderio, $\mathcal{H}$-Matrix Based Solvers for 3D Elastodynamic Boundary Integral Equations, PhD Thesis, Paris-Saclay University, 2017.
[20] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd edition, JHU Press, 2012.
[21] L. Grasedyck, Theorie und Anwendungen Hierarchischer Matrizen, PhD Thesis, Christian-Albrechts Universität Kiel, 2001 (in German).
[22] L. Greengard, J. Huang, V. Rokhlin, S. Wandzura, Accelerating fast multipole methods for the Helmholtz equation at low frequencies, IEEE Comput. Sci. Eng. 5 (3) (1998) 32–38.
[23] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (2) (1987) 325–348.
[24] W. Hackbusch, A sparse matrix arithmetic based on $\mathcal{H}$-matrices, part I: introduction to $\mathcal{H}$-matrices, Computing 62 (2) (1999) 89–108.
[25] W. Hackbusch, Hierarchical Matrices: Algorithms and Analysis, vol. 49, Springer, 2015.
[26] W. Hackbusch, Z.P. Nowak, On the fast matrix multiplication in the boundary element method by panel clustering, Numer. Math. 54 (4) (1989) 463–491.
[27] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, M. Tygert, Randomized algorithms for the low-rank approximation of matrices, Proc. Natl. Acad. Sci. USA 104 (51) (2007) 20167–20172.
[28] B. Lizé, Résolution directe rapide pour les éléments finis de frontière en électromagnétisme et acoustique : $\mathcal{H}$-matrices. Parallélisme et applications industrielles, PhD thesis, 2014, Paris 13 (in French).
[29] M. Messner, M. Schanz, An accelerated symmetric time-domain boundary element formulation for elasticity, Eng. Anal. Bound. Elem. 34 (11) (2010) 944–955.
[30] M. Messner, M. Schanz, E. Darve, Fast directional multilevel summation for oscillatory kernels based on Chebyshev interpolation, J. Comput. Phys. 231 (4) (2012) 1175–1196.
[31] A. Milazzo, I. Benedetti, M.H. Aliabadi, Hierarchical fast BEM for anisotropic time-harmonic 3-D elastodynamics, Comput. Struct. 96 (2012) 9–24.
[32] S. Rjasanow, L. Weggler, Matrix valued adaptive cross approximation, Math. Methods Appl. Sci. (2016).
[33] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (2) (1985) 187–207.
[34] S. Sauter, Variable order panel clustering, Computing 64 (3) (2000) 223–261.
[35] S.A. Sauter, C. Schwab, Boundary Element Methods, Springer Series in Computational Mathematics, vol. 39, Springer-Verlag, 2011.
[36] M. Stolper, Computing and compression of the boundary element matrices for the Helmholtz equation, J. Numer. Math. 12 (1) (2004) 55–75.
[37] T. Takahashi, A wideband fast multipole accelerated boundary integral equation method for time-harmonic elastodynamics in two dimensions, Int. J. Numer. Methods Eng. 91 (5) (2012) 531–551.