NNT : 2017SACLY002

Thèse de doctorat de l'Université Paris-Saclay préparée à l'ENSTA ParisTech

École Doctorale n°574 EDMH
Mathématiques Hadamard

Mathématiques appliquées

Par

M. Luca Desiderio

$\mathcal{H}$-matrix based Solvers for 3D Elastodynamic Boundary Integral Equations

Thèse présentée et soutenue à Palaiseau, le 27/01/2017

Composition du Jury :

Mme Alessandra Aimi	Université de Parme, Italie	Examinatrice
M. François Alouges	École Polytechnique	Examinateur
Mme Stéphanie Chaillat	ENSTA-ParisTech	Co-directrice de thèse
M. Patrick Ciarlet	ENSTA-ParisTech	Directeur de thèse
M. Eric Darrigrand	Université Rennes 1	Rapporteur
M. René-Édouard Plessix	Shell Global Solutions	Invité
M. Martin Schanz	Université de Graz, Autriche	Rapporteur

universite PARIS-SACLAY
École doctorale de mathématiques Hadamard (EDMH)

Title: $\mathcal{H}$-matrix based Solvers for 3D Elastodynamic Boundary Integral Equations

Keywords: Boundary Element Method, $\mathcal{H}$-matrices, Adaptive Cross Approximation, Randomized Singular Value Decomposition, 3D Elastodynamics, Forced Vibration Problems.

Abstract: This thesis focuses on the theoretical and numerical study of fast methods to solve the equations of 3D elastodynamics in frequency-domain, in view of the optimization of the convergence of seismic inversion problems. We use the Boundary Element Method (BEM) as discretization technique, in association with the hierarchical matrices ($\mathcal{H}$-matrices) technique for the fast solution of the resulting linear system. The BEM is based on a boundary integral formulation which requires the discretization of the only domain boundaries. Thus, this method is well suited to treat seismic wave propagation problems. A major drawback of classical BEM is that it results in dense matrices, which leads to high memory requirement ($\mathcal{O}(N^2)$, if N is the number of degrees of freedom) and computational costs. Therefore, the simulation of realistic problems is limited by the number of degrees of freedom. Several fast BEMs have been developed to improve the computational efficiency. We propose a fast $\mathcal{H}$-matrix based direct BEM solver. Even though the concept of $\mathcal{H}$-matrices is simple to understand, its implementation requires significant algorithmic developments, such as the product of matrices with different structures (compressed or not). Another delicate point is the use of the low-rank approximation techniques in the context of vectorial problems. A numerical study has been done to implement the $\mathcal{H}$-matrix technique. An original contribution of this work is the investigation of the expected low rank in the case of oscillatory kernels. Furthermore, we have analyzed the influence of the various parameters of the method in acoustics and 3D elastodynamics, in order to calibrate their optimal values. Finally, we have treated the problem of the propagation of a seismic wave in an elastic half-space subject to point force applied to the surface.

Titre : Solveurs fondés sur la méthode des $\mathcal{H}$-matrices pour les équations intégrales en élastodynamique 3D

Mots clés : Éléments Finis de Frontière, $\mathcal{H}$-matrices, Approximation Adaptive Croisée, Décomposition en Valeurs Singulières Aléatoire, Élastodynamique 3D, Problèmes de Vibrations Forcées.

Résumé : Cette thèse porte sur l'étude théorique et numérique des méthodes rapides pour résoudre les équations de l'élastodynamique 3D en domaine fréquentiel, en vue d'optimiser la convergence des problèmes d'inversion sismique. La méthode repose sur l'utilisation des éléments finis de frontière (BEM) pour la discrétisation et sur les techniques de matrices hiérarchiques ($\mathcal{H}$-matrices) pour l'accélération de la résolution du système linéaire. La BEM, qui correspond à la résolution numérique des équations intégrales de frontière, présente l'avantage de ne nécessiter que la discrétisation de la frontière du domaine de calcul considéré. De plus, elle permet de simuler des milieux étendus en évitant la forte dispersion numérique associée à d'autres schémas. Cette méthode est donc bien adaptée pour le calcul de la propagation d'ondes sismiques. Les difficultés sont liées d'une part aux besoins de stockage en mémoire de l'ordre de N^2 (N étant le nombre de degrés de liberté) et d'autre part au coût de la résolution. La simulation de problèmes réalistes est donc limitée par le nombre de degrés de liberté que peut traiter sur une machine donnée. Afin de dépasser ces limites, des méthodes BEMs rapides ont été développées. Nous avons proposé un solveur direct pour le BEMs en utilisant une factorisation LU et un stockage hiérarchique. Si le concept des $\mathcal{H}$-matrices est simple à comprendre, sa mise en œuvre requiert des développements algorithmiques importants tels que la gestion de la multiplication de matrices représentées par des structures différentes (compressées ou non). Un autre point délicat est l'utilisatioin des méthodes d'approximations par matrices compressées (de rang faible) dans le cadre des problèmes vectoriels. Une étude algorithmique a donc été faite pour mettre en œuvre la méthode des $\mathcal{H}$-matrices. Nous avons par ailleurs estimé théoriquement le rang faible attendu pour les noyaux oscillants, ce qui constitue une nouveauté, et montré que la méthode est utilisable en élastodynamique. En outre on a étudié l'influence des divers paramètres de la méthode en acoustique et en élastodynamique 3D, à fin de calibrer leur valeurs numériques optimales. En fin, nous avons étudié un problème de propagation d'une onde sismique dans un demi-espace élastique soumis à une force ponctuelle en surface.

Quelquefois je regarde en arrière
et je pense à l'avenir:
le sort, la réalité, la mer... Paris.
Mon enfance innocente:
les loisirs, l'amour,
les selfies, la célébrité...
la recherche.

La vie est magnifique
pourquoi tu te la compliques?

Lalla

À Pollyanna in dark, stra wow!

Remerciements

Personne ne m'en voudra si je ne respecte pas la tradition et si je commence par remercier toute ma famille, pour m'avoir soutenu de proche ou de loin pendant ces trois années. Une pensée particulière pour mon père (tu as gagné notre pari!), pour ma mère et pour mon grand-père Pietro, qui est toujours avec moi.

D'abord, une thèse est une expérience scientifique qui ne peut pas se bien développer sans les directeurs de thèse. Je remercie donc Patrick Ciarlet et Stéphanie Chaillat pour tout le temps que ils m'ont consacré et pour m'avoir montré une façon différente de faire la recherche. Je remercie chaleureusement aussi Eric Darrigrand et Martin Schanz qui ont accepté de rapporter ma thèse. Ils ont contribué de façon significative à ma mise en confiance sur la qualité des travaux que j'ai menés pendant ces trois dernières années. Je suis très honoré de la présence de François Alouges et Alessandra Aimi au sein de mon jury. Je remercie particulièrement cette dernière de m'avoir encouragé et soutenu, elle restera pour toujours "ma maman mathématique". Je tiens notamment à mentionner les discussions instructives que j'ai eues avec René-Édouard Plessix et Jean Virieux lors de réunions d'avancement de ma thèse, qui m'ont permis d'approfondir mon sujet. Merci à Mauro Diligenti, les félicitations duquel m'ont fait pleurer de joie. Pendant ma thèse, j'ai eu la chance extraordinaire de travailler dans un environnement stimulant et convivial, et j'en remercie collectivement tous les membres de l'UMA. Une pensée particulière pour David Lefèvre qui a changé ma prospective en me disant: *le mieux de ta vie n'est pas encore arrivé.* Merci à Francesco Russo et Hasna Zidani, avec lesquels j'ai passé des très bons moments au dehors du labo. Je remercie Maurice Diamantini qui a toujours été de bon conseil et Corrinne Chen pour nos exchanges culinaires franco-italiens. Enfin, quelques mots aussi pour remercier Myriam Brettes, le depart de laquelle a été vraiment très dur. Je souhaite faire un clin d'œil à tous mes collègues doctorants et jeunes docteurs que j'ai côtoyés au long de ma thèse. En particulier, merci à Laure Pesudo, la très méchante Sonya Maslovskaya et Alicia Simon-Petit, des filles exceptionnelles, sans lesquelles je ne sais pas si je serais arrivé à tenir le coup.

Il ne faut pas oublier qu'une thèse est aussi une expérience humaine. Je me suis mis en question, je suis tombé et je me suis relevé, j'ai pleuré et j'ai ri, j'ai vu le négatif mais je suis arrivé à me concentrer sur le positif grace à mes amis. Je pense en particulier à Vanna, que je connais depuis presque une vie. Merci pour être toujours si present et attentive. Merci pour avoir partagé avec moi la belle personne que tu es. Merci pour me comprendre avec un seul regard. Mais, surtout, merci pour être venue pour moi et non pour Laurona. J'espère que nous serons *il più bel spettacolo dopo il big bang (altro che Lady Gaga)* encore longtemps. Je remercie Achille Sassi et je réponds à sa question: c'est nous qui allons couper nos chaînes! Un grand merci est pour Giancarlo Attolini, mon "socio" et mon rocher depuis 5 ans, pour les merveilleuses aventures que nous avons vécues ensemble. Davide, Marco, Ema (sisters!!!!), Fausto, Dave et Francesco, je ne serais jamais arrivé à aboutir ma thèse sans nos soirées au Bataclan, sans nos verres au Cox, sans nos dîners au 16 rue Petit, sans nos pizzas chez O'scia... sans vous! Vous avez été vraiment ma famille italienne à Paris et des amis précieux, merci! Pendant la dernière ligne droit j'ai été soutenu par Andrea, Marco et Giuseppe, trois italiens qui venaient d'arriver à Paris. C'est grace à eux si je n'ai pas lâché prise et je ne suis pas resté seul. Je remercie Margherita et Francesca parce que elles ont toujours trouvé les bons mots pour me faire sourire. Discrètement, je dis merci à Thomas, qui a appris à me faire confiance. Merci aussi à Eva Grasso et Elena Ceccarelli, qui m'ont donné la force pour résister, à Alessandro Pascale, mon psy, et aux amis du club Feydeau.

Comme toutes les adventures, ma thèse a aussi eu une bande originale. Je remercie Lalla Bittch, Immanuel Casto, Cristina d'Avena, Lady Gaga, Madonna, Rihanna, Britney Spears, Kylie Minogue, Katy Perry, Anna Tatangelo, Patty Pravo, Loredana Berté, Ornella Vannoni, Orietta Berti, Rino Gaetano, Pupo, Gigi d'Alessio, Tony Tammaro, Alessandra Amoroso, Emma, J-Ax, Fedez, Sabrina Salerno, Jo Squillo, Mika, Tiziano Ferro, Eros Ramazzotti, Marco Mengoni, Vasco Rossi, Renato Zero, Jovannotti, le amiche dell'Abruzzo (les vrais: Giorgia, Elisa, Laura Pausini, Gianna Nannini e Fiorella Mannoia) et Maria Defilippi (parce que il faut toujours la remercier). Mais aussi à Puccini, Verdi, Mozart, Beethoven, Mahler, Chopin et Maria Callas.

Pour finir merci à qui n'a pas eu la force pour rester à mes côtes et s'enfui... tu m'as aidé à prendre confiance en moi même et à comprendre que je peux y arriver tout seul!

Contents

Introduction **3**

1 Elastodynamic Fundamental Equations and Boundary Integral Formulation **11**

1.1 Introduction 12

1.2 The Basics of Elastodynamics 12

1.2.1 Linear Elastodynamic System 14

1.2.2 Derivation of the P- and S-waves Velocities 15

1.2.3 Body and Surface Waves 16

1.2.4 Frequency-Domain Elastodynamic Equations 19

1.2.5 Constitutive Law for a Linear Viscoelastic Medium 20

1.3 Boundary Element Methods for 3D Elastodynamics 21

1.3.1 Integral Representation and Boundary Integral Equation 22

1.3.2 Boundary Element Method 24

1.3.3 BEM System 26

1.3.4 Solution Strategies for BEM Systems 27

2 Principle of $\mathcal{H}$-matrices **29**

2.1 Introduction 30

2.2 General Description of $\mathcal{H}$-Matrices 30

2.2.1 Motivation of an $\mathcal{H}$-Matrix Representation of BEM Matrices . . 31

2.2.2 Clustering of the unknowns 34

2.2.3 Subdivision of the matrix 35

2.2.4 Admissibility Condition 37

2.3 Algorithms to Perform Low-Rank Approximations 38

2.3.1 Truncated Singular Value Decomposition 38

2.3.2 Fully-Pivoted Adaptive Cross Approximation 38

2.3.3 Partially-Pivoted Adaptive Cross Approximation 42

2.3.4 Recompression of the Admissible Blocks 43

2.4 Extension of $\mathcal{H}$-matrix to BEM for Elastodynamics 44

2.4.1 Application of ACA to Oscillatory Kernels: Theoretical Estimates 45

2.4.2 Vectorial Partial-Pivoted Adaptive Cross Approximation 48

2.5 $\mathcal{H}$-matrix Based Solvers 49

2.5.1 $\mathcal{H}$-matrix Based Iterative Solver 49

2.5.2 $\mathcal{H}$-matrix Arithmetic 50
2.5.3 $\mathcal{H}$-matrix Based Direct Solver 63
2.6 Conclusions 65

3 Application of $\mathcal{H}$-matrices to BEM for 3D Frequency Domain Elastodynamics 67
3.1 Introduction 68
3.2 Scattering Problem 68
3.2.1 Traces, Integral Representation Formula and Integral Equations 70
3.2.2 Boundary Element Method 72
3.3 $\mathcal{H}$-matrices: Computational Aspects 72
3.3.1 Compression Rate 74
3.3.2 Accuracy of the $\mathcal{H}$-matrix representation 76
3.3.3 Accuracy of the $\mathcal{H}$-matrix/vector multiplication 77
3.4 Scattering of a Plane Wave by a Rigid Sphere 78
3.4.1 Diffraction of P-waves by a Unit Sphere 80
3.4.2 Numerical Study of the Complexity 81
3.4.3 $\mathcal{H}$-matrix Based Iterative Solver 84
3.4.4 $\mathcal{H}$-matrix Based Direct Solver 88
3.4.5 Error Estimate to Certify the Results of the $\mathcal{H}$-LU Direct Solver 91
3.5 Parallelization 96
3.6 Conclusions 97

4 Application of Fast BEMs to Forced Vibration Problems 99
4.1 Introduction 100
4.2 Seismic Method 100
4.3 Forced Vibration Problem 102
4.3.1 Multi-Domain BEM Formulation 103
4.3.2 Hierarchical Multi-Domain BEM Formulation 106
4.4 Green's Function of a Homogeneous Half-Space 106
4.4.1 Limitations of the Vector Version of the ACA for Half-Space Problems 108
4.4.2 Randomized Singular Value Decomposition 109
4.4.3 $\mathcal{H}$-matrix based BEM for Half-Space Problems 110
4.5 Elastic Half-Space with a Semi-Spherical Alluvial Basin 119
4.6 Conclusions 121

5 Conclusions and Directions for Future Work 123
5.1 Conclusions 124
5.2 Directions for Future Work 125

Introduction

The modeling of seismic wave propagation to understand complex phenomena such as site-effects or soil-structure interaction is an active area of research. The difficulties are related to the complexity of the system to model and the large spatial scale of problems. The development of efficient numerical approaches to simulate visco-elastic wave propagation in complex media is crucial for many topics going from understanding the geodynamics of the Earth, the management of underground resources as well as the mitigation of seismic risk.

Historical Background

Early historical records contain references to earthquakes as far back as 2000 B.C.E. Mentions to major earthquakes on the Dead Sea fault system are present in the Bible and in the Epic of Gilgamesh, where supernatural elements are imported in the narration with a natural tendency to exaggeration in describing such phenomena [19].

Thales of Miletos (624 to 546 B.C.E.) was the first to abandon the mythological explanations, expanding the theory of a natural cause (water) for earthquakes. Afterwards some ancient writers on natural philosophy (as Thucidides, Strabo, Seneca, Livy and Pliny) offered rational explications of earthquake phenomena. In particular, Aristotle (384 to 322 B.C.E.) speculated that earthquakes are caused by air trapped inside the Earth, as it is trying to escape. This theory was destined to influence European thought until the seventeenth century.

In 1660 Robert Hooke laid the foundation for the theory of elasticity, stating the one-dimensional linear stress-strain relationship. After almost a century (November 1^{st}, 1755) the Lisbon earthquake changed dramatically man's outlook on the phenomenon of earthquakes. This tragic event was the starting point of the pioneering studies of John Michell (1761), who firmly established that earthquakes originate within the Earth and they propagate in the form of waves of finite velocity (he estimated that the earthquake waves after Lisbon earthquake had travelled outward at 530 m/sec).

The advent of elastodynamics began in 1821, when Navier derived the elastodynamic displacement equation, using the moduli of elasticity defined by Young, the first to recognize shear stress as an elastic strain. In 1828, Poisson established the existence of longitudinal and shear waves (body waves) in elastic solids, whose physical interpretation is due to Stokes (1845). In a theoretical study, Lord Rayleigh (1885) discovered another type of elastic waves, subsequently known as Rayleigh waves (surface waves), accommodated by a homogeneous elastic medium at its boundary. The mathematical results of Poisson and Lord Rayleigh were confirmed ten years later (1897), when Oldhan identified on earthquake recording the three types of waves predicted. At the beginning of twentieth century (1911) Love explained the occurrence of transversely polarized surface waves, not included in the theories of Lord Rayleigh. These so-called Love waves allowed to derive estimates of the thickness of the Earth's crust and its rigidity. Love's work gave rise to a large number of mathematical investigations, yielding much information on the structure of continents and oceans.

Modelling Seismic Wave Propagation

Since the 1960s, when electronic computers became generally available in universities, the development of numerical schemes to solve elastodynamic problems is of great importance for providing quantitative informations for earthquake disaster prevention and mitigation, for mapping shallow underground structures, such as buried valleys, and for hydrocarbon exploration and extraction. All the classical numerical methods have been employed to handle seismic wave propagation. In particular, we remark the distinction presented in [167] between *differential* or *strong implementations* (for example the Finite Difference Method and the Pseudospectral Method), i.e. numerical methods based on the *velocity-stress formulation* of the equation of motion in its differential form subject to certain boundary conditions, and *integral* or *weak implementations* (for example the Finite Element Method, the Spectral Element Method and the Discontinuous Galerkin Method), i.e. numerical methods based on an integral implementation of the equation of motion that contains the boundary conditions implicitly. Let us review first methods that require a meshing of the volume where the waves propagate.

Finite Difference Method. Perhaps the most intuitively appealing numerical technique is the Finite Difference Method (FDM) [123, 165, 166]. In this approach, the computational domain is covered by a space-time grid, that is by a set of discrete grid positions in space and time. The first-order space and time derivatives in velocity-stress formulation of the equation of motion are approximated by taking differences between neighboring grid points. The original differential equation is thus replaced by a system of algebraic equations, having three basic properties: consistency with the original differential equations, stability and convergence of the numerical solution. These properties have to be analyzed prior to the numerical calculation. To limit numerical grid dispersion and anisotropy, one typically uses a fourth-order scheme in space, and a second-order scheme in time [116]. The main advantage of the FDM is its relative ease of implementation but an improper application of the method can give very inaccurate results. However, when properly treated, the results provided are valid in the interior of the domain. Since less accurate one-sided difference operators come in to play near the domain boundary, the implementation of free surface boundary conditions is often problematic. For this reason surface waves can be poorly simulated [75, 137, 150]. To improve this situation, optimal or compact finite difference operators are developed [7, 70, 179, 178].

Pseudospectral Method. The need for more accurate spatial derivative operators is the motivation of the Pseudospectral Method (PM). Starting from the velocity-stress formulation of the equation of motion, the components of the velocity field and the stress tensor are expanded in term of known global basis functions, for example Fourier basis functions [66, 67, 107, 109, 170], Chebyshev basis functions [31, 94, 95, 108, 110, 161, 163] or a combination of Fourier and Chebyshev basis functions [162]. Alternatively, it is possible to consider directly the equation of motion and use wavelets as basis functions [89]. In both these cases, the space derivatives are computed in the wavenumber-domain with the aid of fast transforms. Since this derivative operation is based on information from all available grid points, it is very accurate and allows to obtain a good precision when smooth isotropic and anisotropic models are considered. Unfortunately, like the FDM, the main demerit of PM is related to the poor representation of surface waves. The PM has been used to address local [30] as well as global [93] seismic wave propagation problems but, due to the use of global basis functions, this technique is limited to smooth media and it is not appropriate when sharp discontinuities are present in the model. To avoid the numerical noise appearing in this case, it is possible to use a *domain decomposition* approach [169].

Finite Element Method. The Finite Element Method (FEM) for partial differential equations is very popular in science and engineering [92, 176], but has found limited application in seismology in general. The offset of the FEM is the so-called *weak* or *variational formulation* of the problem, that is equivalent to the strong formulation. The computational domain is then decomposed into disjoint subdomains, called the *elements*. Within each element the unknown fields are approximated by polynomials of low order (typically piecewise linear functions) and continuity between the elements is imposed explicitly. The problem then reduces to a space-discrete system for the polynomials coefficients, where the so-called *mass* and *stiffness matrices* are present. Despite its capability to correctly account for irregular geometries and to design complicated meshes capturing realistic 3D models, applications of the pure FEM to seismic wave propagation problems are comparatively rare [122, 164], because the method tends to suffer from numerical grid dispersion and anisotropy, which is highly undesirable for phenomena involving real dispersion and anisotropy. Moreover, the mass matrix in FEM is not diagonal, which makes its inversion computationally expensive. FEM is actively being used in the context of regional simulations [1, 11, 21, 68], but it is not used for simulations at the scale of the globe. In the context of regional simulations, only an area of interest is considered and an artificial boundary is introduced at a given distance from it. If we assign Dirichlet or Neumann boundary conditions on this artificial boundary, we observe a reflection of the outgoing waves back into the computational domain [57]. To avoid this problem it is possible to impose Absorbing Boundary Conditions (ABCs) [46, 145] or Perfectly Matched Layers (PMLs) [12, 49, 65, 104].

Spectral Element Method. Used for two decades in computational fluid dynamics [140], the Spectral Element Method (SEM) is half-way between the FEM and the PM, combining the advantages of both approaches while avoiding many of their drawbacks. Like in FEM, after meshing the model, the geometry of each element is defined using low-degree polynomials. The main differences between FEM and SEM are the polynomial degree of the basis functions used to represent the vector fields and the test functions, and the choice of the integration rule. In FEM it is typical to use low-degree basis functions and Gaussian quadrature. In SEM, one uses higher-degree basis functions and Gauss-Lobatto-Legendre quadrature to obtain a higher resolution. In this way, the resulting mass matrix is diagonal and it is trivially inverted. In this respect, the SEM is related to FEM when mass lumping is used to avoid the costly resolution of the non-diagonal system resulting from the use of Gaussian quadrature [48]. Alternatively Chebyshev polynomials can be used as basis functions [144], establishing a direct link to Chebyshev PMs. SEM has been applied to problems related to 3D local or regional [64, 98, 99, 101, 106, 158] and global [40, 41, 42, 100, 102, 103, 105] seismic wave propagation.

Discontinuous Galerkin Method. In 1971 a new class of FEM was proposed for the numerical solution of the nuclear transport problem [146]. This method, namely Discontinuous Galerkin Method (DGM), allows for solutions that are discontinuous across element boundaries, because neighboring elements are linked by numerical fluxes and not by continuity constraints. DGMs for seismic wave propagation have been developed only recently [60, 59, 58, 63, 96, 129, 130].

On the other hand, they are methods that require only meshing of the boundary of the volume where the waves propagate (and possibly of the interfaces between different media).

Boundary Element Method

The Boundary Element Method (BEM) has emerged as an efficient alternative to all these numerical methods to treat several classes of partial differential equations [44, 54, 90, 171]. The main advantage of the BEM is that only the domain boundaries are discretized, so that it allows to handle external problems with bounded boundaries as easily as internal problems.

The most important feature of BEM is a reformulation of the original differential equations and boundary conditions as *Boundary Integral Equations* (BIEs), describing the problem only by an equation with known and unknown *boundary states.* For problem described by elliptic differential equations, a Fredholm integral equation is obtained. Time dependent problems described by parabolic or hyperbolic equations give rise to Volterra integral equations. In order to obtain such reformulation, the equation should have a *fundamental solution*, also known as *Green's function.* Many physical phenomena are described by a differential equation $\mathcal{L}u(x) = f(x)$, involving a differential operator $\mathcal{L}$

that determines a physical unknown u when there is a source f of that physical quantity. When the source term is a unit magnitude load (Dirac Delta distribution), the solution of this equation is called Green's function of the operator $\mathcal{L}$. In other words, the Green function characterizes the response of a system to the presence of a point source.

Traditionally, there have been two different ways to derive BIEs [27, 28]. The *indirect approach* [88] uses certain fictitious density functions, assumed to be distributed over the boundary and having no specific physical significance, to represent the solution. These density functions are sought first using numerical solution procedures, those in turn, are used to arrive at quantities of physical interest. On the other hand, the *direct approach* uses the actual physical variables present in the given problem to form the BIEs, via reciprocal or Betti theorem [53, 148]. Although formally different, these two approaches have been proved equivalent [26].

Once the reformulation of the problem as a BIE is obtained, the idea of the BEM is to partition the boundary of the domain into a number of sub-boundaries, referred to as boundary elements. Parallel to this, the discretization of the BIEs is based on the *nodal collocation method* in which an approximated BIE is evaluated on the interpolation nodes. Besides collocation method, the *Nyström method* is an alternative technique. The numerical solution of BIE is sought by replacing the integral with a representative weighted sum (quadrature). The last approach implements in BEM discretization procedure using one of the weighted residual techniques called *Galerkin method.* Unlike the collocation and Nyström procedures, the Galerkin approach does not involve specific point satisfying the BIE but requires double integrals: in this sense, it is a *variational formulation.* However, the collocation method can be considered as a particular case of the Galerkin BEM, for which the test functions are Dirac distributions. All the three approaches lead to a *linear system*, whose matrix (non-symmetric in the case of collocation method) is in general *dense.* This is an essential difference from FEM, where the mass and the stiffness matrices are sparse.

Early formulations of the BEM for elastodynamic problems are due to Cruse, Rizzo [55] and Dominguez [61] for frequency-domain, and Mansur [126], Wheeler and Sternberg [172] for time-domain. Recent developments of time-domain BEM are due to Antes, Banjai and Schanz [10, 157] and to Aimi, Diligenti, Frangi and Guardasoni [3]. Instead, in frequency-domain some progress is due to Xiao, Ye, Chai and Zang [173, 174], and to Phan, Guduru, Gray and Salvadori [143].

Some interesting examples of application of BEM in elastodynamics can be found in the work of Sánchez-Sesma and Campillo [153, 154] to compute the 2D seismic response of topographical irregularities on the surface of a homogeneous half-space, in the work of Pederson [141] to compute the 3D response of 2D topographies, in the work of Sánchez-Sesma and Luzón to study 3D alluvial valleys [155] and topographies [121], and in the work of Luco and Barros to obtain the 3D response of an infinitely long canyon [120, 119].

Although the use of the BEM reduces the spatial problem dimension (only the discretization of the domain boundaries is required), it leads to a fully-populated and, using the collocation approach, non-symmetric matrix. When a direct method is used to solve the BEM system, the operation count is proportional to $\mathcal{O}(N^3)$ and the memory requirement is proportional to $\mathcal{O}(N^2)$, with N being the number of degrees of freedom (DOF). When an iterative solver is used, the memory requirement remains the same and the computing time is proportional to $\mathcal{O}(N_{\text{it}}N^2)$, where N_{it} denotes the total number of iterations necessary to reach convergence. Thus, the traditional implementation of this method on ordinary laptops is prohibitive for large problems ($\mathcal{O}(10^6)$ DOF) and, therefore, is restricted to problems of small size, typically $\mathcal{O}(10^4)$ DOF.

Fast BEMs

The history of fast methods to overcome the drawback of the fully-populated BEM matrices started in 1985 when Vladimir Rokhlin introduced the Fast Multipole Method for the 2D Laplace equation in integral form [151]. Then, in 1987 Leslie Greengard applied the algorithm in the context of multi particles simulations [76]. The capabilities of the fast multipole algorithm to accelerate the boundary element method, i.e. Fast Multipole accelerated Boundary Element Method (FM-BEM) and its recursive variant the Multi Level FMBEM (ML-FMBEM) have rapidly progressed during the last decades, allowing many engineering applications requiring large BEM models in various fields (plasma physics, fluid dynamics, molecular dynamics and celestial mechanics), including in 3D elastodynamics [35, 37] and 3D visco-elastodynamics [74]. The FMM requires analytic closed-form expression of the fundamental solution to approximate the integral operators and the solution of the BEM system is done with an iterative solver (i.e. GMRES) of complexity $\mathcal{O}(n_{\text{iter}}\mu(N))$ in terms of CPU time (where $\mu(N)$ represents the cost of the matrix-vector product for N degrees of freedom and n_{iter} is the number of iterations of the iterative solver). In 3D elastodynamics, although μ has been shown to be $\mu(N) = N \log N$, the iteration count becomes the main limitation to use the FM-BEM on large-scale problems [39]. Since, in the context of the FMM, the influence matrix is never explicitly assembled in totality, the definition of an efficient preconditioner is not an easy task. Algebraic preconditioners (SPAI, ILU, diagonal, ...) have been adapted to this constraint but have been shown to have real but limited efficiency. A possible explanation is that they are based on the discretized system.

At the end of the 80's, Hackbusch presented the *Panel Clustering* [85], another fast approach closely related to the single-level FMM. Applied to the potential [4] and linear elastostatic problems [87], it is considered to be the ancestor of the *Hierarchical Matrices* ($\mathcal{H}$-matrices) [81], introduced to compute a data-sparse representation of some special dense matrices (for example matrices resulting from the discretization of nonlocal operators). The $\mathcal{H}$-matrix technique has its largest application field in BEM, because they are a very efficient way to approximate dense matrices. An $\mathcal{H}$-matrix is a data-sparse representation of a matrix, consisting of a collection of exactly computed full-matrices (of small sizes) and approximated low-rank matrices (of various sizes).

In combination with efficient algorithms to compute low-rank approximations, such as Adaptive Cross Approximation (ACA, see [13] for Nyström and [18] for collocation matrices), $\mathcal{H}$-matrices are purely algebraic tools. Thus, this technique is an alternative to tackle problems for which analytical expressions of the Green's functions are not available. Hence, the procedure is problem independent. The extension of the classical matrix arithmetic to $\mathcal{H}$-matrices allows to define fast solver. In particular, the $\mathcal{H}$-matrix/vector product can be used within an iterative solver, in order to accelerate the classical matrix/vector product and, consequently, to reduce the time of each iteration. In computational mechanics, the method has been successfully applied to various problems (in [51] the method has been applied to the layered half-space elastodynamics fundamental solutions to study soil-structure interactions, while in [132] it has been applied to study anisotropic elastodynamics media). Recent works (see [32] for electrodynamic, [80] for electromagnetic and [159] for electrical integral equations) have proposed the development of fast direct solvers based on $\mathcal{H}$-matrices. The advantage of this solution strategy consists in the possibility of treating problems with multiple right hand sides.

Aims and Outline of this Thesis

The main goal of this thesis is to develop alternative fast, stable and accurate frequency-domain collocation based BEMs, to increase the capabilities of the standard method in the context of 3D elastic wave propagation and forced-vibration problems. To this end, $\mathcal{H}$-matrix based iterative and direct solvers for the Boundary Element Method, already developed in other areas such as acoustic and electromagnetism, are extended to 3D elastodynamics. The main difficulty consists in analyzing the capabilities of the method when dealing with oscillatory kernels, since the previous theoretical studies concern the Laplace kernel function. This work is a further step at POEMS (CNRES, INRIA, ENSTA-ParisTech) toward the development of fast solvers for visco-elastic waves in anisotropic media. In the future, other applications of the present work will be performed: soil-structure interaction, inverse problems, ...

This thesis is divided into five chapters.

Chapter 1 introduces the dissertation by presenting the basic principles of elastodynamics and of the boundary element method.

Chapter 2 presents the principles of $\mathcal{H}$-matrices and Adaptive Cross Approximation. The construction of $\mathcal{H}$-matrix based iterative and direct solvers is shown. Particular attention is given to the $\mathcal{H}$-matrix arithmetic, whose implementation requires the study of 27 different cases. The original contributions of this chapter are the investigation of the capability of the ACA to produce low-rank approximations in the case of oscillatory kernels and the extension of the standard ACA to vectorial problems.

Chapter 3 treats the $\mathcal{H}$-matrix based iterative and direct solvers for 3D elastodynamics.

Their computational performances are assessed through numerical examples, concerning the diffraction of an incident, vertical P-wave by a unit sphere. The $\mathcal{H}$-matrix based iterative solver is compared with a Fast Multipole based solver and an error estimate is proposed to certify the quality of the solution of the $\mathcal{H}$-matrix based direct solver, when the analytical solution is unknown.

Chapter 4 focuses on forced-vibration problems. The formulation is extended to multi-domain scenarios by developing a $\mathcal{H}$-matrix based BEM-BEM coupling approach suitable for 3D piecewise-homogeneous media. We treat the problem of the computation of the Green's displacement functions of an homogeneous elastic half-space, i.e. the propagation of a seismic wave in an elastic half-space subject to a point force applied to the surface. Then, we consider a half-space with topographic irregularities.

Chapter 5 gives some directions for future work opened by this thesis: viscoelasticity, anisotropy, parallelization, preconditioning, coupling with other numerical methods, forward solver for inverse problems.

1 Elastodynamic Fundamental Equations and Boundary Integral Formulation

Contents

1.1 Introduction 12

1.2 The Basics of Elastodynamics 12

1.2.1 Linear Elastodynamic System 14

1.2.2 Derivation of the P- and S-waves Velocities 15

1.2.3 Body and Surface Waves 16

1.2.4 Frequency-Domain Elastodynamic Equations 19

1.2.5 Constitutive Law for a Linear Viscoelastic Medium 20

1.3 Boundary Element Methods for 3D Elastodynamics 21

1.3.1 Integral Representation and Boundary Integral Equation . . . 22

1.3.2 Boundary Element Method 24

1.3.3 BEM System 26

1.3.4 Solution Strategies for BEM Systems 27

1.1 Introduction

This Chapter aims at introducing the elastodynamic fundamental equations and its boundary integral reformulation. The Chapter is organized as follows. In Section 1.2 the relationships between stress and strain in terms of displacement and material parameters are given [6]. These in turn represent the deformation of a body. Starting from the Cauchy first law of motion, the Navier-Cauchy equations are derived, both in time-domain and in frequency-domain. The elastodynamics problem for an elastic body is formulated as a boundary value problem and a physical interpretation of the boundary conditions is presented [136]. Subsequently, the theory for P- and S-waves is explained and the Rayleigh wave propagating on the surface of a half-space is considered [135]. Love waves are also introduced [6]. Finally, we present the constitutive law for a linear visco-elastic medium. Section 1.3 is devoted to the reformulation of elastodynamic problem as a Boundary Integral Equation (BIE) and to the Boundary Element Method (BEM) to compute its solution. We choose the direct approach to reformulate the elastodynamic problem as BIE [6] and we consider the BEM discretization procedure followed by Chaillat in [33], previously introduced by Bonnet in [22].

1.2 The Basics of Elastodynamics

We consider an elastodynamic medium, i.e. an infinite, homogeneous, continuous and isotropic elastic solid, whose constitutive properties are defined by two independent elastic moduli, i.e. *shear modulus* μ and *Poisson's ratio* ν, related to the *Lamé parameter* $\lambda := {}^{2\mu\nu}/_{1-2\nu}$. Since the material is homogeneous, μ and ν are independent of the position. The motion of the medium is assumed to satisfy conservation of mass, meaning that the mass of the body is the same at all times. Furthermore, the mass density ϱ is defined at every point in the body as the mass divided by the volume.

Throughout this work, the Einstein summation convention is used, i.e. summation is performed over repeated indices.

Assuming that in a fixed Cartesian coordinates system $\mathbf{x} = (x_1, x_2, x_3)^\top$ the medium occupies a region $\Omega \subset \mathbb{R}^3$, having surface boundary $\Gamma := \partial\Omega$ connected and globally Lipschitz continuous with outward unit normal vector $\mathbf{n} = \mathbf{n}(\mathbf{x})$, the time-domain equation of motion in x_i-direction, i.e. the *Cauchy first law of motion*, may be written

$$\frac{\partial}{\partial x_j}\sigma_{ij}(\mathbf{x},t) + \varrho F_i(\mathbf{x},t) = \varrho \ddot{u}_i(\mathbf{x},t), \qquad i = 1,2,3 \tag{1.1}$$

where $\sigma_{ij}(\mathbf{x},t)$ are the cartesian components of the *Cauchy stress tensor*, $u_i(\mathbf{x},t)$ are the components of the *displacement field* and $F_i(\mathbf{x},t)$ is a given *body-force distribution* in coordinate direction i. Finally, $\ddot{u}_i(\mathbf{x},t)$ is the second order local time derivative of the displacement, i.e. the *local acceleration*.

The stress tensor $\sigma_{ij}(\mathbf{x}, t)$ is a very useful quantity because it allows to compute the force acting on the surface Γ per unit of surface area. This force is called *traction* and it is represented by a vector $\mathbf{t}(\mathbf{x}, t)$, whose components are given by

$$t_i(\mathbf{x}, t) := \sigma_{ij}(\mathbf{x}, t) n_j(\mathbf{x}) \quad i = 1, 2, 3. \tag{1.2}$$

Since the conservation of angular momentum is required [71, 125], the stress tensor results symmetric:

$$\sigma_{ij}(\mathbf{x}, t) = \sigma_{ji}(\mathbf{x}, t). \tag{1.3}$$

For general medium the stress in the medium depends in a complicated way on the deformation of the medium, due to the variation of the displacement with position (a constant displacement vector does not generate an internal deformation). The deformation of the medium is described by the *Green-Lagrange strain tensor* which, for small strains, reduces to the symmetric tensor

$$\varepsilon_{ij}(\mathbf{x}, t) := \frac{1}{2}\left[\frac{\partial u_i}{\partial x_j}(\mathbf{x}, t) + \frac{\partial u_j}{\partial x_i}(\mathbf{x}, t)\right], \qquad i, j = 1, 2, 3. \tag{1.4}$$

The small strain tensor $\varepsilon_{ij}(\mathbf{x}, t)$ is related to the Cauchy stress tensor $\sigma_{ij}(\mathbf{x}, t)$ throughout the *Hooke's law* (constitutive equation)

$$\sigma_{ij}(\mathbf{x}, t) = C_{ijhl}\varepsilon_{hl}(\mathbf{x}, t), \qquad i, j = 1, 2, 3 \tag{1.5}$$

where C_{ijhl} is a fourth order *stiffness tensor*, generalizing the concept of the spring constant of a simple spring to three-dimensional elastic medium.

In Hooke's law (1.5) the strain is assumed to be small enough that stress and strain depend linearly on each other. Such a medium is called *linear*. Of course, problems exist in which the nonlinear behaviour can not be disregarded, i.e. when the excitation leads to large strain (for example in the earthquake source region, where nonlinear fracture behavior governs the response of the medium).

In general a tensor of rank 4 in three dimensions has 81 components but the stiffness tensor has only 21 independent elements. From relation (1.3) it is possible to deduce the symmetry of C_{ijhl} when the first two indices are exchanged, i.e. $C_{ijhl} = C_{jihl}$. Since the strain tensor $\varepsilon_{ij}(\mathbf{x}, t)$ is symmetric, the stiffness tensor is also symmetric in the last two indices, i.e. $C_{ijhl} = C_{ijlh}$. Energy considerations [5] imply that the stiffness tensor is also symmetric for exchange of the first and the last pair of indices, i.e. $C_{ijhl} = C_{hlij}$. The symmetry properties of the stiffness tensor are summarized by

$$C_{ijhl} = C_{jihl} = C_{ijlh} = C_{hlij}. \tag{1.6}$$

Combining Equation (1.5) and Equation (1.4) and using the symmetry of the stiffness tensor in the first pair of indices, the relation between the stress and the displacement can be written as

$$\sigma_{ij}(\mathbf{x},t) = C_{ijhl}\frac{\partial u_l}{\partial x_h}(\mathbf{x},t), \qquad i,j = 1,2,3. \tag{1.7}$$

In the case of an isotropic medium, the stiffness tensor does not depend on some preferred direction, but only on μ and λ and the *Kronecker delta* δ_{ij}:

$$C_{ijhl} := \lambda\delta_{ij}\delta_{hl} + \mu(\delta_{ih}\delta_{jl} + \delta_{il}\delta_{jh}), \qquad i,j,h,l = 1,2,3. \tag{1.8}$$

1.2.1 Linear Elastodynamic System

Inserting Equation (1.7) and Equation (1.8) in Equation (1.1), we have the *Navier-Cauchy equations* of motion [2], i.e.

$$(\lambda+\mu)\frac{\partial^2 u_j}{\partial x_i \partial x_j}(\mathbf{x},t) + \mu\frac{\partial^2 u_i}{\partial x_j^2}(\mathbf{x},t) + \varrho F_i(\mathbf{x},t) = \varrho\ddot{u}_i(\mathbf{x},t), \qquad i = 1,2,3. \tag{1.9}$$

Introducing the vector displacement $\mathbf{u}(\mathbf{x},t) = (u_1,u_2,u_3)^\top(\mathbf{x},t)$ and the body force vector per unit volume $\mathbf{F}(\mathbf{x},t) = (F_1,F_2,F_3)^\top(\mathbf{x},t)$, Equation (1.9) can be written as

$$(\lambda+\mu)\boldsymbol{\nabla}\boldsymbol{\nabla}\cdot\mathbf{u}(\mathbf{x},t) + \mu\boldsymbol{\Delta}\mathbf{u}(\mathbf{x},t) + \varrho\mathbf{F}(\mathbf{x},t) = \varrho\ddot{\mathbf{u}}(\mathbf{x},t), \tag{1.10}$$

where $\boldsymbol{\nabla}$ and $\boldsymbol{\Delta}$ denote the Nabla and the Laplace operators, respectively.

Using the identity $\boldsymbol{\Delta}\mathbf{u}(\mathbf{x},t) = \boldsymbol{\nabla}\boldsymbol{\nabla}\cdot\mathbf{u}(\mathbf{x},t) - \boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\mathbf{u})(\mathbf{x},t)$, in which $\times$ denotes the vector cross-product, an alternative form of Equation (1.10) is

$$(\lambda+2\mu)\boldsymbol{\nabla}\boldsymbol{\nabla}\cdot\mathbf{u}(\mathbf{x},t) - \mu\boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\mathbf{u})(\mathbf{x},t) + \varrho\mathbf{F}(\mathbf{x},t) = \varrho\ddot{\mathbf{u}}(\mathbf{x},t). \tag{1.11}$$

This expression represents a linear system of hyperbolic partial differential equations for the dependent variables u_1, u_2 and u_3. For a well-posed problem, conditions have to be prescribed at the initial time $t = 0$ (*initial conditions*) and at the domain boundary (*boundary conditions*).

The initial conditions give the solution at the instant from which the physical system evolves:

$$\begin{aligned}\mathbf{u}(\mathbf{x},0) &= \mathbf{u}_0(\mathbf{x})\\ \dot{\mathbf{u}}(\mathbf{x},0) &= \mathbf{v}_0(\mathbf{x}),\end{aligned}$$

with initial rest ($\mathbf{u}_0(\mathbf{x}) = \mathbf{v}_0(\mathbf{x}) = \mathbf{0}$) frequently assumed in practice.

Boundary conditions can be of different types, depending if they impose the solution (displacements $\mathbf{u}(\mathbf{x},t)$) or its derivative (traction $\mathbf{t}(\mathbf{x},t)$). Among the others, we cite the *Dirichlet* boundary conditions ($\mathbf{u}(\mathbf{x},t)$), *Neumann* boundary conditions ($\mathbf{t}(\mathbf{x},t)$), *Robin* boundary conditions (linear combination of $\mathbf{u}(\mathbf{x},t)$ and $\mathbf{t}(\mathbf{x},t)$), mixed boundary conditions (different boundary conditions are used on different parts of the boundary).

1.2.2 Derivation of the P- and S-waves Velocities

The concept of waves is fundamental to elastodynamics, as it provides a convenient way to describe the mathematics and the physics of the transmission of energy. In physics, waves are usually divided into *progressive* and *standing* waves. Seismic waves are also of these two types but we focus on the progressive seismic waves that propagate away from seismic sources and ignore the standing seismic waves, that represent vibrations of the Earth as a whole.

Expressing the displacement field and the body force field as a combination of two simpler fields facilitates the analysis of solutions of the elastodynamic equations. With the aid of *Helmholtz's decomposition theorem* the vectors $\mathbf{u}(\mathbf{x},t)$ and $\mathbf{F}(\mathbf{x},t)$ can be expressed as

$$\begin{aligned} \mathbf{u}(\mathbf{x},t) &:= \boldsymbol{\nabla}\varphi(\mathbf{x},t) + \boldsymbol{\nabla}\times\boldsymbol{\psi}(\mathbf{x},t) && (1.12)\\ \mathbf{F}(\mathbf{x},t) &:= \boldsymbol{\nabla}\Phi(\mathbf{x},t) + \boldsymbol{\nabla}\times\boldsymbol{\Psi}(\mathbf{x},t), && (1.13) \end{aligned}$$

where $\varphi(\mathbf{x},t)$ and $\Phi(\mathbf{x},t)$ are called *scalar potentials*, while $\boldsymbol{\psi}(\mathbf{x},t)$ and $\boldsymbol{\Psi}(\mathbf{x},t)$ are called *vectorial potentials*. Both the scalar and the vectorial potentials are everywhere continuous and differentiable at interior points where the respective vector field is also continuous. The conditions $\boldsymbol{\nabla}\cdot\boldsymbol{\psi}(\mathbf{x},t) = \mathbf{0}$ and $\boldsymbol{\nabla}\cdot\boldsymbol{\Psi}(\mathbf{x},t) = \mathbf{0}$ may be assumed without loss of generality. These are also known as the *gauge conditions*.

The Helmholtz's decomposition theorem stated above is valid even for infinite, or open, domains, provided that the vector fields $\mathbf{u}(\mathbf{x},t)$ and $\mathbf{F}(\mathbf{x},t)$ vanish with a rate of at least $1/\|\mathbf{x}\|$ for $\|\mathbf{x}\| \to \infty$. This constitutes a *radiation condition* at infinity.

Taking the divergence of Equation (1.11) and remembering that the divergence of the curl of a vector field is always equal to zero, we obtain

$$(\lambda+2\mu)\boldsymbol{\Delta}(\boldsymbol{\nabla}\cdot\mathbf{u})(\mathbf{x},t) + \varrho\boldsymbol{\nabla}\cdot\mathbf{F}(\mathbf{x},t) = \varrho\frac{\partial^2}{\partial t^2}(\boldsymbol{\nabla}\cdot\mathbf{u})(\mathbf{x},t). \tag{1.14}$$

Since from Helmholtz's decomposition it holds

$$\begin{aligned} \boldsymbol{\nabla}\cdot\mathbf{u}(\mathbf{x},t) &:= \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\varphi)(\mathbf{x},t) = \boldsymbol{\Delta}\varphi(\mathbf{x},t) && (1.15)\\ \boldsymbol{\nabla}\cdot\mathbf{F}(\mathbf{x},t) &:= \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}\Phi)(\mathbf{x},t) = \boldsymbol{\Delta}\Phi(\mathbf{x},t), && (1.16) \end{aligned}$$

Equation (1.14) can be recast into the equivalent form

$$\boldsymbol{\Delta}\left((\lambda+2\mu)\boldsymbol{\Delta}\varphi + \varrho\Phi - \varrho\ddot{\varphi}\right)(\mathbf{x},t) = 0. \tag{1.17}$$

Instead, taking the curl of Equation (1.11) and remembering that the curl of the gradient of a scalar field is always equal to zero, we have

$$-\mu\boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\mathbf{u}))(\mathbf{x},t)+\varrho\boldsymbol{\nabla}\times\mathbf{F}(\mathbf{x},t)=\varrho\frac{\partial^2}{\partial t^2}(\boldsymbol{\nabla}\times\mathbf{u})(\mathbf{x},t). \tag{1.18}$$

From the Helmholtz's decomposition and the gauge conditions, it follows

$$\begin{aligned}
\boldsymbol{\nabla}\times\mathbf{u}(\mathbf{x},t) &:= \boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\boldsymbol{\psi})(\mathbf{x},t)=-\boldsymbol{\Delta}\boldsymbol{\psi}(\mathbf{x},t) && (1.19)\\
\boldsymbol{\nabla}\times\mathbf{F}(\mathbf{x},t) &:= \boldsymbol{\nabla}\times(\boldsymbol{\nabla}\times\boldsymbol{\Psi})(\mathbf{x},t)=-\boldsymbol{\Delta}\boldsymbol{\Psi}(\mathbf{x},t) && (1.20)
\end{aligned}$$

and Equation (1.18) can be rewritten as

$$\boldsymbol{\Delta}\left(\mu\boldsymbol{\Delta}\boldsymbol{\psi}+\varrho\boldsymbol{\Psi}-\varrho\ddot{\boldsymbol{\psi}}\right)(\mathbf{x},t)=\mathbf{0}. \tag{1.21}$$

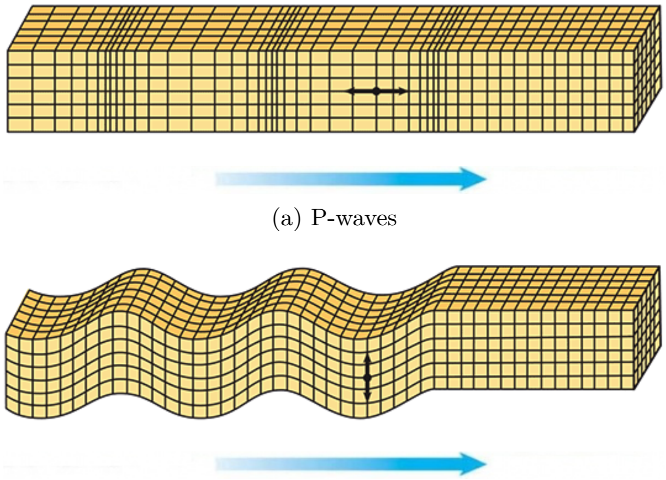
Important particular solutions of Equation (1.17) and Equation (1.21) can be obtained from

$$\begin{aligned}
\boldsymbol{\Delta}\varphi(\mathbf{x},t) &= \frac{1}{c_P^2}\left[\ddot{\varphi}(\mathbf{x},t)-\Phi(\mathbf{x},t)\right] \qquad \text{with} \quad c_P:=\sqrt{\frac{\lambda+2\mu}{\varrho}}\\
\boldsymbol{\Delta}\boldsymbol{\psi}(\mathbf{x},t) &= \frac{1}{c_s^2}\left[\ddot{\boldsymbol{\psi}}(\mathbf{x},t)-\Psi(\mathbf{x},t)\right] \qquad \text{with} \quad c_S:=\sqrt{\frac{\mu}{\varrho}}
\end{aligned} \tag{1.22}$$

i.e φ satisfies the scalar wave equation in three dimensions with phase speed of wave propagation given by c_P, while $\boldsymbol{\psi}$ satisfies the vectorial wave equation in three dimensions with phase speed of wave propagation given by c_S. In this way, the nature of two different wave types has been revealed.

1.2.3 Body and Surface Waves

It is now evident that in a homogeneous medium two *body waves* types propagate independently from each other: the *compressional waves* with velocity c_P and the *shear waves* with velocity c_S.



(a) P-waves

(b) S-waves

Figure 1.1: Body waves: (a) Primary or P-waves and (b) Secondary or S-waves (Image Credits: http://www.kgs.ku.edu).

Recalling that for most real materials the shear modulus and the Poisson's ratio are both real positive quantities (in general $-1 < \nu < {}^1\!/_2$ but only auxetics have negative Poisson's ratio), if we compare c_P and c_S we observe that $c_P > c_S$. The compressional waves reach an observation point before the shear waves. For this reason the seismological names for compressional and shear waves are *P-waves* (or *primary waves*, Figure 1.1a) and *S-waves* (or *secondary waves*, Figure 1.1b), respectively. Since in many elastic materials Poisson's relation holds ($\lambda = \mu$), the relation $c_P = c_S\sqrt{3}$ is frequently used [136]. In liquid or gases, only P-waves (sound waves) can propagate since $\mu = 0$.

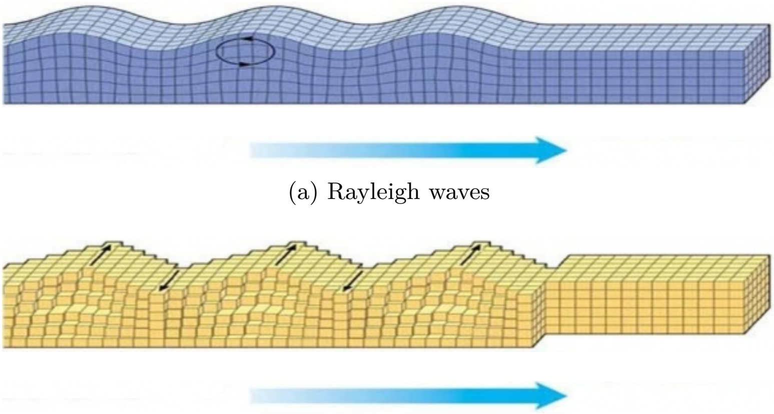
Looking at Equation (1.12), we note that P-waves are free of rotation, while S-waves are free of divergence. This means that P-waves consist of changes in the volume (dilatations) of the medium as they pass through it. Each particle vibrates around its equilibrium position, parallel to the wave propagation direction (primarily longitudinally polarized). For this reason P-waves are often called *longitudinal waves.* Instead, the S-waves consist of shear deformations and rigid rotations. Since the particle motion is perpendicular to the wave propagation direction (primarily transversely polarised), S-waves are also called *transverse waves* and their contribution to the total field is usually further decomposed into a component in a horizontal plane (*SH-waves*) and a component in a vertical plane (*SV-waves*).

The body waves are the only waves that can exist in an infinite elastodynamic medium [111]. However, the Earth can not be considered as a homogeneous medium but can be approximated by piecewise homogeneous media, whose thicknesses are irregular due to several abrupt and gradual variations. This model represents a first approximation of a realistic configuration, due to the presence of heterogeneities inside each homogeneous layer, that have a significant effect on the wave propagation.

Since the materials on the two side of these *surfaces of discontinuity* are supposed homogeneous and isotropic, boundary conditions have to be imposed on the interfaces between two domains with different properties [136]. On solid-solid boundaries (such as Moho discontinuity), both the traction $\mathbf{t}(\mathbf{x}, t)$ and the displacement $\mathbf{u}(\mathbf{x}, t)$ need to be continuous, whereas on fluid-solid boundaries (such as the Gutenberg and Lehmann discontinuities), both the traction $\mathbf{t}(\mathbf{x}, t)$ and the normal component of the displacement $\mathbf{u}(\mathbf{x}, t)$ need to be continuous.

Body waves incident on one side of a discontinuity give rise to both a *refracted* wave on the other side and a *reflected* wave on the same side. Even if the incident wave is purely longitudinal or purely transverse, the refracted and reflected waves are in general superpositions of P- and S-waves, propagating in different directions. If the angle of incidence becomes large enough, at the material interface there are special types of waves which do not penetrate into the bulk of the materials, but decay exponentially with the distance [136]. These waves are known as *surface waves.*

The simplest way to study the surface waves consist in considering the ground as an homogeneous and isotropic elastic medium, in which only the boundary between air and soil exists and all the others are assumed infinitely far away. This model is called a *half-space* [6]. On the so-called *free-surface*, the traction should vanish, so a *traction-free* Neumann boundary condition has to be imposed, i.e. $\mathbf{t}(\mathbf{x}, t) = \mathbf{0}$.



(a) Rayleigh waves

(b) Love waves

Figure 1.2: Surface waves: (a) Rayleigh waves and (b) Love waves (Image Credits: http://www.kgs.ku.edu).

If Equations (1.10) are associated with this boundary condition, other solutions to the Navier-Cauchy equations of motion rise. These solutions are surface waves, known as *Rayleigh waves* (Figure 1.2a), whose velocity c_R is given by the following equation [135]:

$$\left(2 - \frac{c_R^2}{c_S^2}\right)^2 = 4\left(1 - \frac{c_R^2}{c_P^2}\right)^{1/2}\left(1 - \frac{c_R^2}{c_S^2}\right)^{1/2}. \tag{1.23}$$

Rayleigh waves are slower than the S-waves, typically with a phase speed of around $c_R = 0.9c_S$ [136]. For this reason they are recorded even later than shear waves originated in the same point. The particles vibrate in the plane perpendicular to the free surface as they pass through it. The resulting motion can be regarded as a combination of the P- and the SV-vibrations and describes an ellipse with its major axis vertical and minor axis in direction of wave propagation.

Since surface waves polarized in the horizontal plane perpendicularly to the direction of propagation, i.e. surface waves of the SH type, are present on seismograms, the half-space is insufficient to model the ground structure. To describe these waves, called *Love waves* (Figure 1.2b), we have to consider a medium that consists of a layer of a constant thickness, lying on a half-space. Both the layer and the half-space are assumed homogeneous, isotropic and perfectly elastic. Love waves move faster than the other surface waves in the surface layer, but slower than shear waves in the bulk. Thus, they arrive before Rayleigh waves originating at the same point. In earthquakes Love waves are the most destructive because of the shearing motion of the surface layer, not very well tolerated by buildings [6].

1.2.4 Frequency-Domain Elastodynamic Equations

To treat elastodynamic problems it is possible to follow two different approaches, namely, *time-domain* and *frequency-domain* approaches. In the first the physical problem is directly solved in the real time domain, thus one can observe the phenomenon as it evolves. Instead, in the frequency-domain approach the problem is solved at a series of sampling frequencies and the results are then transformed back into time-domain.

The frequency-domain formulation and the time-domain formulation are related by the *Fourier transform*. The Fourier transform of an integrable function $g : \mathbb{R} \to \mathbb{C}$ is defined by:

$$\tilde{g}(\omega) = \mathcal{F}\Big(g(t)\Big) := \int_{-\infty}^{+\infty} g(t)e^{-\imath\omega t}dt. \tag{1.24}$$

Under suitable conditions, g is determined by $\tilde{g}$ via the *inverse transform*:

$$g(t) = \mathcal{F}^{-1}\Big(\tilde{g}(\omega)\Big) := \frac{1}{2\pi}\int_{-\infty}^{+\infty} \tilde{g}(\omega)e^{\imath\omega t}d\omega. \tag{1.25}$$

Since we are interested in harmonic variations of the body forces and the boundary conditions in time, we consider the analysis of elastodynamic problems with respect to frequency rather than time. Thus, the displacement field is in the form

$$\mathbf{u}(\mathbf{x}, t) = \hat{\mathbf{u}}(\mathbf{x})e^{\imath\omega t},$$

where $\hat{\mathbf{u}}(\mathbf{x})$ is an *amplitude function*, $\omega = {}^{2\pi}/_{T}$ is the *circular frequency* and T denotes the *period*. From now on, $\mathbf{u}(\mathbf{x}) = \hat{\mathbf{u}}(\mathbf{x})$ with abuse of language.

Taking into account these considerations, the Cauchy first law of motion can be rewritten for time-harmonic problems in the following vector form:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{x}) + \varrho\mathbf{F}(\mathbf{x}) + \varrho\omega^2\mathbf{u}(\mathbf{x}) = \mathbf{0}, \tag{1.26}$$

while the displacement governing equation for linear elasticity (Equation (1.10)) is reformulated as an elliptic partial differential equation:

$$(\lambda + \mu)\boldsymbol{\nabla\nabla} \cdot \mathbf{u}(\mathbf{x}) + \mu\boldsymbol{\Delta}\mathbf{u}(\mathbf{x}) + \varrho\omega^2\mathbf{u}(\mathbf{x}) + \varrho\mathbf{F}(\mathbf{x}) = \mathbf{0}. \tag{1.27}$$

In frequency-domain the potential φ and $\boldsymbol{\psi}$ satisfy the scalar and vector *Helmholtz's equation* respectively, i.e.

$$\begin{aligned} \boldsymbol{\Delta}\varphi(\mathbf{x}) + \kappa_P^2\varphi(\mathbf{x}) &= -\frac{1}{c_P^2}\Phi(\mathbf{x}) \qquad \text{with} \quad \kappa_P := \frac{\omega}{c_P} \\ \boldsymbol{\Delta}\boldsymbol{\psi}(\mathbf{x}) + \kappa_S^2\boldsymbol{\psi}(\mathbf{x}) &= -\frac{1}{c_s^2}\boldsymbol{\Psi}(\mathbf{x}) \qquad \text{with} \quad \kappa_S := \frac{\omega}{c_S}. \end{aligned} \tag{1.28}$$

The wavelength of a α-wave is $\lambda_\alpha = c_\alpha T$, $\alpha = P, S$. However, in the case of harmonic waves, the spatial variation may advantageously be described in terms of wavenumbers κ_α, related to λ_α by the relation $\kappa_\alpha = {}^{2\pi}/_{\lambda_\alpha}$.

1.2.5 Constitutive Law for a Linear Viscoelastic Medium

In real materials, *anelastic* effects such as friction between particles or grains, molecular collisions or irreversible intercrystal heat flux, lead to the *material damping*, a decay or attenuation of seismic waves, due to a dissipation of energy (conversion of mechanical energy into thermic energy). To take into account this phenomenon in a linear model, there is a need of a generalization of the constitutive law which retains linearity and supposes that stress depends not only on the strain at a given instant, but also on the strain history. By the so-called *hereditary integral*, the *viscoelastic law* can be written [45] as:

$$\sigma_{ij}(\mathbf{x}, t) = \int_{-\infty}^{t} C_{ijhl}(t - \tau)\dot{\varepsilon}_{hl}(\mathbf{x}, t)d\tau, \tag{1.29}$$

where $\sigma_{ij}(\mathbf{x}, t)$ is the incremental stress. The integral present in this relation represents a time convolution of the relaxation tensor $C_{ijhl}(t)$ and the strain rate $\varepsilon_{hl}(\mathbf{x}, t)$.

Since stress can depend only on past times, $C_{ijhl}(t)$ must vanish for negative values of t, i.e. it must be a casual function of time.

For a homogeneous isotropic viscoelastic medium, the relaxation tensor is written in terms of the Heaviside step function $H(t)$ and two time-dependent Lamé-type coefficients:

$$C_{ijhl}(t) := \left[\delta_{ij}\delta_{hl}\lambda(t) + (\delta_{ih}\delta_{jl} + \delta_{il}\delta_{jh})\mu(t)\right] H(t), \qquad i,j,h,l = 1,2,3. \tag{1.30}$$

In frequency domain, using the convolution theorem, Equation (1.29) is equivalent to Equation (1.5) with the only difference that λ and μ are complex-valued constants or frequency-dependent parameters [62], given by

$$\mu = \mathrm{Re}(\mu)(1 + \imath 2\beta_\mu) \qquad \text{and} \qquad \lambda = \mathrm{Re}(\lambda)(1 + \imath 2\beta_\lambda), \tag{1.31}$$

where β_μ and β_λ are the *material damping ratios* which allow to distinguish the contribution of the damping of S- and P-waves.

1.3 Boundary Element Methods for 3D Elastodynamics

In this work we want to solve elastodynamic problems using the Boundary Element Method (BEM), that requires a reformulation of the problem as a Boundary Integral Equation (BIE). The first step to obtain such a reformulation consists in establishing a mutual relationship between two elastodynamic states $(u_i^{(1)}, F_i^{(1)}, \sigma_{ij}^{(1)})$ and $(u_i^{(2)}, F_i^{(2)}, \sigma_{ij}^{(2)})$, both satisfying Equations (1.26), i.e.

$$\frac{\partial}{\partial x_j}\sigma_{ij}^{(1)}(\mathbf{x}) + \varrho F_i^{(1)}(\mathbf{x}) + \varrho\omega^2 u_i^{(1)}(\mathbf{x}) = 0 \quad i = 1,2,3 \tag{1.32}$$

$$\frac{\partial}{\partial x_j}\sigma_{ij}^{(2)}(\mathbf{x}) + \varrho F_i^{(2)}(\mathbf{x}) + \varrho\omega^2 u_i^{(2)}(\mathbf{x}) = 0 \quad i = 1,2,3. \tag{1.33}$$

Applying the two states as weight functions in the governing equation for the other state, we obtain:

$$u_i^{(2)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(1)}(\mathbf{x}) + \varrho u_i^{(2)}(\mathbf{x})F_i^{(1)}(\mathbf{x}) + \varrho\omega^2 u_i^{(2)}(\mathbf{x})u_i^{(1)}(\mathbf{x}) = 0, \quad i = 1,2,3 \tag{1.34}$$

$$u_i^{(1)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(2)}(\mathbf{x}) + \varrho u_i^{(1)}(\mathbf{x})F_i^{(2)}(\mathbf{x}) + \varrho\omega^2 u_i^{(1)}(\mathbf{x})u_i^{(2)}(\mathbf{x}) = 0, \quad i = 1,2,3. \tag{1.35}$$

Subtracting the first equation to the second and integrating over the domain Ω, we have

$$\int_\Omega \left[u_i^{(1)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(2)}(\mathbf{x}) - u_i^{(2)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(1)}(\mathbf{x})\right]d\mathbf{x} = \varrho\int_\Omega \left[u_i^{(2)}(\mathbf{x})F_i^{(1)}(\mathbf{x}) - u_i^{(1)}(\mathbf{x})F_i^{(2)}(\mathbf{x})\right]d\mathbf{x}. \tag{1.36}$$

Application of the *divergence theorem* to the first term of the last equation provides:

$$\begin{aligned}\int_\Omega u_i^{(1)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(2)}(\mathbf{x})d\mathbf{x} &= \int_\Gamma t_i^{(2)}(\mathbf{x})u_i^{(1)}(\mathbf{x})d\Gamma_\mathbf{x} - \int_\Omega \sigma_{ij}^{(2)}(\mathbf{x})\frac{\partial u_i^{(1)}}{\partial x_j}(\mathbf{x})d\mathbf{x}\\ \int_\Omega u_i^{(2)}(\mathbf{x})\frac{\partial}{\partial x_j}\sigma_{ij}^{(1)}(\mathbf{x})d\mathbf{x} &= \int_\Gamma t_i^{(1)}(\mathbf{x})u_i^{(2)}(\mathbf{x})d\Gamma_\mathbf{x} - \int_\Omega \sigma_{ij}^{(1)}(\mathbf{x})\frac{\partial u_i^{(2)}}{\partial x_j}(\mathbf{x})d\mathbf{x}.\end{aligned} \tag{1.37}$$

Since the symmetry properties of the elastic constitutive equation allow us to conclude

$$\sigma_{ij}^{(1)}(\mathbf{x})\frac{\partial u_i^{(2)}}{\partial x_j}(\mathbf{x}) = \sigma_{ij}^{(2)}(\mathbf{x})\frac{\partial u_i^{(1)}}{\partial x_j}(\mathbf{x}), \tag{1.38}$$

we can formulate the *reciprocity theorem*:

$$\int_\Gamma \left[t_i^{(2)}(\mathbf{x})u_i^{(1)}(\mathbf{x}) - t_i^{(1)}(\mathbf{x})u_i^{(2)}(\mathbf{x})\right]d\Gamma_{\mathbf{x}} = \varrho\int_\Omega \left[u_i^{(2)}(\mathbf{x})F_i^{(1)}(\mathbf{x}) - u_i^{(1)}(\mathbf{x})F_i^{(2)}(\mathbf{x})\right]d\mathbf{x}. \tag{1.39}$$

Obviously, in case of unbounded domains, this relation holds provided that the two states satisfy the radiation condition at infinity.

1.3.1 Integral Representation and Boundary Integral Equation

If one of the two elastodynamic states involved in Equations (1.39) is selected as the *fundamental solution*, or *Green's function*, i.e. the response at an *observation point* $\mathbf{x}$ due to a unit magnitude load acting at a *source point* $\mathbf{y}$, we can formulate the boundary integral representation of the displacement field.

In the case of general three-dimensional elastic wave propagation we are dealing with displacement and surface traction vector fields. Hence, whereas the displacement field $\mathbf{u}(\mathbf{x})$ is a vector field with components $u_i(\mathbf{x})$, the corresponding Green's function is a tensor field $\mathbf{U}(\mathbf{x}, \mathbf{y}; \omega)$ with the double-indexed components $U_i^k(\mathbf{x}, \mathbf{y}; \omega)$. Similarly, since the Cauchy stress tensor $\boldsymbol{\sigma}(\mathbf{x})$ is a second order tensor field with components $\sigma_{ij}(\mathbf{x})$, the corresponding Green's function is a tensor field $\boldsymbol{\Sigma}(\mathbf{x}, \mathbf{y}; \omega)$ with the triple-indexed components $\Sigma_{ij}^k(\mathbf{x}, \mathbf{y}; \omega)$.

Because the fundamental solution is strongly dependent on the boundary conditions, in this work we consider the time-harmonic elastodynamics Green's functions for a homogeneous full-space, i.e. unbounded domain. These fundamental solutions are available in closed form and can be written as [22]

$$\begin{aligned} U_i^k(\mathbf{x}, \mathbf{y}; \omega) &:= \frac{1}{4\pi\mu r}\left[\delta_{ik}A_1 + r_{,i}r_{,k}A_2\right] \\ \Sigma_{ij}^k(\mathbf{x}, \mathbf{y}; \omega) &:= \frac{1}{4\pi r^2}\left[2r_{,i}r_{,k}r_{,j}A_3 + (\delta_{ik}r_{,j} + \delta_{jk}r_{,i})A_4 + \delta_{ij}r_{,k}A_5\right], \end{aligned} \tag{1.40}$$

where δ_{ij} stands for the Kronecker' symbol, $\mathbf{r} := \mathbf{x} - \mathbf{y}$ and $r_{,i}$ is the derivative of $r = \|\mathbf{x} - \mathbf{y}\|$ with respect to x_i.

Using the notations $x_P := \kappa_P r$, $x_S := \kappa_S r$ and $\gamma = {c_S}/{c_P}$, the coefficients in (1.40) A_1, A_2, A_3, A_4, A_5 are defined by

$$\begin{aligned}
A_1 &:= \left(1+\frac{\imath}{x_S}-\frac{1}{x_S^2}\right)e^{\imath x_S}-\gamma^2\left(\frac{\imath}{x_P}-\frac{1}{x_P^2}\right)e^{\imath x_P},\\
A_2 &:= \left(\frac{3}{x_S^2}-\frac{3\imath}{x_S}-1\right)e^{\imath x_S}-\gamma^2\left(\frac{3}{x_P^2}-\frac{3\imath}{x_P}-1\right)e^{\imath x_P},\\
A_3 &:= \left(\frac{\imath 15}{x_S}-\frac{15}{x_S^2}+6-\imath x_S\right)e^{\imath x_S}-\gamma^2\left(\frac{\imath 15}{x_P}-\frac{15}{x_P^2}+6-\imath x_P\right)e^{\imath x_P},\\
A_4 &:= (\imath x_S-1)\,e^{\imath x_S}+2A_2,\\
A_5 &:= \left(1-2\gamma^2\right)(\imath x_P-1)e^{\imath x_P}+2A_2.
\end{aligned} \tag{1.41}$$

In 2001, Yoshida [175] proposed a reformulation of the fundamental solutions (1.40) in terms of derivatives of the Green's function for the *Helmholtz equation* with wavenumber κ_α, i.e.

$$G(\mathbf{x},\mathbf{y};\kappa_\alpha) := \frac{e^{\imath\kappa_\alpha r}}{4\pi r} \qquad \alpha = P, S.$$

These reformulations read

$$\begin{aligned}
U_i^k(\mathbf{x},\mathbf{y};\omega) &:= \frac{1}{\kappa_S^2\mu}\left[(\delta_{qs}\delta_{ik}-\delta_{qk}\delta_{is})\frac{\partial}{\partial x_q}\frac{\partial}{\partial y_s}G(\mathbf{x},\mathbf{y};\kappa_S)+\frac{\partial}{\partial x_i}\frac{\partial}{\partial y_k}G(\mathbf{x},\mathbf{y};\kappa_P)\right],\\
\Sigma_{ij}^k(\mathbf{x},\mathbf{y};\omega) &:= C_{ijhl}\frac{\partial}{\partial y_l}U_h^k(\mathbf{x},\mathbf{y};\omega).
\end{aligned} \tag{1.42}$$

Defining the first state in Equations (1.39) as the state of physical displacements u_i, specific body forces F_i and surface traction t_i, and choosing the second state as the fundamental solution state, for all $\mathbf{x} \in \Omega$ we have [22]

$$\gamma(\mathbf{x})u_k(\mathbf{x}) = \int_\Gamma \left[U_i^k(\mathbf{x},\mathbf{y};\omega)t_i(\mathbf{y})-T_i^k(\mathbf{x},\mathbf{y};\omega)u_i(\mathbf{y})\right]d\Gamma_{\mathbf{y}}+\varrho\int_\Omega U_i^k(\mathbf{x},\mathbf{y};\omega)F_i(\mathbf{y})d\mathbf{x}, \tag{1.43}$$

where the components of the tensor field $\mathbf{T}(\mathbf{x},\mathbf{y};\omega)$ are defined as

$$T_i^k(\mathbf{x},\mathbf{y};\omega) := \Sigma_{ij}^k(\mathbf{x},\mathbf{y};\omega)n_j(\mathbf{x}).$$

In Equations (1.43), known as *Somigliana identity*, $\gamma(\mathbf{x})$ is a scalar constant only depending on the local geometry at the observation point $\mathbf{x}$, and it is given by the size, the shape and the spatial orientation of the interior solid angle at $\mathbf{x}$. Hence, $\gamma(\mathbf{x}) = 1$ for any interior point within the domain Ω and $\gamma(\mathbf{x}) = 0$ at an exterior point.

Since a singularity occurs in $\mathbf{y} = \mathbf{x}$ when $\mathbf{x} \in \Gamma$, allowing $\mathbf{x} \in \Omega$ to reach the boundary with the help of a limiting process [77], the frequency-domain BIE for the general three-dimensional elastodynamics problem in absence of body forces can be written in the form

$$\gamma_{ki}(\mathbf{x})u_i(\mathbf{x}) = \int_\Gamma U_i^k(\mathbf{x},\mathbf{y};\omega)t_i(\mathbf{y})d\Gamma_\mathbf{y} - (\mathcal{C.P.V}) \int_\Gamma T_i^k(\mathbf{x},\mathbf{y};\omega)u_i(\mathbf{y})d\Gamma_\mathbf{y} \quad \mathbf{x} \in \Gamma, \tag{1.44}$$

where the free term $\gamma_{ki}(\mathbf{x})$ depends on the Poisson's ratio ν [77] and is equal to $\gamma_{ki}(\mathbf{x}) := {}^1\!/\!{}_2\delta_{ki}$ in the usual case where Γ is smooth at $\mathbf{x}$, and is otherwise a known (second-order tensor-valued) function of the local geometry of Γ at $\mathbf{x}$.

Equations (1.44) involves the evaluation of the Cauchy principal value ($\mathcal{C.P.V.}$) integrals of the strongly singular Green's tractions. In the following, a regularized boundary integral equation is employed, in which the evaluation of the ($\mathcal{C.P.V.}$) integrals is avoided [29, 56, 113, 139]. Since the singularity of the static $T_i^k(\mathbf{x},\mathbf{y})$ and dynamic $T_i^k(\mathbf{x},\mathbf{y};\omega)$ fundamental tractions are known to be identical, we consider

$$\int_\Gamma T_i^k(\mathbf{x},\mathbf{y};\omega)u_i(\mathbf{y})d\Gamma_\mathbf{y} = \int_\Gamma \left[T_i^k(\mathbf{x},\mathbf{y};\omega) - T_i^k(\mathbf{x},\mathbf{y})\right] u_i(\mathbf{y})d\Gamma_\mathbf{y} + \int_\Gamma T_i^k(\mathbf{x},\mathbf{y})u_i(\mathbf{y})d\Gamma_\mathbf{y} \tag{1.45}$$

The first integral in the right side of the previous equation is non-singular and its numerical integration is performed using classical Gaussian integration [78]. The remaining integral involving the static fundamental solution is singular, but can be evaluated analytically for some special configurations (for example triangular elements, [33]).

From now on, we refer to Equation (1.44) as

$$\gamma_{ki}(\mathbf{x})u_i(\mathbf{x}) = \int_\Gamma U_i^k(\mathbf{x},\mathbf{y};\omega)t_i(\mathbf{y})d\Gamma_\mathbf{y} - \int_\Gamma T_i^k(\mathbf{x},\mathbf{y};\omega)u_i(\mathbf{y})d\Gamma_\mathbf{y} \quad \mathbf{x} \in \Gamma, \tag{1.46}$$

where the described regularization process is applied.

1.3.2 Boundary Element Method

In order to use the collocation method, we take N_C *collocation nodes* $\mathbf{x}^c \in \Gamma$ and enforce Equation (1.46) on these points, obtaining:

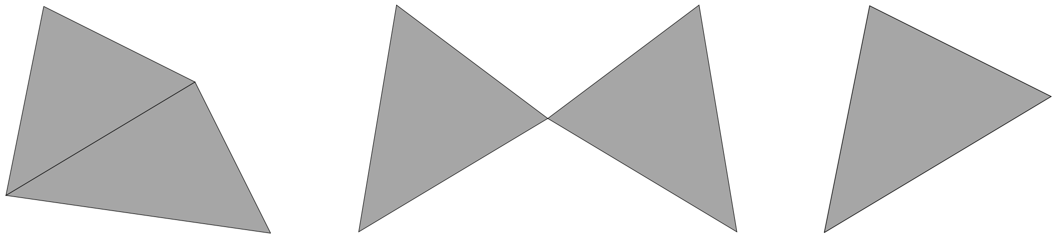
$$\gamma_{ki}(\mathbf{x}^c)u_i(\mathbf{x}^c) = \int_\Gamma U_i^k(\mathbf{x}^c,\mathbf{y};\omega)t_i(\mathbf{y})d\Gamma_\mathbf{y} - \int_\Gamma T_i^k(\mathbf{x}^c,\mathbf{y};\omega)u_i(\mathbf{y})d\Gamma_\mathbf{y}, \quad 1 \le c \le N_C. \tag{1.47}$$

The discretization of this equation begins with an approximation Γ_h of the surface Γ, i.e.

$$\Gamma \simeq \Gamma_h = \bigcup_{e=1}^{N_E} E_e, \tag{1.48}$$

using a system of surface elements E_e, such that the resulting *mesh* of Γ is denoted by $\mathcal{T}_h(\Gamma) := \{E_e,\ e = 1,\dots,N_E\}$, where N_E is the total number of elements. The index h

represents the grain size of the mesh, i.e. the maximal diameter of an element in $\mathcal{T}_h(\Gamma)$. In principle, the boundary elements may take a variety of forms but usually triangular or quadrilateral elements are used.



(a) Common edge (b) Common vertex (c) Coincident case

Figure 1.3: Conforming triangulation: intersection of two triangles.

The mesh of elements needs to be geometrically *conforming*, i.e, the intersection of two elements must be either empty, or an element, or a vertex, or a common side, and from now on, the mesh considered is assumed to be conforming.

An important technical issue in BEM is the *normal orientation* of the elements. In the standard convention the ordering of the element nodes is such that the normals are always exterior to the domain.

Equation (1.47) then takes the form of a sum of elementary integrals:

$$\gamma_{ki}(\mathbf{x}^c)u_i(\mathbf{x}^c) = \sum_{e=1}^{N_E} \left[\int_{E_e} U_i^k(\mathbf{x}^c, \mathbf{y}; \omega) t_i(\mathbf{y}) d\Gamma_{\mathbf{y}} - \int_{E_e} T_i^k(\mathbf{x}^c, \mathbf{y}; \omega) u_i(\mathbf{y}) d\Gamma_{\mathbf{y}} \right]. \tag{1.49}$$

Unknown displacement and traction fields over each element are interpolated independently through shape functions. Therefore, appropriate functions have to be used, in order to achieve a good accuracy and convergence rate, weighted against computational costs.

In the following, we consider linear interpolation shape functions for the components of both the displacement and the traction fields [33]. Furthermore, only triangular elements are used and we refer to $\mathcal{T}_h(\Gamma)$ as *triangulation.*

Cartesian points $\mathbf{y}$ within a given element E_e are mapped to points $\boldsymbol{\xi} = (\xi_1, \xi_2)^\top$ in a reference element E_{ref} based upon the transformation

$$\mathbf{y}(\boldsymbol{\xi}) := \sum_{p=1}^{3} M^p(\boldsymbol{\xi})\mathbf{y}^p. \tag{1.50}$$

The *control points* or *anchors* $\mathbf{y}^p$ define the geometry of the triangle. The functions M^p provide a mean for mapping and are given by

$$\begin{array}{lcl} M^1(\boldsymbol{\xi}) & := & 1-\xi_1-\xi_2, \\ M^2(\boldsymbol{\xi}) & := & \xi_1, \\ M^3(\boldsymbol{\xi}) & := & \xi_2. \end{array} \tag{1.51}$$

In practice, it is advantageous to use *isoparametric* triangles, i.e. use the same functions M^1, M^2 and M^3 also as shape functions for the approximation of the components of the displacement field, thus

$$\begin{aligned} u_i(\mathbf{y}) &\simeq \sum_{p=1}^{3} \tilde{u}_i^{m(e,p)} M^p(\boldsymbol{\xi}) \quad \mathbf{y}\in E_e, \\ t_i(\mathbf{y}) &\simeq \sum_{p=1}^{3} \tilde{t}_i^{m(e,p)} M^p(\boldsymbol{\xi}) \quad \mathbf{y}\in E_e, \end{aligned} \tag{1.52}$$

where $\tilde{u}_i^{m(e,p)}$ and $\tilde{u}_i^{m(e,p)}$ are the nodal values at the anchors $\mathbf{y}^m$. Since in the mesh the grid points are shared among neighboring triangles, we need to distinguish between the grid points that define an element (the local mesh) and all grid points in the model (the global mesh), many of which are shared among several elements. The index $m(e,p)$ represents a mapping between grid points in the local mesh and grid points in the global mesh.

In the light of these considerations, Equation (1.49) becomes

$$\begin{aligned} \gamma_{ki}(\mathbf{x}^c)u_i(\mathbf{x}^c) = & \sum_{e=1}^{N_E}\sum_{p=1}^{3}\left[\int_{E_e} M^p(\boldsymbol{\xi})U_i^k(\mathbf{x}^c,\mathbf{y};\omega)d\Gamma_{\mathbf{y}}\right]\tilde{t}_i^{m(e,p)} + \\ & -\sum_{e=1}^{N_E}\sum_{p=1}^{3}\left[\int_{E_e} M^p(\boldsymbol{\xi})T_i^k(\mathbf{x}^c,\mathbf{y};\omega)d\Gamma_{\mathbf{y}}\right]\tilde{u}_i^{m(e,p)}. \end{aligned} \tag{1.53}$$

1.3.3 BEM System

Equation (1.53) can be recast in the following vector form

$$\boldsymbol{\gamma}(\mathbf{x}^c)\mathbf{u}(\mathbf{x}^c) = \sum_{e=1}^{N_E}\left\{\left[\int_{E_e}\mathbf{U}(\mathbf{x}^c,\mathbf{y};\omega)\cdot\mathbf{M}(\boldsymbol{\xi})d\Gamma_{\mathbf{y}}\right]\tilde{\mathbf{t}}^e - \left[\int_{E_e}\mathbf{T}(\mathbf{x}^c,\mathbf{y};\omega)\cdot\mathbf{M}(\boldsymbol{\xi})d\Gamma_{\mathbf{y}}\right]\tilde{\mathbf{u}}^e\right\}, \tag{1.54}$$

where

$$\begin{aligned} \tilde{\mathbf{u}}^e &:= \left[\tilde{u}_1^{m(1,e)}\ \tilde{u}_1^{m(2,e)}\ \tilde{u}_1^{m(3,e)}\ \tilde{u}_2^{m(1,e)}\ \tilde{u}_2^{m(2,e)}\ \tilde{u}_2^{m(3,e)}\ \tilde{u}_3^{m(1,e)}\ \tilde{u}_3^{m(2,e)}\ \tilde{u}_3^{m(3,e)}\right]^\top \\ \tilde{\mathbf{t}}^e &:= \left[\tilde{t}_1^{m(1,e)}\ \tilde{t}_1^{m(2,e)}\ \tilde{t}_1^{m(3,e)}\ \tilde{t}_2^{m(1,e)}\ \tilde{t}_2^{m(2,e)}\ \tilde{t}_2^{m(3,e)}\ \tilde{t}_3^{m(1,e)}\ \tilde{t}_3^{m(2,e)}\ \tilde{t}_3^{m(3,e)}\right]^\top \end{aligned} \tag{1.55}$$

are vectors containing all the 9 nodal values respectively of the displacement and the traction field on E_e, and $\mathbf{M}(\boldsymbol{\xi})$ is a matrix given by

$$\mathbf{M}(\boldsymbol{\xi}) := \begin{bmatrix} M^1(\boldsymbol{\xi}) & M^2(\boldsymbol{\xi}) & M^3(\boldsymbol{\xi}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M^1(\boldsymbol{\xi}) & M^2(\boldsymbol{\xi}) & M^3(\boldsymbol{\xi}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & M^1(\boldsymbol{\xi}) & M^2(\boldsymbol{\xi}) & M^3(\boldsymbol{\xi}) \end{bmatrix}. \tag{1.56}$$

The equation for each collocation node may be assembled into a global system of N_c linear equations for the entire domain, i.e.

$$\mathbf{H}\tilde{\mathbf{u}} = \mathbf{K}\tilde{\mathbf{t}}, \tag{1.57}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ are vectors storing all nodal values of the displacements and tractions respectively, while $\mathbf{H}$ and $\mathbf{K}$ are the matrices of influence coefficients.

If the boundary conditions are given in term of a Dirichlet datum along one part Γ_D of the boundary and a Neumann datum on the remaining part Γ_N of the boundary, it is obvious that the vectors $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ are known on Γ_D and Γ_N respectively, i.e. $\tilde{\mathbf{u}} = \tilde{\mathbf{u}}_D$ on Γ_D and $\tilde{\mathbf{t}} = \tilde{\mathbf{t}}_N$ on Γ_N. Therefore, the system (1.57) may be written as

$$\begin{bmatrix} \mathbf{H}_{DD} & \mathbf{H}_{DN} \\ \mathbf{H}_{ND} & \mathbf{H}_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_D \\ \tilde{\mathbf{u}}_N \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{DD} & \mathbf{K}_{DN} \\ \mathbf{K}_{ND} & \mathbf{K}_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{t}}_D \\ \tilde{\mathbf{t}}_N \end{bmatrix}. \tag{1.58}$$

Upon rearrangement, the final result can be written as

$$\begin{bmatrix} -\mathbf{K}_{DD} & \mathbf{H}_{DN} \\ -\mathbf{K}_{ND} & \mathbf{H}_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{t}}_D \\ \tilde{\mathbf{u}}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{H}_{DD} & \mathbf{K}_{DN} \\ -\mathbf{H}_{ND} & \mathbf{K}_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_D \\ \tilde{\mathbf{t}}_N \end{bmatrix}, \tag{1.59}$$

whose compact form is

$$\mathbf{A}\mathbf{z} = \mathbf{b}, \tag{1.60}$$

where the *influence matrix* $\mathbf{A}$, the unknown vector $\mathbf{z}$ and the *Right Hand Side* (RHS) $\mathbf{b}$ are given by

$$\mathbf{A} := \begin{bmatrix} -\mathbf{K}_{DD} & \mathbf{H}_{DN} \\ -\mathbf{K}_{ND} & \mathbf{H}_{NN} \end{bmatrix}, \qquad \mathbf{z} := \begin{bmatrix} \tilde{\mathbf{t}}_D \\ \tilde{\mathbf{u}}_N \end{bmatrix} \quad \text{and} \quad \mathbf{b} := \begin{bmatrix} -\mathbf{H}_{DD} & \mathbf{K}_{DN} \\ -\mathbf{H}_{ND} & \mathbf{K}_{NN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_D \\ \tilde{\mathbf{t}}_N \end{bmatrix}. \tag{1.61}$$

In order to have a square system, on each triangle the collocation has to be performed at the vertices (*nodal collocation*), because both the displacement and the traction fields are unknown on this element.

1.3.4 Solution Strategies for BEM Systems

In Equation (1.60), $\mathbf{A}$ is a *full* matrix because, as the involved boundary integral operators are non-local, its entries do not vanish in general. For this reason, the dimensional advantage of BEM with respect to domain discretization methods is offset.

The computing time and the memory requirements for the evaluation of the matrix entries increase quadratically with N (N^2 operations and N^2 units of storage are required).

Two classes of methods for solving system (1.60) are of interest: *direct methods* (such as the LU factorization) and *iterative methods* (GMRES [152] being the usual choice in case of non-symmetric matrices). In a direct method, the system matrix is transformed or factorized into a simpler form, involving diagonal or triangular matrices. If numerical rounding errors are not considered, the exact solution is obtained in a finite number of arithmetic operations. On the other hand, iterative methods compute a *sequence* of approximate solutions, which converges to the exact solution to the limit, i.e. in practice until a desired accuracy is obtained. Their computational cost is of the order of $\mathcal{O}(N^2)$ operations for each iteration, to be compared with an overall cost of the order of $\mathcal{O}(N^3)$ operations needed by direct methods. Direct methods can therefore become competitive with iterative methods provided the total number of iterations N_{it} depends on N. Furthermore, direct solvers have advantages when the number of right hand sides N_{rhs} is large.

Due to the current available computer capacity, the implementation of the standard BEM on laptops is restricted to problems of rather small size, i.e. not exceeding $N = \mathcal{O}(10^4)$, because the total cost of the problem (cost of building the matrix plus cost of solving the system) is at least of the order of $\mathcal{O}(N^2)$. However, applications of the BEM to large models (typically $N = \mathcal{O}(10^6)$) requires procedures that are *fast* and *avoid the explicit storage* of the system matrix. The idea is to compress the informations and give a *data-sparse* representation of $\mathbf{A}$, whose amount of storage is no longer quadratic but logarithmic-linear or almost linear. These procedures are known as *fast BEM* techniques. In the last decades, several fast BEM have been published, including the Fast Multipole Method (FMM) [151] and the Hierarchical Matrices ($\mathcal{H}$-matrices) [81].

2 Principle of $\mathcal{H}$-matrices

Contents

2.1 Introduction 30

2.2 General Description of $\mathcal{H}$-Matrices 30

2.2.1 Motivation of an $\mathcal{H}$-Matrix Representation of BEM Matrices . 31

2.2.2 Clustering of the unknowns 34

2.2.3 Subdivision of the matrix 35

2.2.4 Admissibility Condition 37

2.3 Algorithms to Perform Low-Rank Approximations 38

2.3.1 Truncated Singular Value Decomposition 38

2.3.2 Fully-Pivoted Adaptive Cross Approximation 38

2.3.3 Partially-Pivoted Adaptive Cross Approximation 42

2.3.4 Recompression of the Admissible Blocks 43

2.4 Extension of $\mathcal{H}$-matrix to BEM for Elastodynamics 44

2.4.1 Application of ACA to Oscillatory Kernels: Theoretical Estimates 45

2.4.2 Vectorial Partial-Pivoted Adaptive Cross Approximation 48

2.5 $\mathcal{H}$-matrix Based Solvers 49

2.5.1 $\mathcal{H}$-matrix Based Iterative Solver 49

2.5.2 $\mathcal{H}$-matrix Arithmetic 50

2.5.3 $\mathcal{H}$-matrix Based Direct Solver 63

2.6 Conclusions 65

2.1 Introduction

In the introductory Chapter 1, the standard boundary element method (BEM) has been presented and shown to be well suited to deal with problems in spatially-extended regions (idealized as unbounded), typically not only of seismology but also of acoustics and electromagnetics. In fact, the BEM is based on a boundary integral formulation that accounts implicitly for the radiation condition at infinity. Numerically, it requires only the discretization of the domain boundaries (i.e a surface for 3-D problems) and possible interfaces. However, the BEM leads to a fully-populated coefficient matrix, which makes the implementation of traditional BEM prohibitive for large-scale realistic problems. It is thus essential to develop alternative, faster strategies that still allow to exploit the known advantages of BEM.

Initially introduced in the context of potential problems [83], the hierarchical matrices are a very efficient way to treat dense matrices [73, 81]. Due to their purely algebraic nature, in addition to the memory advantages of storing such matrices, approximations of the usual matrix operations (addition, multiplication, inversion, etc...) can be computed (with logarithmic-linear complexity in case of potential problems [73]) and allow to define fast solvers. In particular, the $\mathcal{H}$-matrix/vector product accelerates the classical matrix/vector product and, consequently, reduces the time of each iteration of an iterative solver, which is the most expensive task for these solvers. Since the convergence rate (number of iterations) of iterative solvers usually deteriorates with the increase of the problem size, in many applications a direct solution strategy is an interesting option and $\mathcal{H}$-matrix arithmetic has been used to do an approximated LU decomposition, in order to reduce the numerical effort of a direct solver. There is a vast literature on the applications of $\mathcal{H}$-matrices. In particular, this technique has been successfully employed to solve elliptic Helmholtz [9], Laplace [138] and Lamé [124] problems.

This Chapter aims at describing the principles of $\mathcal{H}$-matrices and presenting the principles of $\mathcal{H}$-matrix based iterative and direct solvers. The Chapter is organized as follows. Section 2.2 gives a general description of $\mathcal{H}$-matrices, illustrating how to establish a hierarchical separation of a matrix into sub-blocks, and introducing the concepts of cluster tree, block cluster-tree and admissibility condition. Section 2.3 is devoted to the algorithms to compute low-rank approximations, needed for the reduction of memory requirements. In Section 2.4 we consider the extension to vectorial oscillatory problems. Finally, in Section 2.5 iterative and direct solvers based on $\mathcal{H}$-matrices are presented.

2.2 General Description of $\mathcal{H}$-Matrices

$\mathcal{H}$-matrices have been introduced by Hackbusch [81] to compute a data-sparse representation of some special dense matrices (for example matrices resulting from the discretization of non-local operators). The key features of $\mathcal{H}$-matrices are:

1. a hierarchical separation of the space, leading to a block decomposition of the

matrix;

2. a simple tool to carefully determine a priori which blocks are approximately of low-rank, i.e. can be approximated by low-rank matrices, since the explicit computation of the ranks of the sub-matrices would be too expensive;

3. a method to compute a low-rank approximation of blocks that are approximately of low-rank.

With these three ingredients, it is possible to define fast solvers for matrices having a hierarchical representation. Using low-rank approximation, the memory requirements and the cost of a matrix-vector product are reduced [25]. In addition, using $\mathcal{H}$-matrix arithmetic [83], it is possible to derive fast direct solvers.

2.2.1 Motivation of an $\mathcal{H}$-Matrix Representation of BEM Matrices

The possibility of an $\mathcal{H}$-matrix representation of the influence matrix A coming from the discretization of a non-local operator, is linked to the concept of low-rank matrices.

Before providing a definition of such matrices, we recall that the *rank* of a matrix $M \in \mathbb{C}^{m\times n}$ corresponds to the common value of the *column rank*, i.e. the maximum number of linearly independent column vectors, and the *row rank*, i.e. the maximum number of linearly independent row vectors. Usually, M is said to be *full-rank* if its rank equals the minimum between m and n. Otherwise, M is said to be *rank deficient*. Supposing that M is rank deficient with rank k, it can be represented in the so-called *outer-product form*, i.e. there exist two matrices $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$ such that

$$M = UV^H. \tag{2.1}$$

where the matrix V^H is the conjugate transpose of V. The storage requirement for M is therefore reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(k(m+n))$. This leads to a formal definition of *low-rank* matrices, i.e. rank deficient matrices such that $k(m+n) << mn$. It is known that low-rank matrices allow always a representation in outer-product form.

Briefly, we recall that the SVD of a matrix $M \in \mathbb{C}^{m\times n}$ with rank $r \leq \min\{m,n\}$ is a factorization of the form

$$M := \sum_{l=1}^{r} \mathbf{u}_l s_l \mathbf{v}_l^H = USV^H, \tag{2.2}$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_r$ are the first eigenvectors of MM^H (*left singular vectors* forming the matrix $U \in \mathbb{C}^{m\times r}$), $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are the first eigenvectors of M^HM (*right singular vectors* forming the matrix $V \in \mathbb{C}^{n\times r}$) and $s_1 > s_2 > \ldots > s_k$ are the square roots of the eigenvalues of M^HM (*singular values* placed on the diagonal of the diagonal matrix $S \in \mathbb{C}^{r\times r}$).

The SVD is linked to the Frobenius norm $\|\cdot\|_{\mathrm{F}}$ and to the spectral norm $\|\cdot\|_2$, because it holds:

$$\|M\|_2^2 = s_1^2 \leq \sum_{l=1}^{r} s_l^2 = \|M\|_F^2. \tag{2.3}$$

In terms of relative error of approximation, an alternative notion of rank is introduced [15], based on a required accuracy $\varepsilon > 0$, in order to compute a low-rank approximation of matrices that are approximately of low-rank. The ε-rank $k := k(\varepsilon)$ of the matrix M in a unitary invariant norm $\|\cdot\|$ (for example Frobenius norm or spectral norm) is defined as

$$k(\varepsilon) := \min\{k \text{ such that } \|M - M_k\| \leq \varepsilon\|M\|\}, \tag{2.4}$$

where M_k is the Singular Value Decomposition (SVD) of M truncated to the k largest singular values, i.e.

$$M_k := \sum_{l=1}^{k} \mathbf{u}_l s_l \mathbf{v}_l^H \tag{2.5}$$

Since the matrices coming from the discretization of non local operators are invertible and, consequently, of full rank, representing exactly the entire matrix A by a low-rank matrix is not feasible. However, it is possible to establish a *hierarchical separation* of the matrix into sub-blocks, such that some of those are approximately of low-rank. In order to simply illustrate this hierarchical reppresentation, we consider the kernel function $G_h : [-1,1] \times [-1,1] \to \mathbb{C}$ defined as

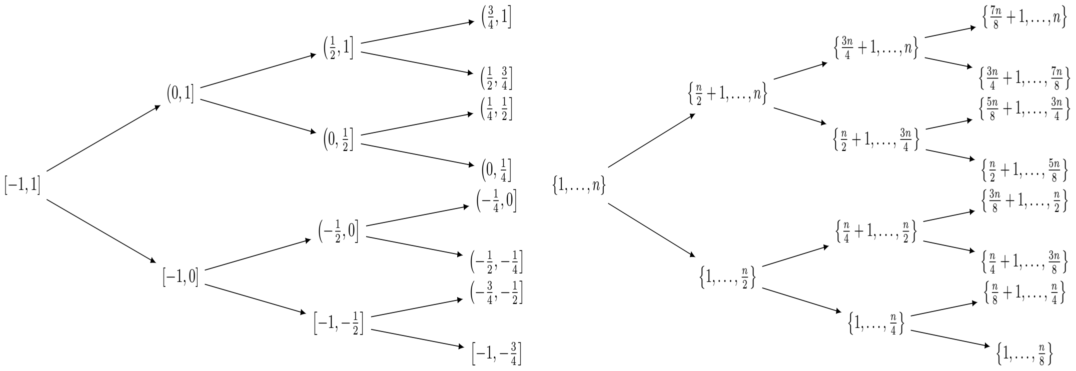
$$G_h(x,y;\kappa) = \frac{e^{\imath\kappa|x-y|}}{4\pi|x-y| + h}, \tag{2.6}$$

i.e. the Helmholtz fundamental solution modified by introducing the parameter $h > 0$, in order to avoid the singularity occurring as $x \to y$. We choose h such that it equals the discretization step used to discretize the interval $[-1,1]$. Since we want 10 discretization points per wave-length (as standard procedure), we fix the wave-number $\kappa = 100\pi$ and put $h = {}^{\pi}\!/_{5\kappa}$. With this value of h we obtain $n = 1000$ discretization points uniformly discretized and given by:

$$x_i = -1 + (i-1)h, \quad i = 1, \ldots, n. \tag{2.7}$$

We call G the $n \times n$ matrix with entries $G_{ij} = G_h(x_i, x_j; \kappa)$ and $I := \{1, \ldots, n\}$ the set of degrees of freedom of the problem. Since G has full rank, which is not surprising remembering the near-singular nature of G_h, we cluster the interval $[-1,1]$ into two sub-intervals $[-1,0]$ and $(0,1]$. Remembering that each element $i \in I$ is bijectively mapped to one discretization point x_i, the set I is naturally partitioned into two subsets $\{1, \ldots, {}^{n}\!/_{2}\}$ and $\{{}^{n}\!/_{2}{+}1, \ldots, n\}$, inducing a partition of matrix G in four sub-blocks, that are the candidates for checking in which parts of the matrix low-rank approximations are possible. This procedure have to be applied to each of these four sub-blocks.

In Figure 2.1, we give an example of the partition of the interval $[-1, 1]$ (2.1a) and, consequently, of the set $I = \{1, \ldots, n\}$ (2.1b). The sub-intervals and the sub-sets obtained are organized in two binary trees with four levels.

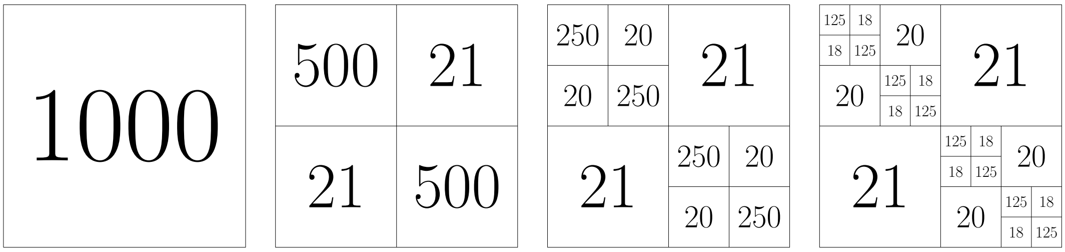

$[-1,1]$
$(0,1]$
$[-1,0]$
$(\frac{1}{2},1]$
$(0,\frac{1}{2}]$
$(-\frac{1}{2},0]$
$[-1,-\frac{1}{2}]$
$(\frac{3}{4},1]$
$(\frac{1}{2},\frac{3}{4}]$
$(\frac{1}{4},\frac{1}{2}]$
$(0,\frac{1}{4}]$
$(-\frac{1}{4},0]$
$(-\frac{1}{2},-\frac{1}{4}]$
$(-\frac{3}{4},-\frac{1}{2}]$
$[-1,-\frac{3}{4}]$
$\{1,\ldots,n\}$
$\{\frac{n}{2}+1,\ldots,n\}$
$\{1,\ldots,\frac{n}{2}\}$
$\{\frac{3n}{4}+1,\ldots,n\}$
$\{\frac{n}{2}+1,\ldots,\frac{3n}{4}\}$
$\{\frac{n}{4}+1,\ldots,\frac{n}{2}\}$
$\{1,\ldots,\frac{n}{4}\}$
$\{\frac{7n}{8}+1,\ldots,n\}$
$\{\frac{3n}{4}+1,\ldots,\frac{7n}{8}\}$
$\{\frac{5n}{8}+1,\ldots,\frac{3n}{4}\}$
$\{\frac{n}{2}+1,\ldots,\frac{5n}{8}\}$
$\{\frac{3n}{8}+1,\ldots,\frac{n}{2}\}$
$\{\frac{n}{4}+1,\ldots,\frac{3n}{8}\}$
$\{\frac{n}{8}+1,\ldots,\frac{n}{4}\}$
$\{1,\ldots,\frac{n}{8}\}$

(a) Partition of the domain of the problem

(b) Partition of the set of degrees of freedom

Figure 2.1: Partition of the interval $[-1, 1]$ and, consequently, of the set of degrees of freedom $I = \{1, \ldots, n\}$ (n=1000).

In Figure 2.2, we present the structure of the $\mathcal{H}$-matrix obtained. From the first diagram we see that the whole matrix has full rank. The second diagram in Figure 2.2 shows the ranks of the four blocks obtained splitting the interval $[-1, 1]$ in two sub-intervals, and consequently partitioning the set of row and column indices of the matrix in two subsets. These ranks are computed by the MATLAB function *rank* with the default tolerance. We observe that the off-diagonal blocks G_{12} and G_{21} are approximately of low-rank, while the diagonal blocks G_{11} and G_{22} are full-rank. We subdivide these blocks further (third and fourth diagrams). The amount of entries required for storing the matrix at each level of the approximation is: 1 000 000 (the whole matrix), 542 000 (almost the 50% of the original matrix), 292 000 (almost the 30% of the original matrix) and 243 000 (almost the 25% of the original matrix). Clearly, we have a drastic reduction in storage requirements.



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

Figure 2.2: Ranks of the sub-blocks of the matrix corresponding to $G_h(x, y; \kappa)$ (n=1000)

In Figures 2.3 and 2.4, we present the singular values of the diagonal block G_{11} and of off-diagonal block G_{12} in Step 2 of Figure 2.2. We observe that the approximation of G_{11} would require a lot of informations, since all the singular values are significant, i.e. far from the machine epsilon. Instead, the singular values of G_{12} decrease very quickly and a good approximation is achieved considering only a few of them in (2.5).

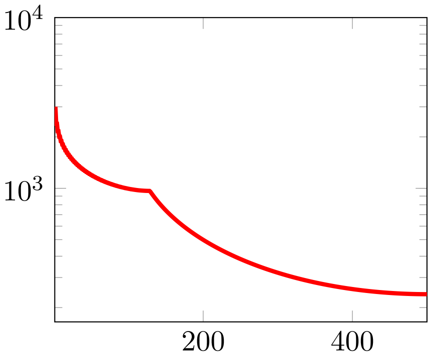


Figure 2.3: Singular values of diagonal blocks (Step 2 of Figure (2.2))

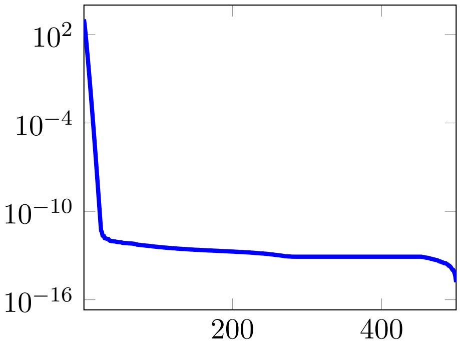


Figure 2.4: Singular values of off-diagonal blocks (Step 2 of Figure (2.2))

Thanks to this simple toy model, we have investigated the possibility of decomposing a matrix A coming from the discretization of a non-local operator into a system of blocks which can be approximated by low-rank matrices. This operation is very subtle because each low-rank approximation introduces an error that could propagate on the $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$ of the whole matrix A. To avoid this phenomenon, each low-rank approximation has to be computed according to an acceptable tolerance, that allows to guarantee the error bound

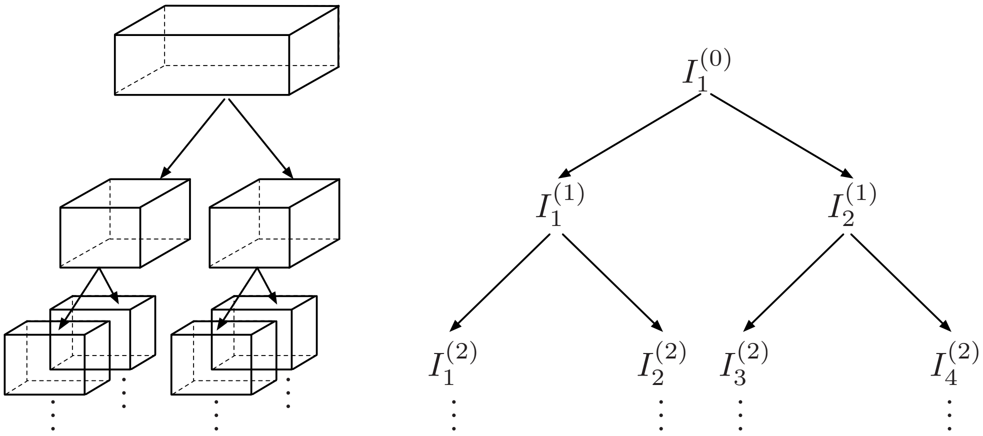
$$\|A - A_{\mathcal{H}}\| \leq \varepsilon \|A\|, \tag{2.8}$$

where $\|\cdot\|$ denotes an unitarily invariant norm and $\varepsilon > 0$ a given accuracy.

2.2.2 Clustering of the unknowns

As explained in the previous Section, the first step prior to the partition of the matrix A consists in a partitioning of the domain, that induces a partitioning of the set I of degrees of freedom. The purpose is to obtain cluster of degrees of freedom such that constitutive indices correspond to points interacting at close range.

In a realistic 3D scenario, the clustering requires a data structure called *cluster tree*, i.e. a full binary tree with root I, where the leaves consist of a relatively small number $N_{\text{LEAF}} \in \mathbb{N}$ of indices. Each node of the cluster tree, from now on denoted by T_I, defines a subset of indices $\tau \subset I$, corresponding to a sub-domain partition of the domain (see Figure 2.5b). If $|\tau| \leq N_{\text{LEAF}}$, τ is a *leaf* and it has no sons. Otherwise, τ has two sons $\tau_1, \tau_2 \subset I$ and it is their disjoint union. The levels of T_I are counted beginning with the root and its *depth* $L(I)$ is determined by N_{LEAF}.



(a) Partition of the domain

(b) Cluster Tree T_I

Figure 2.5: Clustering of the degrees of freedom: (a) partition of the degrees of freedom in the domain and (b) corresponding binary tree.

By induction, it is easy to prove [14] that the number of nodes in a cluster tree is bounded by the dimension of the problem, i.e.

$$|T_I| \leq 2|I| - 1. \tag{2.9}$$

In theory there are more than $|I|$ possible cluster trees for the set I, all generated by recursive subdivisions of I [25]. In the most common strategies each cluster $\tau \in T_I$ is enclosed into a simple structure, called *bounding box*, and it is divided into two sons using a *separation plane*, orthogonal to a chosen direction (see Figure 2.5a).

Depending on the position of this plane, we have two types of cluster trees:

- a *balanced cluster tree*, i.e. a cluster tree where each non-leaf cluster is split into two sons of approximately equal cardinality;
- a *geometric cluster tree*, i.e. a cluster tree where each non-leaf cluster is split into two sons whose bounding boxes have approximately equal volume.

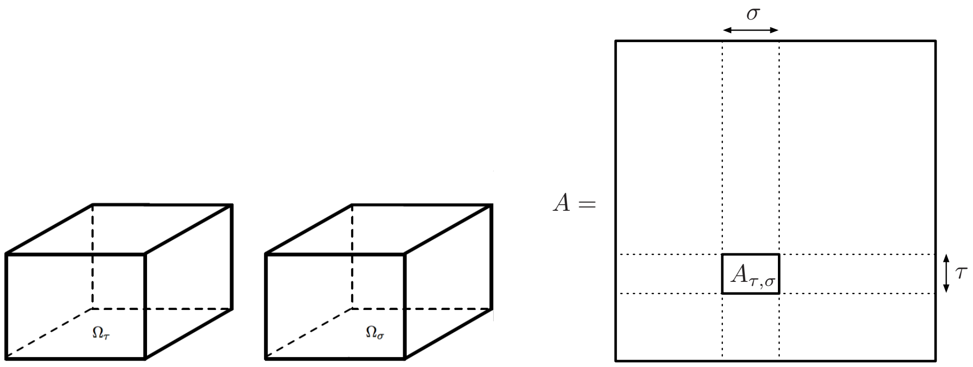
It is important to remark that, during the construction of the cluster tree, a renumbering of the degrees of freedom occurs.

When the geometry of the domain is regular and its discretization is uniform, these two approaches are equivalent. In this work we only consider geometric clusters tree.

2.2.3 Subdivision of the matrix

The next step consists in a subdivision of the matrix A into a system of non-overlapping sub-blocks. A uniform partition based on the cluster tree T_I would be inefficient, since the number of all possible candidates is $|T_I|^2 = |I|^2$. Additionally, such a partition is not fine enough such that most of the blocks can be successfully approximated by low-rank matrices. For this reason a hierarchical representation is introduced.

Starting from the initial matrix A, a block cluster representation $T_{I\times I}$ of the matrix A is performed by going through the cluster tree T_I. Each node of $T_{I\times I}$ contains a pair (τ,σ) of clusters of T_I and defines a block of A (Figure 2.6a). τ is called *row-cluster* and σ is called *column-cluster*. Starting from the initial matrix, each block is recursively subdivided until its row-cluster or its column-cluster is a leaf of T_I [25].



(a) Clustering of the unknowns

(b) Corresponding block in the matrix

Figure 2.6: 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.

The set $\mathrm{Leaf}(T_{I\times I})$ of all the leaves of $T_{I\times I}$, that are nodes which do not have any sons, realizes a partition, among whose elements we have to find which parts of matrix A can be well approximated by a low-rank matrix and which parts have to be stored as dense matrices. Since the computation of the rank of a matrix is a very expensive procedure (based on the SVD), this aim is accomplished introducing an *admissibility condition*, i.e. a boolean function $Adm : T_I \times T_I \to \{\mathrm{true}, \mathrm{false}\}$. If $Adm(\tau,\sigma) = \mathrm{true}$, the cluster pair (τ,σ) (or equivalent the corresponding block of $T_{I\times I}$) corresponds to a block that can be approximated by a low-rank matrix and it is marked as *admissible*. Otherwise, it is a full-rank matrix and it is marked as *non-admissible*.

Defining the sets

$$\begin{aligned}\mathrm{Leaf}^+(T_{I\times I}) &:= \{(\tau,\sigma) \in T_{I\times I} : Adm(\tau,\sigma) = \mathrm{true}\}\\ \mathrm{Leaf}^-(T_{I\times I}) &:= \{(\tau,\sigma) \in T_{I\times I} : Adm(\tau,\sigma) = \mathrm{false}\},\end{aligned} \tag{2.10}$$

we can look at the partition $\mathrm{Leaf}(T_{I\times I})$ as their disjoint union, i.e.

$$\mathrm{Leaf}(T_{I\times I}) = \mathrm{Leaf}^+(T_{I\times I}) \cup \mathrm{Leaf}^-(T_{I\times I}). \tag{2.11}$$

2.2.4 Admissibility Condition

$\mathcal{H}$-matrices are efficient representations for matrices coming from the discretization of asymptotically smooth kernels. In this case, the method is well-documented and estimates are provided for the Taylor expansion and interpolation errors. A kernel function $K(\mathbf{x}, \mathbf{y})$ is said *asymptotically smooth* if there exist two real constants $c_1, c_2 > 0$ and a singular degree $\beta \in \mathbb{N}$ such that for any $z \in \{x_i, y_i \ : \ i = 1, 2, 3\}$ and any $\alpha \in \mathbb{N}$ it holds:

$$|\partial_z^\alpha K(\mathbf{x}, \mathbf{y})| \leq \alpha! c_1 (c_2 \|\mathbf{x} - \mathbf{y}\|)^{-\alpha-\beta}, \quad \mathbf{x} \neq \mathbf{y}. \tag{2.12}$$

The admissibility condition was defined in the context of the data-sparse approximation of matrices resulting from the discretization of the boundary integral equations involving this type of kernels [25]. The admissibility condition depends on a parameter $\eta < 0$ and it is given by

$$Adm(\tau, \sigma) = \text{true} \quad \Leftrightarrow \min\{\text{diam}(\tau), \text{diam}(\sigma)\} \leq \eta \text{dist}(\tau, \sigma), \tag{2.13}$$

where the *diameter* of a cluster $\tau \in T_I$ is defined as

$$\text{diam}(\tau) := \max_{i,j \in \tau} \|\mathbf{x}_i - \mathbf{x}_j\|, \tag{2.14}$$

while the *distance* between two clusters $\tau, \sigma \in T_I$ is given by

$$\text{dist}(\tau, \sigma) := \min_{i \in \tau, j \in \sigma} \|\mathbf{x}_i - \mathbf{x}_j\|. \tag{2.15}$$

Since the computation of these quantities can be too expensive, for practical implementations it is convenient to replace the clusters τ and σ with their bounding boxes B_τ and B_σ (see Figure 2.7). Thus, the admissibility condition becomes:

$$Adm(\tau, \sigma) = \text{true} \quad \Leftrightarrow \min\{\text{diam}(B_\tau), \text{diam}(B_\sigma)\} \leq \eta \text{dist}(B_\tau, B_\sigma). \tag{2.16}$$

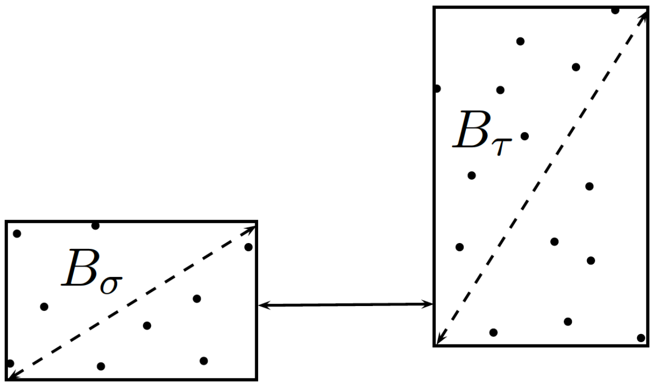


Figure 2.7: Illustration of the admissibility condition with bounding boxes.

2.3 Algorithms to Perform Low-Rank Approximations

Informally, an $\mathcal{H}$-matrix representation of the matrix A is a structure $A_{\mathcal{H}}$ where the matrix blocks $A_{\tau,\sigma}$ corresponding to a non-admissible cluster pair $(\tau, \sigma) \in \text{Leaf}^{-}(T_{I\times I})$ are stored as dense matrices and the matrix blocks $A_{\tau,\sigma}$ corresponding to an admissible cluster pair $(\tau, \sigma) \in \text{Leaf}^{+}(T_{I\times I})$ are approximated by low-rank matrices and stored in outer-product form (2.1). The storage requirement of the matrix blocks $A_{\tau,\sigma}$ with cluster pair $(\tau, \sigma) \in \text{Leaf}^{-}(T_{I\times I})$ is influenced by the (small) parameter N_{LEAF}, since $|\tau| \simeq |\sigma| \simeq N_{\text{LEAF}}$. On the other hand, the storage requirement of the matrix blocks $A_{\tau,\sigma}$ with cluster pair $(\tau, \sigma) \in \text{Leaf}^{+}(T_{I\times I})$ is determined by the rank of its low-rank approximation. Thus, the crucial point for computing the $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$ is an accurate and rank revealing algorithm to perform low-rank approximations of the admissible blocks.

2.3.1 Truncated Singular Value Decomposition

The best possible low-rank approximation (with rank k) of the matrix $M \in \mathbb{C}^{m\times n}$ with rank $r \le \min\{m, n\}$ is given by its truncated SVD (Eckart-Young Theorem), i.e.

$$M_k := \sum_{l=1}^{k} \mathbf{u}_l s_l \mathbf{v}_l^H = USV^H, \qquad \text{with } k < r. \tag{2.17}$$

Unfortunately, the computation of the truncated SVD is too expensive, in the order of $\mathcal{O}(\max(m, n) \min^2(m, n))$. In addition, it requires the knowledge of all the entries of M, i.e. an assembling cost of the order of $\mathcal{O}(mn)$. For this reason, in the context of the $\mathcal{H}$-matrices, the use of the truncated SVD would be equivalent to assembly the complete matrix.

2.3.2 Fully-Pivoted Adaptive Cross Approximation

The Adaptive Cross Approximation (ACA) [17, 18] offers an interesting alternative to the SVD since it produces a quasi-optimal low-rank approximation. The starting point of the ACA is that every matrix $M \in \mathbb{C}^{m\times n}$ 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.

The matrix M is split into $M = S_k + R_k$, where S_k denotes the rank k approximation of M and R_k denotes the residual to be minimized. Starting from

$$S_0 := 0 \quad \text{and} \quad R_0 := M \tag{2.18}$$

at each iteration the pivot $\gamma_{k+1} = (R_k)_{ij}^{-1}$ is the inverse of the largest entry in R_k (see [15] for further details about the right choice of the initial pivot γ_1). At each step, the information is shifted iteratively from the residual to the approximant, by subtracting the outer product of the pivot row and the column from R_k and adding it to S_k:

$$\begin{array}{rcl} R_{k+1} & := & R_k - \mathbf{u}_{k+1}\mathbf{v}_{k+1}^\top, \\ S_{k+1} & := & S_k + \mathbf{u}_{k+1}\mathbf{v}_{k+1}^\top \end{array} \tag{2.19}$$

with the i-th row vector and the j-th column vector defined as

$$\mathbf{v}_{k+1} = \gamma_{k+1}(R_k)_{i,\cdot} \quad \text{and} \quad \mathbf{u}_{k+1} = \gamma_{k+1}(R_k)_{\cdot,j}. \tag{2.20}$$

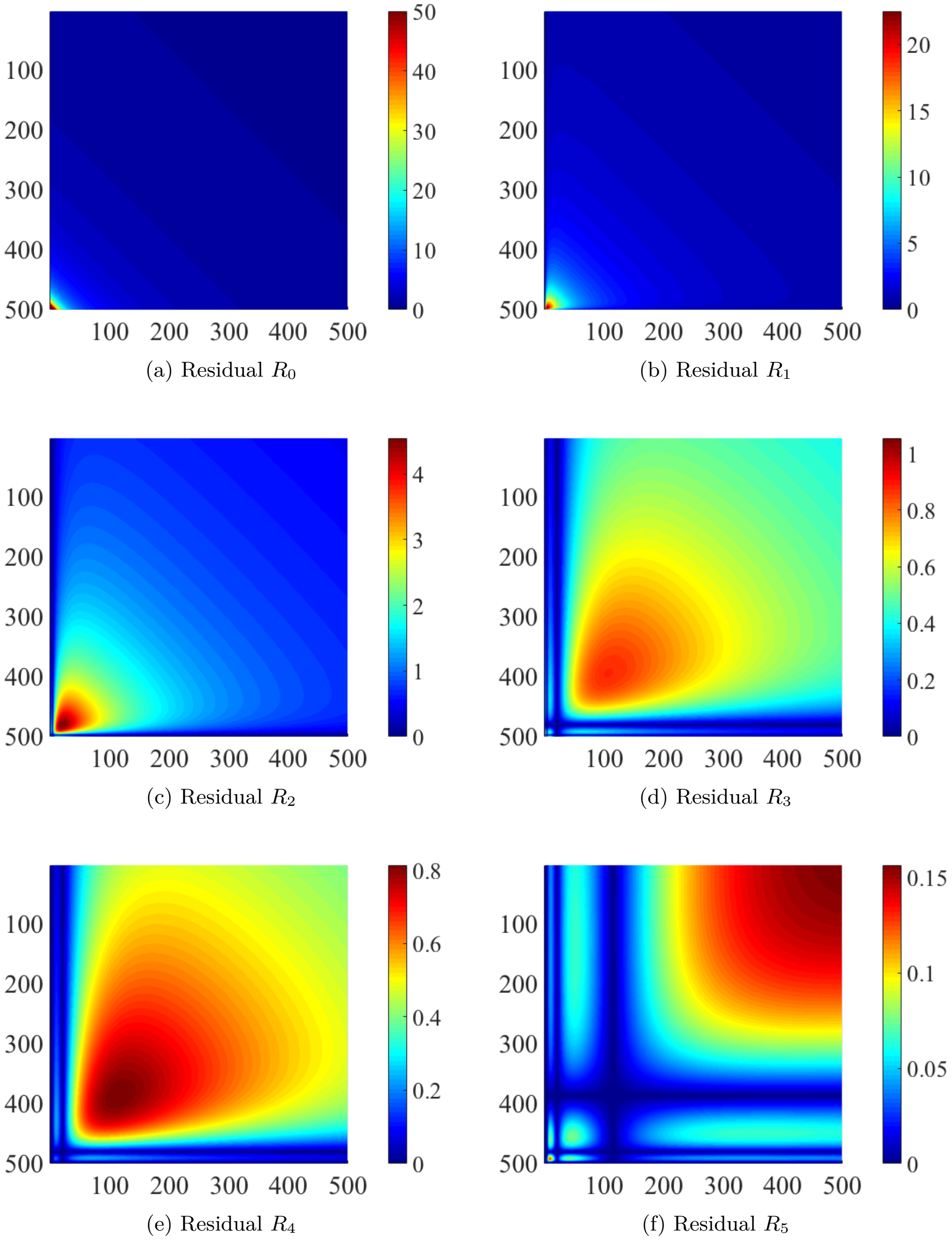
In this way the residual R_k is minimized and the rank of the approximation S_k is increased step by step. Finally, the approximation stops if the following criterion holds:

$$\|M - S_k\|_F \leq \varepsilon_{\text{ACA}}\|M\|_F, \tag{2.21}$$

where $\varepsilon_{\text{ACA}} > 0$ is a parameter of the method.

The efficiency of the fully-pivoted ACA is illustrated using the off-diagonal block G_{12} of the matrix $G_{ij} = G_h(x_i, x_j; \kappa)$ obtained in Step 2 of Figure 2.2. In Figure 2.8 and Figure 2.9, the successive minimizations of the residual R_k are shown. Note that the exactly computed crosses are shown in black, while the remaining color scales are adapted to the actual values of the residuals and, therefore, are different for all pictures. In Table 2.1, we report the indices of the row and the column pivot, the value of the pivot and the relative error at each iteration of the algorithm. All these results correspond to the value $\varepsilon_{\text{ACA}} = 10^{-4}$ for the threshold in the stopping criterion (2.21). In particular we observe that the first five steps of the algorithm remove rows and columns in the left lower part of the matrix. Further iterations quickly reduce the relative error of approximation in the Frobenius' norm. As we expect, the value of the pivot decreases at each iteration.

The complexity of the fully-pivoted ACA to generate an approximation of rank k is $\mathcal{O}(kmn)$ (at each iteration $\mathcal{O}(mn)$ operations are required to update the approximant S_k and the residual R_k). Similarly to the SVD, it requires the computation of all the entries of M to find the pivot. Thus, the ACA in the fully-pivoted form does not fulfill our requirements.



(a) Residual R_0

(b) Residual R_1

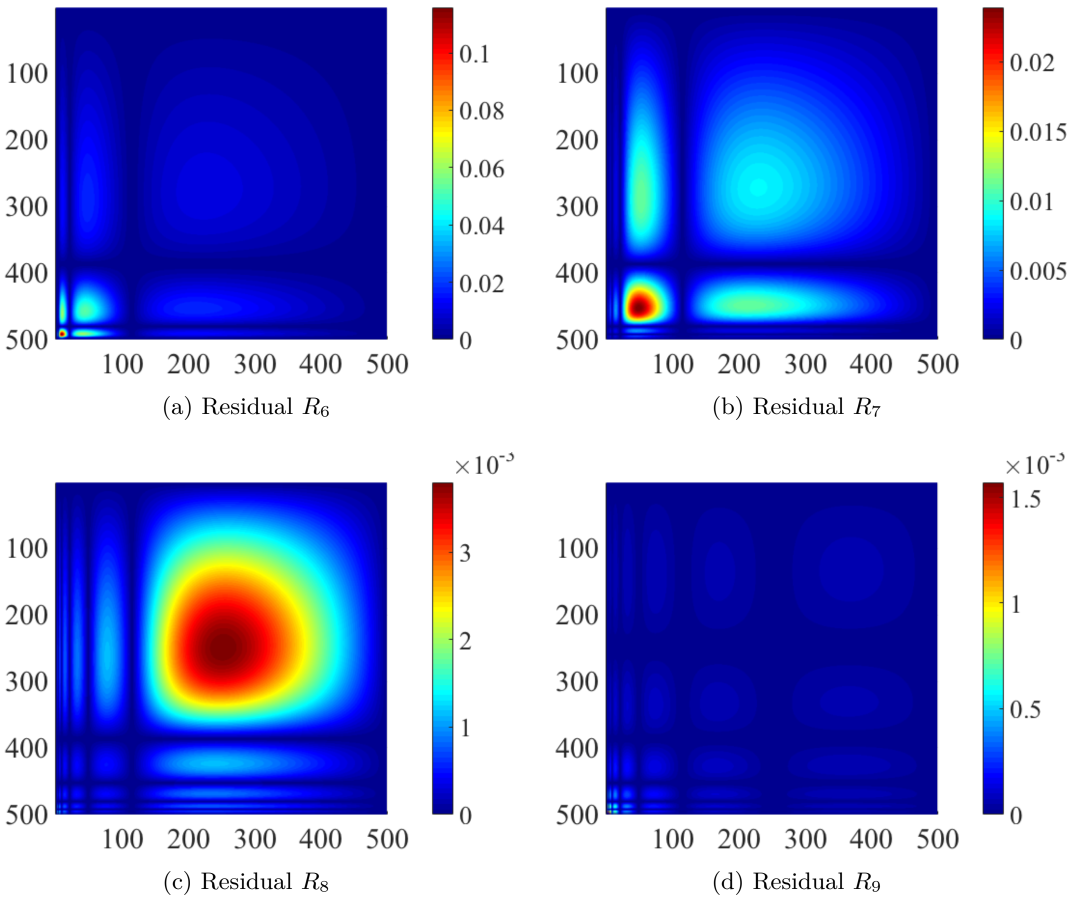
(c) Residual R_2

(d) Residual R_3

(e) Residual R_4

(f) Residual R_5

Figure 2.8: From top to bottom and from left to right, residuals $R_0 - R_5$ produced by fully-pivoted ACA of the off-diagonal block G_{12} of matrix G (Step 2 of Figure 2.2).



(a) Residual R_6

(b) Residual R_7

(c) Residual R_8

(d) Residual R_9

Figure 2.9: From top to bottom and from left to right, residuals $R_6 - R_9$ produced by fully-pivoted ACA of the off-diagonal block G_{12} of matrix G (Step 2 of Figure 2.2).

Step	Pivot row	Pivot column	$1/\gamma_k$	Relative Error
1	500	1	$2.50 \times 10^{+2}$	6.94×10^{-1}
2	497	4	$2.25 \times 10^{+1}$	4.80×10^{-1}
3	480	21	$4.55 \times 10^{+0}$	2.32×10^{-1}
4	499	2	$1.05 \times 10^{+0}$	2.21×10^{-1}
5	386	115	8.13×10^{-1}	3.41×10^{-2}
6	1	500	1.56×10^{-1}	3.31×10^{-3}
7	492	9	1.16×10^{-1}	1.95×10^{-3}
8	452	49	2.39×10^{-2}	6.09×10^{-4}
9	249	252	3.79×10^{-3}	1.80×10^{-5}

Table 2.1: Fully-pivoted ACA of the off-diagonal block G_{12} of matrix G (Step 2 of Figure 2.2).

2.3.3 Partially-Pivoted Adaptive Cross Approximation

The *partially-pivoted* ACA proposes a different approach to choose the pivot, avoiding the assembly of the complete matrix. The idea is to maximize the residual only for one of the two indices and to keep the other one fixed. With this strategy, only one row or one column is assembled at each iteration. More precisely, at iteration k, given S_k and assuming the row index i^* is known the algorithm is given by the following six steps:

1. generation of the rows $\mathbf{a} := M^\top \mathbf{e}_{i^*}$ and $R_k^\top \mathbf{e}_{i^*} := \mathbf{a} - \sum_{l=1}^{k} (\mathbf{u}_l)_{i^*} \mathbf{v}_l$
2. find the column index $j^* := \arg\max_j |(R_k)_{i^*j}|$ and compute $\gamma_{k+1} := ((R_k)_{i^*j^*})^{-1}$
3. generation of the columns $\mathbf{b} := M\mathbf{e}_{j^*}$ and $R_k \mathbf{e}_{j^*} := \mathbf{b} - \sum_{l=1}^{k} (\mathbf{v}_l)_{j^*} \mathbf{u}_l$
4. find the next row index $i^* := \arg\max_i |(R_k)_{ij^*}|$
5. compute vectors $\mathbf{u}_{k+1} := \gamma_{k+1} R_k \mathbf{e}_{j^*}$ and $\mathbf{v}_{k+1} := R_k^\top \mathbf{e}_{i^*}$
6. update the approximation $S_{k+1} := S_k + \mathbf{u}_{k+1}\mathbf{v}_{k+1}^\top$.

With this approach, approximants and residuals are not computed explicitly nor stored, so the stopping criterion (2.21) needs to be adapted. The common choice is a stagnation-based error estimate. The algorithm stops when the new rank-1 approximation does not improve the accuracy of the approximation. Since $S_{k+1} - S_k = \mathbf{u}_{k+1}\mathbf{v}_{k+1}^\top$, the stopping criterion now reads:

$$\|\mathbf{u}_{k+1}\|_2 \|\mathbf{v}_{k+1}\|_2 \le \varepsilon_{\mathrm{ACA}} \|S_{k+1}\|_F, \tag{2.22}$$

where

$$\|S_{k+1}\|_F^2 = \|S_k\|_F^2 + 2\sum_{l=1}^{k} \mathbf{u}_{k+1}^\top \mathbf{u}_l \mathbf{v}_l^\top \mathbf{v}_{k+1} + \|\mathbf{u}_{k+1}\|_F^2 \|\mathbf{v}_{k+1}\|_F^2. \tag{2.23}$$

Table 2.2 illustrates the behavior of the partially-pivoted ACA algorithm. We report again the indices of the row and the column pivot, the value of the pivot and the relative error at each iteration of the algorithm. All these results correspond to the value $\varepsilon_{\mathrm{ACA}} = 10^{-4}$ for the threshold in the stopping criterion (2.22). We note that the numerical rank is bigger than the numerical rank computed by full-pivoted ACA with the same threshold. This phenomenon is not strange and it is linked to the choice of the pivot. In fact, in the fully-pivoted version the pivot γ_{k+1} is chosen as the maximum entry in modulus of the residual R_k, while in the partially-pivoted ACA it is chosen as the maximum entry in modulus of a computed row or column of the residual R_k. However, the two numerical ranks are of the same order.

Step	Pivot row	Pivot column	$1/\gamma_k$	Relative Error
1	1	1	$7.83 \times 10^{+2}$	$7.79 \times 10^{+0}$
2	500	24	$5.21 \times 10^{+2}$	7.42×10^{-1}
3	499	2	$3.27 \times 10^{+1}$	5.73×10^{-1}
4	498	5	$6.51 \times 10^{+0}$	4.10×10^{-1}
5	495	172	$4.32 \times 10^{+0}$	4.82×10^{-2}
6	482	10	9.99×10^{-1}	2.75×10^{-2}
7	469	61	7.58×10^{-1}	1.01×10^{-2}
8	431	3	2.55×10^{-1}	4.86×10^{-3}
9	404	500	4.44×10^{-2}	2.71×10^{-4}
10	312	296	5.19×10^{-3}	2.12×10^{-5}

Table 2.2: Partially-pivoted ACA of the off-diagonal block G_{12} of matrix G (Step 2 of Figure 2.2)

If we approximate a matrix $M \in \mathbb{C}^{m\times n}$, the update of the approximation S_k and the computation of its Frobenius norm require $\mathcal{O}(k^2(m+n))$ operations, while the computation of matrix entries requires $\mathcal{O}(k(m+n))$ operations. Thus, the partially-pivoted ACA has a reduced computational cost of $\mathcal{O}(k^2(m+n)+k(m+n))$ compared to the cost of the fully-pivoted ACA. When we consider the dense matrices resulting from the discretization of non-local operators, this complexity scales like $\mathcal{O}(k(m+n))$, because the costs for generating the matrix entries dominate the algebraic transformations in the algorithm. In fact, the sums and the multiplications involved in each iteration of partially-pivoted ACA are much faster than the evaluation of non-local operator.

Since the partially-pivoted ACA is a heuristic method, there exist counter-examples where it fails [24] and variants to improve the robustness of the method, such as the improved ACA (ACA+) and the Hybrid Cross Approximation [14, 25]. However, we show with numerical examples in the next Chapter that the basic partially-pivoted ACA is sufficient for our applications.

2.3.4 Recompression of the Admissible Blocks

After a low-rank matrix has been approximated by ACA, a further reduction of the required memory can be achieved by a suitable recompression scheme, so that the amount of memory required for the final storage is lower than the one needed for the ACA generated matrices. In fact, ACA does not provide the best numerical rank but a first approximation, that can be improved, without losing accuracy, performing a SVD recompression.

Each matrix $M \in \mathbb{C}^{m\times n}$, with $m \geq n$, can be factorized as the product of an unitary matrix $Q \in \mathbb{C}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{C}^{m\times n}$. As the bottom $m-n$ rows of R consist of zeros, it is often useful to partition Q and R in the following way:

$$M = \left[\begin{array}{cc} Q_1 & Q_2 \end{array} \right] \cdot \left[\begin{array}{c} R_1 \\ 0 \end{array} \right] = Q_1 R_1, \tag{2.24}$$

where $Q_1 \in \mathbb{C}^{m\times n}$ and $Q_2 \in \mathbb{C}^{m\times(m-n)}$ have orthogonal columns and $R_1 \in \mathbb{C}^{n\times n}$ is an upper triangular matrix. Equation (2.24) is called *reduced QR-decomposition* of M.

Starting from $M \simeq UV^H$ with $U \in \mathbb{C}^{m\times k_{\text{ACA}}}$ and $V \in \mathbb{C}^{n\times k_{\text{ACA}}}$ (where $k_{\text{ACA}} \in \mathbb{N}$ is the rank provided by the ACA), we compute the reduced QR-decomposition of matrices U and V in $\mathcal{O}(k_{\text{ACA}}^2(m+n))$ operations, such that

$$U = Q_U R_U \quad \text{and} \quad V = Q_V R_V. \tag{2.25}$$

Next, we assemble the small matrix $R_U R_V^H \in \mathbb{C}^{k_{\text{ACA}}\times k_{\text{ACA}}}$ in $\mathcal{O}(k_{\text{ACA}}^3)$ operations and compute a truncated SVD of this $k_{\text{ACA}} \times k_{\text{ACA}}$ matrix to obtain its best approximant of rank $k_{\text{SVD}} < k_{\text{ACA}}$ according to a prescribed accuracy $\varepsilon_{\text{SVD}} > 0$, i.e.

$$R_U R_V^H \simeq \Phi\Sigma\Psi^H. \tag{2.26}$$

The cost of the evaluation of this SVD is $\mathcal{O}(k_{\text{ACA}}^3)$.

We deduce the truncated SVD of M, since:

$$M \simeq UV^H = Q_U(R_U R_V^H)Q_V^H \simeq (Q_U\Phi)\Sigma(Q_V\Psi)^H. \tag{2.27}$$

The cost of the products $Q_U\Phi$ and $Q_V\Psi$ is $\mathcal{O}(k_{\text{SVD}}^2(m+n))$.

Finally, the total cost of this recompression procedure is $\mathcal{O}(k_{\text{SVD}}^2(m+n) + k_{\text{ACA}}^3)$ (see [16] for further details).

2.4 Extension of $\mathcal{H}$-matrix to BEM for Elastodynamics

The 3D elastodynamics Green's functions are linear combinations of derivatives of the Helmholtz fundamental solution, that is not asymptotically smooth in the sense of relation (2.12). On the other hand, it can be decomposed as follows

$$G(\mathbf{x},\mathbf{y};\kappa) = \frac{e^{\imath\kappa\|\mathbf{x}-\mathbf{y}\|}}{4\pi\|\mathbf{x}-\mathbf{y}\|} := f(\mathbf{x},\mathbf{y})e^{\imath\kappa\|\mathbf{x}-\mathbf{y}\|}. \tag{2.28}$$

Since $f(\mathbf{x},\mathbf{y})$ is an asymptotically smooth function, it has been proved [9] that there exist two real constants $c_1, c_2 > 0$ and a singular degree $\beta \in \mathbb{N}$ such that for any $z \in \{x_i, y_i \ : \ i = 1,2,3\}$ and any $\alpha \in \mathbb{N}$ it holds:

$$|\partial_z^\alpha G(\mathbf{x},\mathbf{y};\kappa)| \leq \alpha! c_1 (1+\kappa\|\mathbf{x}-\mathbf{y}\|)^\alpha (c_2\|\mathbf{x}-\mathbf{y}\|)^{-\alpha-\beta}, \quad \mathbf{x} \neq \mathbf{y}. \tag{2.29}$$

From this inequality, it follows that we need to distinguish two cases [17]. In the *low-frequency* regime, i.e. when κdiam(Ω) is small, the Helmholtz fundamental solution is asymptotically smooth because $\kappa\|\mathbf{x}-\mathbf{y}\|$ is small for all $\mathbf{x},\mathbf{y}\in\Omega$. But in the *high frequency* regime the methods developed for asymptotically smooth kernels could be not efficient.

2.4.1 Application of ACA to Oscillatory Kernels: Theoretical Estimates

There exist a lot of works on ACA [18, 23] for asymptotically smooth kernels but the case of Helmholtz fundamental solution has been paid less attention. We can mention the work [160] where a collocation approach is considered for Helmholtz problems up to 20 480 DOFs and where encouraging numerical results are presented.

The capability of the ACA to produce low-rank approximations can be related to the concept of *degenerate functions* [15]. 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. Noting Ω_1 and Ω_2 two domains of $\mathbb{R}^3$, we are looking for an approximation of $G:\Omega_1\times\Omega_2\rightarrow\mathbb{C}$ such that

$$G_k(\mathbf{x},\mathbf{y};\kappa)=\sum_{l=1}^{k}u_l(\mathbf{x};\kappa)v_l(\mathbf{y};\kappa)\qquad\forall\mathbf{x}\in\Omega_1\text{ and }\forall\mathbf{y}\in\Omega_2.\tag{2.30}$$

Assuming that $\mathbf{x}\in\Omega_1$ and $\mathbf{y}\in\Omega_2$ with $\|\mathbf{x}\|>\|\mathbf{y}\|$, for Helmholtz kernel it is possible to prove [72] the following error estimates

$$|G(\mathbf{x},\mathbf{y};\kappa)-G_k(\mathbf{x},\mathbf{y};\kappa)|\leq c\frac{e^{\imath\kappa\|\mathbf{x}\|}}{4\pi\|\mathbf{x}\|}k^2\left(\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|}\right)^2,\tag{2.31}$$

where $G_k(\mathbf{x},\mathbf{y};\kappa)$ is the *multipole expansion* of $G(\mathbf{x},\mathbf{y};\kappa)$, given in terms of the *Legendre polynomials* P_l $(l\geq 0)$, i.e.

$$G_k(\mathbf{x},\mathbf{y};\kappa):=\imath\pi\sum_{l=0}^{k-1}\left(l+\frac{1}{2}\right)\frac{H^{(1)}_{l+\frac{1}{2}}(\kappa\|\mathbf{x}\|)}{\|\mathbf{x}\|^{1/2}}\frac{J_{l+\frac{1}{2}}(\kappa\|\mathbf{y}\|)}{\|\mathbf{y}\|^{1/2}}P_l(\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}).\tag{2.32}$$

Here, we use the notation $\hat{\mathbf{x}}:={}^{\mathbf{x}}/_{\|\mathbf{x}\|}$ for $\mathbf{x}\neq\mathbf{0}$, $H^{(1)}_l$ denotes the l-th *Hankel function* of first kind and J_l denotes the l-th *Bessel function.*

The degeneracy becomes visible by the *addition theorem* of spherical harmonics:

$$P_l(\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}):=\frac{4\pi}{2l+1}\sum_{|m|\leq l}Y_l^m(\hat{\mathbf{x}})Y_l^{-m}(\hat{\mathbf{y}}),\tag{2.33}$$

with the spherical harmonics Y_l^m of order l and degree m $(|m|\leq l)$.

It is clear that the error estimate (2.31) deteriorates in the high-frequency regime.

Another way to get a degenerate kernel is to perform a *Taylor expansion*. Of course this method is only used to mathematically justify the degeneracy, because in practice we use the ACA after discretization.

Following the method presented in [15] (see Lemma 3.15), let $\boldsymbol{\xi}_{\Omega_2}$ be the *Chebyshev center* of the set Ω_2, i.e. the center of a ball with minimum radius ϱ_{Ω_2} containing Ω_2 (see Figure 2.10).

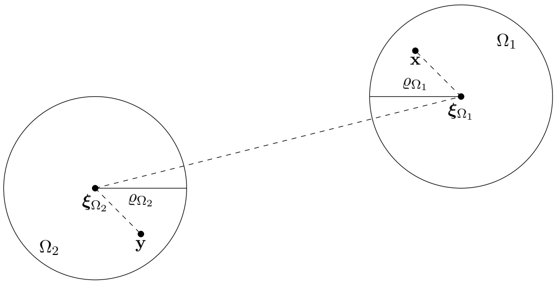


Figure 2.10: Chebyshev centers of domains Ω_1 and Ω_2.

In order to guarantee that $\|\mathbf{x}\| > \|\mathbf{y}\|$ holds for all $\mathbf{x} \in \Omega_1$ in the coordinate system with origin in $\boldsymbol{\xi}_{\Omega_2}$, we have to require

$$\eta \text{dist}(\boldsymbol{\xi}_{\Omega_2}, \Omega_1) \geq \varrho_{\Omega_2}, \quad \text{with } \eta < 1. \tag{2.34}$$

For asymptotically smooth kernels, the estimate (2.12) is used to prove exponential convergence of the Taylor series if Ω_1 and Ω_2 are sufficiently far enough away one from the other.

Since Helmholtz kernel $G : \Omega_1 \times \Omega_2 \to \mathbb{C}$ is analytic with respect to its second argument $\mathbf{y} \neq \mathbf{x}$, it is decomposed into the following Taylor expansion:

$$G(\mathbf{x}, \mathbf{y}; \kappa) = \sum_{|\alpha|<k} \frac{1}{\alpha!} \partial_y^\alpha G(\mathbf{x}, \boldsymbol{\xi}_{\Omega_2}; \kappa)(\mathbf{y} - \boldsymbol{\xi}_{\Omega_2})^\alpha + R_k(\mathbf{x}, \mathbf{y}; \kappa), \tag{2.35}$$

where $\alpha! := \alpha_1!\alpha_2!\alpha_3!$ for the multi-index $\alpha := (\alpha_1, \alpha_2, \alpha_3)$. The remainder $R_k(\mathbf{x}, \mathbf{y}; \kappa)$ converges to zero as k tends to infinity, even if the convergence rate depends on the respective positions of $\mathbf{x}$, $\mathbf{y}$ and $\boldsymbol{\xi}_{\Omega_2}$.

It is clear that the remainder satisfies

$$|R_k(\mathbf{x}, \mathbf{y}; \kappa)| \leq \sum_{|\alpha|\geq k} \frac{1}{\alpha!} |\partial_y^\alpha G(\mathbf{x}, \boldsymbol{\xi}_{\Omega_2}; \kappa)| \cdot |(\mathbf{y} - \boldsymbol{\xi}_{\Omega_2})^\alpha|. \tag{2.36}$$

From inequality (2.29) it follows

$$|R_k(\mathbf{x}, \mathbf{y}; \kappa)| \leq c_1 \sum_{|\alpha|\geq k} \frac{|\alpha|!}{\alpha!} \frac{(1 + \kappa\|\mathbf{x} - \boldsymbol{\xi}_{\Omega_2}\|)^{|\alpha|}}{(c_2\|\mathbf{x} - \boldsymbol{\xi}_{\Omega_2}\|)^{|\alpha|+\beta}} \cdot |(\mathbf{y} - \boldsymbol{\xi}_{\Omega_2})^\alpha| \tag{2.37}$$

or equivalently

$$|R_k(\mathbf{x},\mathbf{y};\kappa)| \leq \frac{c_1}{(c_2\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|)^\beta}\sum_{l=k}^{\infty}\left(\frac{\gamma}{\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|}\right)^l \sum_{|\alpha|=l}\binom{l}{\alpha}|(\mathbf{y}-\boldsymbol{\xi}_{\Omega_2})^\alpha| \tag{2.38}$$

having set

$$\gamma := \frac{1+\kappa\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|}{c_2}. \tag{2.39}$$

Due to

$$\sum_{|\alpha|=l}\binom{l}{\alpha}|\boldsymbol{\xi}|^\alpha = \left(\sum_{i=1}^{3}|\xi_i|\right)^l \leq 3^{l/2}\|\boldsymbol{\xi}\|^l, \tag{2.40}$$

we have

$$|R_k(\mathbf{x},\mathbf{y};\kappa)| \leq \frac{c_1}{(c_2\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|)^\beta}\sum_{l=k}^{\infty}\left(\sqrt{3}\gamma\frac{\|\mathbf{y}-\boldsymbol{\xi}_{\Omega_2}\|}{\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|}\right)^l. \tag{2.41}$$

By the definition of $\varrho\Omega_2$, it follows:

$$|R_k(\mathbf{x},\mathbf{y};\kappa)| \leq \frac{c_1}{(c_2\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|)^\beta}\sum_{l=k}^{\infty}\left(\sqrt{3}\gamma\frac{\varrho\Omega_2}{\mathrm{dist}(\boldsymbol{\xi}_{\Omega_2},\Omega_1)}\right)^l. \tag{2.42}$$

To assure convergence of the series, the parameter $\eta<1$ introduced in (2.34) needs to be chosen such that

$$\sqrt{3}\gamma\frac{\varrho\Omega_2}{\mathrm{dist}(\boldsymbol{\xi}_{\Omega_2},\Omega_1)} < 1. \tag{2.43}$$

Provided that this conditions holds, we have

$$|R_k(\mathbf{x},\mathbf{y};\kappa)| \leq \frac{c_1}{(c_2\|\mathbf{x}-\boldsymbol{\xi}_{\Omega_2}\|)^\beta}\frac{(\sqrt{3}\gamma\eta)^k}{1-\sqrt{3}\gamma\eta}. \tag{2.44}$$

As a result, the Taylor expansion of the Helmholtz kernel converges exponentially with convergence rate $\sqrt{3}\gamma\eta<1$ and $k\sim|\log\varepsilon|$ is the rank required to achieve an approximation with accuracy $\varepsilon>0$.

Remark that in the low-frequency case, $\gamma\sim{}^1\!/\!_{c_2}$ such that the condition is similar to asymptotically smooth kernels. Instead, in the high-frequency case the parameter η in the admissibility condition (2.13) should be adapted to the frequency to keep the required k to achieve a constant accuracy ε, by keeping $\kappa\mathrm{diam}(\Omega)$ to a constant value. In other words, if the frequency doubles, i.e. $\kappa=2\kappa_0$, the size of the admissible blocks should be divided by two to keep the product constant. An admissible block A_0 at frequency κ_0 should be decomposed into a 2×2 block matrix at frequency κ, i.e.

$$A_0 = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}. \tag{2.45}$$

From error estimates (2.31) and condition (2.44) we deduce that the low-rank approximation of A_0 is of the form

$$\begin{bmatrix} U_1 V_1^H & U_2 V_2^H \\ U_3 V_1^H & U_4 V_2^H \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix} \cdot \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}^H. \tag{2.46}$$

This means that if η is kept constant, i.e. it is not adapted to the frequency, then the rank of the admissible blocks depends linearly on the frequency.

Another approach to avoid the linear dependence of the rank with the frequency is the concept of $\mathcal{H}^2$-matrices (see [82]). However, for our applications we should stay in low/mid-frequency regime, such that the $\mathcal{H}$-matrix structure is sufficient.

2.4.2 Vectorial Partial-Pivoted Adaptive Cross Approximation

One of the specificity of this work is to consider $\mathcal{H}$-matrices and ACA in the context of vectorial problems. 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 entries in a given column. But for vectorial problems in $\mathbb{R}^d$, each pair of nodes on the mesh does not define a single entry in the complete matrix but rather a $d \times d$ sub-matrix. This happens for example for 3D elastodynamics where the Green's tensors are 3×3 matrices. Messner and Schanz claim that *a straightforward generalization of the scalar-value ACA to the matrix-valued version is not possible* [131]. They proposed to partition the complete system matrix into 3×3 block matrices and for each 3×3 block matrix they used the scalar ACA. This approach is well suited for iterative solvers but can not be adapted in the context of direct solvers.

Before describing the solution we adopted, we report two ideas tested and explain why they did not work. For this purpose we define the notion of *block row* (respectively *block column*), i.e. a row whose entries are 3×3 matrices, as opposed to *scalar row* (respectively *scalar column*), i.e. a classical row.

The first idea consists in applying the fully-pivoted ACA in a block row or column: once the first block row is generated, we look for the largest non zero entry in modulus and choose the respective scalar row and column as the first scalar row and column pivot; we then find the largest non zero entry in modulus in the selected scalar column pivot and define the next block row as the block row to which this entry belongs; we repeat this procedure until the ACA stopping criterion (2.22) is satisfied. However, this technique does not select the best scalar rows and column to subtract to the residuum R_k and to add to the approximant S_k.

In the second proposed approach, we do not consider scalar rows or columns: we select the 3×3 matrix with the largest norm in the first block row and define the first block column pivot as the block column to which this matrix belongs; the next block row pivot is the one corresponding to the 3×3 matrix with the largest norm in the previously selected block column pivot; we repeat this procedure until the ACA stopping criterion (2.22) is satisfied. The convergence of this method fails because for no norm it is guaranteed that a proper pivot can be found.

We propose instead to consider the eigenvalues of every 3×3 matrix in a block row/column in order to choose the pivot. In practice, we compute the eigenvalues of every 3×3 matrix and select the smallest one in modulus. The next pivot is given by the index of the 3×3 matrix with the largest eigenvalue among the selected ones. In this case the choice of pivot is stable and the convergence of the algorithm is ensured, because the 3×3 matrix defining each pivot is non singular and its determinant is not too small. This technique ensures a good low-rank approximation of an admissible block and it is used to obtain the results presented in the Chapter 3.

2.5 $\mathcal{H}$-matrix Based Solvers

Since the hierarchical matrix framework is an algebraic approach, it is possible to define a matrix-vector product and a matrix arithmetics (addition, multiplication and inversion) for the hierarchical matrices [73, 115]. As a consequence, once the $\mathcal{H}$-matrix representation of the influence matrix is computed, it is easy to derive an $\mathcal{H}$-matrix based solver.

2.5.1 $\mathcal{H}$-matrix Based Iterative Solver

A number of iterative procedures has been developed for the approximate solution of a linear system of equations. The oldest and simplest iterative methods are the so called stationary iterative methods (or relaxation methods), such as the Jacobi Method, the Gauss-Seidel Method and the Successive Over-Relaxation (SOR) Method. While these methods are simple to derive, implement and analyze, convergence is only guarantee for a limited class of matrices. For this reason, these methods have largely been supplanted by more sophisticated procedures, such as Krylov subspace methods. The prototypical method in this class is the Conjugate Gradient Method (CGM). Other methods are the Generalized Minimal Residual Method (GMRES) and the Biconjugate Gradient Method (BiCG). In particular, GMRES (initially proposed by Saad and Schultz [152]) is the usual choice in the case of non-symmetric systems, as the BEM systems obtained by using the collocation approach. In order to define an iterative solver, the matrix/vector product is the only operation needed.

The $\mathcal{H}$-matrix/vector product can be used to accelerate the classical matrix/vector product and, consequently, to reduce the time of each iteration of a GMRES based iterative solver, which is the most expansive task for this type of solution method.

The $\mathcal{H}$-matrix/vector product is performed hierarchically by going through the block cluster tree $T_{I\times I}$. At the leaf level, there are two possibilities for a sub-block $A_{\tau,\sigma} \in \mathbb{C}^{m\times n}$:

- if the cluster pair $(\tau,\sigma) \in \text{Leaf}^{-}(T_{I\times I})$, $A_{\tau,\sigma}$ does not admit a low-rank approximation and the standard matrix-vector product is used with cost $\mathcal{O}(mn)$;

- if the cluster pair $(\tau,\sigma) \in \text{Leaf}^{+}(T_{I\times I})$, $A_{\tau,\sigma}$ admits a low-rank approximation and its representation in outer-product form is used to perform the product. The cost of this part of the matrix-vector product is reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(k(m+n))$, where k is the numerical rank of the block.

Obviously, the accuracy of this matrix-vector product is related to the overall accuracy of the $\mathcal{H}$-matrix representation and to the accuracy of the low-rank approximations.

2.5.2 $\mathcal{H}$-matrix Arithmetic

As the efficiency of an $\mathcal{H}$-matrix based iterative solver relies on an efficient $\mathcal{H}$-matrix based matrix-vector product, the efficiency of an $\mathcal{H}$-matrix based direct solver relies on an efficient $\mathcal{H}$-matrix arithmetic. Since the $\mathcal{H}$-matrix representation relies on the low-rank approximations, it is natural to generalize the matrix/matrix addition and multiplication first in the context of low-rank matrices and, then, extend these operations to the $\mathcal{H}$-matrices.

2.5.2.1 Algebra of Low-rank Matrices

Summation. Let us consider two matrices $M_1 = U_1V_1^H \in \mathbb{C}^{m\times n}$ and $M_2 = U_2V_2^H \in \mathbb{C}^{m\times n}$ with low-ranks $k_1, k_2 \in \mathbb{N}$ respectively. The sum of these matrices is given by

$$\tilde{M} = M_1 + M_2 = U_1V_1^H + U_2V_2^H = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix} = UV^H \in \mathbb{C}^{m\times n} \tag{2.47}$$

Without any manipulation the rank of $\tilde{M}$ is $k_1 + k_2$. However, further compression can be obtained by using the following recompression procedure:

1. calculate a reduced QR-decomposition of $U = Q_{\text{U}}R_{\text{U}}$, with $Q_{\text{U}} \in \mathbb{C}^{m\times(k_1+k_2)}$ and $R_{\text{U}} \in \mathbb{C}^{(k_1+k_2)\times(k_1+k_2)}$;

2. calculate a reduced QR-decomposition of $V = Q_{\text{V}}R_{\text{V}}$, with $Q_{\text{V}} \in \mathbb{C}^{n\times(k_1+k_2)}$ and $R_{\text{V}} \in \mathbb{C}^{(k_1+k_2)\times(k_1+k_2)}$;

3. calculate a truncated SVD of $R_{\text{U}}R_{\text{V}}^{\top} = \tilde{\Phi}\tilde{\Lambda}\tilde{\Psi}^H$, in order to obtain its best approximant of rank $k^{'} < k_1 + k_2$ according to a given accuracy $\varepsilon > 0$;

4. set $\tilde{U} := Q_{\text{U}}\tilde{\Phi}\tilde{\Lambda} \in \mathbb{C}^{m\times k^{'}}$ and $\tilde{V} := Q_{\text{V}}R_{\text{V}}\tilde{\Psi} \in \mathbb{C}^{n\times k^{'}}$.

It results that the recompressed addition between low-rank matrices is not an exact operation but a *formatted* operation, i.e. it is based on the given accuracy $\varepsilon > 0$, and its total cost is $\mathcal{O}((k_1 + k_2)(k_1 + k_2 + k^{'})(m + n) + (k_1 + k_2)^3)$. This technique is a good compromise between a controlled loss of precision and memory savings, because the rank is reduced.

Multiplication. The multiplication of a low-rank matrix $M = UV^H \in \mathbb{C}^{m\times n}$, with $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$, by an arbitrary matrix R from the left (in this case $R \in \mathbb{C}^{s\times m}$) or the right (in this case $R \in \mathbb{C}^{n\times s}$) yields again to a low-rank matrix

$$\begin{array}{lllllllll} M \cdot R & = & (UV^H) \cdot R & = & U \cdot (R^H V)^H & := & U\tilde{V}^H & \text{with} & U \in \mathbb{C}^{m\times k}, \tilde{V} \in \mathbb{C}^{s\times k} \\ R \cdot M & = & R \cdot (UV^H) & = & (RU) \cdot V^H & := & \tilde{U}V^H & \text{with} & \tilde{U} \in \mathbb{C}^{s\times k}, V \in \mathbb{C}^{n\times k}. \end{array} \tag{2.48}$$

To calculate this product, we perform the matrix-vector multiplication $R^H V_i$ for the k columns of V with the conjugate transposed of R, or RU_i for the k columns of U with the matrix R. The total cost of this procedure is $\mathcal{O}(kns)$ in the case of the multiplication from right, and $\mathcal{O}(knm)$ in the case of the multiplication from left.

2.5.2.2 Algebra of $\mathcal{H}$-Matrices

After these considerations about the low-rank matrices, we explain how to multiply two blocks $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ of an $\mathcal{H}$-matrix and add the result to another block $A_{\tau,\sigma}$, i.e. $A_{\tau,\sigma} \leftarrow A_{\tau,\sigma} + A_{\tau,\zeta} \cdot A_{\zeta,\sigma} := A_{\tau,\sigma} + \tilde{A}_{\tau,\sigma}$. First of all, we need to assure the compatibility between the blocks, since they can be: an $\mathcal{H}$-matrix block, a full matrix block or a low-rank block. For this reason, we have to distinguish 27 possible cases, grouped in 3 scenarios:

1. $A_{\tau,\sigma}$ is a full matrix. This is the simplest case because the conversion of the matrix $\tilde{A}_{\tau,\sigma}$ into a full matrix is obtained using the standard product between matrices. For sake of clarity, we treat separately the nine sub-cases.

   1.1 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both full matrices.

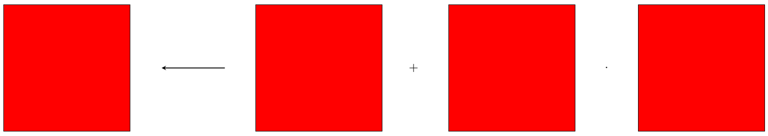


Figure 2.11: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a full matrix.

   The matrix $\tilde{A}_{\tau,\sigma}$ is computed and added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

   1.2 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V^H_{\zeta,\sigma}$ is a low-rank matrix.

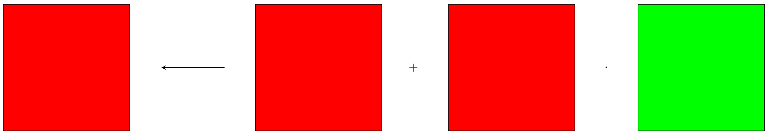


Figure 2.12: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V^H_{\zeta,\sigma}) = (A_{\tau,\zeta}U_{\zeta,\sigma})V^H_{\zeta,\sigma} := \tilde{U}_{\zeta,\sigma}V^H_{\zeta,\sigma}.$$

This matrix is converted into a full matrix performing the product between $\tilde{U}_{\zeta,\sigma}$ and $V^H_{\zeta,\sigma}$, and then it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.3 $A_{\tau,\zeta} = U_{\tau,\zeta}V^H_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

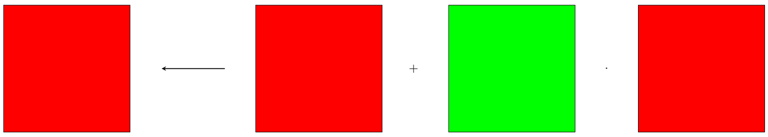


Figure 2.13: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V^H_{\tau,\zeta})A_{\zeta,\sigma} = U_{\tau,\zeta}(V^H_{\tau,\zeta}A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}^H_{\tau,\zeta}.$$

This matrix is converted into a full matrix performing the product between $U_{\tau,\zeta}$ and $\tilde{V}^H_{\tau,\zeta}$, and then it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.4 $A_{\tau,\zeta} = U_{\tau,\zeta}V^H_{\tau,\zeta}$ and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V^H_{\zeta,\sigma}$ are both low-rank matrices.

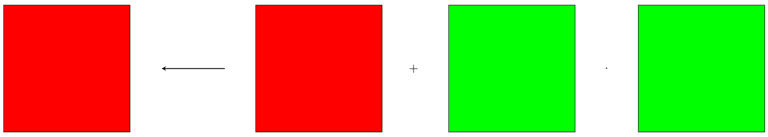


Figure 2.14: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (U_{\tau,\zeta}(V_{\tau,\zeta}^H U_{\zeta,\sigma}))V_{\zeta,\sigma}^H := (U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H)V_{\zeta,\sigma}^H = \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H.$$

This matrix is converted into a full matrix performing the product between $\tilde{U}_{\zeta,\sigma}$ and $V_{\zeta,\sigma}^H$, and then it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.5 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

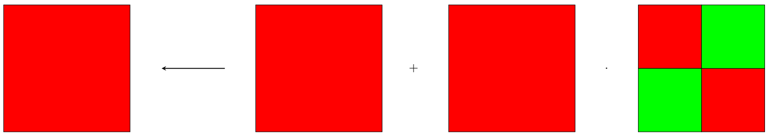


Figure 2.15: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

In order to compute the matrix $\tilde{A}_{\tau,\sigma}$, we use the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ to each row of $A_{\tau,\zeta}$. In this way, $\tilde{A}_{\tau,\sigma}$ is a full matrix and it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.6 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

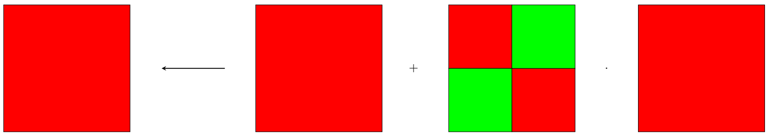


Figure 2.16: $A_{\tau,\sigma}$ is a full matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

In order to compute the matrix $\tilde{A}_{\tau,\sigma}$, we use the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ to each column of $A_{\zeta,\sigma}$. In this way, $\tilde{A}_{\tau,\sigma}$ is a full matrix and it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.7 $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

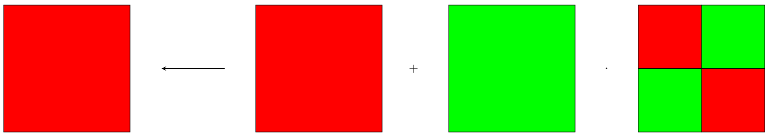


Figure 2.17: $A_{\tau,\sigma}$ is full matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form, i.e

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)A_{\zeta,\sigma} = U_{\tau,\zeta}(V_{\tau,\zeta}^H A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ to each row of $V_{\tau,\zeta}^H$. $\tilde{A}_{\tau,\sigma}$ is converted into a full matrix performing the product between $U_{\tau,\zeta}$ and $\tilde{V}_{\tau,\zeta}^H$, and then it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.8 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$ is a low-rank matrix.

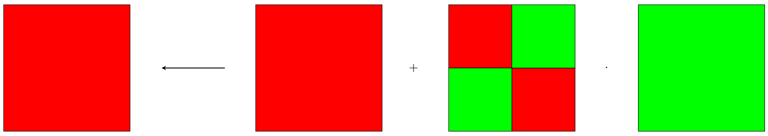


Figure 2.18: $A_{\tau,\sigma}$ is full matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form, i.e.

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (A_{\tau,\zeta}U_{\zeta,\sigma})V_{\zeta,\sigma}^H := \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ to each column of $U_{\zeta,\sigma}$. $\tilde{A}_{\tau,\sigma}$ is converted into a full matrix performing the product between $\tilde{U}_{\zeta,\sigma}$ and $V_{\zeta,\sigma}^H$, and then it is added to $A_{\tau,\sigma}$ using the standard matrix arithmetic.

1.9 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both $\mathcal{H}$-matrices.

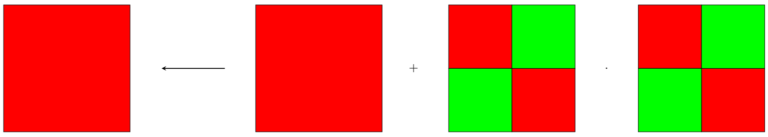


Figure 2.19: $A_{\tau,\sigma}$ is full matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

We convert the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ *into a low-rank matrix*, using the technique explained in Subsection 2.5.2.3, and we follow the procedure described in sub-case 1.7.

2. $A_{\tau,\sigma}$ is a low-rank matrix. In this eventuality, we compute $\tilde{A}_{\tau,\sigma} := A_{\tau,\zeta} \cdot A_{\zeta,\sigma}$ in the outer-product form and we use the formatted addition between low-rank matrices to add $\tilde{A}_{\tau,\sigma}$ to $A_{\tau,\sigma}$. The result is not exact but approximated according to a given accuracy $\varepsilon > 0$. For sake of clarity, we treat separately the nine sub-cases.

2.1 $A_{\tau,\zeta} = U_{\tau,\zeta}V^H_{\tau,\zeta}$ and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V^H_{\zeta,\sigma}$ are both low-rank matrices.

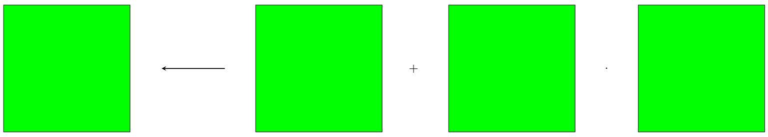


Figure 2.20: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the desired outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V^H_{\tau,\zeta})(U_{\zeta,\sigma}V^H_{\zeta,\sigma}) = (U_{\tau,\zeta}(V^H_{\tau,\zeta}U_{\zeta,\sigma}))V^H_{\zeta,\sigma} := (U_{\tau,\zeta}\tilde{V}^H_{\tau,\zeta})V^H_{\zeta,\sigma} = \tilde{U}_{\zeta,\sigma}V^H_{\zeta,\sigma}.$$

This matrix is added to $A_{\tau,\sigma} = U_{\tau,\sigma}V^H_{\tau,\sigma}$ using the formatted addition between low-rank matrices.

2.2 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V^H_{\zeta,\sigma}$ is a low-rank matrix.

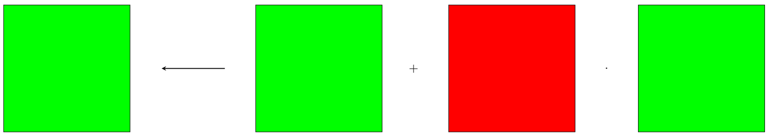


Figure 2.21: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the desired outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V^H_{\zeta,\sigma}) = (A_{\tau,\zeta}U_{\zeta,\sigma})V^H_{\zeta,\sigma} := \tilde{U}_{\zeta,\sigma}V^H_{\zeta,\sigma}.$$

This matrix is added to $A_{\tau,\sigma}$ using the formatted addition between low-rank matrices.

2.3 $A_{\tau,\zeta} = U_{\tau,\zeta}V^H_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

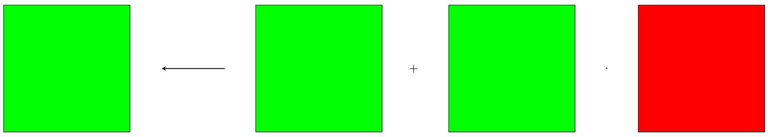


Figure 2.22: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the desired outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)A_{\zeta,\sigma} = U_{\tau,\zeta}(V_{\tau,\zeta}^H A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H.$$

This matrix is added to $A_{\tau,\sigma}$ using the formatted addition between low-rank matrices.

2.4 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$ is a low-rank matrix.

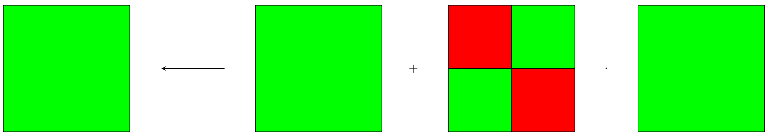


Figure 2.23: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the desired outer-product form, i.e.

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (A_{\tau,\zeta}U_{\zeta,\sigma})V_{\zeta,\sigma}^H := \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ to each column of $U_{\zeta,\sigma}$. $\tilde{A}_{\tau,\sigma}$ is added to $A_{\tau,\sigma}$ using the formatted addition between low-rank matrices.

2.5 $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

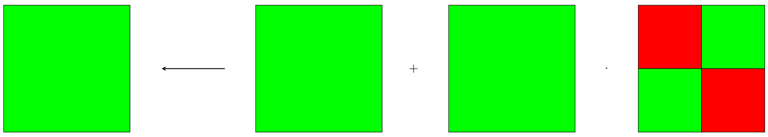


Figure 2.24: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

We observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the desired outer-product form, i.e

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)A_{\zeta,\sigma} = U_{\tau,\zeta}(V_{\tau,\zeta}^H A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ to each row of $V_{\tau,\zeta}^H$. $\tilde{A}_{\tau,\sigma}$ is added to $A_{\tau,\sigma}$ using the formatted addition between low-rank matrices.

2.6 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both full matrices.

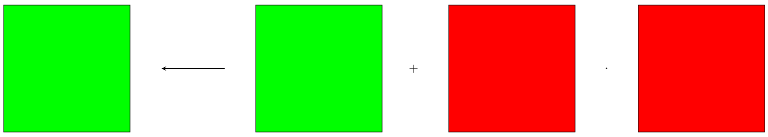


Figure 2.25: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a full matrix.

First, we convert the full matrix $A_{\tau,\zeta}$ in the outer-product form, i.e. $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$, using the truncated SVD according to a given accuracy $\varepsilon > 0$. Then, we follow the procedure described in sub-case 2.3.

2.7 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

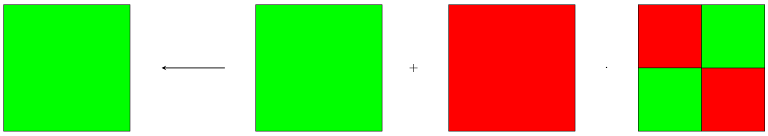


Figure 2.26: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

First, we convert the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ in the outer-product form, i.e. $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$. Then, we follow the procedure described in sub-case 2.2.

2.8 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

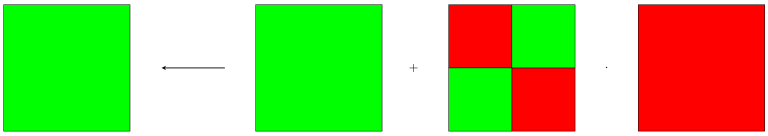


Figure 2.27: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

First, we convert the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ in the outer-product form, i.e. $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$. Then, we follow the procedure described in sub-case 2.3.

2.9 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both $\mathcal{H}$-matrices.

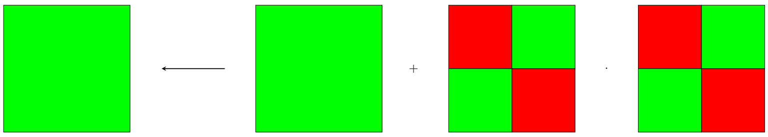


Figure 2.28: $A_{\tau,\sigma}$ is a low-rank matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a $\mathcal{H}$-matrix.

First, we convert the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ in the outer-product form, i.e. $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$. Then, we follow the procedure described in sub-case 2.4.

3. $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix. *The principal difficulty of this case is the conversion of* $\tilde{A}_{\tau,\sigma}$ *into an* $\mathcal{H}$-matrix *with the same structure than* $A_{\tau,\sigma}$*.* This operation is very delicate, because it introduces a further degree of approximation in the $\mathcal{H}$-matrix representation. For sake of clarity, we treat separately the nine sub-cases.

   3.1 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both $\mathcal{H}$-matrices.

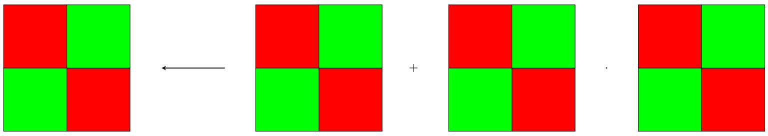


Figure 2.29: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a $\mathcal{H}$-matrix.

The multiplication and the addition are recursively done in the sub-blocks.

   3.2 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$ is a low-rank matrix.

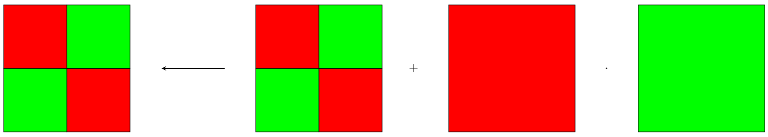


Figure 2.30: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

First, we observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (A_{\tau,\zeta}U_{\zeta,\sigma})V_{\zeta,\sigma}^H := \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H.$$

Then, we convert $\tilde{A}_{\tau,\sigma}$ into an $\mathcal{H}$-matrix with the same structure than $A_{\tau,\sigma}$ (using the technique explained in Subsection 2.5.2.3) and, finally, we follow the procedure described in sub-case 3.1.

3.3 $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

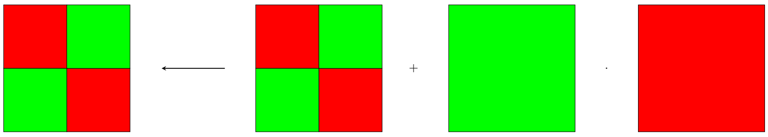


Figure 2.31: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a full matrix.

First, we observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)A_{\zeta,\sigma} = U_{\tau,\zeta}(V_{\tau,\zeta}^H A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H.$$

Then, we convert $\tilde{A}_{\tau,\sigma}$ into an $\mathcal{H}$-matrix with the same structure than $A_{\tau,\sigma}$ and, finally, we follow the procedure described in sub-case 3.1.

3.4 $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$ and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$ are both low-rank matrices.

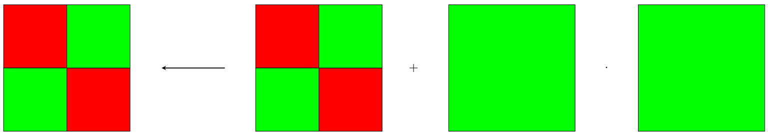


Figure 2.32: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

First, we observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form in a fast way, thanks to the algebra of low-rank matrices, since

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (U_{\tau,\zeta}(V_{\tau,\zeta}^H U_{\zeta,\sigma}))V_{\zeta,\sigma}^H := (U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H)V_{\zeta,\sigma}^H = \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H.$$

Then, we convert $\tilde{A}_{\tau,\sigma}$ into an $\mathcal{H}$-matrix with the same structure than $A_{\tau,\sigma}$ and, finally, we follow the procedure described in sub-case 3.1.

3.5 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$ is a low-rank matrix.

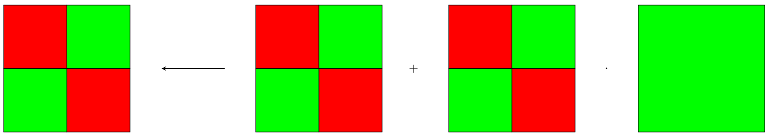


Figure 2.33: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a low-rank matrix.

First, we observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form, i.e.

$$\tilde{A}_{\tau,\sigma} = A_{\tau,\zeta}(U_{\zeta,\sigma}V_{\zeta,\sigma}^H) = (A_{\tau,\zeta}U_{\zeta,\sigma})V_{\zeta,\sigma}^H := \tilde{U}_{\zeta,\sigma}V_{\zeta,\sigma}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\tau,\zeta}$ to each column of $U_{\zeta,\sigma}$. Then, we convert $\tilde{A}_{\tau,\sigma}$ into an $\mathcal{H}$-matrix with the same structure than $A_{\tau,\sigma}$ and, finally, we follow the procedure described in sub-case 3.1.

3.6 $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

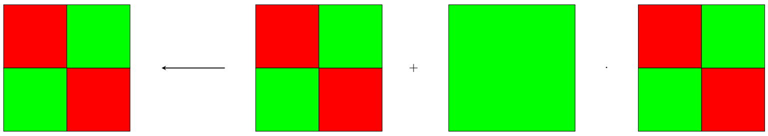


Figure 2.34: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a low-rank matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

First, we observe that the matrix $\tilde{A}_{\tau,\sigma}$ is obtained in the outer-product form, i.e

$$\tilde{A}_{\tau,\sigma} = (U_{\tau,\zeta}V_{\tau,\zeta}^H)A_{\zeta,\sigma} = U_{\tau,\zeta}(V_{\tau,\zeta}^H A_{\zeta,\sigma}) := U_{\tau,\zeta}\tilde{V}_{\tau,\zeta}^H,$$

using the fast $\mathcal{H}$-matrix/vector product to apply the $\mathcal{H}$-matrix $A_{\zeta,\sigma}$ to each row of $V_{\tau,\zeta}^H$. Then, we convert $\tilde{A}_{\tau,\sigma}$ into an $\mathcal{H}$-matrix with the same structure than $A_{\tau,\sigma}$ and, finally, we follow the procedure described in sub-case 3.1.

3.7 $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

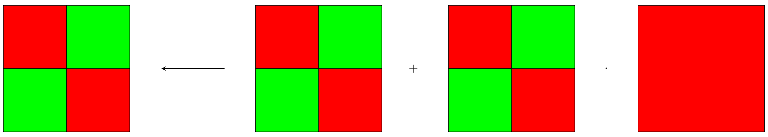


Figure 2.35: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is an $\mathcal{H}$-matrix and $A_{\zeta,\sigma}$ is a full matrix.

First, we convert the full matrix $A_{\zeta,\sigma}$ in the outer-product form, i.e. $A_{\zeta,\sigma} = U_{\zeta,\sigma}V_{\zeta,\sigma}^H$, using the truncated SVD according to a given accuracy $\varepsilon > 0$. Then, we follow the procedure described in sub-case 3.5.

3.8 $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

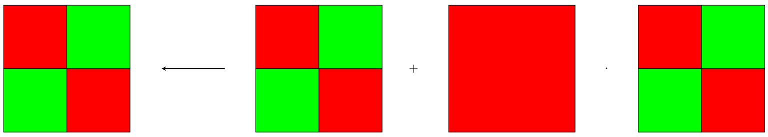


Figure 2.36: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is an $\mathcal{H}$-matrix.

First, we convert the full matrix $A_{\tau,\zeta}$ in the outer-product form, i.e. $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$, using the truncated SVD according to a given accuracy $\varepsilon > 0$. Then, we follow the procedure described in sub-case 3.6.

3.9 $A_{\tau,\zeta}$ and $A_{\zeta,\sigma}$ are both full matrices.

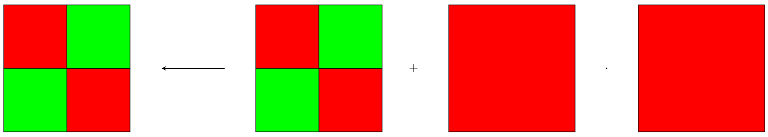


Figure 2.37: $A_{\tau,\sigma}$ is an $\mathcal{H}$-matrix, $A_{\tau,\zeta}$ is a full matrix and $A_{\zeta,\sigma}$ is a full matrix.

First, we convert the full matrix $A_{\tau,\zeta}$ in the outer-product form, i.e. $A_{\tau,\zeta} = U_{\tau,\zeta}V_{\tau,\zeta}^H$, using the truncated SVD according to a given accuracy $\varepsilon > 0$. Then, we follow the procedure described in sub-case 3.3.

2.5.2.3 Conversion of the Matrix Format

In the algebra of the $\mathcal{H}$-matrices a very important role is played by the merging of a $\mathcal{H}$-matrix into a low-rank matrix. We remember that the truncated SVD allows to convert a full matrix into a low-rank matrix. Then, to illustrate how it is possible to merge an $\mathcal{H}$-matrix into a low-rank matrix, we explain this procedure considering four admissible sub-blocks and observing that

$$\begin{bmatrix} U_1V_1^H & U_2V_2^H \\ U_3V_3^H & U_4V_4^H \end{bmatrix} = E_1 + E_2 + E_3 + E_4, \tag{2.49}$$

with

$$E_1 = \begin{bmatrix} U_1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ 0 \end{bmatrix}^H, \; E_2 = \begin{bmatrix} U_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ V_2 \end{bmatrix}^H, \; E_3 = \begin{bmatrix} 0 \\ U_3 \end{bmatrix} \cdot \begin{bmatrix} V_3 \\ 0 \end{bmatrix}^H \text{ and } E_4 = \begin{bmatrix} 0 \\ U_4 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ V_4 \end{bmatrix}^H. \tag{2.50}$$

We first focus our attention on the sum

$$E_1 + E_2 = \left[\begin{array}{c|c} U_1 \in \mathbb{C}^{m_1 \times k_1} & U_2 \in \mathbb{C}^{m_1 \times k_2} \\ \hline 0 \in \mathbb{C}^{m_2 \times k_1} & 0 \in \mathbb{C}^{m_2 \times k_2} \end{array}\right] \cdot \left[\begin{array}{c|c} V_1 \in \mathbb{C}^{n_1 \times k_1} & 0 \in \mathbb{C}^{n_1 \times k_2} \\ \hline 0 \in \mathbb{C}^{n_2 \times k_1} & V_2 \in \mathbb{C}^{n_2 \times k_2} \end{array}\right]^H . \quad (2.51)$$

The rank of $E_1 + E_2$ is $k_1 + k_2$. Then, we apply the following algorithm to compress the result

1. calculate the reduced QR-decomposition $\tilde{U} := [U_1|U_2] = Q_{\tilde{U}} R_{\tilde{U}}$, with $Q_{\tilde{U}} \in \mathbb{C}^{m_1 \times (k_1+k_2)}$ and $R_{\tilde{U}} \in \mathbb{C}^{(k_1+k_2) \times (k_1+k_2)}$ (in the following we consider $R_{\tilde{U}} := \left[R^1_{\tilde{U}} | R^2_{\tilde{U}}\right]$ with $R^1_{\tilde{U}} \in \mathbb{C}^{(k_1+k_2) \times k_1}$ and $R^2_{\tilde{U}} \in \mathbb{C}^{(k_1+k_2) \times k_2}$);

2. calculate the reduced QR-decomposition of $V_1 = Q^1_V R^1_V$, with $Q^1_V \in \mathbb{C}^{n_1 \times k_1}$ and $R^1_V \in \mathbb{C}^{k_1 \times k_1}$;

3. calculate the reduced QR-decomposition of $V_2 = Q^2_V R^2_V$, with $Q^2_V \in \mathbb{C}^{n_2 \times k_2}$ and $R^2_V \in \mathbb{C}^{k_2 \times k_2}$;

4. without assembling the block diagonal matrix

$$R_{\tilde{V}} := \left[\begin{array}{c|c} R^1_V & 0 \\ \hline 0 & R^2_V \end{array}\right] \in \mathbb{C}^{(k_1+k_2) \times (k_1+k_2)},$$

calculate the product

$$R_{\tilde{U}} R^\top_{\tilde{V}} = \left[\begin{array}{c|c} R^1_{\tilde{U}} & R^2_{\tilde{U}} \end{array}\right] \cdot \left[\begin{array}{c|c} R^{1\top}_V & 0 \\ \hline 0 & R^{2\top}_V \end{array}\right] = \left[\begin{array}{c|c} R^1_{\tilde{U}} R^{1\top}_V & R^2_{\tilde{U}} R^{2\top}_V \end{array}\right]$$

5. calculate the truncated SVD $\tilde{\Phi}\tilde{\Lambda}\tilde{\Psi}^H$ of $R_{\tilde{U}} R^\top_{\tilde{V}}$, in order to obtain its best approximation of rank $k^{'} < k_1 + k_2$ according to a given accuracy $\varepsilon > 0$;

6. finally, set

$$U := \left[\begin{array}{c} Q_{\tilde{U}} \\ \hline 0 \end{array}\right] \tilde{\Phi}\tilde{\Lambda} \qquad \text{and} \qquad V := \left[\begin{array}{c|c} Q^1_V & 0 \\ \hline 0 & Q^2_V \end{array}\right] \tilde{\Psi}.$$

In this way, we have a low-rank representation of the matrix $E_1 + E_2$ of rank $k^{'} < k_1 + k_2$. Of course the algorithm described can be applied, with some small modifications, to compute all the sums present in Equation (2.49).

To complete the description of the $\mathcal{H}$-matrix arithmetic, we have to describe how to convert a low-rank matrix $M = UV^H$ into an $\mathcal{H}$-matrix with four admissible blocks. This procedure is very simple if we consider matrices U and V as 2×1 block-matrices, since:

$$UV^H = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}^H = \begin{bmatrix} U_1 V_1^H & U_1 V_2^H \\ U_2 V_1^H & U_2 V_2^H \end{bmatrix}. \quad (2.52)$$

2.5.3 $\mathcal{H}$-matrix Based Direct Solver

One of the advantages of the $\mathcal{H}$-matrix representation is the possibility to derive a fast direct solver. As we know from the standard linear algebra, a LU-factorization of the system matrix is to be preferred to its direct inversion. Briefly, we recall that for all non-singular matrices $M \in \mathbb{C}^{m\times m}$ there is a permutation matrix $P \in \mathbb{C}^{m\times m}$ and a lower $L \in \mathbb{C}^{m\times m}$ and a upper $U \in \mathbb{C}^{m\times m}$ triangular matrices, such that $PM = LU$. Moreover, L can be taken to have ones on its diagonal. This type of factorization is called LU factorization with partial pivoting and can be viewed as the matrix form of the Gaussian elimination. Ignoring pivoting, we also remember that, if M has a LU-factorization $M = LU$, for any partition of the matrix of the form

$$M := \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \tag{2.53}$$

there are corresponding decompositions of the matrices L and U, i.e.

$$L := \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \quad \text{and} \quad U := \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}, \tag{2.54}$$

where the diagonal blocks L_{ii} and U_{ii}, $i = 1, 2$, are lower and upper triangular respectively. Moreover, we have:

$$\begin{aligned} M_{11} &= L_{11}U_{11} \\ M_{12} &= L_{21}U_{11} \\ M_{21} &= L_{12}U_{22} \\ M_{22} &= L_{21}U_{12} + L_{22}U_{21}. \end{aligned} \tag{2.55}$$

Thus, $L_{11}U_{11}$ is a LU factorization of M_{11}.

Let A be the matrix of interest. The proposed $\mathcal{H}$-matrix based direct solution strategy has three components:

- compute an $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$ of A;
- compute an $\mathcal{H}$-matrix based LU-factorization in order to obtain two factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ in $\mathcal{H}$-matrix format, such that $A_{\mathcal{H}} \simeq L_{\mathcal{H}}U_{\mathcal{H}}$;
- use an $\mathcal{H}$-matrix based backward and forward substitution method to solve the lower triangular system $L_{\mathcal{H}}\mathbf{w} = \mathbf{b}$ for $\mathbf{w}$ and the upper triangular system $U_{\mathcal{H}}\mathbf{z} = \mathbf{w}$ for $\mathbf{z}$, resulting from:

$$\mathbf{b} = A\mathbf{z} \simeq A_{\mathcal{H}}\mathbf{z} \simeq L_{\mathcal{H}}U_{\mathcal{H}}\mathbf{z} = L_{\mathcal{H}}\mathbf{w} \quad \text{with } \mathbf{w} = U_{\mathcal{H}}\mathbf{z}. \tag{2.56}$$

2.5.3.1 $\mathcal{H}$-LU Factorization

In the light of (2.54), to extend the LU-factorization to the $\mathcal{H}$-matrices, we make the assumption that the factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ have the same $\mathcal{H}$-matrix structure than $A_{\mathcal{H}}$

(Figure 2.38).

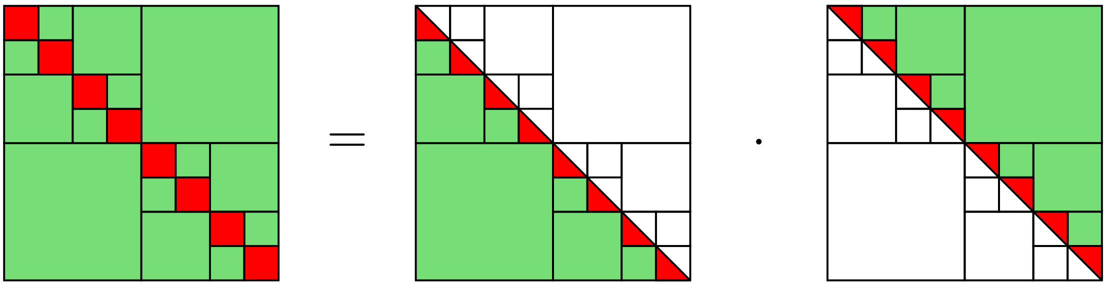


Figure 2.38: Assumption on the structure of the $\mathcal{H}$-LU factorization: the factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ have the same $\mathcal{H}$-matrix structure than $A_{\mathcal{H}}$.

Provided all diagonal leaf blocks are non-singular full matrices, we use an $\mathcal{H}$-matrix block $A_{\tau,\sigma}$ to illustrate the $\mathcal{H}$-LU factorization process. If the cluster pair $(\tau, \sigma) \notin \text{Leaf}(T_{I\times I})$, the corresponding block $A_{\tau,\sigma}$ can be subdivided into four sub-blocks, such that

$$A_{\tau,\sigma} = \begin{bmatrix} A_{\tau_1,\sigma_1} & A_{\tau_1,\sigma_2} \\ A_{\tau_2,\sigma_1} & A_{\tau_2,\sigma_2} \end{bmatrix} = \begin{bmatrix} L_{\tau_1,\sigma_1} & 0 \\ L_{\tau_2,\sigma_1} & L_{\tau_2,\sigma_2} \end{bmatrix} \cdot \begin{bmatrix} U_{\tau_1,\sigma_1} & U_{\tau_1,\sigma_2} \\ 0 & U_{\tau_2,\sigma_2} \end{bmatrix}. \tag{2.57}$$

where τ_1 and τ_2 are the children of τ, while σ_1 and σ_2 are the children of σ in the cluster tree T_I. For such a block matrix, the LU factorization is decomposed into four steps:

1. recursively call the $\mathcal{H}$-LU decomposition until the cluster pair $(\tau_1, \sigma_1) \in \text{Leaf}^{-}(T_{I\times I})$, to get the factors L_{τ_1,σ_1} and U_{τ_1,σ_1} such that $A_{\tau_1,\sigma_1} = L_{\tau_1,\sigma_1} U_{\tau_1,\sigma_1}$ (classical partial pivoted LU-factorization of full matrices);

2. compute U_{τ_1,σ_2} solving the lower triangular matrix-equation $A_{\tau_1,\sigma_2} = L_{\tau_1,\sigma_1} U_{\tau_1,\sigma_2}$, with L_{τ_1,σ_1} obtained in the first step;

3. compute L_{τ_2,σ_1} solving the upper triangular matrix-equation $A_{\tau_2,\sigma_1} = L_{\tau_2,\sigma_1} U_{\tau_1,\sigma_1}$, with U_{τ_1,σ_1} obtained in the first step;

4. recursively call $\mathcal{H}$-LU decomposition until the cluster pair $(\tau_2, \sigma_2) \in \text{Leaf}^{-}(T_{I\times I})$, to get the factors L_{τ_2,σ_2} and U_{τ_2,σ_2} such that $A_{\tau_2,\sigma_2} - L_{\tau_2,\sigma_1} U_{\tau_1,\sigma_2} = L_{\tau_1,\sigma_1} U_{\tau_1,\sigma_1}$ (classical pivoted LU-factorization of full matrices, whose existence is assured by the hypothesis of non-singularity of the full diagonal leaf blocks).

During the LU-factorization process, a further degree of compression is introduced by addition and multiplication in step 4. These operations are not between classical matrices but between $\mathcal{H}$-matrices and, consequently, they induce various modifications on the $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$. In particular, the conversion of the structure of a block requires operations that are formatted, i.e based on a prescribed accuracy $\varepsilon_{\text{LU}} > 0$.

2.5.3.2 Solving an $\mathcal{H}$-matrix Triangular System for a Matrix

In step 2 and step 3 of the $\mathcal{H}$-LU factorization, a method to solve a triangular $\mathcal{H}$-matrix system for a matrix is needed. Such procedure can be derived in a similar fashion as $\mathcal{H}$-LU decomposition, using the hierarchical block structure. Solving the triangular system $LX = B$ for a right-hand side matrix B and a lower triangular matrix L in the $\mathcal{H}$-matrix format is done recursively. If the cluster pair $(\tau, \sigma) \notin \text{Leaf}(T_{I\times I})$, the corresponding block $L_{\tau,\sigma}$ can be subdivided into four sub-blocks

$$L_{\tau,\sigma} = \begin{bmatrix} L_{\tau_1,\sigma_1} & 0 \\ L_{\tau_2,\sigma_1} & L_{\tau_2,\sigma_2} \end{bmatrix}, \tag{2.58}$$

inducing a similar subdivision on the matrix X and B, i.e.

$$X_{\tau,\sigma} = \begin{bmatrix} X_{\tau_1,\sigma_1} & X_{\tau_1,\sigma_2} \\ X_{\tau_2,\sigma_1} & X_{\tau_2,\sigma_2} \end{bmatrix} \quad \text{and} \quad B_{\tau,\sigma} = \begin{bmatrix} B_{\tau_1,\sigma_1} & B_{\tau_1,\sigma_2} \\ B_{\tau_2,\sigma_1} & B_{\tau_2,\sigma_2} \end{bmatrix}. \tag{2.59}$$

For such sub-block matrices, we have to solve the two triangular systems $L_{\tau_1,\sigma_1}X_{\tau_1,\sigma_1} = B_{\tau_1,\sigma_1}$ and $L_{\tau_1,\sigma_1}X_{\tau_1,\sigma_2} = B_{\tau_1,\sigma_2}$, which yield X_{τ_1,σ_1} and X_{τ_1,σ_2}, and afterwards the two triangular systems $L_{\tau_2,\sigma_2}X_{\tau_2,\sigma_1} = B_{\tau_2,\sigma_1} - L_{\tau_2,\sigma_1}X_{\tau_1,\sigma_1}$ and $L_{\tau_2,\sigma_2}X_{\tau_2,\sigma_2} = B_{\tau_2,\sigma_2} - L_{\tau_2,\sigma_1}X_{\tau_1,\sigma_2}$, which yield X_{τ_2,σ_1} and X_{τ_2,σ_2}.

Of course, all the additions and multiplications performed during this process, are between $\mathcal{H}$-matrices and, consequently, have to be computed according to the prescribed accuracy ε_{LU}.

2.5.3.3 $\mathcal{H}$-matrix based Backward Substitution Method

Once the $\mathcal{H}$-matrix based LU factorization is computed, solving the triangular system $L_{\mathcal{H}}\mathbf{w} = \mathbf{b}$ for the right-hand side $\mathbf{b}$ and the lower triangular matrix $L_{\mathcal{H}}$ in the $\mathcal{H}$-matrix format is done recursively. If the cluster pair $(\tau, \sigma) \notin \text{Leaf}(T_{I\times I})$, the corresponding block $L_{\tau,\sigma}$ can be subdivided into four sub-blocks, such that

$$\begin{bmatrix} L_{\tau_1,\sigma_1} & 0 \\ L_{\tau_2,\sigma_1} & L_{\tau_2,\sigma_2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \tag{2.60}$$

For such sub-block matrix, we have to solve the system $L_{\tau_1,\sigma_1}\mathbf{w}_1 = \mathbf{b}_1$, which yields $\mathbf{w}_1$ and afterwards the system $L_{\tau_2,\sigma_2}\mathbf{w}_2 = \mathbf{b}_2 - L_{\tau_2,\sigma_1}\mathbf{w}_1$, which yields $\mathbf{w}_2$.

A similar procedure is applied to solve the upper triangular system $U_{\mathcal{H}}\mathbf{z} = \mathbf{w}$.

2.6 Conclusions

In this second Chapter, we have presented the $\mathcal{H}$-matrix technique to provide a data-sparse representation of dense matrices. The construction of an $\mathcal{H}$-matrix requires several steps. First, a hierarchical cluster tree for the set of the degrees of freedom is derived. Its nodes should correspond in the matrix to positions which are close in the physical space. At the root level, the cluster consists of the complete set. Each cluster is recursively partitioned into two sons. Second, admissible and non-admissible cluster pairs are identified (block cluster tree), in order to understand which blocks of the matrix have to be represented as either low-rank or dense matrices.

The crucial point for the application of $\mathcal{H}$-matrices is an efficient and easy-to-use rank revealing algorithm. Among the several approaches developed, the Adaptive Cross Approximation (ACA) represents an alternative to the efficient but expensive strategy based on the Singular Value Decomposition (SVD). Unfortunately, the ACA does not provide the best numerical low-rank but a good first approximation, that can be improved without losing accuracy performing a SVD recompression. There exist a lot of works on ACA for asymptotically smooth kernels, we have investigated the capability of the ACA to produce low-rank approximations in the case of oscillatory kernels. The most important result of this Chapter is the extension of the standard ACA to vectorial problems, suited for both iterative and direct solver, in order to utilize $\mathcal{H}$-matrices and the ACA in the context of 3D elastodynamics.

The main feature of $\mathcal{H}$-matrices used in association with ACA is that this technique is completely algebraic and, consequently, it is possible to define a matrix/vector product and a matrix arithmetics (addition, multiplication, inversion, etc...) for the hierarchical matrices. In particular, we have explained how to use the $\mathcal{H}$-matrix/vector product to accelerate the classical matrix/vector product, in order to reduce the time of each iteration of a GMRES based iterative solver. We have paid a particular attention to the algebra of $\mathcal{H}$-matrices, detailing how to calculate additions and multiplications between such matrices. We have observed that these operations are not exact but formatted, i.e. based on a prescribed accuracy. Finally, we have presented an $\mathcal{H}$-matrix based direct solution strategy, consisting in an $\mathcal{H}$-matrix based LU factorization and an $\mathcal{H}$-matrix based backward and forward substitution method.

3 Application of $\mathcal{H}$-matrices to BEM for 3D Frequency Domain Elastodynamics

Contents

3.1 Introduction 68

3.2 Scattering Problem 68

3.2.1 Traces, Integral Representation Formula and Integral Equations 70

3.2.2 Boundary Element Method 72

3.3 $\mathcal{H}$-matrices: Computational Aspects 72

3.3.1 Compression Rate 74

3.3.2 Accuracy of the $\mathcal{H}$-matrix representation 76

3.3.3 Accuracy of the $\mathcal{H}$-matrix/vector multiplication 77

3.4 Scattering of a Plane Wave by a Rigid Sphere 78

3.4.1 Diffraction of P-waves by a Unit Sphere 80

3.4.2 Numerical Study of the Complexity 81

3.4.3 $\mathcal{H}$-matrix Based Iterative Solver 84

3.4.4 $\mathcal{H}$-matrix Based Direct Solver 88

3.4.5 Error Estimate to Certify the Results of the $\mathcal{H}$-LU Direct Solver 91

3.5 Parallelization 96

3.6 Conclusions 97

3.1 Introduction

The application of hierarchical matrices to 3-D elastodynamic BEM is recent. In 2010, an accelerated $\mathcal{H}$-matrix elastodynamic time-domain boundary element formulation has been presented [131], using the convolution quadrature method to convert the time-domain problem into a system of decoupled Laplace-domain problems. In this approach the full-space elastodynamic fundamental solutions are employed with consequent discretization of the free-surface to model wave propagation in a half-space. In the same year, the advantage of the hierarchical format and hierarchical arithmetic have been exploited in the context of 3-D elastodynamic crack problems (see [20], where a dual boundary element method formulated in the Laplace-domain is used), and in 2012 the method have been proposed for anisotropic elastodynamics in the frequency domain [132]. In 2013, the hierarchical matrices have been applied to boundary element methods for elastodynamics based on Green's functions for a horizontally layered half-space [51]. A year later, the coupling of the finite element and this fast boundary element methods have been discussed [52] in the context of soil-structure interaction problems in the frequency domain.

This Chapter aims at extending the capabilities of $\mathcal{H}$-matrix based iterative and direct solvers to 3-D time-harmonic elastodynamics, and at studying their properties since all the previous published works only apply the hierarchical matrix technique. The Chapter is organized as follows. In Section 3.2 classical concepts pertaining to BEMs applied to scattering problems are recalled, paying particular attention to the theory of single- and double-layer potentials [50]. Several crucial computational and implementation issues are addressed in Section 3.3, in order to understand how to set up the parameters necessary to compute an $\mathcal{H}$-matrix representation of the elastodynamic BEM matrix. Finally, the accuracy and computational performances of the proposed $\mathcal{H}$-matrix based BEM are assessed in Section 3.4 on numerical examples treating the diffraction of incident vertical P-waves by an unit sphere. A comparison between the $\mathcal{H}$-matrix based and a Fast-Multipole based iterative solver is presented [37], while the accuracy of the $\mathcal{H}$-matrix based direct solver is certified with an error estimate.

3.2 Scattering Problem

Let us consider a bounded domain Ω^- in $\mathbb{R}^3$ representing an impenetrable body (obstacle) with a closed boundary $\Gamma := \partial\Omega^-$ of class $\mathcal{C}^2$ at least. Let Ω^+ denote the associated exterior domain $\mathbb{R}^3 \setminus \overline{\Omega^-}$ and $\mathbf{n}$ the outer unit normal vector to the boundary Γ. The Lamé parameters μ and λ and the density ϱ are positive constants. The propagation of time-harmonic waves in a three-dimensional isotropic and homogeneous elastic medium is modeled by the Navier equation:

$$\mathrm{div}\boldsymbol{\sigma}(\mathbf{u}) + \varrho\omega^2\mathbf{u} = \mathbf{0}, \tag{3.1}$$

where $\omega > 0$ is the frequency. In the case of 3D isotropic elastodynamics, the stress

$\boldsymbol{\sigma}(\mathbf{u})$ and the strain $\boldsymbol{\varepsilon}(\mathbf{u})$ tensors are given by:

$$\boldsymbol{\sigma}(\mathbf{u}) := \lambda(\mathrm{div}\mathbf{u})\mathbf{I}_3 + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) \quad \text{and} \quad \boldsymbol{\varepsilon}(\mathbf{u}) := \frac{1}{2}\left([\boldsymbol{\nabla}\mathbf{u}] + [\boldsymbol{\nabla}\mathbf{u}]^\top\right), \tag{3.2}$$

where $\mathbf{I}_3$ is the 3-by-3 identity matrix and $[\boldsymbol{\nabla}\mathbf{u}]$ is the matrix whose j-th column is the gradient of the j-th component of $\mathbf{u}$.

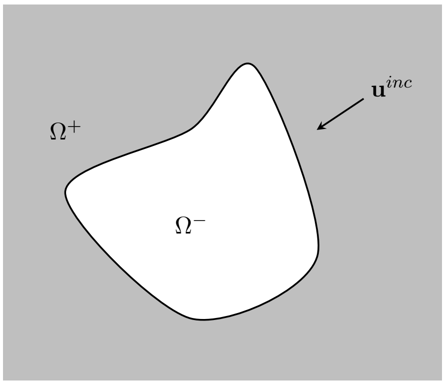


Figure 3.1: Scattering problem: notations.

The field $\mathbf{u}$ can be decomposed into a longitudinal field $\mathbf{u}_P := \nabla\psi_P$ and a transverse field $\mathbf{u}_S := \mathbf{u} - \mathbf{u}_P = \mathbf{curl}\ \boldsymbol{\psi}_S$. The scalar and vector potentials ψ_P and $\boldsymbol{\psi}_S$ satisfy respectively

$$\begin{cases} \psi_P = -\kappa_P^{-2}\mathrm{div}\mathbf{u} \\ \Delta\psi_P + \kappa_P^2\psi_P = 0 \end{cases} \quad \text{and} \quad \begin{cases} \boldsymbol{\psi}_S = -\kappa_S^{-2}\mathbf{curl}\ \mathbf{u} \\ \Delta\boldsymbol{\psi}_S + \kappa_S^2\boldsymbol{\psi}_S = \mathbf{0} \end{cases}, \tag{3.3}$$

where $\kappa_P^2 = \varrho\omega(\lambda + 2\mu)^{-1}$ and $\kappa_S^2 = \varrho\omega^2\mu^{-1}$ are the P and S wave-numbers. The wave-length are defined by $\lambda_\alpha := {}^{2\pi}/_{\kappa_\alpha}$, with $\alpha =$ P,S.

The *scattering problem* is formulated as follows: given an incident wave field $\mathbf{u}^{inc}$ which is assumed to solve the Navier equation in absence of any scatterer, find the displacement $\mathbf{u}$ solution to the Navier Equation (3.1) in Ω^+ which satisfies the Dirichlet boundary condition on Γ:

$$\mathbf{u}^{tot}_{|\Gamma} = \mathbf{u}_{|\Gamma} + \mathbf{u}^{inc}_{|\Gamma} = \mathbf{0}. \tag{3.4}$$

In addition, the field $\mathbf{u}$ has to satisfy the *Kupradze radiation conditions* at infinity:

$$\lim_{r\to\infty} r\left(\frac{\partial\psi_P}{\partial r} - \imath\kappa_P\psi_P\right) = 0 \quad \text{and} \quad \lim_{r\to\infty} r\left(\frac{\partial\boldsymbol{\psi}_S}{\partial r} - \imath\kappa_S\boldsymbol{\psi}_S\right) = \mathbf{0}, \tag{3.5}$$

uniformly in all directions.

3.2.1 Traces, Integral Representation Formula and Integral Equations

We denote by $H^s_{loc}(\overline{\Omega^+})$ and $H^s(\Gamma)$ the standard (local in the case of exterior domain) complex valued Hilbertian Sobolev spaces of the order $s \in \mathbb{R}$, defined on $\overline{\Omega^+}$ and Γ respectively (with the convention $H^0 = L^2$). Spaces of vector functions will be denoted by boldface letters, thus $\mathbf{H}^s := (H^s)^3$. We set

$$\mathbf{\Delta}^*\mathbf{u} := \text{div}\boldsymbol{\sigma}(\mathbf{u}) = (\lambda + 2\mu)\boldsymbol{\nabla}\text{div } \mathbf{u} - \mu\mathbf{curl}(\mathbf{curl}\ \mathbf{u}) \tag{3.6}$$

and introduce the energy space

$$\mathbf{H}^1_\pm(\mathbf{\Delta}^*) := \{\mathbf{u} \in \mathbf{H}^1_{loc}(\overline{\Omega^\pm})\ :\ \mathbf{\Delta}^*\mathbf{u} \in \mathbf{L}^2_{loc}(\overline{\Omega^\pm})\}. \tag{3.7}$$

The Neumann trace for elastodynamic problems, defined by $\mathbf{t} := \mathbf{Tu}$, is given by the traction operator

$$\mathbf{T} := 2\mu\frac{\partial}{\partial\mathbf{n}} + \lambda\mathbf{n}\text{div} + \mu\mathbf{n} \times \mathbf{curl}. \tag{3.8}$$

We recall that we have $\mathbf{u}_{|\Gamma} \in \mathbf{H}^{1/2}(\Gamma)$ and $\mathbf{t}_{|\Gamma} \in \mathbf{H}^{-1/2}(\Gamma)$ for all $\mathbf{u} \in \mathbf{H}^1_\pm(\mathbf{\Delta}^*)$.

If $\mathbf{u}^+ \in \mathbf{H}^1_+(\mathbf{\Delta}^*)$ and $\mathbf{u}^- \in \mathbf{H}^1_-(\mathbf{\Delta}^*)$ are solutions to the Navier Equation (3.1) in Ω^+ and Ω^- respectively, and $\mathbf{u}^+$ satisfies Kupradze radiation conditions (3.5), the integral representation of the field is given by

$$\mathcal{D}[\mathbf{u}]_\Gamma(\mathbf{x}) - \mathcal{S}[\mathbf{t}]_\Gamma(\mathbf{x}) = \begin{cases} \mathbf{u}^+(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega^+ \\ \mathbf{u}^-(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega^- \end{cases}, \tag{3.9}$$

where the square brackets $[\mathbf{u}]_\Gamma := \mathbf{u}^-_{|\Gamma} - \mathbf{u}^+_{|\Gamma}$ and $[\mathbf{t}]_\Gamma := \mathbf{t}^-_{|\Gamma} - \mathbf{t}^+_{|\Gamma}$ denote discontinuities across the interface. The *single-layer potential operator* $\mathcal{S}$ and the *double-layer potential operator* $\mathcal{D}$ are respectively defined by

$$\mathcal{S}\boldsymbol{\varphi} := \int_\Gamma \mathbf{U}(\cdot,\mathbf{y})\boldsymbol{\varphi}(\mathbf{y})d\Gamma_y \quad \text{and} \quad \mathcal{D}\boldsymbol{\psi} := \int_\Gamma [\mathbf{T}_y\mathbf{U}(\cdot,\mathbf{y})]^\top \boldsymbol{\psi}(\mathbf{y})d\Gamma_y. \tag{3.10}$$

In (3.10), the fundamental solution of the Navier equation is a 3-by-3 matrix valued function expressed by

$$\mathbf{U}(\mathbf{x},\mathbf{y}) := \frac{1}{\varrho\omega^2}\left(\mathbf{curl}\ \mathbf{curl}_x\left[\frac{e^{\imath\kappa_S\|\mathbf{x}-\mathbf{y}\|}}{4\pi\|\mathbf{x}-\mathbf{y}\|}\mathbf{I}_3\right] - \boldsymbol{\nabla}_x\text{div}_x\left[\frac{e^{\imath\kappa_P\|\mathbf{x}-\mathbf{y}\|}}{4\pi\|\mathbf{x}-\mathbf{y}\|}\mathbf{I}_3\right]\right). \tag{3.11}$$

The tensor $\mathbf{T}_y\mathbf{U}(\mathbf{x},\mathbf{y})$ is obtained by applying the traction operator $\mathbf{T}_y$ (i.e. the operator $\mathbf{T}$ where the derivatives are applied with respect to variable y) to each column of $\mathbf{U}(\mathbf{x},\mathbf{y})$.

The potentials $\mathcal{S}$ and $\mathcal{D}$ are continuous from $\mathbf{H}^{-1/2}(\Gamma)$ and $\mathbf{H}^{1/2}(\Gamma)$, respectively to $\mathbf{H}^1_-(\mathbf{\Delta}^*) \cup \mathbf{H}^1_+(\mathbf{\Delta}^*)$. For any $\boldsymbol{\varphi} \in \mathbf{H}^{-1/2}(\Gamma)$ and $\boldsymbol{\psi} \in \mathbf{H}^{1/2}(\Gamma)$, 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 Neumann $\gamma_1^\pm$ traces of $\mathcal{S}$ and $\mathcal{D}$ are given by

$$\gamma_0^\pm \mathcal{S} = S, \quad \gamma_1^\pm \mathcal{S} = \mp\frac{1}{2}I + D' \quad \text{and} \quad \gamma_0^\pm \mathcal{D} = \pm\frac{1}{2}I + D, \tag{3.12}$$

where I is the identity operator and the operators S, D and $D^{'}$ are given by

$$\begin{aligned} S\boldsymbol{\varphi}(\mathbf{x}) &:= \int_\Gamma \mathbf{U}(\mathbf{x},\mathbf{y})\boldsymbol{\varphi}(\mathbf{y})d\Gamma_y, \\ D\boldsymbol{\psi}(\mathbf{x}) &:= \int_\Gamma \left[\mathbf{T}_y\mathbf{U}(\mathbf{x},\mathbf{y})\right]^\top \boldsymbol{\psi}(\mathbf{y})d\Gamma_y, \\ D^{'}\boldsymbol{\varphi}(\mathbf{x}) &:= \int_\Gamma \left[\mathbf{T}_x\mathbf{U}(\mathbf{x},\mathbf{y})\right]^\top \boldsymbol{\varphi}(\mathbf{y})d\Gamma_y. \end{aligned} \tag{3.13}$$

Since we are considering a Dirichlet problem, with the particular choice of solution $\mathbf{u}^- = -\mathbf{u}^{inc}$ and $\mathbf{u}^+ = \mathbf{u}$ we observe that

$$\begin{aligned} [\mathbf{u}]_\Gamma &= -\mathbf{u}^{inc}_{|\Gamma} - \mathbf{u}_{|\Gamma} = \mathbf{0} \\ [\mathbf{t}]_\Gamma &= -\mathbf{t}^{inc}_{|\Gamma} - \mathbf{t}_{|\Gamma} := -\mathbf{t}^{tot}. \end{aligned} \tag{3.14}$$

Thus, the integral representation of the unknown field $\mathbf{u}$ can be rewritten as

$$\mathbf{u}(\mathbf{x}) = \mathcal{S}\mathbf{t}^{tot}(\mathbf{x}), \quad \mathbf{x} \in \Omega^+ \tag{3.15}$$

and, consequently, we get the following single-layer potential representation of the total field

$$\mathcal{S}\mathbf{t}^{tot}(\mathbf{x}) + \mathbf{u}^{inc}(\mathbf{x}) = \begin{cases} \mathbf{u}^{tot}(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega^+ \\ \mathbf{0} & \text{if } \mathbf{x} \in \Omega^- \end{cases}. \tag{3.16}$$

Since the integral representation (3.16) ensures that $\mathbf{u}^{tot}$ is solution to the Navier Equation (3.1) in $\Omega^+ \cup \Omega^-$ and satisfies Kupradze radiation conditions (3.5), we have to determine the unknown $\mathbf{t}^{tot}$ such that $\mathbf{u}^{tot}$ satisfies also the Dirichlet boundary condition (3.4). In order to obtain an integral equation for this unknown, we apply the potential theory that we recalled, to the relation

$$\mathcal{S}\mathbf{t}^{tot}(\mathbf{x}) + \mathbf{u}^{inc}(\mathbf{x}) = \mathbf{0}, \qquad \mathbf{x} \in \Omega^-. \tag{3.17}$$

In particular, applying the trace operator γ_0^- to (3.17), we obtain

$$S\mathbf{t}^{tot}(\mathbf{x}) = -\mathbf{u}^{inc}_{|\Gamma}(\mathbf{x}), \quad \mathbf{x} \in \Gamma \tag{3.18}$$

while, applying the trace operator γ_1^- to (3.17), we obtain

$$\left(\frac{1}{2}I + D'\right)\mathbf{t}^{tot}(\mathbf{x}) = -\mathbf{t}^{inc}_{|\Gamma}(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \tag{3.19}$$

3.2.2 Boundary Element Method

The numerical solution of the boundary integral equations (3.18) or (3.19) is based on a discretization of the surface Γ into N_E isoparametric boundary elements. Continuous, piecewise-linear approximation of each component of the traction, based on the three-noded triangular boundary elements, is considered. The N_I traction approximation nodes thus defined also serve as collocation points. This discretization process transforms (3.18) or (3.19) into a square complex-valued matrix equation of size $N = 3N_I$ of the form

$$A\mathbf{z} = \mathbf{b}, \tag{3.20}$$

where the N-vector $\mathbf{z}$ collects the sought degrees of freedom (DOFs), namely the nodal traction components, while the N-vector $\mathbf{b}$ arises from the imposed incident wave-field. Setting up the matrix A classically requires the computation of all element integrals for each collocation point, thus requiring a computational time of the order $\mathcal{O}(N^2)$.

The influence matrix A is fully-populated. For this reason, on personal laptops, standard BEM is usually restricted to models of size not exceeding $N = \mathcal{O}(10^4)$. BEM problems can be solved by means of direct or iterative algorithms. The overall computational cost of a direct method is of the order of $\mathcal{O}(N^3)$ operations, to be compared with a cost of the order of $\mathcal{O}(N^2)$ operations for each iteration needed by an iterative method. Since the total cost of the problem is at least of the order of $\mathcal{O}(N^2)$, the application of the BEM to large models requires a reduction not only of the memory requirements of the matrix A, but also of the CPU time to solve (3.20). The Hierarchical Matrices ($\mathcal{H}$-matrices) technique introduced in Chapter 2 is known in many other fields as a very efficient approach for achieving these objectives. It is therefore chosen as the basis for the present acceleration of the elastodynamic BEM.

3.3 $\mathcal{H}$-matrices: Computational Aspects

The efficiency of the $\mathcal{H}$-matrix representation of the BEM matrix A is affected by two parameters: the cardinality of the leaf-clusters N_{LEAF} and the parameter η in the admissible condition (2.16) that we recall here:

$$\min\{\text{diam}(B_\tau), \text{diam}(B_\sigma)\} \leq \eta \text{dist}(B_\tau, B_\sigma),$$

where B_ζ is the ball enclosing all the discretization points corresponding to the matrix-indices in $\zeta = \tau, \sigma$. In particular, N_{LEAF} influences the number of levels in the cluster tree T_I and, consequently, the number of blocks in the block-cluster tree $T_{I\times I}$. On the other hand, η is responsible for the number of admissible blocks and, therefore, for the memory savings. Theoretical works on $\mathcal{H}$-matrices recommend the choice of the parameter $\eta < 1$, in order to achieve the optimal complexity of storage [84]. In practical implementations, it has been observed that values of η between $1/2$ and $3/2$ are more appropriate to deal with Laplace problems [138], while electromagnetic problems required $\eta \geq 3$ [118].

To investigate how to set these two parameters N_{LEAF} and η in the context of elastodynamics, we consider an isotropic bar with constitutive properties $\mu = \varrho = 1$ and $\nu = {}^{1}\!/_{3}$, whose geometrical configuration is shown in Figure 3.2.

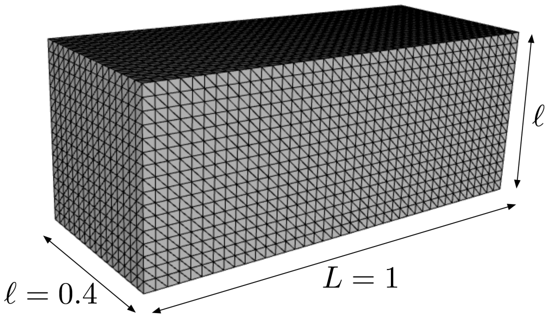


Figure 3.2: Bar geometry and example of mesh.

Seven circular frequencies ω are considered, adjusted in order to have meshes with a fixed density $d_{\lambda_{\text{S}}} = 10$ points per S-wavelength, i.e. $\omega := {}^{2\pi}\!/_{d_{\lambda_{\text{S}}} a}$, where $a > 0$ is the mean mesh length. In Table 3.1 we report the number of degrees of freedom N considered and the corresponding non-dimensional frequencies $\phi_{\text{S}} = {}^{\omega L}\!/_{c_{\text{S}}\pi}$.

N	2 310	9 222	20 742	36 870	57 606	82 950	112 902
ϕ_{S}	2	4	6.5	8	10	12	14

Table 3.1: Isotropic bar (elastodynamics): number of degrees of freedom and corresponding non-dimensional frequency.

In the following of this Section, we focus our attention on the matrix A resulting from the discretization of the single-layer potential operator S, defined by

$$S\boldsymbol{\varphi}(\mathbf{x}) := \int_{\Gamma} \mathbf{U}(\mathbf{x},\mathbf{y};\omega)\boldsymbol{\varphi}(\mathbf{y})d\Gamma_y. \tag{3.21}$$

The computation of the entries of the matrix A involves the evaluation of the 3D elastodynamic Green's function $\mathbf{U}(\mathbf{x},\mathbf{y};\omega)$, which is a linear combination of derivatives of the Helmholtz fundamental solution. In a first step, we consider the matrix corresponding to the evaluation of a modified version of the Helmholtz fundamental solution, i.e.

$$G_h(\mathbf{x},\mathbf{y};\kappa) = \frac{e^{\imath\kappa\|\mathbf{x}-\mathbf{y}\|}}{4\pi\|\mathbf{x}-\mathbf{y}\| + h} \tag{3.22}$$

where we have introduced the parameter $h > 0$ to avoid the singularity that occurs when $\mathbf{x} \to \mathbf{y}$. As this singularity is only present in the diagonal blocks, which are non-admissible (and hence the full storage is used), it does not affect the memory savings and the accuracy of the $\mathcal{H}$-matrix representation. In the following, we choose h to be equal to the discretization step used to discretize the bar of Figure 3.2, with a fixed density of $d_\lambda = 10$ points per wavelength. The non-dimensional frequencies $\phi = {}^{L\kappa}/_{2\pi}$ considered are given in Table 3.2.

N	3 074	19 202	49 154	110 594	150 530	307 202	480 002	940 802
ϕ	3.5	8.8	14.1	21.3	24.4	35.7	43.5	62.5

Table 3.2: Isotropic bar (Helmholtz): number of degrees of freedom and corresponding non-dimensional frequency.

The results corresponding to the discretization matrix of the Helmholtz Green's function (acoustic toy-problem) are provided by a MATLAB code. In order to treat the matrix resulting from the discretization of the single-layer potential operator (elastodynamic toy-problem), this code has been used as prototype for a FORTRAN 90 code, subsequently merged into the code COFFEE, developed at the laboratory POEMS (ENSTA-ParisTech) to handle full elastodynamic problems [1].

All the results presented in this Chapter have been obtained on a 3.5-GHz Intel Xeon CPU ES-2637 with 768 Gb of RAM.

3.3.1 Compression Rate

As explained in Chapter 2, an $\mathcal{H}$-matrix representation of a matrix coming from the discretization of a non-local operator, is a hierarchical structure where the matrix blocks corresponding to a non-admissible cluster pair $(\tau, \sigma) \in \text{Leaf}^-(T_{I\times I})$ are stored as dense matrices and the matrix blocks corresponding to an admissible cluster pair $(\tau, \sigma) \in \text{Leaf}^+(T_{I\times I})$ are stored as low-rank matrices (with rank k). Informations on the memory saving using $\mathcal{H}$-matrices is provided by the study of the compression rate $T(\mathcal{H})$ of the $\mathcal{H}$-matrix representation, defined as the ratio between the size of the compressed matrix and the size of the complete (not compressed) matrix, i.e.

$$T(\mathcal{H}) := \frac{1}{N^2}\left(\sum_{(\tau,\sigma)\in\text{Leaf}^+(T_{I\times I})} k\left(|\tau| + |\sigma|\right) + \sum_{(\tau,\sigma)\in\text{Leaf}^-(T_{I\times I})} |\tau|\cdot|\sigma|\right). \tag{3.23}$$

The ACA algorithm computes adaptively the rank required to guarantee a prescribed accuracy $\varepsilon_{\text{ACA}} > 0$ (usually the intrinsic stopping criterion (2.21) based on the variation

[1] The code COFFEE is a Fortran 90 based program that solves 3D linear visco-elastodynamic problems in the frequency domain by the Boundary Element Method accelerated by Fast Multipole Method (FM-BEM). This code is able to integrate the singularities of the Green's displacement and traction. Our contribution (25 000 lines) consists in the implementation of the $\mathcal{H}$-matrices technique. The code, written by Chaillat [33], is now able to assemble the $\mathcal{H}$-matrix representation of the BEM matrix and solve the BEM system with both an iterative (based on GMRES accelerated by the fast $\mathcal{H}$-matrix/vector product) and with a direct (accelerated by the fast $\mathcal{H}$-LU factorization) solution strategy.

of the Frobenius norm in consecutive approximations is used). Unfortunately, ACA does not provide the best numerical rank (i.e. the lowest to achieve a prescribed accuracy) but a good first approximation. Thus, a truncated SVD recompression is applied to each admissible block of the $\mathcal{H}$-matrix representation with a parameter $\varepsilon_{\text{SVD}} > 0$ (Subsection 2.3.4).

The first stage of our numerical study of the compression rate is relative to the acoustic toy-problem with kernel $G_h(\mathbf{x}, \mathbf{y}; \kappa)$. In Figure 3.3, we plot the compression rate obtained when we compute the $\mathcal{H}$-matrix representation of the matrix coming from the discretization of the function $G_h(\mathbf{x}, \mathbf{y}; \kappa)$, for the frequency range $\kappa = 4\pi$ to 80π and, consequently, $N = 2\ 310$ to $921\ 606$ (see Table 3.2). In this case no recompression with truncated SVD has been performed. As suggested by acoustic studies [118], the $\mathcal{H}$-matrix representation is constructed with a minimum number of elements $N_{\text{LEAF}} = 100$ in the hierarchical cluster tree and the parameter $\eta = 3$ in the admissibility condition.

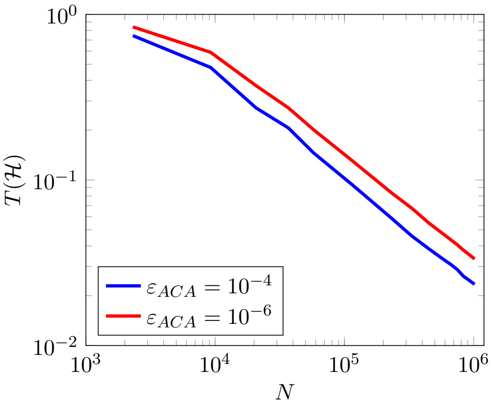


Figure 3.3: Discretization of the modified 3D Helmholtz fundamental solution: compression rate obtained with two different values of the parameter ε_{ACA}.

We note that for both values $\varepsilon_{\text{ACA}} = 10^{-4}$ and $\varepsilon_{\text{ACA}} = 10^{-6}$, the compression rate improves when we increase the frequency and, consequently, the problem size. In particular, for problems of almost 1 million unknowns only 2% of the entries of the original matrix is needed in its $\mathcal{H}$-matrix representation. After these encouraging results for the 3D acoustic Green's function, we consider the following studies on the matrix A, coming from the discretization of single-layer potential operator defined in (3.21).

In Tables 3.3 - 3.5, we present the dependency of the compression rates $T_{\text{B}}(\mathcal{H})$ and $T_{\text{A}}(\mathcal{H})$, respectively before and after recompression, as a function of N_{LEAF} and η, with the fixed values $\varepsilon_{\text{ACA}} = \varepsilon_{\text{SVD}} = 10^{-4}$. We observe that for all values of N_{LEAF} neither compression rates $T_{\text{B}}(\mathcal{H})$ or $T_{\text{A}}(\mathcal{H})$ are modified, if η is greater than 3. As in the context of electromagnetic problems [118], a good choice of these two parameters seems to be $N_{\text{LEAF}} = 100$ and $\eta = 3$, because they ensure good compression rates, due to an appropriate number of blocks in the block-cluster tree (consequence of $N_{\text{LEAF}} = 100$) and a proper selection of the admissible blocks (consequence of $\eta = 3$). To validate these values, it is necessary to investigate the efficiency of the $\mathcal{H}$-matrix representation and $\mathcal{H}$-matrix/vector product.

	$\eta = 1$		$\eta = 3$		$\eta = 5$	
N	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$
2 310	0.877	0.824	0.741	0.684	0.741	0.684
9 222	0.604	0.508	0.417	0.318	0.417	0.318
20 742	0.409	0.333	0.277	0.219	0.277	0.219
36 870	0.292	0.231	0.184	0.142	0.184	0.142
57 606	0.238	0.196	0.144	0.112	0.144	0.112
82 950	0.184	0.144	0.111	0.085	0.111	0.085
112 902	0.158	0.126	0.093	0.072	0.093	0.072

Table 3.3: Dependency of $T_{\mathrm{B}}(\mathcal{H})$ and $T_{\mathrm{A}}(\mathcal{H})$ on η ($N_{\mathrm{LEAF}} = 75$ and $\varepsilon_{\mathrm{ACA}} = 10^{-4}$).

	$\eta = 1$		$\eta = 3$		$\eta = 5$	
N	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$
2 310	0.909	0.888	0.745	0.694	0.745	0.694
9 222	0.696	0.653	0.477	0.416	0.477	0.416
20 742	0.414	0.342	0.278	0.223	0.278	0.223
36 870	0.333	0.285	0.205	0.171	0.205	0.171
57 606	0.244	0.202	0.146	0.115	0.146	0.115
82 950	0.210	0.177	0.119	0.096	0.119	0.096
112 902	0.156	0.128	0.094	0.074	0.094	0.074

Table 3.4: Dependency of $T_{\mathrm{B}}(\mathcal{H})$ and $T_{\mathrm{A}}(\mathcal{H})$ on η ($N_{\mathrm{LEAF}} = 100$ and $\varepsilon_{\mathrm{ACA}} = 10^{-4}$).

	$\eta = 1$		$\eta = 3$		$\eta = 5$	
N	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$	$T_{\mathrm{B}}(\mathcal{H})$	$T_{\mathrm{A}}(\mathcal{H})$
2 310	0.909	0.887	0.754	0.694	0.754	0.694
9 222	0.862	0.855	0.643	0.632	0.643	0.632
20 742	0.686	0.669	0.400	0.369	0.400	0.369
36 870	0.348	0.302	0.212	0.179	0.212	0.179
57 606	0.305	0.273	0.175	0.150	0.175	0.150
82 950	0.281	0.256	0.152	0.134	0.152	0.134
112 902	0.203	0.179	0.105	0.087	0.105	0.087

Table 3.5: Dependency of $T_{\mathrm{B}}(\mathcal{H})$ and $T_{\mathrm{A}}(\mathcal{H})$ on η ($N_{\mathrm{LEAF}} = 200$ and $\varepsilon_{\mathrm{ACA}} = 10^{-4}$).

3.3.2 Accuracy of the $\mathcal{H}$-matrix representation

When we replace the matrix A (resulting from the discretization of the single-layer potential operator) with its $\mathcal{H}$-matrix representation, the reduction of storage costs comes from low-rank approximations computed by the ACA. To understand how it affects the overall accuracy of the $\mathcal{H}$-matrix representation and to validate the parameters $N_{\text{LEAF}} = 100$ and $\eta = 3$, we study the relative error

$$\varepsilon_1 := \frac{\|A - A_{\mathcal{H}}\|_{\text{F}}}{\|A\|_F}. \tag{3.24}$$

In Table 3.6, we represent the dependency of ε_1 on the parameter ε_{ACA}, used in the stopping criteria (2.22). The circular frequencies ω corresponding to the considered problem sizes are reported in Table 3.1. We see that the level of error introduced by ACA can be controlled by the parameter ε_{ACA}. It represents a good measure of the error on the approximation of the whole BEM matrix.

N	$\varepsilon_{\text{ACA}} = 10^{-4}$	$\varepsilon_{\text{ACA}} = 10^{-6}$	$\varepsilon_{\text{ACA}} = 10^{-9}$
2 310	5.23×10^{-5}	5.27×10^{-7}	4.72×10^{-10}
9 222	1.25×10^{-4}	1.49×10^{-6}	1.65×10^{-9}
20 742	1.48×10^{-4}	1.64×10^{-6}	1.92×10^{-9}
36 870	2.12×10^{-4}	2.78×10^{-6}	2.31×10^{-9}
57 606	2.83×10^{-4}	3.11×10^{-6}	2.73×10^{-9}
82 950	3.30×10^{-4}	3.17×10^{-6}	3.13×10^{-9}
112 902	3.66×10^{-4}	3.41×10^{-6}	3.85×10^{-9}

Table 3.6: Dependency of ε_1 on ε_{ACA} ($N_{\text{LEAF}} = 100$ and $\eta = 3$)

In the context of Helmholtz problem [17], it has been observed that the achieved error of the $\mathcal{H}$-matrix approximation via the ACA deteriorates for a growing number of degrees of freedom. In our applications this phenomenon does not occur. However, in order to achieve a desired accuracy, it is better to choose a value of ε_{ACA} which is one order of magnitude higher than our goal.

3.3.3 Accuracy of the $\mathcal{H}$-matrix/vector multiplication

The accuracy of the $\mathcal{H}$-matrix/vector product is related to the precision of the low-rank approximations. For this reason, we introduce the relative error

$$\varepsilon_2 := \frac{\|\mathbf{y} - \mathbf{y}_{\mathcal{H}}\|_2}{\|\mathbf{y}\|_2}, \tag{3.25}$$

where $\mathbf{y} := A\mathbf{w}$ is the result of the product of the matrix A with an arbitrary random vector $\mathbf{w}$ and $\mathbf{y}_{\mathcal{H}} := A_{\mathcal{H}}\mathbf{w}$ is the result of the product between the $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$ and the vector $\mathbf{w}$.

Table 3.7 gives the precision of the $\mathcal{H}$-matrix/vector product for three values of ε_{ACA}: 10^{-4}, 10^{-6} and 10^{-9}. The circular frequencies ω corresponding to the considered problem sizes are reported in Table 3.1. No recompression with SVD has been performed. We observe that the accuracy of the $\mathcal{H}$-matrix/vector product is related to the overcall accuracy of the $\mathcal{H}$-matrix representation. In fact, comparing Table 3.6 and Table 3.7, we observe the same order of accuracy. In particular, the precision obtained by the ACA gives informations on the accuracy of the $\mathcal{H}$-matrix/vector product. Note that we have tested this accuracy for different random vectors $\mathbf{w}$.

N	$\varepsilon_{\text{ACA}} = 10^{-4}$	$\varepsilon_{\text{ACA}} = 10^{-6}$	$\varepsilon_{\text{ACA}} = 10^{-9}$
2 310	4.05×10^{-5}	1.89×10^{-7}	1.62×10^{-10}
9 222	2.18×10^{-4}	1.32×10^{-6}	1.80×10^{-9}
20 742	1.76×10^{-4}	1.34×10^{-6}	1.12×10^{-9}
36 870	4.92×10^{-4}	2.57×10^{-6}	1.93×10^{-9}
57 606	2.88×10^{-4}	4.44×10^{-6}	2.68×10^{-9}
82 950	4.26×10^{-4}	4.34×10^{-6}	4.08×10^{-9}
112 902	7.17×10^{-4}	6.60×10^{-6}	6.26×10^{-9}

Table 3.7: Dependency of ε_2 on the parameter ε_{ACA} ($N_{\text{LEAF}} = 100$ and $\eta = 3$).

3.4 Scattering of a Plane Wave by a Rigid Sphere

So far, we have only considered the $\mathcal{H}$-matrix representation of the BEM matrix of the problem modeled by (3.18). Now, we focus our attention on the time-harmonic scattering problem of a plane wave by a rigid elastic sphere of radius $R > 0$, in order to analyze the performances of the $\mathcal{H}$-matrix based solvers. We recall here the analytical solution for this problem, presented in [38]. Since the Neumann trace $\mathbf{t}_{|\Gamma}$ of the displacement $\mathbf{u}$ solution to the exterior problem (3.1)-(3.4)-(3.5) can be expressed analytically in terms of vector spherical harmonics, in the following Y_l^m denotes the spherical harmonics of order $l \in \mathbb{N}$ and degree $m \in \mathbb{N}$ ($|m| \leq l$), J_l denotes the spherical Bessel functions of the first kind and order $l \in \mathbb{N}$ and $H_l^{(1)}$ denotes the spherical Hankel function of the first kind and order $l \in \mathbb{N}$. These functions allow to introduce the scalar functions:

$$\begin{aligned} u_{l,m}^{(1)}(\kappa_{\text{P}}, \mathbf{x}) &:= J_l(\kappa_{\text{P}}|\mathbf{x}|)Y_l^m(\hat{\mathbf{x}}) \\ u_{l,m}^{(3)}(\kappa_{\text{P}}, \mathbf{x}) &:= H_l^{(1)}(\kappa_{\text{P}}|\mathbf{x}|)Y_l^m(\hat{\mathbf{x}}) \end{aligned} \tag{3.26}$$

and the vector function

$$\begin{aligned} \mathbf{M}_{l,m}^{(k)}(\kappa_{\text{S}}, \mathbf{x}) &:= \mathbf{curl}\left(\mathbf{x}u_{l,m}^{(k)}(\kappa_{\text{S}}, \mathbf{x})\right) \\ \mathbf{N}_{l,m}^{(k)}(\kappa_{\text{S}}, \mathbf{x}) &:= \frac{1}{\imath\kappa_{\text{S}}}\mathbf{curl}\left(\mathbf{M}_{l,m}^{(1)}(\kappa_{\text{S}}, \mathbf{x})\right) \end{aligned} \qquad k = 1, 3 \tag{3.27}$$

In (3.26), we use the notation $\hat{\mathbf{x}} := {}^{\mathbf{x}}/_{\|\mathbf{x}\|} \in \mathbb{S}^2$ (unit sphere) for $\mathbf{x} \neq \mathbf{0}$. We consider an incident plane wave of the form

$$\mathbf{u}^{inc}(\mathbf{x}) = \frac{1}{\mu}e^{\imath\kappa_{\text{S}}(\mathbf{x}\cdot\mathbf{d})}(\mathbf{d} \times \mathbf{p}) \times \mathbf{d} + \frac{1}{\lambda + 2\mu}e^{\imath\kappa_{\text{P}}(\mathbf{x}\cdot\mathbf{d})}(\mathbf{d} \cdot \mathbf{p})\mathbf{d}, \tag{3.28}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the polarization vector and $\mathbf{d} \in \mathbb{S}^2$ is the propagation vector. Such incidence plane wave admits the following series expansion:

$$\mathbf{u}^{inc}(\mathbf{x}) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left[\alpha_l^{(1)}\mathbf{M}_{l,m}^{(1)}(\kappa_{\mathrm{S}},\mathbf{x}) + \beta_l^{(1)}\mathbf{N}_{l,m}^{(1)}(\kappa_{\mathrm{S}},\mathbf{x})\right] + \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\gamma_l^{(1)}\boldsymbol{\nabla} u_{l,m}^{(1)}(\kappa_{\mathrm{P}},\mathbf{x}) \tag{3.29}$$

with coefficients

$$\begin{aligned}
\alpha_l^{(1)} &:= \frac{1}{\mu}\frac{4\pi\imath^l}{l(l+1)}\mathbf{curl}_{\mathbb{S}^2}\overline{Y_l^m(\mathbf{d})}\cdot\mathbf{p}\\
\beta_l^{(1)} &:= \frac{1}{\mu}\frac{4\pi\imath^l}{l(l+1)}\boldsymbol{\nabla}_{\mathbb{S}^2}\overline{Y_l^m(\mathbf{d})}\cdot\mathbf{p}\\
\gamma_l^{(1)} &:= -\frac{4\pi\imath^l}{\kappa_{\mathrm{P}}(\lambda+2\mu)}\overline{Y_l^m(\mathbf{d})}\mathbf{d}\cdot\mathbf{p}
\end{aligned} \tag{3.30}$$

Since the vector spherical harmonics form a complete orthonormal system in the Hilbert space $\mathbf{L}^2(\mathbb{S}^2)$, the scattered wave is given by

$$\mathbf{u}(\mathbf{x}) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left[\alpha_l^{(3)}\mathbf{M}_{l,m}^{(3)}(\kappa_{\mathrm{S}},\mathbf{x}) + \beta_l^{(3)}\mathbf{N}_{l,m}^{(3)}(\kappa_{\mathrm{S}},\mathbf{x})\right] + \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\gamma_l^{(3)}\boldsymbol{\nabla} u_{l,m}^{(3)}(\kappa_{\mathrm{P}},\mathbf{x}) \tag{3.31}$$

where

$$\gamma_0^{(3)} := -\frac{J_0'(\kappa_{\mathrm{P}}R)}{H_0^{(1)'}(\kappa_{\mathrm{P}}R)}\gamma_0^{(1)} \tag{3.32}$$

and for $m \geq 1$

$$\begin{bmatrix}\alpha_l^{(3)}\\ \beta_l^{(3)}\\ \gamma_l^{(3)}\end{bmatrix} = -\begin{bmatrix}a_l^{(3)} & 0 & 0\\ 0 & b_l^{(3,1)} & c_l^{(3,1)}\\ 0 & b_l^{(3,2)} & c_l^{(3,2)}\end{bmatrix}\cdot\begin{bmatrix}a_l^{(1)} & 0 & 0\\ 0 & b_l^{(1,1)} & c_l^{(1,1)}\\ 0 & b_l^{(1,2)} & c_l^{(1,2)}\end{bmatrix}\cdot\begin{bmatrix}\alpha_l^{(1)}\\ \beta_l^{(1)}\\ \gamma_l^{(1)}\end{bmatrix}. \tag{3.33}$$

In this last equation we have $a_l^{(k)} := z_l^{(k)}(\kappa_{\mathrm{S}}R)$ and

$$\begin{aligned}
b_l^{(k,1)} &:= \frac{1}{\imath\kappa_{\mathrm{S}}R}\left[z_l^{(k)}(\kappa_{\mathrm{S}}R) + \kappa_{\mathrm{S}}Rz_l^{(k)'}(\kappa_{\mathrm{S}}R)\right]\\
b_l^{(k,2)} &:= \frac{l(l+1)}{\imath\kappa_{\mathrm{S}}R}z_l^{(k)}(\kappa_{\mathrm{S}}R)\\
c_l^{(k,1)} &:= \frac{1}{R}z_l^{(k)}(\kappa_{\mathrm{P}}R)\\
c_l^{(k,2)} &:= \kappa_P z_l^{(k)'}(\kappa_{\mathrm{P}}R).
\end{aligned} \tag{3.34}$$

To compute the traction trace of $\mathbf{u}$, i.e.

$$\mathbf{t}_{|\Gamma}(\mathbf{x}) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left[\alpha_l^{\mathrm{ex}}\mathbf{curl}_{\mathbb{S}^2}Y_l^m(\hat{\mathbf{x}}) + \beta_l^{\mathrm{ex}}\boldsymbol{\nabla}_{\mathbb{S}^2}Y_l^m(\hat{\mathbf{x}})\right] + \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\gamma_l^{\mathrm{ex}}\hat{\mathbf{x}}Y_l^m(\hat{\mathbf{x}}), \tag{3.35}$$

we use the following formulas for $k = 1, 3$:

$$\begin{aligned}
\mathbf{T}_x\mathbf{M}^{(k)}_{l,m}(\kappa,\mathbf{x}) :=& \mu\left[\kappa z_l^{(k)'}(\kappa|\mathbf{x}|) - \frac{z_l^{(k)}(\kappa|\mathbf{x}|)}{|\mathbf{x}|}\right]\mathbf{curl}_{\mathbb{S}^2}Y_l^m(\hat{\mathbf{x}})\\
\mathbf{T}_x\mathbf{N}^{(k)}_{l,m}(\kappa,\mathbf{x}) :=& \frac{\mu}{\imath\kappa|\mathbf{x}|^2}\cdot\left[-\left(\kappa^2|\mathbf{x}|^2 - 2l(l+1)+2\right)z_l^{(k)}(\kappa|\mathbf{x}|) - 2\kappa|\mathbf{x}|z_l^{(k)'}(\kappa|\mathbf{x}|)\right]\boldsymbol{\nabla}_{\mathbb{S}^2}Y_l^m(\hat{\mathbf{x}})+\\
&+2\mu\frac{l(l+1)}{\imath\kappa|\mathbf{x}|}\left[\kappa z_l^{(k)'}(\kappa|\mathbf{x}|) - \frac{z_l^{(k)}(\kappa|\mathbf{x}|)}{|\mathbf{x}|}\right]\hat{\mathbf{x}}Y_l^m(\hat{\mathbf{x}})
\end{aligned} \tag{3.36}$$

and

$$\begin{aligned}
\mathbf{T}_x\boldsymbol{\nabla}u^{(k)}_{l,m}(\kappa,\mathbf{x}) :=& \frac{2\mu}{|\mathbf{x}|}\left[\kappa z_l^{(k)'}(\kappa|\mathbf{x}|) - \frac{z_l^{(k)}(\kappa|\mathbf{x}|)}{|\mathbf{x}|}\right]\boldsymbol{\nabla}_{\mathbb{S}^2}Y_l^m(\hat{\mathbf{x}})+\\
&+\left[-\frac{2\mu}{|\mathbf{x}|^2}\left(\left(\kappa^2|\mathbf{x}|^2 - l(l+1)\right)z_l^{(k)}(\kappa|\mathbf{x}|) + 2\kappa|\mathbf{x}|z_l^{(k)'}(\kappa|\mathbf{x}|)\right)\right] + \hat{\mathbf{x}}Y_l^m(\hat{\mathbf{x}})\\
&-\lambda\kappa^2 z_l^{(k)}(\kappa|\mathbf{x}|)\hat{\mathbf{x}}.
\end{aligned} \tag{3.37}$$

3.4.1 Diffraction of P-waves by a Unit Sphere

To demonstrate the accuracy and the efficiency of the $\mathcal{H}$-matrix based solvers for 3D elastodynamics, we consider the diffraction of vertical incident plane P-waves ($\mathbf{p} = \mathbf{d} = (0, 0, -1)^\top$) by a unit sphere ($R = 1$ in Figure 3.4). The material properties are again fixed to $\mu = \varrho = 1$ and $\nu = 1/3$.

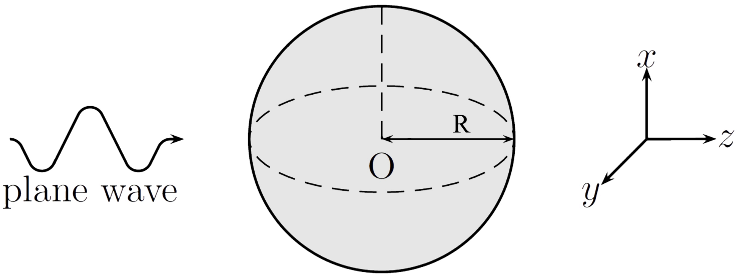


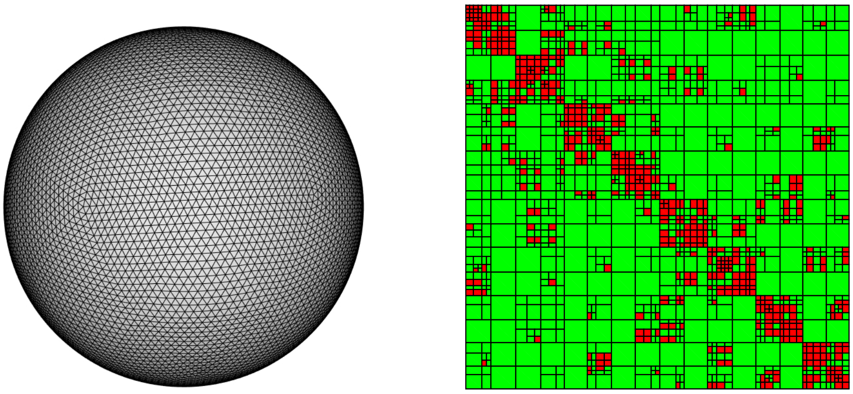
Figure 3.4: Diffraction of an incident P-wave by a unit sphere: geometry and notations.

This scattering problem is modeled by the Boundary Integral Equation (3.18) or (3.19). The surface of the sphere is discretized with a density of $d_{\lambda_S} = 10$ degrees of freedom per S-wavelength. Six circular frequencies $\omega = \omega(d_{\lambda_{\mathrm{S}}})$ are considered, with respective problem sizes $N = N(d_{\lambda_{\mathrm{S}}})$ and corresponding non-dimensional frequencies $\phi_{\mathrm{S}} = {}^{\omega R}/_{c_{\mathrm{S}}\pi}$ reported in Table 3.8.

N	7 686	30 726	183 099	490 629	763 638	985 818
ω	8	16	40	66	84	92
ϕ_S	2.5	5.1	12.7	21.0	26.7	29.3

Table 3.8: Diffraction of an incident P-wave by a unit sphere: dimensions of the problem, circular frequency and non-dimensional frequencies.

In the light of the numerical results of Sections 3.3.2 and 3.3.3, in all the following numerical examples the cluster tree T_I is built with a stopping criteria $N_{\text{LEAF}} = 100$ and the constant in the admissibility condition is set to $\eta = 3$. The admissibility condition depends only on the geometry of the domain and on the parameter η. In Figure 3.5, we give an illustration of the block repartition in the $\mathcal{H}$-matrix representation corresponding to the simple geometry of the unit sphere. We consider a problem of small size ($N = 30\ 726$) in order to better illustrate how $\eta = 3$ leads to large low-rank blocks (green blocks) and a small number of full blocks (red blocks).



(a) Mesh

(b) $\mathcal{H}$-matrix representation

Figure 3.5: Diffraction of an incident P-wave by a unit sphere: mesh of the sphere and corresponding $\mathcal{H}$-matrix representation ($N = 30\ 726$).

3.4.2 Numerical Study of the Complexity

The efficiency of $\mathcal{H}$-matrix based solvers depends on the possible storage reduction obtained by low-rank approximations. As explained in [73], due to the use of a geometrical cluster tree, the storage requirement is of the order of

$$N_{st}(A) \simeq \max\{k_{\text{MAX}}, N_{\text{LEAF}}\}|I| \log |I|, \tag{3.38}$$

where k_{MAX} is the maximum rank among the low-rank approximations of the admissible blocks. In the context of BEM, $|I| = N$ such that storage requirement is

$$N_{st}(A) \simeq \max\{k_{\text{MAX}}, N_{\text{LEAF}}\} N \log N. \tag{3.39}$$

For asymptotically smooth kernels, a constant k_{MAX} is used for all problem size without sacrificing accuracy, and hence ignored in the theoretical complexity analysis. Instead, for oscillatory kernels we have shown in Subsection 2.4.1 that k_{MAX} scales linearly with respect to the wave number κ. Additionally, the mesh size h used to discretize the boundary has to be chosen such that $\kappa h << 1 \;\Leftrightarrow\; \kappa << h^{-1}$ for a sufficient accuracy of the solution. If we assume that $h^{-1} \sim N^{1/2}$, we have $k_{\text{MAX}} \sim h^{-1} \sim N^{1/2}$. Hence, it is worthwhile to investigate the maximum rank of low-ranks blocks obtained in the case of the simple-layer and the double-layer operators. The theoretical complexity analysis is compared against the recorded memory requirements (number of entries that we have to store). These numerical experiences aim in particular at studying k_{MAX} as a function of N and at exploring the possibility to treat elastodynamic BEM matrices with logarithmic-linear complexity.

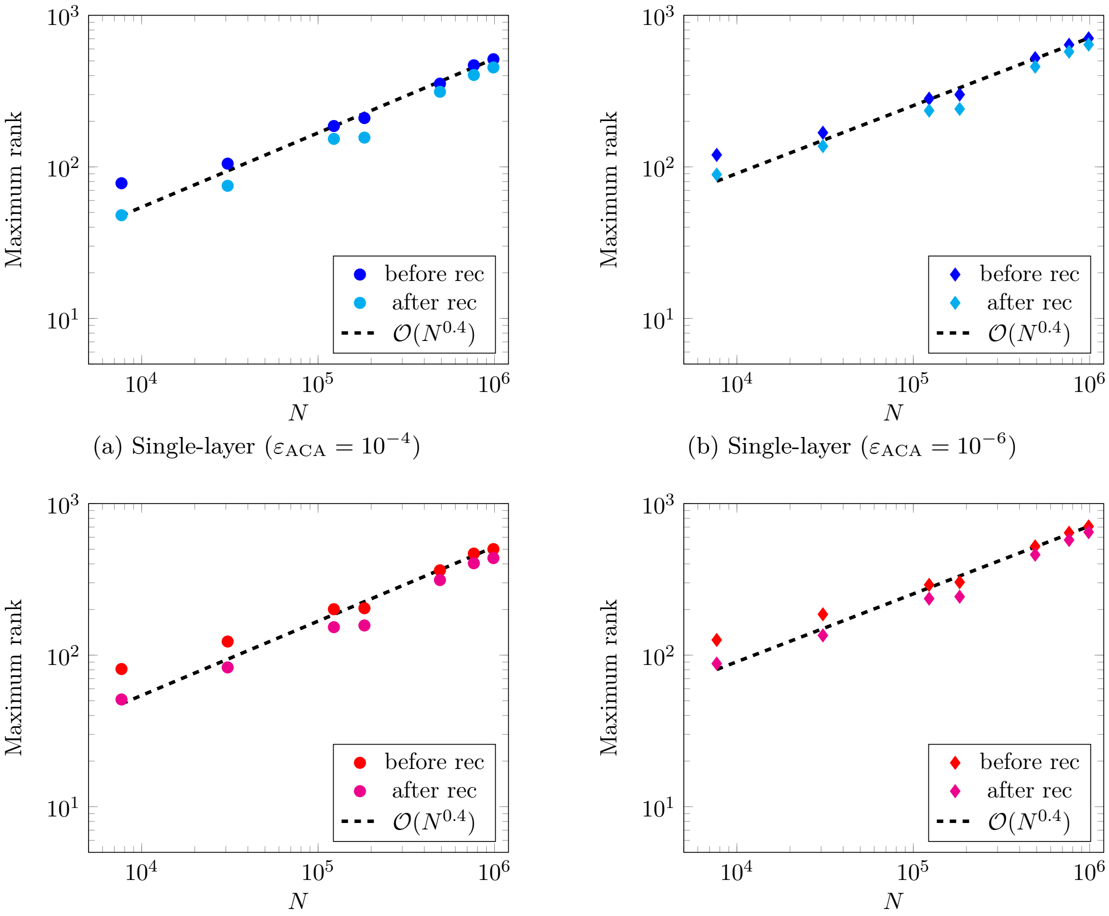
The following results involve the numerical solution of (3.18) and (3.19). We noticed in Subsection 2.4.2 that, since ACA gives an almost optimal numerical rank, the optimum rank has to be revealed performing a recompression with truncated SVD. This technique is applied to each admissible block of the $\mathcal{H}$-matrix representation with a parameter ε_{SVD} always chosen equal to ε_{ACA}.

In Figure 3.6, the maximum rank in the admissible blocks obtained numerically in the case of single-layer and double-layer operators is reported for two values of ε_{ACA}, both before and after recompression with the truncated SVD. The results indicate that k_{MAX} evolves as $\mathcal{O}(N^{0.4})$ when we increase the number of degrees of freedom N and, consequently, the frequency. Therefore, for this elastodynamic problem we find the same behavior observed in the case of 3-D Helmholtz problems [17], even if a small regression is observed. In particular, for large N no significant differences are observed between the value of k_{MAX} obtained performing ACA with $\varepsilon_{\text{ACA}} = 10^{-4}$ and $\varepsilon_{\text{ACA}} = 10^{-6}$. For the two considered values of ε_{ACA}, k_{MAX} given by the recompression with the truncated SVD is of the same order than the numerical maximum rank revealed by the ACA, during the approximation of the low-rank blocks of the $\mathcal{H}$-matrix representation.

Combining these observations on k_{MAX} and the theoretical estimate (3.39), we expect the memory requirement of the elastodynamic $\mathcal{H}$-matrix based BEM to be of the order of $\mathcal{O}(N^{1.4} \log N)$. In practice, it is reduced to $\mathcal{O}(N^{1.25} \log N)$, as shown in Figure 3.7 where the numerical results for the single- and the double-layer operators are lower than this bound.

We conclude that $\mathcal{H}$-matrix representations of the elastodynamic BEM matrices can be achieved with almost logarithmic linear memory cost for both the values $\varepsilon_{\text{ACA}} = 10^{-4}$ and $\varepsilon_{\text{ACA}} = 10^{-6}$. Since this last parameter influences the accuracy of the $\mathcal{H}$-matrix, further considerations on the adequate choice require the knowledge of how the error of the approximation of the matrix propagates on the solution of the BEM system.

The cumulative CPU time for the computation of the $\mathcal{H}$-matrix representation of the matrix A is the last aspect that we investigate. In Figure 3.8, we present the values recorded, which are of course code- and computer-dependent. We note that the parameter ε_{ACA} has small influence on the computational time, that evolves as $\mathcal{O}(N^{1.5}\log N)$ for both the single-layer and the double-layer operators. We observe that this behavior is almost equal to the optimal $\mathcal{O}(k_{\text{MAX}}N\log N)$. Comparing the results obtained for the memory requirement and for the CPU time, we see that, although the latter is more affected by the evolution of k_{MAX} than the former, it is still lower than its theoretical bound. Furthermore code-optimizations are possible but a parallelization of the code is needed (Section 3.5), in order to drastically reduce the computation time required for the $\mathcal{H}$-matrix representation.



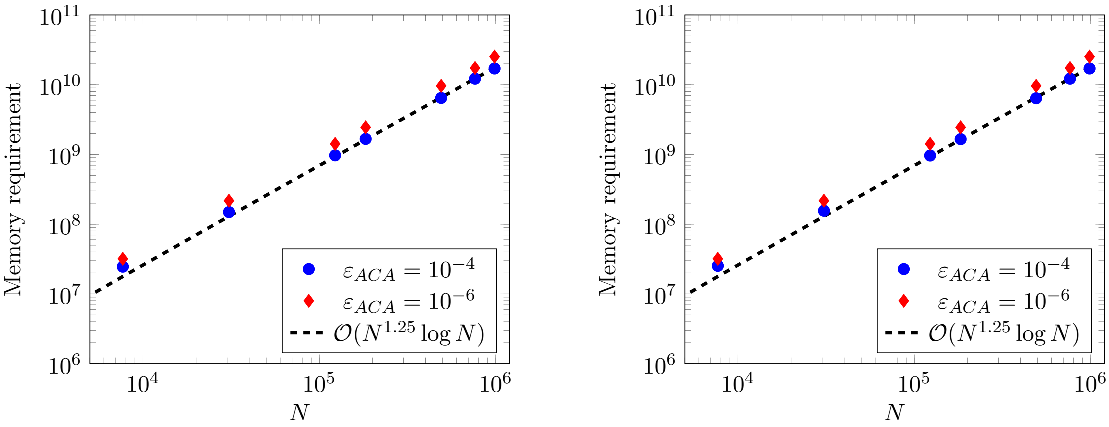
(a) Single-layer ($\varepsilon_{\text{ACA}} = 10^{-4}$)

(b) Single-layer ($\varepsilon_{\text{ACA}} = 10^{-6}$)

(c) Double-layer ($\varepsilon_{\text{ACA}} = 10^{-4}$)

(d) Double-layer ($\varepsilon_{\text{ACA}} = 10^{-6}$)

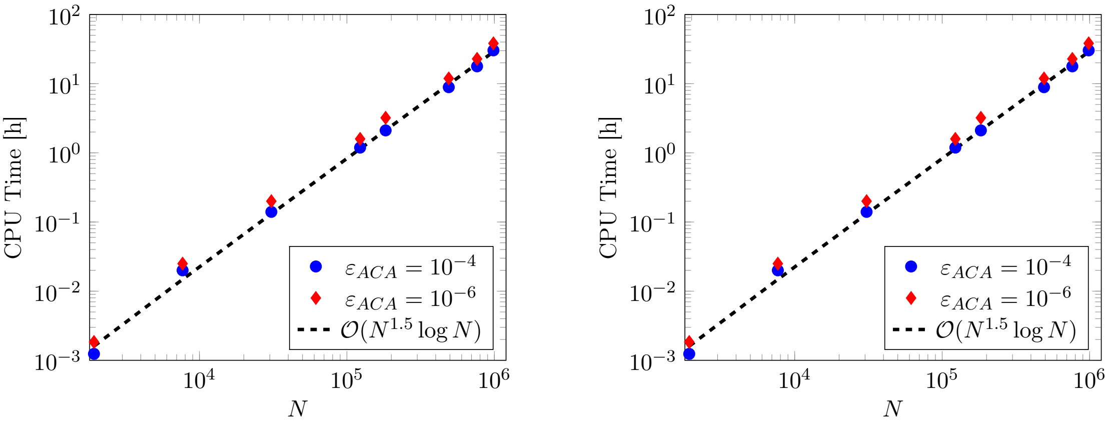
Figure 3.6: Diffraction of an incident P-wave by a unit sphere: maximum rank for fixed density, before and after recompression with truncated SVD.



(a) Single-layer

(b) Double-layer

Figure 3.7: Diffraction of an incident P-wave by a unit sphere: memory requirement (number of entries that we have to store) for the elastodynamic $\mathcal{H}$-matrix based BEM.



(a) Single-layer

(b) Double-layer

Figure 3.8: Diffraction of an incident P-wave by a unit sphere: CPU times (in hours) to compute the $\mathcal{H}$-matrix representation of the BEM matrix.

In the light of these considerations and remembering the algebraic nature of the hierarchical matrices, we have all the elements to extend the fast $\mathcal{H}$-matrix based solvers, introduced in Subsections 2.5.1 and 2.5.3, to 3-D elastodynamic BEM.

3.4.3 $\mathcal{H}$-matrix Based Iterative Solver

The first solver proposed for 3-D elastodynamic BEM is an iterative solver, using the fast $\mathcal{H}$-matrix/vector product to accelerate the GMRES based iterative solver. The accuracy of the solver is studied with the relative error defined by

$$\varepsilon_{\mathcal{H}} := \frac{\|\mathbf{t}^{tot} - \mathbf{t}^{tot}_{\mathcal{H}}\|_2}{\|\mathbf{t}^{tot}\|_2}, \tag{3.40}$$

where $\mathbf{t}^{tot}$ is the analytical solution of (3.18) or (3.19) evaluated at the approximation nodes, and $\mathbf{t}^{tot}_{\mathcal{H}}$ is the numerical solution obtained with the $\mathcal{H}$-matrix based iterative solver. In the following, the surface of the sphere is discretized with a fixed density of $d_{\lambda_{\text{S}}} = 10$ degrees of freedom per S-wavelength.

In Tables 3.9 and 3.10, we report the accuracy of the $\mathcal{H}$-matrix based iterative solver. For both the single-layer and the double-layer operators, a threshold $\varepsilon_{\text{ACA}} = 10^{-4}$ is used in the ACA algorithm to obtain low-rank approximations of the admissible blocks in the $\mathcal{H}$-matrix representation of the BEM matrix. The values of $\varepsilon_{\mathcal{H}}$ show that the error on the solution is kept almost constant ($\varepsilon_{\mathcal{H}}$ varies at most by a factor of two in Table 3.10) in the range analyzed. As a result, it is useless to use a value of $\varepsilon_{\text{GMRES}}$ smaller than ε_{ACA}, because it is impossible to solve the BEM system with a more accurate result than the approximation of the matrix. In addition, the quality of the approximation is better in the case of the single-layer potential. The reason is that the double-layer requires more discretization points to achieve the same level of precision, due to its stronger singularity.

	$\varepsilon_{\mathcal{H}}$		
N	$\varepsilon_{\text{GMRES}} = 10^{-4}$	$\varepsilon_{\text{GMRES}} = 10^{-6}$	$\varepsilon_{\text{GMRES}} = 10^{-9}$
7 686	7.62×10^{-3}	7.62×10^{-3}	7.62×10^{-3}
30 726	5.77×10^{-3}	5.77×10^{-3}	5.77×10^{-3}
183 099	7.96×10^{-3}	7.96×10^{-3}	7.96×10^{-3}
490 629	9.40×10^{-3}	9.40×10^{-3}	9.40×10^{-3}

Table 3.9: Diffraction of an incident P-wave by a unit sphere: accuracy of the $\mathcal{H}$-matrix based iterative solver (Single-layer, $\varepsilon_{\text{ACA}} = 10^{-4}$).

	$\varepsilon_{\mathcal{H}}$		
N	$\varepsilon_{\text{GMRES}} = 10^{-4}$	$\varepsilon_{\text{GMRES}} = 10^{-6}$	$\varepsilon_{\text{GMRES}} = 10^{-9}$
7 686	7.95×10^{-2}	7.95×10^{-2}	7.95×10^{-2}
30 726	1.59×10^{-1}	1.59×10^{-1}	1.59×10^{-1}
183 099	7.59×10^{-2}	7.59×10^{-2}	7.59×10^{-2}
490 629	1.70×10^{-1}	1.70×10^{-1}	1.70×10^{-1}

Table 3.10: Diffraction of an incident P-wave by a unit sphere: accuracy of the $\mathcal{H}$-matrix based iterative solver (Double-layer, $\varepsilon_{\text{ACA}} = 10^{-4}$).

In Figures 3.9 and 3.10, we compare the accuracy of the proposed $\mathcal{H}$-matrix based iterative solver with the accuracy of a Multi Level Fast Multipole based iterative solver [37]. As just explained, we consider only the value $\varepsilon_{\text{GMRES}} = 10^{-4}$, because $\varepsilon_{\text{ACA}} = 10^{-4}$. In the low-frequency regime, the $\mathcal{H}$-matrix based solver gives a good approximation of the analytical solution while the Fast Multipole based iterative solver is less precise. At higher frequencies, the results obtained by the two solvers are similar. This is not surprising because it is known that the reformulation, in terms of plane wave expansions, of the elastodynamic fundamental solutions in terms of products of functions of $\mathbf{x}$ and $\mathbf{y}$ used in FMM (involving Legendre polynomials and spherical Henkel functions of the first kind) deteriorates in the low-frequency case.

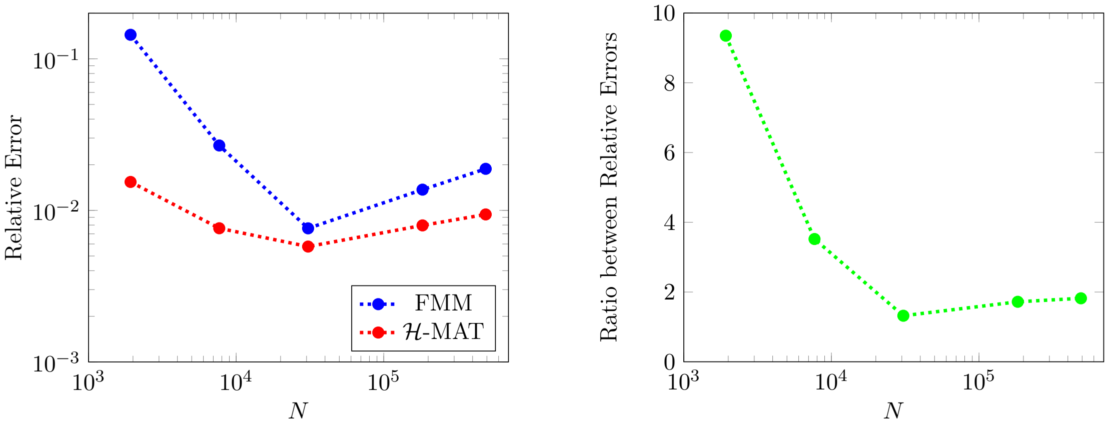


Figure 3.9: Diffraction of an incident P-wave by a unit sphere: comparison between $\mathcal{H}$-matrix and FMM based iterative solvers (Single-layer, $\varepsilon_{\text{GMRES}} = 10^{-4}$).

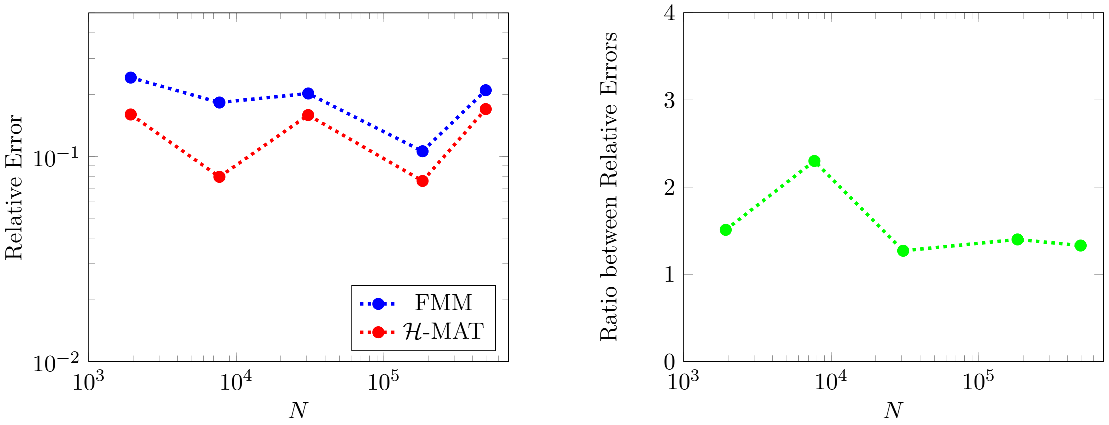
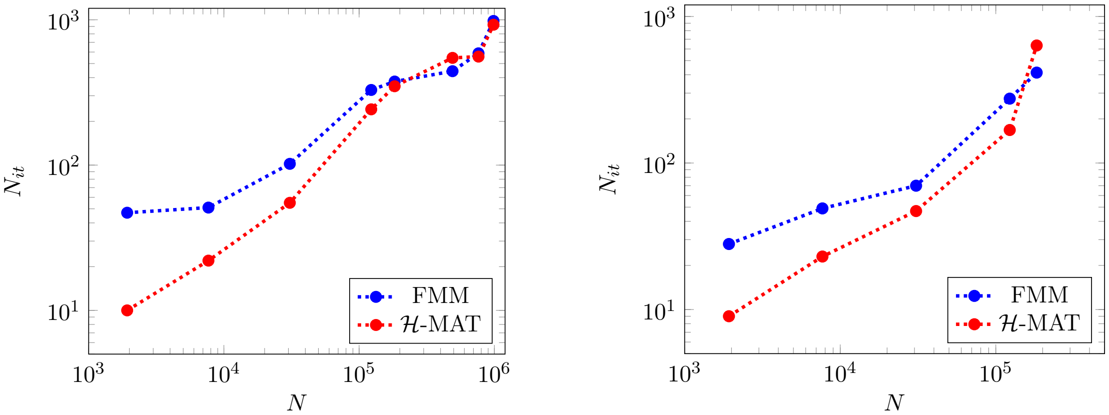


Figure 3.10: Diffraction of an incident P-wave by a unit sphere: comparison between $\mathcal{H}$-matrix and FMM based iterative solvers (Double-layer, $\varepsilon_{\text{GMRES}} = 10^{-4}$).

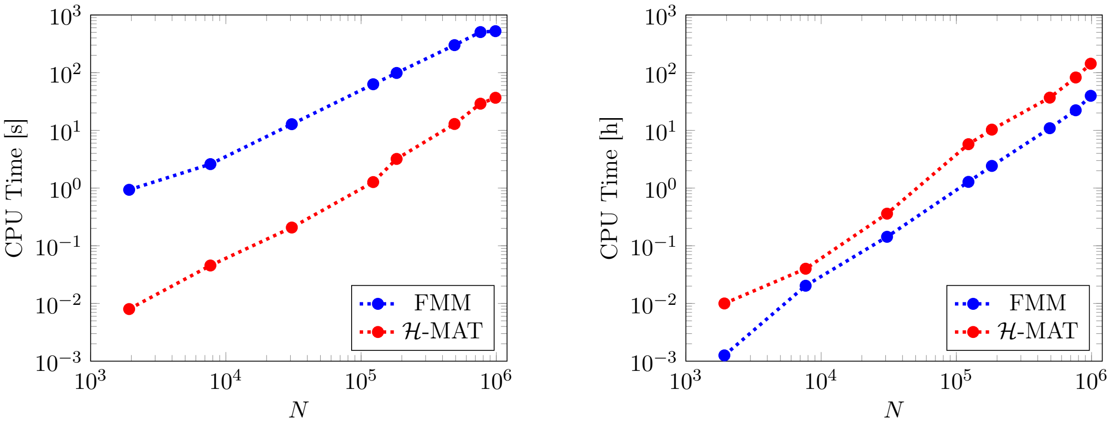
The number of iterations N_{it} required for convergence of the two solvers are reported in Figure 3.11. We observe that in the low-frequency regime the $\mathcal{H}$-matrix based iterative solver converges after a number of iterations smaller than the number of iterations required by the Fast Multipole based iterative solver. At higher frequencies N_{it} is equal for the two solvers and increases due to the combined effect of the mesh refinement and the badly conditioned nature of the BEM matrix. Reducing the iteration count requires a preconditioning strategy, a critical component of the development of efficient iterative BEM solvers. However, it has to be kept in mind that the construction of efficient preconditioners is significantly easier for hierarchical matrices than for Fast Multipole method. In fact, the algebraic nature of $\mathcal{H}$-matrices allows to use the $\mathcal{H}$-matrix arithmetic with a small threshold to compute an $\mathcal{H}$-SPAI (hierarchical Sparse Approximate Inverse) or an $\mathcal{H}$-ILU (hierarchical Incomplete LU decomposition) of the BEM matrix. These preconditioners can be used to accelerate the convergence not only of the $\mathcal{H}$-matrix based iterative solver, but also of the Fast Multipole based iterative solver [115].



(a) Single-layer ($\varepsilon_{\text{GMRES}} = 10^{-4}$) (b) Double-layer ($\varepsilon_{\text{GMRES}} = 10^{-4}$)

Figure 3.11: Diffraction of an incident P-wave by a unit sphere: comparison of the number of iterations of $\mathcal{H}$-matrix and FM based iterative solvers.

In Figure 3.12, we compare the performance of the $\mathcal{H}$-matrix based and the Fast Multipole based iterative solvers. In particular, in Figure 3.12a we show the computation time for a single matrix/vector product, while in Figure 3.12b we present the total computation time to solve the BEM system. These results are relative to the single-layer operator and are computed setting $\varepsilon_{\text{ACA}} = 10^{-4}$ and, consequently, $\varepsilon_{\text{GMRES}} = 10^{-4}$. We observe that the total computation time to solve the BEM system is of the same order for the two solvers (it varies at most by a factor of four), even if the computation time for a single matrix/vector product is significantly faster for $\mathcal{H}$-matrix based iterative solver (almost two orders of magnitude less than the Fast Multipole based iterative solver). This phenomenon can be explained, because the Fast Multipole based iterative solver does not assemble the BEM matrix, while the time needed to assemble the $\mathcal{H}$-matrix representation of the BEM matrix is predominant during the $\mathcal{H}$-matrix based iterative solution process. Although the $\mathcal{H}$-matrix based iterative solver is almost four times slower than the Fast Multipole based iterative solver, the first is more advantageous in the context of elastodynamic BEM problems, if we keep in mind that the implementation and parallelization is much easier for hierarchical matrices compared to fast multipole method. Additionally, the $\mathcal{H}$-matrix based iterative solver is more suitable for dealing with multiple right hand side problems, because the $\mathcal{H}$-matrix representation of the BEM matrix is assembled only once.



(a) Single matrix/vector product

(b) Solution of BEM system

Figure 3.12: Diffraction of an incident P-wave by a unit sphere: comparison between the computation time (a) for a single matrix/vector product and (b) to solve the BEM system (Single-layer operator).

3.4.4 $\mathcal{H}$-matrix Based Direct Solver

The large number of iterations of the $\mathcal{H}$-matrix based iterative solver is the principal motivation for using the $\mathcal{H}$-matrix based arithmetic to develop a fast direct solver or an efficient preconditioner. The $\mathcal{H}$-LU factorization (see Subsection 2.5.3) is particularly suited to fulfill these two tasks.

In Tables 3.11 and 3.12, the precision of the $\mathcal{H}$-matrix based iterative and direct solvers is compared. For both the single-layer and the double-layer operators, a threshold $\varepsilon_{\text{ACA}} = 10^{-4}$ is used in the ACA algorithm to obtain low-rank approximations. The results presented are relative to four different values of ε_{LU}. As explained in Subsection 2.5.3, this parameter is the prescribed accuracy assigned to perform the $\mathcal{H}$-matrix based arithmetic during the $\mathcal{H}$-LU factorization. The same accuracy is observed for the iterative and direct solvers. In the case of the iterative solver, the last three columns of Tables 3.11 and 3.12 prove that it is not judicious to choose ε_{LU} smaller than ε_{ACA}, because the precision of the approximation of the low-rank blocks dominates the error on the system solution. Since in the case of scattering problem modeled by (3.18) the $\mathcal{H}$-matrix based iterative solver need 547 iterations to solve a problem of size $N = 490\ 629$, these results incentivize to treat 3D elastodynamic problems with a direct approach or to use the $\mathcal{H}$-LU factorization as an efficient preconditioner for the $\mathcal{H}$-matrix or Fast Multipole based iterative solvers.

N	Iterative	Direct			
		$\varepsilon_{LU} = 10^{-2}$	$\varepsilon_{LU} = 10^{-4}$	$\varepsilon_{LU} = 10^{-6}$	$\varepsilon_{LU} = 10^{-9}$
7 686	7.62×10^{-3}	7.70×10^{-2}	7.61×10^{-3}	7.61×10^{-3}	7.61×10^{-3}
30 726	5.77×10^{-3}	1.16×10^{-1}	5.77×10^{-3}	5.77×10^{-3}	5.77×10^{-3}
183 099	7.96×10^{-3}	1.87×10^{-1}	7.98×10^{-3}	7.98×10^{-3}	7.98×10^{-3}
490 629	9.40×10^{-3}	2.07×10^{-1}	9.40×10^{-3}	9.40×10^{-3}	9.40×10^{-3}

Table 3.11: Diffraction of an incident P-wave by a unit sphere: accuracy of the $\mathcal{H}$-matrix based solvers (Single-layer, $\varepsilon_{\text{ACA}} = 10^{-4}$).

N	Iterative	Direct			
		$\varepsilon_{LU} = 10^{-2}$	$\varepsilon_{LU} = 10^{-4}$	$\varepsilon_{LU} = 10^{-6}$	$\varepsilon_{LU} = 10^{-9}$
7 686	7.95×10^{-2}	1.39×10^{-1}	7.95×10^{-2}	7.95×10^{-2}	7.95×10^{-2}
30 726	1.59×10^{-1}	7.50×10^{-1}	1.59×10^{-1}	1.59×10^{-1}	1.59×10^{-1}
183 099	7.59×10^{-2}	7.09×10^{-1}	7.59×10^{-2}	7.59×10^{-2}	7.59×10^{-2}
490 629	1.70×10^{-1}	7.77×10^{-1}	1.70×10^{-1}	1.70×10^{-1}	1.70×10^{-1}

Table 3.12: Diffraction of an incident P-wave by a unit sphere: accuracy of the $\mathcal{H}$-matrix based solvers (Double-layer, $\varepsilon_{\text{ACA}} = 10^{-4}$).

Now, we investigate the CPU time to compute the $\mathcal{H}$-LU factorization and to solve the BEM system. In Tables 3.13 and 3.14, we report the recorded CPU times to compute the $\mathcal{H}$-LU factorization, for both the single-layer and the double-layer operators. We consider only the value $\varepsilon_{\text{LU}} = 10^{-4}$, because $\varepsilon_{\text{ACA}} = 10^{-4}$. We note that the dense BEM matrices can be factorized in modest CPU time, modest memory usage and with prescribed accuracy satisfied. In this light, the $\mathcal{H}$-matrix based direct solver is again very advantageous when we treat problems with multiple right hand sides.

N	$T(\mathcal{H})$	$\mathcal{H}$-LU (CPU Time [h])
7 686	4.71×10^{-1}	3.00×10^{-2}
30 726	2.06×10^{-1}	3.80×10^{-1}
183 099	5.89×10^{-2}	$1.59 \times 10^{+1}$
490 629	3.29×10^{-2}	$6.34 \times 10^{+1}$

Table 3.13: Diffraction of an incident P-wave by a unit sphere: Compression rate and CPU time to compute the $\mathcal{H}$-LU factorization (in hours). Single-layer, $\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}} = 10^{-4}$.

N	$T(\mathcal{H})$	$\mathcal{H}$-LU (CPU Time [h])
7 686	4.97×10^{-1}	4.00×10^{-2}
30 726	2.26×10^{-1}	4.40×10^{-1}
183 099	6.00×10^{-2}	$1.73 \times 10^{+1}$
490 629	3.45×10^{-2}	$6.79 \times 10^{+1}$

Table 3.14: Diffraction of an incident P-wave by a unit sphere: Compression rate and CPU time to compute the $\mathcal{H}$-LU factorization (in hours). Double-layer, $\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}} = 10^{-4}$.

Finally, in Tables 3.15 and 3.16, we present the recorder CPU times to solve the BEM system, for both the single-layer and the double-layer operators. Again, we consider only the value $\varepsilon_{\text{LU}} = 10^{-4}$, because $\varepsilon_{\text{ACA}} = 10^{-4}$. In particular, we observe that the CPU times to solve the BEM system is negligible, compared with the CPU time to compute the $\mathcal{H}$-LU factorization.

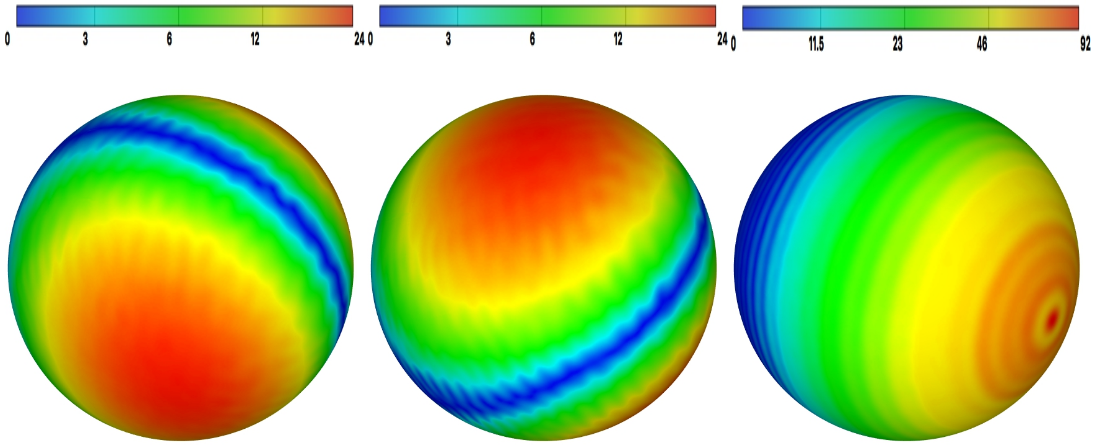
N	Sol (CPU Time [s])	${}^{\text{Sol}}/_{\mathcal{H}\text{-LU}}$
7 686	4.80×10^{-2}	4.00×10^{-4}
30 726	2.96×10^{-1}	2.16×10^{-4}
183 099	$4.60 \times 10^{+0}$	8.04×10^{-5}
490 629	$1.98 \times 10^{+1}$	8.68×10^{-5}

Table 3.15: Diffraction of an incident P-wave by a unit sphere: CPU time to solve the system (in seconds) and comparison with the CPU time to compute the $\mathcal{H}$-LU factorization. Single-layer, $\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}} = 10^{-4}$.

N	Sol (CPU Time [s])	${}^{\text{Sol}}/_{\mathcal{H}\text{-LU}}$
7 686	5.20×10^{-2}	3.61×10^{-4}
30 726	3.16×10^{-1}	2.00×10^{-4}
183 099	$4.72 \times 10^{+0}$	7.58×10^{-5}
490 629	$2.09 \times 10^{+1}$	8.55×10^{-5}

Table 3.16: Diffraction of an incident P-wave by a unit sphere: CPU time to solve the system (in seconds) and comparison with the CPU time to compute the $\mathcal{H}$-LU factorization. Double-layer, $\varepsilon_{\text{ACA}} = \varepsilon_{\text{LU}} = 10^{-4}$.

With the purpose of giving a purely qualitative representation of the results obtained by the $\mathcal{H}$-matrix based direct solver (single-layer), in Figure 3.13 we show the solution of (3.18), modeling the diffraction of an incident P-wave by a unit sphere. The surface of the sphere is discretized with 372 082 triangular boundary elements and 163 543 collocation nodes are used. Consequently, the number of degrees of freedom of the problems is $N = 490\ 629$. A hierarchical cluster tree is constructed with a minimum number of elements $N_{\text{LEAF}} = 100$, resulting in 4 807 clusters and 13 cluster levels. The parameter $\eta = 3$ is used in the admissibility condition with, consequently, 82 517 block clusters in the block cluster tree. A threshold $\varepsilon_{\text{ACA}} = 10^{-4}$ has been used in the ACA algorithm to obtain low-rank approximations of the blocks corresponding to admissible cluster pairs, while a tolerance $\varepsilon_{\text{LU}} = 10^{-4}$ is used in the $\mathcal{H}$-LU factorization.



(a) $|u_1|$ (b) $|u_2|$ (c) $|u_3|$

Figure 3.13: Diffraction of an incident P-wave by a unit sphere: graphic representation of the magnitude of the three components of the solution of (3.18).

3.4.5 Error Estimate to Certify the Results of the $\mathcal{H}$-LU Direct Solver

The $\mathcal{H}$-LU direct solver is based on a heuristic method (the Adaptive Cross Approximation) to perform the low-rank approximations of the admissible blocks. Additionally, during the $\mathcal{H}$-LU factorization a further degree of approximation is introduced, due to the computation of addition and multiplication not only between classical matrices, but also between $\mathcal{H}$-matrices. As explained in Subsection 2.5.2, given three compatible cluster pairs (τ, σ), (τ, ζ), (ζ, σ) in the cluster tree $T_{I \times I}$, to compute $A_{\tau,\sigma} \leftarrow A_{\tau,\sigma} - A_{\tau,\zeta} A_{\zeta,\sigma}$ we have to distinguish between 27 different cases, many of which require operations that are not exact but approximated up to a given accuracy ε_{LU} (for example, the conversion of a full matrix or an $\mathcal{H}$-matrix into a low-rank matrix). Furthermore, the $\mathcal{H}$-LU factorization process is performed going up and down in the $\mathcal{H}$-matrix representation and, consequently, the error may propagate from lower to upper levels. For these reasons, it is important to propose a simple and efficient way to certify the results obtained, when the analytical solution of the problem is unknown.

We start considering an initial system $A\mathbf{z} = \mathbf{b}$. We denote $\tilde{\mathbf{z}}$ the approximated solution of $A\mathbf{z} = \mathbf{b}$, obtained by replacing A with its $\mathcal{H}$-matrix representation $A_{\mathcal{H}}$, i.e. $\tilde{\mathbf{z}}$ is the numerical solution of the linear system $A_{\mathcal{H}}\mathbf{z} = \mathbf{b}$. Since

$$\mathbf{b} - A\tilde{\mathbf{z}} = \mathbf{b} - A_{\mathcal{H}}\tilde{\mathbf{z}} + A_{\mathcal{H}}\tilde{\mathbf{z}} - A\tilde{\mathbf{z}}, \tag{3.41}$$

we observe that

$$\frac{\|\mathbf{b}-A\tilde{\mathbf{z}}\|_2}{\|\mathbf{b}\|_2} \leq \frac{1}{\|\mathbf{b}\|_2}\left(\|\mathbf{b}-A_{\mathcal{H}}\tilde{\mathbf{z}}\|_2 + \|A-A_{\mathcal{H}}\|_2\|\tilde{\mathbf{z}}\|_2\right). \tag{3.42}$$

As for every square matrix we have $\|\cdot\|_2 \leq \|\cdot\|_F$, the following estimate holds:

$$\frac{\|\mathbf{b}-A\tilde{\mathbf{z}}\|_2}{\|\mathbf{b}\|_2} \leq \frac{1}{\|\mathbf{b}\|_2}\left(\|\mathbf{b}-A_{\mathcal{H}}\tilde{\mathbf{z}}\|_2 + \|A-A_{\mathcal{H}}\|_F\|\tilde{\mathbf{z}}\|_2\right) := \frac{1}{\|\mathbf{b}\|_2}\left(\delta + \delta_{\mathcal{H}}\|\tilde{\mathbf{z}}\|_2\right). \tag{3.43}$$

Above, $\delta_{\mathcal{H}} := \|A-A_{\mathcal{H}}\|_F$ estimates the accuracy of the $\mathcal{H}$-matrix representation, i.e. the precision of the approximation of the low-rank blocks influenced by the parameter ε_{ACA}; $\delta := \|\mathbf{b}-A_{\mathcal{H}}\tilde{\mathbf{z}}\|_2$ measures the quality of the solution and, consequently, gives informations on the stability of the $\mathcal{H}$-LU factorization, influenced by the parameter ε_{LU}. The evaluations of δ, $\|\mathbf{b}\|_2$ and $\|\tilde{\mathbf{z}}\|_2$ reduce to the computation of the norm of a vector. The evaluation of $\delta_{\mathcal{H}}$ is performed in Frobenius norm to reduce the computational cost [2], and this term does not depend on the right-hand side, meaning that for multiple right-hand sides the computation of this error estimate is not expensive. In the following, we refer to the proposed error estimate as

$$\mathcal{I}(\delta_{\mathcal{H}};\delta) := \frac{1}{\|\mathbf{b}\|_2}\left(\delta + \delta_{\mathcal{H}}\|\tilde{\mathbf{z}}\|_2\right). \tag{3.44}$$

Acoustic Case. The first numerical test performed to validate the proposed error estimate consists in considering the matrix A coming from the evaluation of the modified Green's function $G_h(\mathbf{x},\mathbf{y};\kappa)$, defined in (3.22), on the discretization nodes used to discretize the bar in Figure 3.2. The value of h is chosen equal to the discretization step and the density of the mesh is $d_\lambda = 10$ points per wavelength. In this case the right-hand side is an arbitrary $\mathbf{b} \in \mathbb{R}^N$. The $\mathcal{H}$-matrix representation of A is constructed with a minimum number of elements $N_{\text{LEAF}} = 100$ in the hierarchical cluster tree and the parameter $\eta = 3$.

In Table 3.17, we present the true error, the proposed error estimate, the error on the solution of the system and the accuracy of the $\mathcal{H}$-matrix representation, for a frequency range $\kappa = 4\pi - 16\pi$ and, consequently, $N = 2\ 310 - 36\ 870$. Different values of the parameters ε_{ACA} and ε_{LU} are considered. We observe that $\mathcal{I}(\delta_{\mathcal{H}};\delta)$ represents a good measure of the true error. Additionally, the error on the $\mathcal{H}$-matrix representation $\|A-A_{\mathcal{H}}\|_F$ affects mainly the error estimate. In particular, the parameter ε_{ACA} is a good measure of the error on the system solution, even when ε_{LU} is smaller. In fact, relaxing ε_{ACA}, the quality of the solution becomes worse. These considerations allow to conclude that it is not necessary to choose ε_{LU} smaller than ε_{ACA}.

Elastodynamic Case. Now, we validate the proposed error estimate for the direct solver, considering the problem of the diffraction of an incident plane P-wave by a unit sphere, modeled by (3.18) and (3.19). In this case, the circular frequency ω is adapted to the mesh length to keep the density of points per S-wavelength fixed for different

[2] (3.43) obviously holds with $\delta_{\mathcal{H}} := \|A-A_{\mathcal{H}}\|_2$.

meshes. Thus, an increase of the number of degrees of freedom N leads to an increase of the dimensions γ_{λ_S} compared to the S-wavelength.

In Tables 3.18 and 3.19, we present the true error, the proposed error estimate, the error on the solution of the system and the accuracy of the $\mathcal{H}$-matrix representation, for both the single-layer and the double-layer operators. We set the parameters $\varepsilon_{\text{ACA}} = 10^{-4}$ or $\varepsilon_{\text{ACA}} = 10^{-6}$ and $\varepsilon_{\text{LU}} = 10^{-4}$ or $\varepsilon_{\text{LU}} = 10^{-6}$, in order to study their influence on the error estimate. First of all we note that the level of error introduced by the ACA, i.e. $\|A - A_{\mathcal{H}}\|_F$, and the quality of the solution, i.e. $\|\mathbf{b} - A_{\mathcal{H}}\tilde{\mathbf{z}}\|_2$, can be controlled respectively by ε_{ACA} (as we expect from Table 3.6) and ε_{LU}. Then, since the error on the $\mathcal{H}$-matrix representation affects mainly $\mathcal{I}(\delta_{\mathcal{H}};\delta)$, we expect that it is impossible to solve the system and to obtain a more accurate result than the approximation of the matrix (as we expect from Tables 3.11 and 3.12). Thus, it is not necessary to choose ε_{LU} smaller than ε_{ACA}, because the precision of the approximation of the low-rank blocks represents a good measure of the error on the system solution. Also, in order to achieve a desired accuracy, we choose a value of ε_{ACA} and ε_{LU} which is one order of magnitude higher than our goal. All these considerations are valid for both the single-layer and the double-layer operators. We conclude this analysis observing that $\mathcal{I}(\delta_{\mathcal{H}};\delta)$ represents a good measure of the true error, because they are of the same order ($\mathcal{I}(\delta_{\mathcal{H}};\delta)$ varies at most by a factor of two).

N	κ	$\varepsilon_{ACA},\ \varepsilon_{LU}$	$\frac{\lVert b-A\tilde{\mathbf{z}}\rVert_2}{\lVert b\rVert_2}$	$\mathcal{I}(\delta_{\mathcal{H}};\delta)$	$\lVert b - A_{\mathcal{H}}\tilde{\mathbf{z}}\rVert_2$	$\lVert A_{\mathcal{H}} - A\rVert_F$
2 310	4π	10^{-4}, 10^{-4}	6.97×10^{-15}	5.01×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
9 222	8π	10^{-4}, 10^{-4}	9.89×10^{-6}	3.57×10^{-4}	8.71×10^{-5}	1.17×10^{-4}
20 742	13π	10^{-4}, 10^{-4}	1.07×10^{-5}	4.83×10^{-4}	1.08×10^{-4}	1.33×10^{-4}
36 870	16π	10^{-4}, 10^{-4}	1.63×10^{-5}	7.57×10^{-4}	1.89×10^{-4}	2.84×10^{-4}
2 310	4π	10^{-6}, 10^{-4}	6.97×10^{-15}	5.01×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
9 222	8π	10^{-6}, 10^{-4}	3.99×10^{-6}	8.96×10^{-6}	4.97×10^{-4}	1.22×10^{-6}
20 742	13π	10^{-6}, 10^{-4}	5.02×10^{-6}	1.14×10^{-5}	6.27×10^{-4}	1.56×10^{-6}
36 870	16π	10^{-6}, 10^{-4}	6.00×10^{-6}	2.92×10^{-5}	6.93×10^{-4}	2.01×10^{-6}
2 310	4π	10^{-4}, 10^{-6}	6.97×10^{-15}	5.01×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
9 222	8π	10^{-4}, 10^{-6}	9.94×10^{-6}	4.02×10^{-4}	3.13×10^{-7}	1.17×10^{-4}
20 742	13π	10^{-4}, 10^{-6}	1.14×10^{-5}	5.31×10^{-4}	7.89×10^{-6}	1.33×10^{-4}
36 870	16π	10^{-4}, 10^{-6}	1.68×10^{-5}	6.73×10^{-4}	9.04×10^{-6}	2.84×10^{-4}
2 310	4π	10^{-6}, 10^{-6}	6.97×10^{-15}	5.01×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
9 222	8π	10^{-6}, 10^{-6}	4.09×10^{-6}	9.95×10^{-6}	5.55×10^{-6}	1.22×10^{-6}
20 742	13π	10^{-6}, 10^{-6}	5.52×10^{-5}	3.27×10^{-5}	6.78×10^{-6}	1.56×10^{-6}
36 870	16π	10^{-6}, 10^{-6}	6.61×10^{-5}	5.83×10^{-5}	8.40×10^{-6}	2.01×10^{-6}

Table 3.17: Discretization of the modified 3D Helmholtz Green's function: error estimate for fixed density of points.

N	$\omega\ (d_{\lambda_s})$	$\varepsilon_{ACA},\ \varepsilon_{LU}$	$\frac{\lVert b-A\tilde{\mathbf{z}}\rVert_2}{\lVert b\rVert_2}$	$\mathcal{I}(\delta_{\mathcal{H}};\delta)$	$\lVert b-A_{\mathcal{H}}\tilde{\mathbf{z}}\rVert_2$	$\lVert A_{\mathcal{H}}-A\rVert_F$
1 926	3(13)	10^{-4}, 10^{-4}	3.61×10^{-15}	3.21×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
7 686	3(27)	10^{-4}, 10^{-4}	9.94×10^{-6}	4.63×10^{-4}	5.84×10^{-5}	7.21×10^{-5}
30 726	3(55)	10^{-4}, 10^{-4}	1.18×10^{-5}	6.37×10^{-4}	1.40×10^{-4}	9.93×10^{-5}
1 926	7(6)	10^{-4}, 10^{-4}	3.87×10^{-15}	4.07×10^{-15}	1.79×10^{-14}	8.56×10^{-17}
7 686	7(12)	10^{-4}, 10^{-4}	3.78×10^{-5}	1.39×10^{-3}	2.30×10^{-4}	9.59×10^{-5}
30 726	7(24)	10^{-4}, 10^{-4}	3.97×10^{-5}	1.70×10^{-3}	5.35×10^{-4}	1.18×10^{-4}
1 926	14(3)	10^{-4}, 10^{-4}	4.67×10^{-15}	5.69×10^{-15}	2.27×10^{-14}	7.16×10^{-17}
7 686	14(6)	10^{-4}, 10^{-4}	1.04×10^{-4}	3.89×10^{-3}	5.99×10^{-4}	1.37×10^{-4}
30 726	14(12)	10^{-4}, 10^{-4}	1.29×10^{-4}	5.50×10^{-3}	1.39×10^{-3}	1.95×10^{-4}
1 926	3(13)	10^{-6}, 10^{-4}	3.61×10^{-15}	3.21×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
7 686	3(27)	10^{-6}, 10^{-4}	4.65×10^{-6}	9.69×10^{-6}	5.88×10^{-5}	7.92×10^{-7}
30 726	3(55)	10^{-6}, 10^{-4}	5.50×10^{-6}	1.29×10^{-5}	1.39×10^{-4}	1.16×10^{-6}
1 926	7(6)	10^{-6}, 10^{-4}	3.87×10^{-15}	4.07×10^{-15}	1.79×10^{-14}	8.56×10^{-17}
7 686	7(12)	10^{-6}, 10^{-4}	1.87×10^{-5}	3.40×10^{-5}	2.37×10^{-4}	1.07×10^{-6}
30 726	7(24)	10^{-6}, 10^{-4}	2.17×10^{-5}	4.13×10^{-5}	5.50×10^{-4}	1.37×10^{-6}
1 926	14(3)	10^{-6}, 10^{-4}	4.67×10^{-15}	5.69×10^{-15}	2.73×10^{-14}	7.15×10^{-17}
7 686	14(6)	10^{-6}, 10^{-4}	4.85×10^{-5}	9.42×10^{-5}	6.14×10^{-4}	1.62×10^{-6}
30 726	14(12)	10^{-6}, 10^{-4}	5.65×10^{-5}	1.16×10^{-4}	1.43×10^{-3}	2.12×10^{-6}
1 926	3(13)	10^{-4}, 10^{-6}	3.61×10^{-15}	3.21×10^{-15}	1.66×10^{-14}	9.36×10^{-17}
7 686	3(27)	10^{-4}, 10^{-6}	9.01×10^{-6}	4.59×10^{-4}	2.97×10^{-7}	7.21×10^{-5}
30 726	3(55)	10^{-4}, 10^{-6}	1.11×10^{-5}	5.61×10^{-4}	9.27×10^{-6}	8.53×10^{-5}
1 926	7(6)	10^{-4},10^{-6}	4.06×10^{-15}	3.87×10^{-15}	1.79×10^{-14}	8.56×10^{-17}
7 686	7(12)	10^{-4}, 10^{-6}	3.34×10^{-5}	1.37×10^{-3}	9.72×10^{-7}	9.59×10^{-5}
30 726	7(24)	10^{-4}, 10^{-6}	3.38×10^{-5}	1.68×10^{-3}	2.26×10^{-6}	1.18×10^{-4}
1 926	14(3)	10^{-4}, 10^{-6}	4.67×10^{-15}	5.69×10^{-15}	2.27×10^{-14}	7.16×10^{-17}
7 686	14(6)	10^{-4}, 10^{-6}	9.20×10^{-5}	3.85×10^{-3}	2.60×10^{-6}	1.37×10^{-4}
30 726	14(12)	10^{-4}, 10^{-6}	1.17×10^{-4}	5.44×10^{-3}	5.81×10^{-6}	1.95×10^{-4}

Table 3.18: Diffraction of an incident P-wave by a unit sphere: error estimate for fixed frequency (Single-layer)

N	$\omega\ (d_{\lambda_s})$	$\varepsilon_{ACA},\ \varepsilon_{LU}$	$\frac{\lVert b-A\tilde{\mathbf{z}}\rVert_2}{\lVert b\rVert_2}$	$\mathcal{I}(\delta_{\mathcal{H}};\delta)$	$\lVert b-A_{\mathcal{H}}\tilde{\mathbf{z}}\rVert_2$	$\lVert A_{\mathcal{H}}-A\rVert_F$
1 926	3(13)	10^{-4}, 10^{-4}	3.29×10^{-15}	3.61×10^{-15}	7.67×10^{-14}	2.88×10^{-16}
7 686	3(27)	10^{-4}, 10^{-4}	8.47×10^{-4}	3.85×10^{-4}	9.79×10^{-5}	2.72×10^{-4}
30 726	3(55)	10^{-4}, 10^{-4}	1.03×10^{-5}	4.88×10^{-4}	2.29×10^{-4}	3.37×10^{-4}
1 926	7(6)	10^{-4}, 10^{-4}	3.45×10^{-15}	3.52×10^{-15}	1.87×10^{-13}	3.66×10^{-16}
7 686	7(12)	10^{-4}, 10^{-4}	1.94×10^{-5}	8.98×10^{-4}	1.11×10^{-3}	6.03×10^{-4}
30 726	7(24)	10^{-4}, 10^{-4}	2.17×10^{-5}	1.14×10^{-3}	2.61×10^{-3}	7.61×10^{-4}
1 926	14(3)	10^{-4}, 10^{-4}	3.25×10^{-15}	3.36×10^{-15}	3.38×10^{-13}	5.23×10^{-16}
7 686	14(6)	10^{-4}, 10^{-4}	7.16×10^{-5}	2.50×10^{-3}	6.46×10^{-3}	1.94×10^{-3}
30 726	14(12)	10^{-4}, 10^{-4}	6.92×10^{-5}	3.18×10^{-3}	1.54×10^{-2}	2.43×10^{-3}
1 926	3(13)	10^{-6}, 10^{-4}	3.29×10^{-15}	3.26×10^{-15}	7.67×10^{-14}	2.88×10^{-16}
7 686	3(27)	10^{-6}, 10^{-4}	1.97×10^{-6}	6.58×10^{-6}	1.06×10^{-4}	3.27×10^{-6}
30 726	3(55)	10^{-6}, 10^{-4}	2.25×10^{-6}	8.40×10^{-6}	2.42×10^{-4}	4.27×10^{-6}
1 926	7(6)	10^{-6}, 10^{-4}	3.45×10^{-15}	3.52×10^{-15}	1.87×10^{-13}	3.66×10^{-16}
7 686	7(12)	10^{-6}, 10^{-4}	9.30×10^{-6}	1.98×10^{-5}	1.16×10^{-3}	7.12×10^{-6}
30 726	7(24)	10^{-6}, 10^{-4}	1.07×10^{-5}	2.41×10^{-5}	2.66×10^{-3}	9.08×10^{-6}
1 926	14(3)	10^{-6}, 10^{-4}	3.25×10^{-15}	3.36×10^{-15}	3.38×10^{-13}	5.23×10^{-16}
7 686	14(6)	10^{-6}, 10^{-4}	2.62×10^{-5}	5.05×10^{-5}	6.56×10^{-3}	1.90×10^{-5}
30 726	14(12)	10^{-6}, 10^{-4}	3.14×10^{-5}	6.45×10^{-5}	1.58×10^{-2}	2.54×10^{-5}
1 926	3(13)	10^{-4}, 10^{-6}	3.29×10^{-15}	3.26×10^{-15}	7.67×10^{-14}	2.88×10^{-16}
7 686	3(27)	10^{-4}, 10^{-6}	8.22×10^{-6}	3.84×10^{-4}	6.12×10^{-7}	2.72×10^{-4}
30 726	3(55)	10^{-4}, 10^{-6}	1.01×10^{-5}	4.86×10^{-4}	1.49×10^{-6}	3.37×10^{-4}
1 926	7(6)	10^{-4},10^{-6}	3.45×10^{-15}	3.52×10^{-15}	1.87×10^{-13}	3.66×10^{-16}
7 686	7(12)	10^{-4}, 10^{-6}	1.72×10^{-5}	8.89×10^{-4}	3.82×10^{-6}	6.03×10^{-4}
30 726	7(24)	10^{-4}, 10^{-6}	1.89×10^{-5}	1.13×10^{-3}	9.39×10^{-6}	7.61×10^{-4}
1 926	14(3)	10^{-4}, 10^{-6}	3.25×10^{-15}	3.36×10^{-15}	3.38×10^{-14}	3.38×10^{-16}
7 686	14(6)	10^{-4}, 10^{-6}	6.66×10^{-5}	2.47×10^{-3}	6.24×10^{-5}	1.94×10^{-3}
30 726	14(12)	10^{-4}, 10^{-6}	6.15×10^{-5}	3.15×10^{-3}	6.04×10^{-5}	2.43×10^{-3}

Table 3.19: Diffraction of an incident P-wave by a unit sphere: error estimate for fixed frequency (Double-layer)

3.5 Parallelization

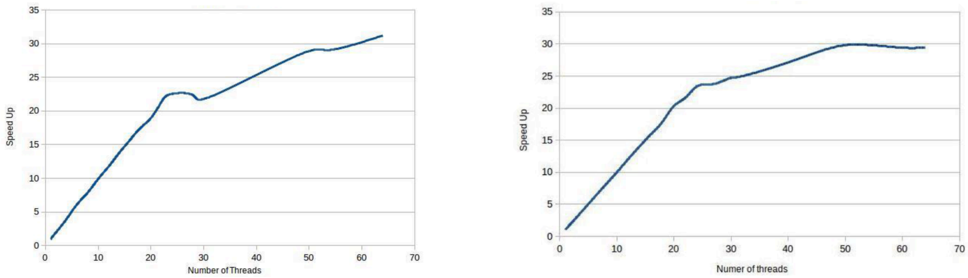
The results presented in this Chapter are relative to a sequential implementation of the $\mathcal{H}$-matrix technique to accelerate the 3D elastodynamic BEM in frequency domain. Even if the CPU times recorded are conformed to theoretical or previous-published results, they are still too large to treat realistic problems. Thus, a parallelization of the code developed is needed. During his internship at laboratory POEMS (ENSTA-ParisTech), Shashank Kumar Anand worked on the parallelization of the construction of the $\mathcal{H}$-matrix representation of the BEM matrix.

There are several parallel programming models in common use. *Shared memory* is an efficient means of passing data between processes. In a shared-memory model, parallel processes share a global address space that they read and write to asynchronously. Asynchronous concurrent access can lead to race conditions and mechanisms such as locks, semaphores and monitors can be used to avoid these. Conventional multi-core processors directly support shared memory, which many parallel programming languages and libraries, such as Cilk, OpenMP and Threading Building Blocks, are designed to exploit.

Shashank Kumar Anand opted for an OpenMP (Open Multi-Processing) implementation. Briefly, we recall that OpenMP uses the fork-join model of parallel execution. All Open MP programs begin as a single process, the master thread. The master thread executes sequentially until the first parallel region construct is encountered. The master thread then creates a team of parallel threads (Fork). The statements in the program that are enclosed by the parallel region construct are then executed in parallel among the various team threads. When the team threads complete the statements in the parallel region construct, they synchronize and terminate, leaving only the master thread (Join). The number of parallel regions and the threads that comprise them are arbitrary.

In Figures 3.14 and 3.15, we report the results obtained on a 2.0-GHz Intel Xeon CPU E7-4820 with 1 TB RAM, having 32 cores with hyper threading. Thus, we have logical 64 threads which mean we can set our maximum number of threads equal to 64. But maximum speed up expected is around 32 (may be slightly more than that) as physical number of cores are 32 in number. The problem considered is the diffraction of an incident plane P-wave by a unit sphere, modeled by the boundary integral equation (3.18). Four different problem sizes are considered: $N = 7\ 686$ (Figure 3.14a), $N = 30\ 726$ (Figure 3.14b), $N = 122\ 886$ (Figure 3.15a) and $N = 183\ 099$ (Figure 3.15b). To accelerate the construction of the $\mathcal{H}$-matrix representation of the BEM matrix, a Dynamic Scheduling strategy was adopted, i.e. each thread will take up a single task only as soon as it gets free. For all the four problem sizes treated, the speed up efficiency is almost around 90%.

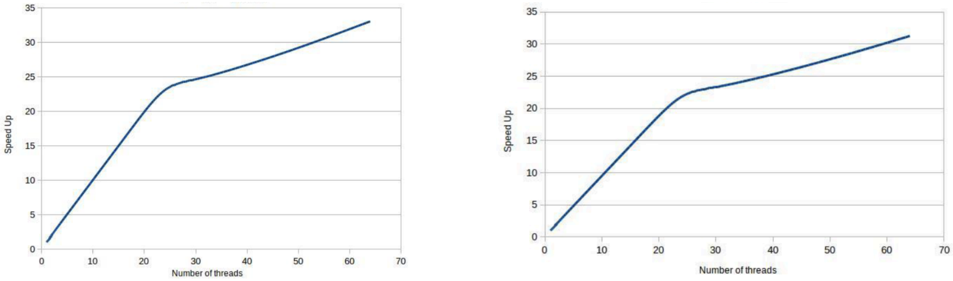
The parallelization of the $\mathcal{H}$-LU factorization is still an open difficult task, because various stages of the algorithm link at least two sub-blocks. Recently, the idea of splitting the $\mathcal{H}$-LU factorization into single tasks and of defining corresponding dependencies to form a Direct Acyclic Graph (DAG), has been presented. This task/DAG based algorithm is able to utilize parallel CPU which are much more efficient compared to the recursive algorithm and, in particular, demonstrates an optimal parallel scaling behavior on many-core systems [112].



(a) Single-layer ($N = 7\ 686$)

(b) Single-layer ($N = 30\ 726$)

Figure 3.14: Diffraction of an incident P-wave by a unit sphere: Dynamic Scheduling Results for the computation of the $\mathcal{H}$-matrix representation of BEM matrix (Parallelization of a Fast Direct Solver for the 3D Elastodynamic Boundary Element Method, Shashank Kumar Anand).



(a) Single-layer ($N = 122\ 886$)

(b) Single-layer ($N = 183\ 099$)

Figure 3.15: Diffraction of an incident P-wave by a unit sphere: Dynamic Scheduling Results for the computation of the $\mathcal{H}$-matrix representation of BEM matrix (Parallelization of a Fast Direct Solver for the 3D Elastodynamic Boundary Element Method, Shashank Kumar Anand).

3.6 Conclusions

In this Chapter, we have developed innovative fast 3D Boundary Element Method solvers based on $\mathcal{H}$-matrices. Taking advantages of published developments for Helmholtz, Laplace and Lamé equations, the $\mathcal{H}$-matrices have been extended to elastodynamics in the frequency domain. The efficiency and the accuracy of the method have been tested in the context of scattering of time-harmonic elastic plane P-waves by a unit sphere, comparing the results with analytical solutions.

Firstly, we have shown that the generalization of the scalar-value Adaptive Cross Approximation (ACA) to the matrix-valued problems presents all the features of standard ACA, i.e. its black-box and self-controlling nature. In particular, the level of error introduced by the ACA can be controlled by one parameter, $\varepsilon_{\mathrm{ACA}}$, representing in practice an accurate measure of the error on the approximation of the whole BEM matrix. However, in order to achieve a desired accuracy, it is better to choose a value of $\varepsilon_{\mathrm{ACA}}$ which is one order of magnitude higher than our goal. Since vectorial ACA gives an almost optimal numerical rank, the optimum rank has to be revealed performing a recompression with truncated SVD. In this way, we have observed a reduction of the memory requirement compared to classical boundary element method. In fact, in our experiments the $\mathcal{H}$-matrix representations of the elastodynamic BEM matrices can be achieved with almost logarithmic linear memory cost, i.e. $\mathcal{O}(N^{1.25}\log N)$.

Due to the purely algebraic nature of $\mathcal{H}$-matrices, approximations of the usual matrix operations (addition, multiplication, inversion, etc...) allow to define fast solvers. We have obtained that the accuracy of the $\mathcal{H}$-matrix/vector product is related to the overall accuracy of the $\mathcal{H}$-matrix representation, because it is impossible to obtain a more accurate result than the approximation of the matrix. In the light of these considerations, the $\mathcal{H}$-matrix/vector product has been used to accelerate the classical matrix/vector product and, consequently, to reduce the time of each iteration of a GMRES based iterative solver, which is the most expensive task for this type of solution method. No preconditioner has been used. The proposed iterative solver has been compared with a Fast Multipole based iterative solver. Although for both solvers a deterioration of the convergence rate (number of iterations) has been observed for large problem sizes, the computation time to solve the BEM system is of the same order for both these iterative solvers. Even if $\mathcal{H}$-matrix based iterative solver is four times slower than the Fast Multipole based iterative solver, it results more advantageous because its implementation and parallelization is easier. In parallel, the $\mathcal{H}$-matrix based arithmetic has been used to approximate the factors of the LU decomposition, in order to reduce the numerical effort of a direct solver. The numerical tests presented have shown that it is not necessary to choose the parameter $\varepsilon_{\mathrm{LU}}$ smaller than $\varepsilon_{\mathrm{ACA}}$, because the precision of the approximation of the low-rank blocks represents a good measure of the error on the system solution. For this last solver an error estimate has been proposed to assess the quality of the solution, when the analytical solution is unknown.

Both the proposed iterative and direct solvers permit to reduce the computational burden in CPU time, for the simulation of wave propagation and allow to run models of size $N = \mathcal{O}(10^6)$ on a personal laptop. In particular, the CPU time observed incentivates to treat 3D elastodynamic problems with a direct approach, especially in the case of problems with multiple right hand sides.

4 Application of Fast BEMs to Forced Vibration Problems

Contents

4.1 Introduction 100

4.2 Seismic Method 100

4.3 Forced Vibration Problem 102

4.3.1 Multi-Domain BEM Formulation 103

4.3.2 Hierarchical Multi-Domain BEM Formulation 106

4.4 Green's Function of a Homogeneous Half-Space 106

4.4.1 Limitations of the Vector Version of the ACA for Half-Space Problems 108

4.4.2 Randomized Singular Value Decomposition 109

4.4.3 $\mathcal{H}$-matrix based BEM for Half-Space Problems 110

4.5 Elastic Half-Space with a Semi-Spherical Alluvial Basin . . 119

4.6 Conclusions 121

4.1 Introduction

In the previous Chapter, we have extended the capabilities of the $\mathcal{H}$-matrices to 3D frequency-domain elastodynamics, validating this technique in the context of the scattering of time-harmonic elastic waves by a bounded obstacle. The presented integral formulations involve the single-layer operator S and the double-layer operator D that are the basis of other integral equations, suitable to treat more general wave propagation phenomena.

This Chapter is aimed at developing an $\mathcal{H}$-matrix based Boundary Element Method (BEM) to simulate forced vibration problems. The Chapter is organized as follows. In Section 4.2 we give the motivation and the industrial context of the problem, recalling the principles of a seismic experiment. Next, in Section 4.3 the mathematical formulation of the problem is presented and we introduce a multi-domain BEM formulation, naturally suited to define $\mathcal{H}$-matrix based solution strategies. The capability of the method to treat a simple forced vibration problem, i.e. the computation of the response of a homogeneous medium to time-harmonic point load, is investigated in Section 4.4. This example is also used to present the randomized Singular Value Decomposition (rand SVD) as an alternative to the vectorial partially-pivoted Adaptive Cross Approximation (ACA) to compute low-rank approximations. Finally, in Section 4.5 we show the efficiency of the $\mathcal{H}$-matrix based BEM to simulate an elastic half-space with topographic irregularities.

4.2 Seismic Method

Detailed informations about the nature and the structure of the shallow surface is required to reduce many uncertainties in oil and gas exploration and production. Unfortunately, little of the planet is accessible to direct observation. In Kola peninsula in Russia drilling has penetrated to depths of up to 12 km ($\simeq$ 0.2% of Earth's radius, 6 371 km), though at great expense. Consequently, most of our understanding is based on indirect inferences, including both geochemical and geophysical observations and laboratory studies of state and properties of materials at high pressures and temperatures.

Parallel to the development of techniques such as computers, GPS positioning, increased number of channels in instrument recording, improvements in processing software, etc., enormous resources (the 95% of all geophysical exploration funds) have been invested in seismic explorations. These sophisticated and powerful geophysical methods, based on the analysis of the information from data obtained by recording elastic wave motion of the ground, enable imaging of geological stratigraphy and structure to depths of several kilometers, and have been fundamental to the mapping of much of the world's hydrocarbon reserves. For this reason, in many major oil companies seismic methods surveying increased exponentially over the past few decades.

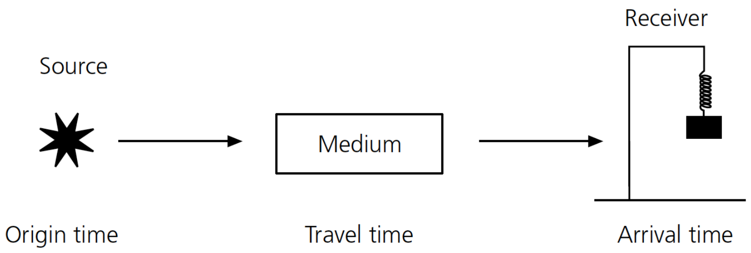


Figure 4.1: Schematic geometry of a seismic experiment

In a seismic data acquisition (see Figure 4.1), an elastic wave-field is emitted by a *seismic source* (there are different kinds of sources in practice but in general impulsive dynamite sources and vibroseis method are considered) at a certain location at the surface of the ground, and propagates in different directions through the surface. Variations in the physical properties of the geological layers cause the seismic signal to reflect and refract at the boundaries of these layers, depending on the velocities (see Table 4.1) and densities of the material on either side of the boundaries. The reflected waves head back towards the surface where *detectors* or *receivers* have been placed to record these waves. Typically, the receivers are deployed in a straight line for 2D work (depth versus distance profiles) or on a grid for 3D work (area versus depth data blocks). Each receiver is a sensitive *geophone* or a combined group of geophones, allowing to transform seismic energy into an electrical voltage. It ordinarily responds to only one component of the ground displacement, velocity or acceleration associated with the passage of a seismic wave (for a seismic reflection survey with P-waves this is the vertical component). A motion-sensitive *transducer* converts ground motion to an electrical signal, that is recorded to a digital *seismograph*. If the *origin time* when the waves left the source is known, their *arrival time* at the receiver gives the *travel time* required to pass through the medium, and hence information about the speed at which they traveled, and thus the physical properties of the medium. After each such *short record* experiment, the source is moved to another location and the measurement is repeated.

One of the key challenges of the shallow surface reconstruction is the presence of *strong noisy events*, i.e. ground-roll or surface waves, that have to be removed from the gathered data before starting the imaging process, since they do not contain any information of the deeper interior of the ground. Historically, two-dimension filters, the so-called *F-K filters*, were used for this type of noise removal. Again for land data, surface and near-surface *topography* can have a tremendous effect on the total response, since the near-surface conditions can vary strongly from location to location. *Static corrections* have been defined as corrections applied to seismic data to compensate for the effects of variations in elevation, weathering thickness, weathering velocity or reference to a datum. The objective is to determine the reflection arrival times which would have been observed if all measurements had been made on a flat plane with no weathering or low-velocity materials presence.

	c_P (km/s)		c_P (km/s)
Unconsolidated materials		*Igneous/Metamorphic rocks*	
Sand (dry)	0.2 - 1.0	Granite	5.5 - 6.0
Sand (water-saturated)	1.5 - 2.0	Gabbro	6.5 - 7.0
Clay	1.0 - 2.5	Ultramafic rocks	7.5 - 8.5
Glacial till (water-saturated)	1.5 - 2.5	Serpentinite	5.5 - 6.5
Permafrost	3.5 - 4.0		
		Pore fluids	
Sedimentary rocks		Air	0.3
Sandstones	2.0 - 6.0	Water	1.4 - 1.5
Tertiary sandstone	2.0 - 2.5	Ice	3.4
Pennant sandstone (Carboniferous)	4.0 - 4.5	Petroleum	1.3 - 1.4
Cambrian quartzite	5.5 - 6.0		
Limestones	2.0 - 6.0	*Other materials*	
Cretaceous chalk	2.0 - 2.5	Steel	6.1
Jurassic oolites and bioclastic limestones	3.0 - 4.0	Iron	5.8
Carboniferous limestones	5.0 - 5.5	Aluminium	6.6
Dolomites	2.5 - 6.5	Concrete	3.6
Salt	4.5 - 5.0		
Anhydrite	4.5 - 6.5		
Gypsum	2.0 - 3.5		

Table 4.1: Compressional waves velocities for various isotropic materials.

4.3 Forced Vibration Problem

To get insight into the combination of waves generated by a localized source, such as an explosion, it is useful to consider a semi-infinite layered half-space consisting of homogeneous, elastic, non- intersecting layers on the top of a homogeneous half-space with topographic irregularities.

Accordingly, letting Ω_{F} denotes the *half-space* $\{\mathbf{x} = (x_1, x_2, x_3)^\top \in \mathbb{R}^3 \;:\; x_3 > 0\}$ bounded by the infinite planar *free-surface* $\Gamma_{\text{F}} := \{\mathbf{x} \in \mathbb{R}^3 \;:\; x_3 = 0\}$, characterized by the unit outward normal vector $\mathbf{n} = (n_1, n_2, n_3)^\top$. Local coordinate system with its origin on Γ_{F} is introduced. Configurations of interest are semi-infinite domains Ω which coincide with Ω_{F} or deviate from it only in a region of finite size (surface irregularities). The boundary Γ_{01} of Ω is thus of the form $\Gamma_{01} := S \cup \Gamma$, where the bounded surface S defines possible topographic irregularities and Γ is the remaining part of Γ_{F} (see Figure 4.2). Note that Γ coincides with Γ_{F} outside of S.

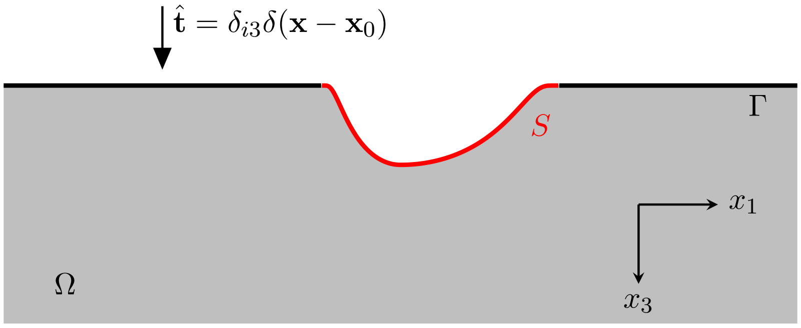

$\hat{\mathbf{t}} = \delta_{i3}\delta(\mathbf{x} - \mathbf{x}_0)$
S
Γ
x_1
Ω
x_3

Figure 4.2: Forced vibration problem: geometry and notations.

The *forced vibration problem* is formulated as follows: given a vertical point unit time-harmonic excitation $\hat{\mathbf{t}}(\mathbf{x})$ (*point load*), applied in the point $\mathbf{x}_0 \in \Gamma$ (see Figure 4.2), find the displacements $\mathbf{u}$ solution to Navier-Cauchy equation of motion in Ω, which satisfies the Neumann boundary condition on Γ_{01}

$$\mathbf{t}_{|\Gamma_{01}}(\mathbf{x}) := \boldsymbol{\sigma}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \hat{\mathbf{t}}(\mathbf{x}) := \delta(\mathbf{x} - \mathbf{x}_0)\mathbf{e}_3, \tag{4.1}$$

where the load $\hat{\mathbf{t}}$ is modeled by the Dirac delta distribution $\delta(\mathbf{x})$.

4.3.1 Multi-Domain BEM Formulation

Since we treat a layered soil, the entire problem domain Ω is divided into M non-overlapping and non-intersecting sub-domains Ω_i, i.e.

$$\Omega := \bigcup_{i=1}^{M} \Omega_i, \tag{4.2}$$

each characterized by its shear modulus μ_i, Poisson's ratio ν_i and mass density ϱ_i. Time-harmonic motions with circular frequency ω_i are assigned in each sub-domain. In the following, $\partial\Omega_i$ denotes the boundary of the sub-domain Ω_i, while Γ_{ij} is the interface between Ω_i and Ω_j, characterized by the unit normal vector $\mathbf{n}_{ij}$, directed outside of Ω_i.

In this way, the original single-domain problem is divided into a group of local problems in M sub-domains (see Figure 4.3, where a simple model is shown).

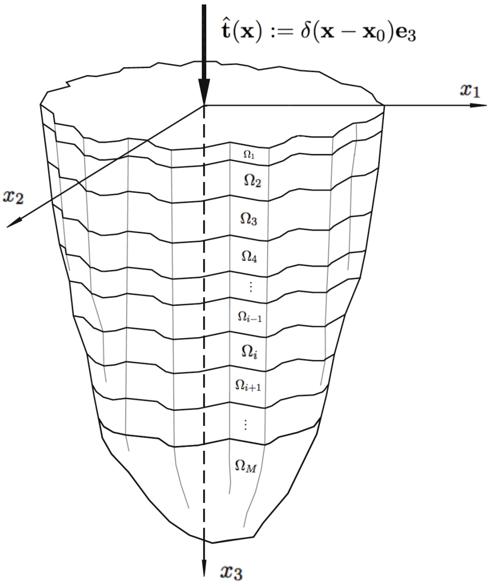


Figure 4.3: Simple model of layer soil: a semi-infinite layered half-space consisting of homogeneous, elastic non-intersecting layers on the top of a homogeneous half-space.

Since for each of these sub-domains the local displacement field $\mathbf{u}_i$ satisfies the Navier-Cauchy equation of motion in the domain Ω_i, each sub-problem can be reformulated as a boundary integral equation

$$\left(-\frac{1}{2}I + D_{\partial\Omega_i}\right)\mathbf{u}_{i_{|\Gamma_i}}(\mathbf{x}) = S_{\partial\Omega_i}\mathbf{t}_{i_{|\Gamma_i}}(\mathbf{x}), \tag{4.3}$$

involving the identity operator I and a linear combination of the single-layer $S_{\partial\Omega_i}$ and the double-layer $D_{\partial\Omega_i}$ operators, defined in Equation (3.13) in the context of scattering problems. In order to relate the unknown fields on the interfaces, additional *coupling conditions* are required. In the case of interfaces between two solid materials, we require that both the traction and the displacement are continuous on Γ_{ij}, i.e.

$$\mathbf{u}_{i_{|\Gamma_{ij}}}(\mathbf{x}) \ := \ \mathbf{u}_{j_{|\Gamma_{ij}}}(\mathbf{x}) \qquad \text{and} \qquad \mathbf{t}_{i_{|\Gamma_{ij}}}(\mathbf{x}) \ := \ -\mathbf{t}_{j_{|\Gamma_{ij}}}(\mathbf{x}). \tag{4.4}$$

We note that the interface conditions are not unique (any linearly independent combinations of the relations in Equation (4.4) can be used as coupling conditions) and different forms of interface conditions can influence the conditioning of the final system. Due to this consideration and in the light of the results of potential problems (see [91]), we choose to consider the following form of the coupling conditions:

$$\begin{array}{rcl} \mathbf{u}_{i_{|\Gamma_{ij}}}(\mathbf{x}) - \mathbf{u}_{j_{|\Gamma_{ij}}}(\mathbf{x}) - \mathbf{t}_{i_{|\Gamma_{ij}}}(\mathbf{x}) - \mathbf{t}_{j_{|\Gamma_{ij}}}(\mathbf{x}) & = & 0 \\ -\mathbf{u}_{i_{|\Gamma_{ij}}}(\mathbf{x}) + \mathbf{u}_{j_{|\Gamma_{ij}}}(\mathbf{x}) - \mathbf{t}_{i_{|\Gamma_{ij}}}(\mathbf{x}) - \mathbf{t}_{j_{|\Gamma_{ij}}}(\mathbf{x}) & = & 0. \end{array} \tag{4.5}$$

The following multi-domain formulation is based on three-noded triangular boundary elements and piecewise linear interpolations of displacements and tractions. The collocation method is used. In each sub-domain Ω_i, the discretization of the integral equation (4.3) gives rise to the matrix equation:

$$K_i\tilde{\mathbf{u}}_i = S_i\tilde{\mathbf{t}}_i \qquad \text{with } K_i = \left(-\frac{1}{2}I + D_i\right), \tag{4.6}$$

where the matrices D_i and S_i are fully populated unsymmetric matrices, whose computation entails the evaluation of the single-layer and the double layer operators, and I is the identity matrix. The vectors $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{t}}_i$ gather the displacement and traction degrees of freedom.

For sake of definiteness, we first focus our attention on the sub-domain Ω_1. Since its boundary $\partial\Omega_1 = \Gamma_{01} \cup \Gamma_{12}$ is composed by the free-surface Γ_{01} and the interface Γ_{12}, we can separate the interactions between Γ_{01} and Γ_{12} and consider Equation (4.6) in the form:

$$\begin{bmatrix} K_{\Gamma_{01}\Gamma_{01}} & K_{\Gamma_{01}\Gamma_{12}} \\ K_{\Gamma_{12}\Gamma_{01}} & K_{\Gamma_{12}\Gamma_{12}} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{u}}_1^{01} \\ \tilde{\mathbf{u}}_1^{12} \end{bmatrix} = \begin{bmatrix} S_{\Gamma_{01}\Gamma_{01}} & S_{\Gamma_{01}\Gamma_{12}} \\ S_{\Gamma_{12}\Gamma_{01}} & S_{\Gamma_{12}\Gamma_{12}} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\mathbf{t}}_1^{01} \\ \tilde{\mathbf{t}}_1^{12} \end{bmatrix}. \tag{4.7}$$

Now, calling $\hat{\mathbf{t}}$ the vector resulting from the evaluation of the function $\hat{\mathbf{t}}(\mathbf{x})$ in the node used to discretize the free-surface Γ_{01}, the application of the boundary condition (4.1) allow us to rewrite the previous equation as the following system of matrix equations:

$$\begin{cases} K_{\Gamma_{0,1}\Gamma_{0,1}}\tilde{\mathbf{u}}_1^{0,1} + K_{\Gamma_{0,1}\Gamma_{1,2}}\tilde{\mathbf{u}}_1^{1,2} - S_{\Gamma_{0,1}\Gamma_{1,2}}\tilde{\mathbf{t}}_1^{1,2} = S_{\Gamma_{0,1}\Gamma_{0,1}}\hat{\mathbf{t}} \\ K_{\Gamma_{1,2}\Gamma_{0,1}}\tilde{\mathbf{u}}_1^{0,1} + K_{\Gamma_{1,2}\Gamma_{1,2}}\tilde{\mathbf{u}}_1^{1,2} - S_{\Gamma_{1,2}\Gamma_{1,2}}\tilde{\mathbf{t}}_1^{1,2} = S_{\Gamma_{1,2}\Gamma_{0,1}}\hat{\mathbf{t}}. \end{cases} \tag{4.8}$$

Similarly, separating the interactions between the interfaces $\Gamma_{i,i-1}$ and $\Gamma_{i,i+1}$ (i=2,. . . ,M-1), in each sub-domain Ω_i we have to solve the system of matrix equations:

$$\begin{cases} K_{\Gamma_{i,i-1}\Gamma_{i,i-1}}\tilde{\mathbf{u}}_i^{i,i-1} + K_{\Gamma_{i,i-1}\Gamma_{i,i+1}}\tilde{\mathbf{u}}_i^{i,i+1} - S_{\Gamma_{i,i-1}\Gamma_{i,i-1}}\tilde{\mathbf{t}}_2^{i,i-1} - S_{\Gamma_{i,i-1}\Gamma_{i,i+1}}\hat{\mathbf{t}}_2^{i,i+1} = 0 \\ K_{\Gamma_{i,i+1}\Gamma_{i,i-1}}\tilde{\mathbf{u}}_i^{i,i-1} + K_{\Gamma_{i,i+1}\Gamma_{i,i+1}}\tilde{\mathbf{u}}_i^{i,i+1} - S_{\Gamma_{i,i+1}\Gamma_{i,i-1}}\tilde{\mathbf{t}}_2^{i,i-1} - S_{\Gamma_{i,i+1}\Gamma_{i,i+1}}\hat{\mathbf{t}}_2^{i,i+1} = 0. \end{cases} \tag{4.9}$$

Finally, in the sub-domain Ω_{M} we have to consider the matrix equation

$$K_{\Gamma_{\mathrm{M,M-1}}\Gamma_{\mathrm{M,M-1}}}\tilde{\mathbf{u}}_{\mathrm{M}}^{\mathrm{M,M-1}} - S_{\Gamma_{\mathrm{M,M-1}}\Gamma_{\mathrm{M,M-1}}}\tilde{\mathbf{t}}_{\mathrm{M}}^{\mathrm{M,M-1}} = 0. \tag{4.10}$$

If we do not eliminate the duplicated unknowns on the interfaces, combining Equations (4.8), Equations (4.9) and Equations (4.10) with the coupling conditions (4.4), the final BEM linear system of equations is

$$A\mathbf{z} = \mathbf{b}, \tag{4.11}$$

where coefficient matrix A is given by:

$$A = \begin{bmatrix} K_{\Gamma_{01}\Gamma_{01}} & K_{\Gamma_{01}\Gamma_{12}} & -S_{\Gamma_{01}\Gamma_{12}} & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ K_{\Gamma_{12}\Gamma_{01}} & K_{\Gamma_{12}\Gamma_{12}} & -S_{\Gamma_{12}\Gamma_{12}} & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & I & -I & -I & -I & 0 & 0 & \dots & 0 & 0 \\ 0 & -I & I & -I & -I & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & K_{\Gamma_{21}\Gamma_{21}} & K_{\Gamma_{21}\Gamma_{23}} & -S_{\Gamma_{21}\Gamma_{21}} & -S_{\Gamma_{21}\Gamma_{23}} & \dots & 0 & 0 \\ 0 & 0 & 0 & K_{\Gamma_{23}\Gamma_{21}} & K_{\Gamma_{23}\Gamma_{23}} & -S_{\Gamma_{23}\Gamma_{21}} & -S_{\Gamma_{23}\Gamma_{23}} & \dots & 0 & 0 \\ 0 & 0 & 0 & I & -I & -I & -I & \dots & 0 & 0 \\ 0 & 0 & 0 & -I & I & -I & -I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & K_{\Gamma_{\mathrm{M,M-1}}\Gamma_{\mathrm{M,M-1}}} & -S_{\Gamma_{\mathrm{M,M-1}}\Gamma_{\mathrm{M,M-1}}} \end{bmatrix}, \tag{4.12}$$

while the unknown vector $\mathbf{z}$ and the right hand side $\mathbf{b}$ are given by

$$\mathbf{z} = \begin{bmatrix} \tilde{\mathbf{u}}_1^{0,1} \\ \tilde{\mathbf{u}}_1^{1,2} \\ \tilde{\mathbf{t}}_1^{1,2} \\ \tilde{\mathbf{u}}_2^{2,1} \\ \tilde{\mathbf{t}}_2^{2,1} \\ \tilde{\mathbf{u}}_2^{2,3} \\ \tilde{\mathbf{t}}_2^{2,3} \\ \vdots \\ \tilde{\mathbf{u}}_{\mathrm{M}}^{\mathrm{M,M-1}} \\ \tilde{\mathbf{t}}_{\mathrm{M}}^{\mathrm{M,M-1}} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} S_{\Gamma_{0,1}\Gamma_{0,1}}\hat{\mathbf{t}} \\ S_{\Gamma_{1,2}\Gamma_{0,1}}\hat{\mathbf{t}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{4.13}$$

Briefly, we remember that all the block-matrices present in (4.12) require a quadratic amount of storage. Thus, if N is the total number of degrees of freedom of the global problem, the amount of memory for the matrix A is $\mathcal{O}(N^2)$. Furthermore, solving equation (4.11) with a direct solver such as a LU-factorization requires a cubic amount of numerical operations ($\mathcal{O}(N^3)$), to be compared with a cost of the order of $\mathcal{O}(N^2)$ operations for each iteration needed by an iterative method. Therefore, the application of classical boundary element method is restricted to problem of small size.

4.3.2 Hierarchical Multi-Domain BEM Formulation

The use of $\mathcal{H}$-matrices provides an efficient way to approximate fully populated matrices (with an arbitrary prescribed accuracy) by means of memory efficient representations. In particular, the proposed multi-domain BEM formulation is naturally suited to define $\mathcal{H}$-matrix based solution strategies, because the entire coefficient matrix is partitioned and decomposed using the Schur component.

Assembling the hierarchical representations $K_i^{\mathcal{H}}$ and $S_i^{\mathcal{H}}$ of the matrices K_i and S_i respectively, allows us to replace equation (4.11) with

$$A_{\mathcal{H}}\mathbf{z} = \mathbf{b}_{\mathcal{H}}. \tag{4.14}$$

Since the construction of an $\mathcal{H}$-matrix representation of matrices K_i and S_i requires a hierarchical subdivision of the initial matrix based on the geometrical considerations, for each sub-domain Ω_i we need only one cluster tree T_I^i and, consequently, only one block-cluster tree $T_{I\times I}^i$. Additionally, we can separate the interactions between the interfaces $\Gamma_{i,i-1}$ and $\Gamma_{i,i+1}$ during the construction of the cluster tree T_I^i, and obtain the required partition of matrix K_i and S_i in a natural way. Once the $\mathcal{H}$-matrix $S_1^{\mathcal{H}}$ is assembled, we use the $\mathcal{H}$-matrix/vector product to accelerate the matrix/vector products needed to compute the right hand side.

4.4 Green's Function of a Homogeneous Half-Space

A special case of forced vibrations is the computation of the Green's function for a homogeneous elastic half-space. This problem harks back to the early twentieth century, when Horace Lamb [114] investigated a half-space subjected to concentrated vertical or horizontal loads at the surface or inside it. The use of complex contour integration in his work yielded the wave motions generated at the surface of the elastic half-space. An important and comprehensive subsequent study in this area is the work of Miller and Pursey [133], in which Lamb's method was adapted to furnish definite integral representations of the field at an arbitrary point in an isotropic half-space due to the stress prescribed on the surface. The first truly complete solutions to Lamb's problem were given by Pekeris [142] and Chao [43], who provided closed-form expressions for the components of motion elicited by a vertical and a horizontal load, respectively, but only when Poisson's ratio is 1/4. This problem was taken up again by Mooney [134], who extended the Pekeris solution to any arbitrary Poisson's ratio, but he did so only for the vertical component while ignoring the radial one. Then in 1979, Richards [147] considered this problem once again and gave a complete set of exact formulae for both loading cases and for any Poisson's ratio.

In the frequency domain, the Green's function for a homogeneous elastic half-space is known [8, 97] and applied in dynamic soil-structure interaction [47, 79]. Since it is expressed in Fourier-Bessel integral form (involving oscillatory integrals over infinite intervals), with no closed-form expression available, its numerical evaluation is complex and time-consuming [128]. Singularity and regularization issues for the half-space Green's function have been presented [149]. A fast multipole formulation has recently been introduced [36].

In the following, we consider the response of a homogeneous half-space Ω_F to a time-harmonic load $\hat{\mathbf{t}}(\mathbf{x}) = \delta_{i3}\delta(\mathbf{x})$, applied on the free-surface Γ_F at the origin $\mathbf{0}$ of the local coordinate system (see Figure 4.4). The material properties are fixed to $\mu = \varrho = 1$ and $\nu = {}^1\!/_3$.

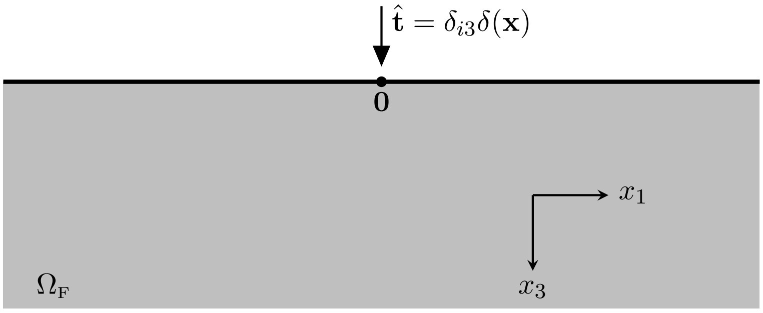


Figure 4.4: Schematic representation of a homogeneous elastic half-space

As we consider a homogeneous domain, the BEM formulation of the problem reduces to

$$\left(-\frac{1}{2}I + D\right)\tilde{\mathbf{u}} = S\tilde{\mathbf{t}}, \tag{4.15}$$

where the solution $\tilde{\mathbf{u}}$ is the response in the direction $\mathbf{e}_i$, at the observation point $\mathbf{x}$ to the load, i.e. it is the Green's displacement function $U_{i3}^{\mathrm{HS}}(\mathbf{x}, \mathbf{0}; \omega)$, corresponding to the component $i3$ of the Green's displacement tensor for a homogeneous half-space, $\mathbf{U}^{\mathrm{HS}}(\mathbf{x}, \mathbf{y}; \omega)$, evaluated in $\mathbf{y} = \mathbf{0}$.

Assembling the hierarchical representations $D_\mathcal{H}$ and $S_\mathcal{H}$ of the matrices D and S respectively, allows us to replace Equation (4.15) with

$$\left(-\frac{1}{2}I + D_\mathcal{H}\right)\tilde{\mathbf{u}} = S_\mathcal{H}\tilde{\mathbf{t}}. \tag{4.16}$$

This last matrix equation differs from the previous considered in Chapter 3, because its right hand side has to be computed performing the fast $\mathcal{H}$-matrix/vector product between $S_{\mathcal{H}}$ and $\tilde{\mathbf{t}}$.

Since the computation of the $\mathcal{H}$-matrix representations of matrices D and S in Equation (4.16) involves integrals over the unbounded surface Γ_{F}, in practice we need to truncate Γ_{F} at $\Gamma_{\mathrm{F}}(L)$, i.e. to bound the unbounded surface by a circle of radius L.

The main difficulty of the extension of the $\mathcal{H}$-matrices to this problem is linked to the quality of the approximation provided by the vector ACA, when we evaluate the double-layer operator D.

4.4.1 Limitations of the Vector Version of the ACA for Half-Space Problems

Looking at equation (4.16), we note that the computation of the entries of matrices $D_{\mathcal{H}}$ and $S_{\mathcal{H}}$ requires the evaluation of the tensors $\mathbf{U}(\mathbf{x},\mathbf{y};\omega)$ and $\mathbf{T}(\mathbf{x},\mathbf{y};\omega)$ on the surface Γ_{F}, whose components are given by:

$$\begin{aligned} U_i^k(\mathbf{x},\mathbf{y};\omega) &:= \frac{1}{4\pi\mu r}\left(A_1\delta_{ik} + A_2 r_{,i} r_{,k}\right) \\ T_i^k(\mathbf{x},\mathbf{y};\omega) &:= \frac{1}{4\pi r}\left[2A_3 r_{,i} r_{,k} r_{,j} + \left(\delta_{ik} r_{,j} + \delta_{jk} r_{,i}\right) A_4 + \delta_{ij} r_{,k} A_5\right] n_j \end{aligned} \tag{4.17}$$

where the coefficients A_1, A_2, A_3, A_4 and A_5 are defined by equations (1.41), while $r_{,i}$ is the derivative of $r = \|\mathbf{x}-\mathbf{y}\|$ with respect to x_i.

Since Γ_{F} is a planar surface with normal $\mathbf{n}(\mathbf{x}) = (0,0,-1)^\top$, we have:

$$T_i^k(\mathbf{x},\mathbf{y};\omega) = -\frac{1}{4\pi r}(\delta_{3k} r_{,i} A_4 + \delta_{i3} r_{,k} A_5). \tag{4.18}$$

Finally, as the third component of the distance between $\mathbf{x}$ and $\mathbf{y}$ vanishes, the symmetric tensor $\mathbf{U}(\mathbf{x},\mathbf{y};\omega)$ corresponds to the 3-by-3 matrix

$$\mathbf{U}(\mathbf{x},\mathbf{y};\omega) = \frac{1}{4\pi\mu r}\begin{bmatrix} A_1 + A_2 r_{,1}^2 & A_2 r_{,1} r_{,2} & 0 \\ A_2 r_{,1} r_{,2} & A_1 + A_2 r_{,2}^2 & 0 \\ 0 & 0 & A_1 \end{bmatrix} \quad \Rightarrow \quad \det(\mathbf{U}) \neq 0, \tag{4.19}$$

while the symmetric tensor $\mathbf{T}(\mathbf{x},\mathbf{y};\omega)$ corresponds to the 3-by-3 matrix

$$\mathbf{T}(\mathbf{x},\mathbf{y};\omega) = -\frac{1}{4\pi r}\begin{bmatrix} 0 & 0 & A_4 r_{,1} \\ 0 & 0 & A_4 r_{,2} \\ A_4 r_{,1} & A_4 r_{,2} & 0 \end{bmatrix} \quad \Rightarrow \quad \det(\mathbf{T}) = 0. \tag{4.20}$$

As result, we can not use the vector ACA to give an $\mathcal{H}$-matrix representation of matrix D on Γ_{F}, because each of its entries is a singular 3-by-3 matrix and, consequently, it is not possible to choose a non-zero pivot (see Subsection 2.4.2).

As we have explained in Section 2.3, the truncated Singular Value Decomposition or the fully-pivoted ACA always provide good approximations of low-rank matrices but they are not interesting options for $\mathcal{H}$-matrices, because they need the knowledge of the complete matrix to approximate. In order to avoid these approximation techniques, we take advantage of the sparse structure of $\mathbf{T}(\mathbf{x}, \mathbf{y}; \omega)$ and we use randomized Singular Value Decomposition algorithm to approximate the admissible blocks.

4.4.2 Randomized Singular Value Decomposition

Randomized algorithms for low-rank approximations of matrices have been an active research topic in the recent years with a vast literature (see [127] for an overview). In particular, the *randomized singular value decomposition* (random SVD) algorithm is a very useful alternative to the vectorial ACA for computing a low-rank approximation of a given numerically low-rank matrix [117]. Given a matrix M of size $m \times n$, the random SVD method computes $k \in \mathbb{N}$, $U \in \mathbb{C}^{m\times k}$, $S \in \mathbb{C}^{k\times k}$ and $V \in \mathbb{C}^{n\times k}$, with singular values $\sigma_1 \leq \sigma_2 \leq \dots \sigma_{\min\{m,n\}}$, such that

$$\|USV^H - M\| \leq \varepsilon_{\text{rSVD}}\sigma_{k+1}, \tag{4.21}$$

where k is the number of singular values greater than a prescribed accuracy $\varepsilon_{\text{rSVD}} > 0$. There exist different versions of this technique but we choose the procedure presented by Halko [86] and used by Chaillat and Biros to treat scalar Helmholtz inverse medium problems [34].

Starting with $k = 1$, we consider $l = k + c$, where $c \in \mathbb{N}$ is a fixed small parameter. The main component of the randomized SVD algorithm is the multiplication of M to a Gaussian random matrix $G \in \mathbb{C}^{n\times l}$ to compute $R := MG \in \mathbb{C}^{m\times l}$. Applying classical singular value decomposition to R, we obtain two orthogonal matrices $\Phi \in \mathbb{C}^{m\times l}$ and $\Psi \in \mathbb{C}^{l\times l}$ and a diagonal matrix $\Lambda \in \mathbb{C}^{l\times l}$ such that $R = \Phi\Lambda\Psi^H$. The first k left singular vectors of R, i.e. the first k columns of Φ, give the matrix $Q \in \mathbb{C}^{m\times k}$. To avoid the need to precompute the matrix rank, we use an error estimate and test if

$$\|R - QQ^H R\| \leq \varepsilon_{\text{rSVD}}\lambda_1 \tag{4.22}$$

where λ_1 is the bigger singular value of R. If this condition is not fullfilled, we increase k and we repeat the algorithm. Otherwise, we compute the matrix $T = Q^H M \in \mathbb{C}^{n\times k}$ and its singular value decomposition $T = WSV^H$, where $W \in \mathbb{C}^{k\times k}$ and $V \in \mathbb{C}^{n\times k}$ are orthogonal matrices and $S \in \mathbb{C}^{r\times r}$ is a diagonal matrix. Finally, we assemble $U = QW \in \mathbb{C}^{m\times r}$.

Since the generation of random numbers is quite efficient, the cost of the Gaussian random matrix G is negligible and, in practice, the cost of each iteration of the described algorithm is $\mathcal{O}(\mu(n, m)l + \mu(n, m)r + ml^2 + nk^2)$, where $\mu(n, m)$ denotes the cost of the application of matrix M. Assuming $m < n$, the complexity is $\mathcal{O}(\mu(n, m)l + k^2 n)$. If we have a dense matrix, the complexity is $\mathcal{O}(lmn)$ but, in practice, we use randomized SVD to approximate sparse matrices. It follows that for our applications the overall complexity of this approximate factorization is $\mathcal{O}(ln)$ for work and $\mathcal{O}(kn)$ for storage, values comparable with work and storage required by the ACA.

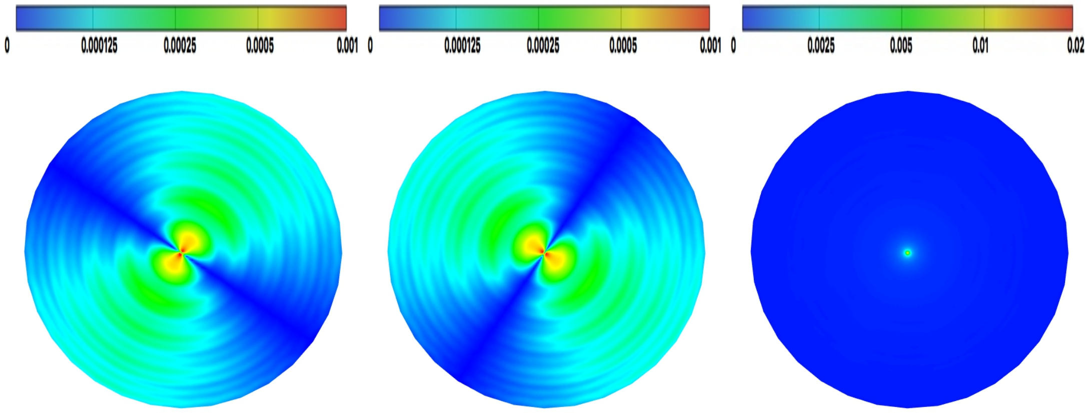
4.4.3 $\mathcal{H}$-matrix based BEM for Half-Space Problems

In the following, we consider a truncated version $\Gamma_{\mathrm{F}}(L)$ of Γ_{F} bounded by a circle of radius $L = 5$ and discretized with a density of $d_{\lambda_{\mathrm{S}}} = 10$ degrees of freedom per S-wavelength. Four circular frequencies ω are considered, with respective problem size N and non-dimensional frequency $\phi_{\mathrm{S}} = {}^{L\omega}/_{c_{\mathrm{S}}\pi}$ reported in Table 4.2.

N	5 475	21 507	85 251	339 459
ω	2.72	5.43	10.85	21.74
ϕ_{S}	4.33	8.64	17.27	34.60

Table 4.2: Green's displacement function of a homogeneous half-space: dimensions of the problem, circular frequency and non-dimensional frequency.

The cluster tree T_I is built with a stopping criteria $N_{\mathrm{LEAF}} = 100$ and the constant in the admissibility condition is set to $\eta = 3$. The low-rank approximations of the admissible blocks are computed using the randomized Singular Value Decomposition with $\varepsilon_{\mathrm{rSVD}} = 10^{-4}$. The SVD recompression is applied to each of these blocks with a parameter $\varepsilon_{\mathrm{SVD}}$ always chosen equal to $\varepsilon_{\mathrm{rSVD}}$. Equation (4.14) is solved using a direct approach based on the $\mathcal{H}$-LU factorization. To perform the $\mathcal{H}$-matrix based arithmetic, the accuracy $\varepsilon_{\mathrm{LU}} = 10^{-4}$ is assigned.



(a) $|u_1|$ (free-surface) (b) $|u_2|$ (free-surface) (c) $|u_3|$ (free-surface)

Figure 4.5: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on the free-surface ($N = 339\ 459$).

In Figure 4.5 we show the solution of Equation (4.16), that is the starting point to analyze the accuracy of the proposed $\mathcal{H}$-matrix BEM. In fact, once the approximated solution of Equation (4.15) is known, in a post-processing step we are able to calculate the displacement field $\mathbf{u}$ at each point of the half-space Ω_{F}, using the integral representation formula:

$$\mathbf{u}(\mathbf{x}) = \int_{\Gamma_{\mathrm{F}}(L)} \mathbf{U}(\mathbf{x},\mathbf{y};\omega)\cdot\mathbf{t}(\mathbf{y})d\Gamma_y - \int_{\Gamma_{\mathrm{F}}(L)} \mathbf{T}(\mathbf{x},\mathbf{y};\omega)\cdot\mathbf{u}(\mathbf{y})d\Gamma_y, \qquad \forall\mathbf{x}\in\Omega_{\mathrm{F}}. \quad (4.23)$$

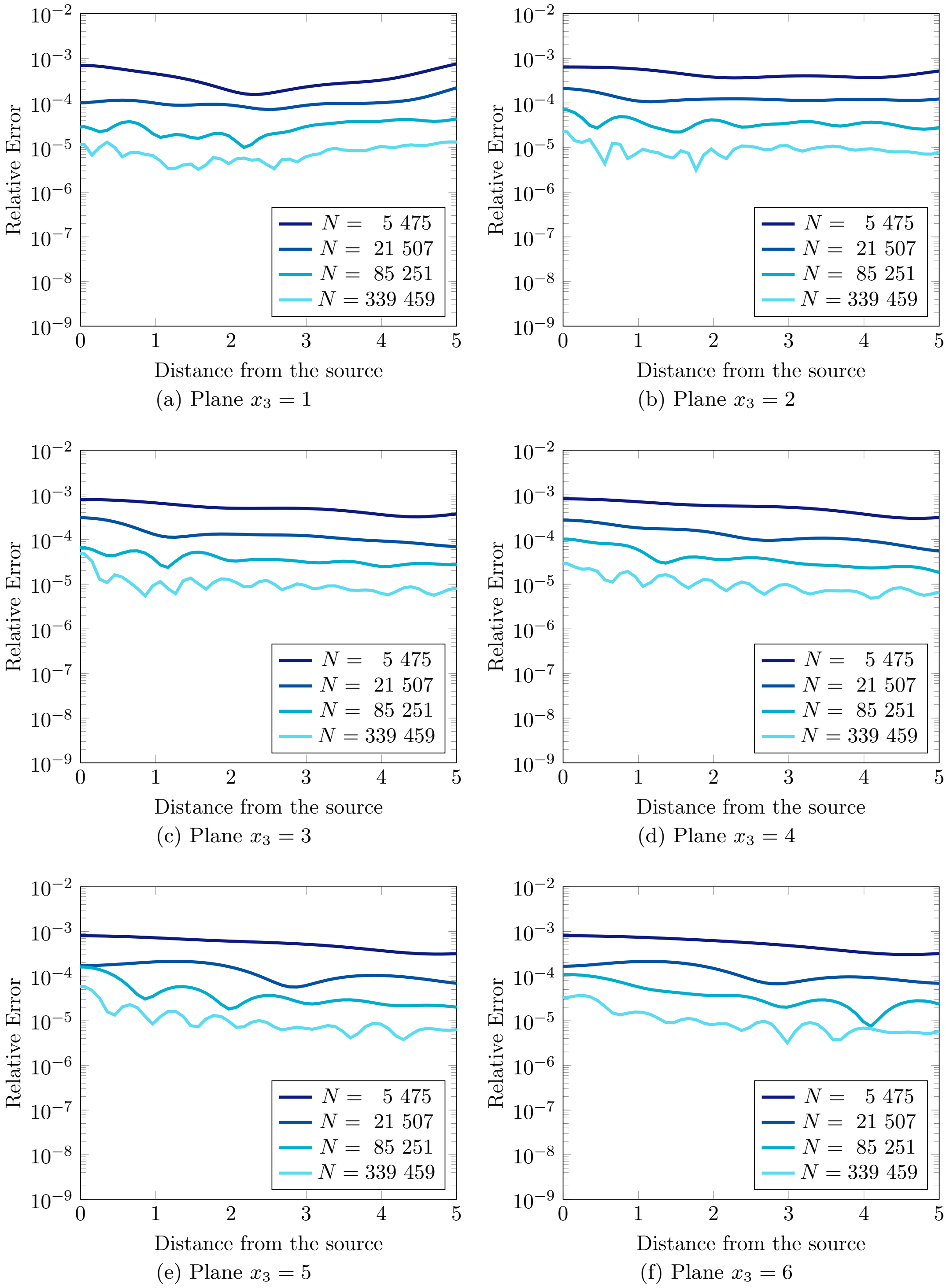
Since the analytical solution $\mathbf{U}^{\mathrm{HS}}_{\cdot 3}$ is not defined on all the free-surface, we compute $\mathbf{u}$ on six different planes parallel to $\Gamma_{\mathrm{F}}(L)$ and we introduce the relative error

$$\varepsilon_{\mathrm{HS}} := \frac{\|\mathbf{U}^{\mathrm{HS}}_{\cdot 3} - \mathbf{u}\|_2}{\|\mathbf{U}^{\mathrm{HS}}_{\cdot 3}\|_2} \quad (4.24)$$

where $\mathbf{U}^{\mathrm{HS}}_{\cdot 3}$ is computed numerically using the code provided by B.B. Guizina (for further details see [79]).

In Figure 4.6, we show the dependency of the relative error $\varepsilon_{\mathrm{HS}}$ on the distance from the center of the plane (the closest point to the source). We observe that increasing the number of the degrees of freedom, and consequently the frequency, we are able to improve the quality of the approximation. These results may be inconsistent with the analogous presented in the context of scattering problems, where the error was almost constant for different problem sizes. In order to explain the improvement of the quality of the solution of the problem treated, we have to keep in mind that the reference solution $\mathbf{U}^{\mathrm{HS}}_{\cdot 3}$ is not exact but computed numerically. For this reason some oscillations of the relative error are present at higher frequencies.

In Figures 4.7-4.12, we present a graphical comparison between the numerical solution and the reference solution. In this case we have discretized $\Gamma_{\mathrm{F}}(L)$ with 225 280 triangular boundary elements and 113 153 collocation noreference solutiondes. Consequently, the number of degrees of freedom of the problem is $N = 339\ 459$. The hierarchical cluster tree consists in 2 924 clusters and 11 cluster levels. Consequently, there are 51 931 block clusters in the block cluster tree.



(a) Plane $x_3 = 1$

(b) Plane $x_3 = 2$

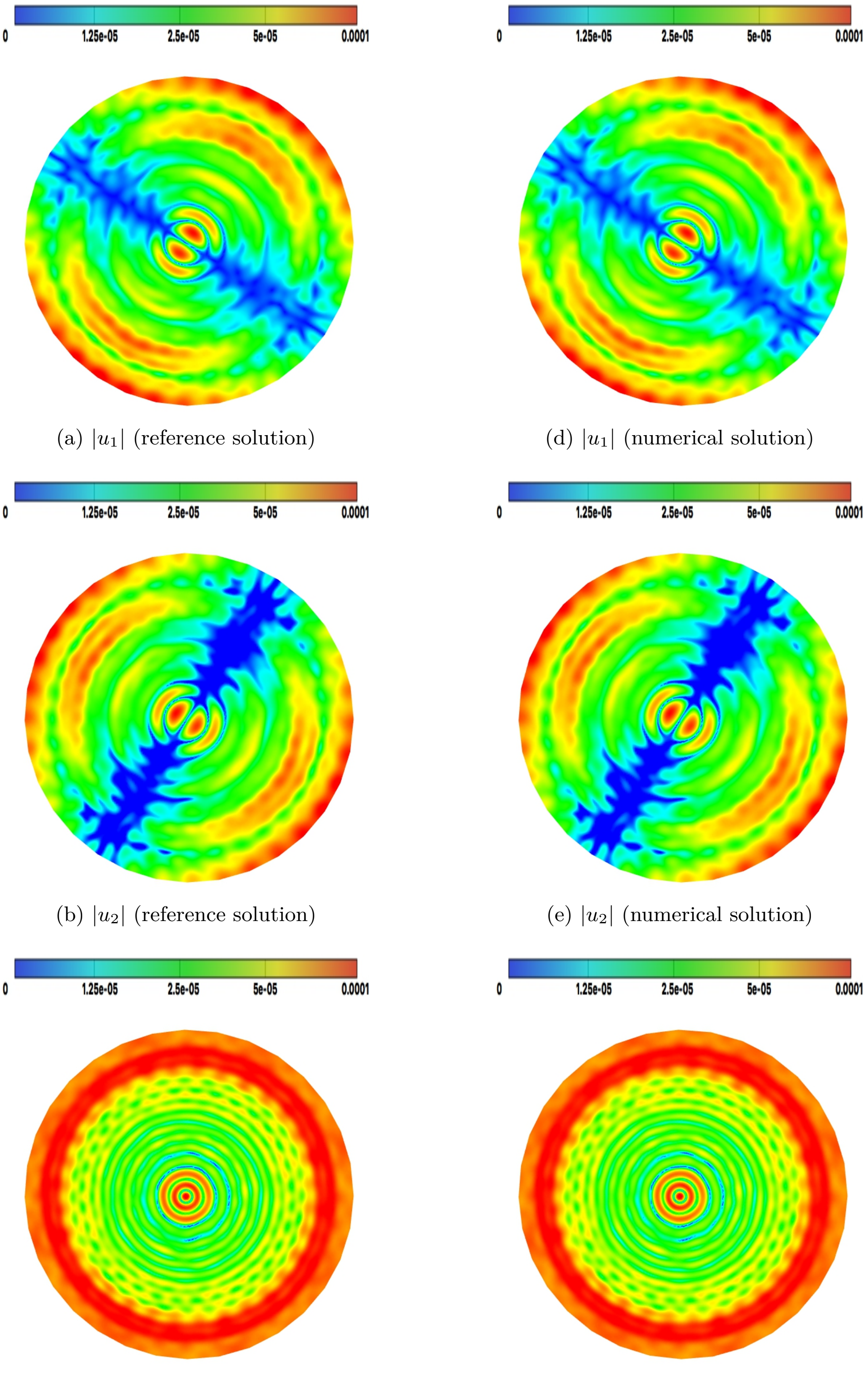
(c) Plane $x_3 = 3$

(d) Plane $x_3 = 4$

(e) Plane $x_3 = 5$

(f) Plane $x_3 = 6$

Figure 4.6: Green's displacement function of a homogeneous half-space: dependence of the relative error on the distance from the source on different planes parallel to the free-surface.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

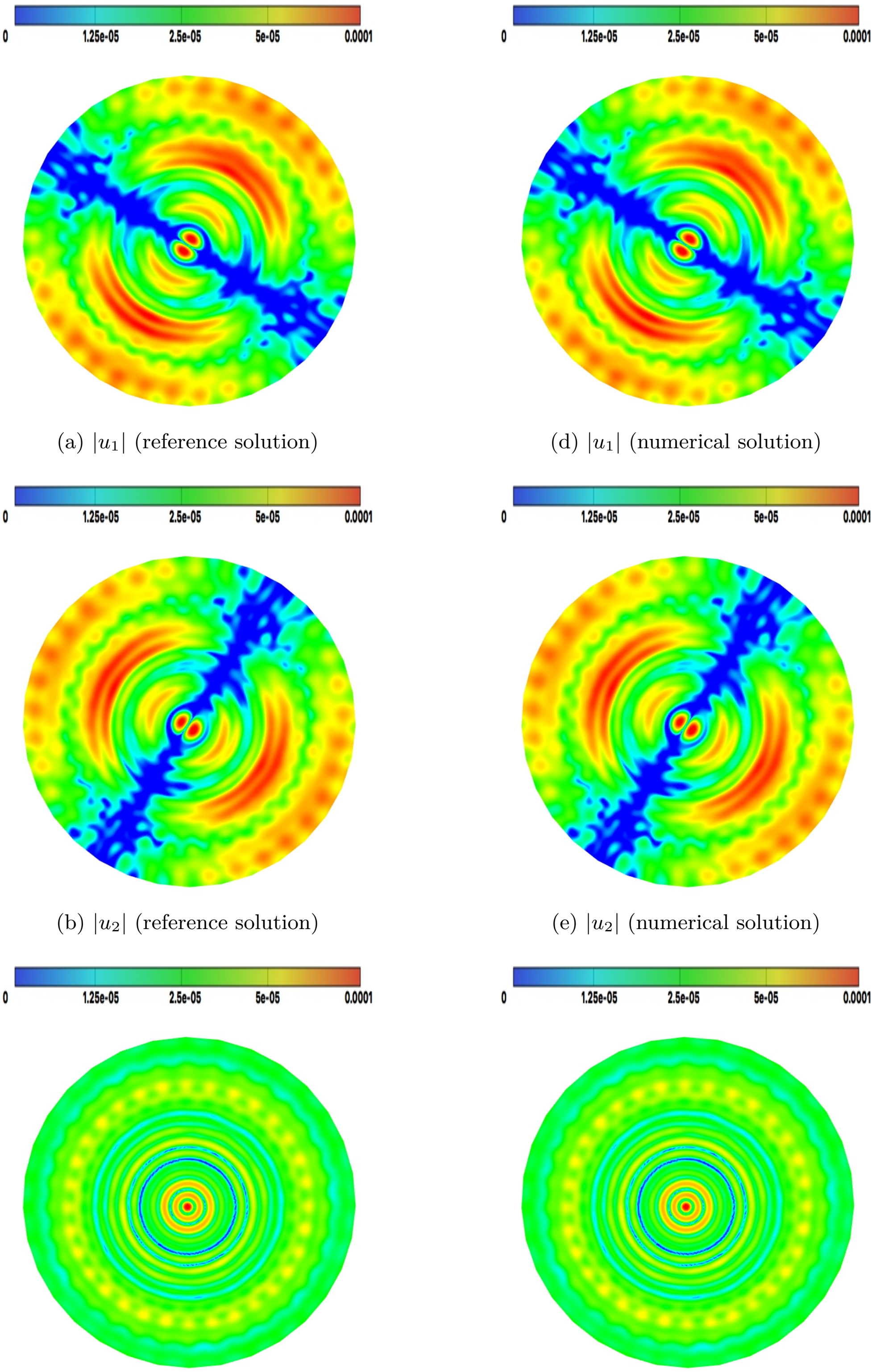
(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.7: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 1$.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

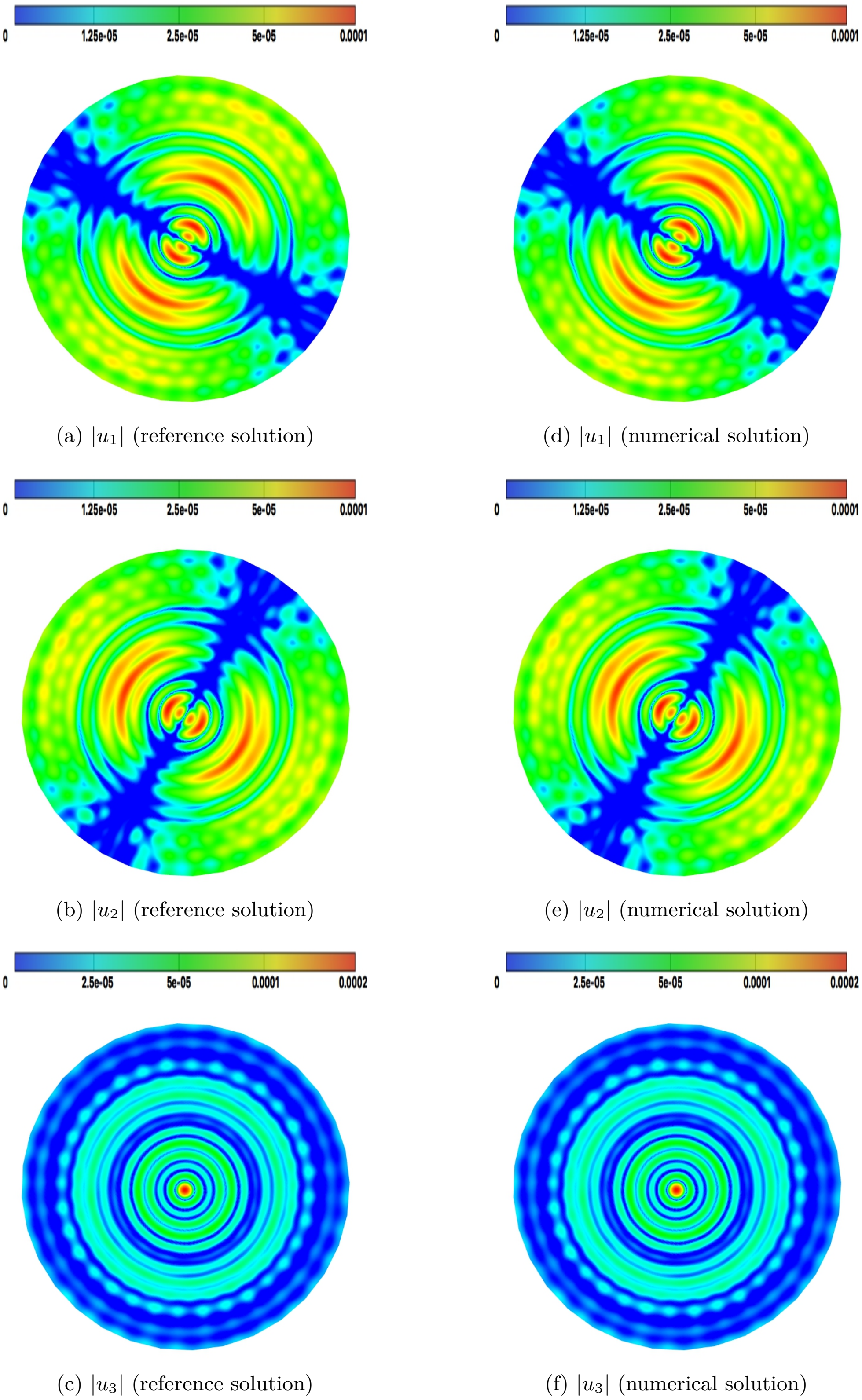
(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.8: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 2$.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

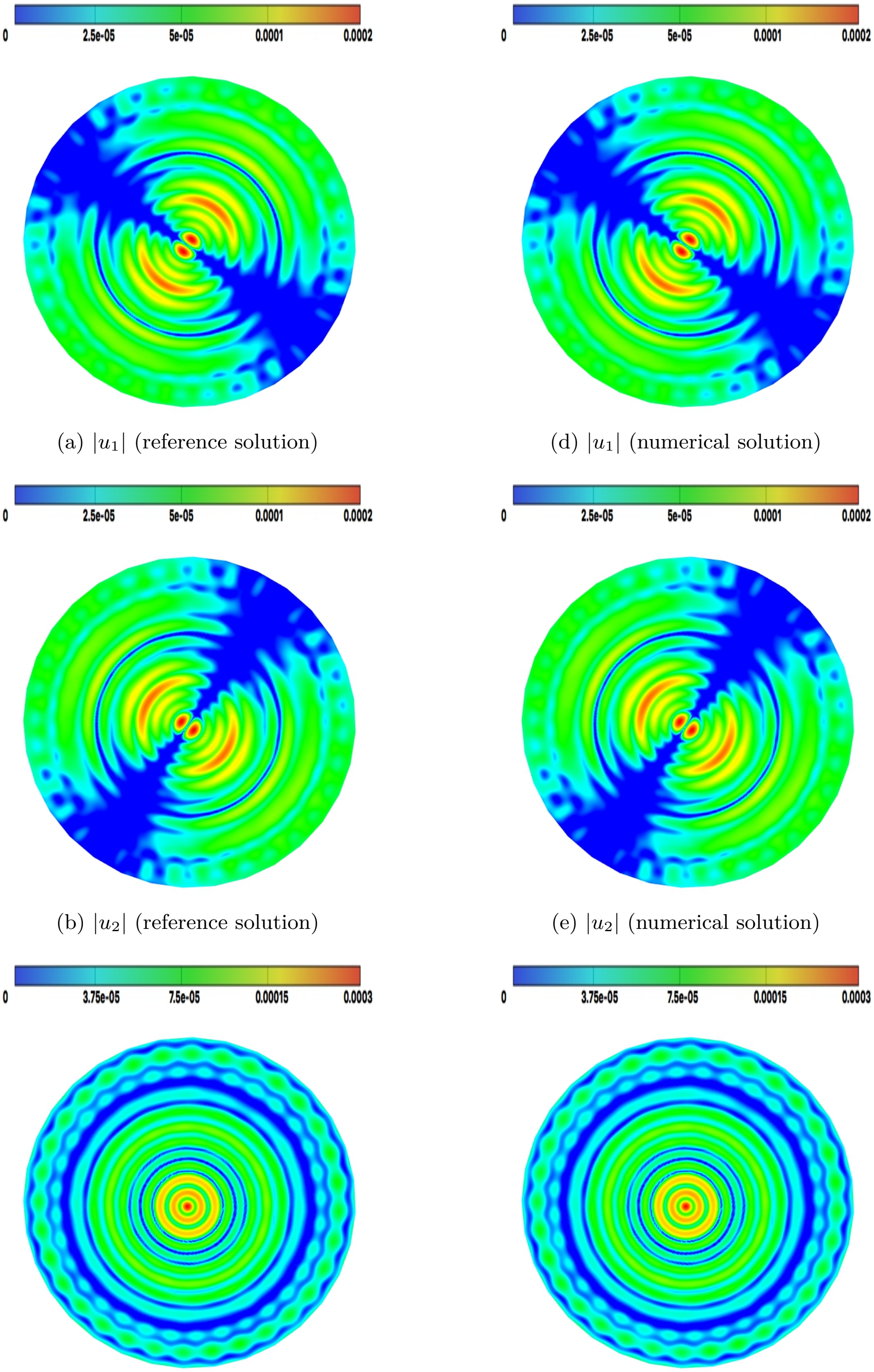
(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.9: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 3$.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

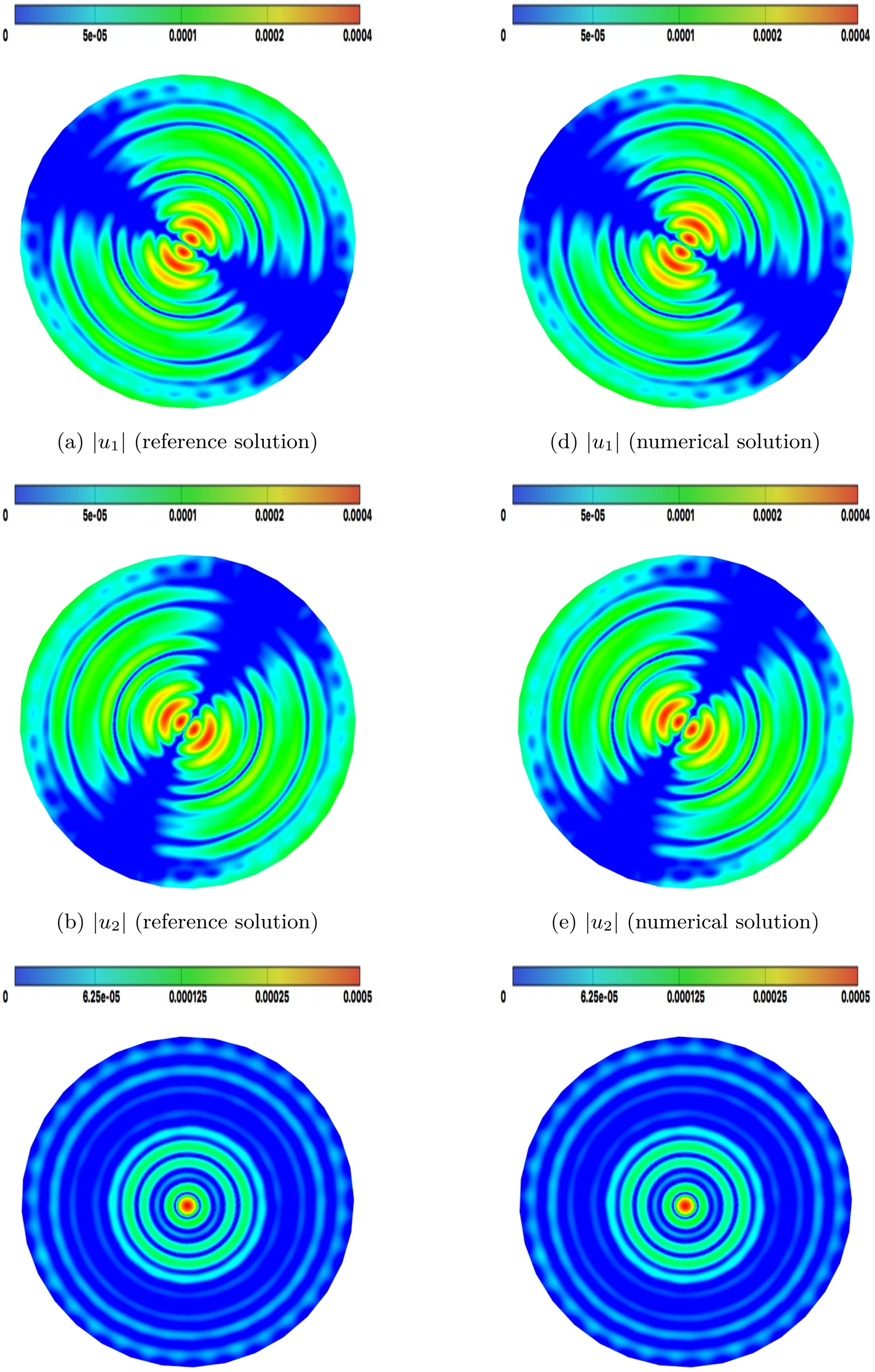
(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.10: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 4$.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

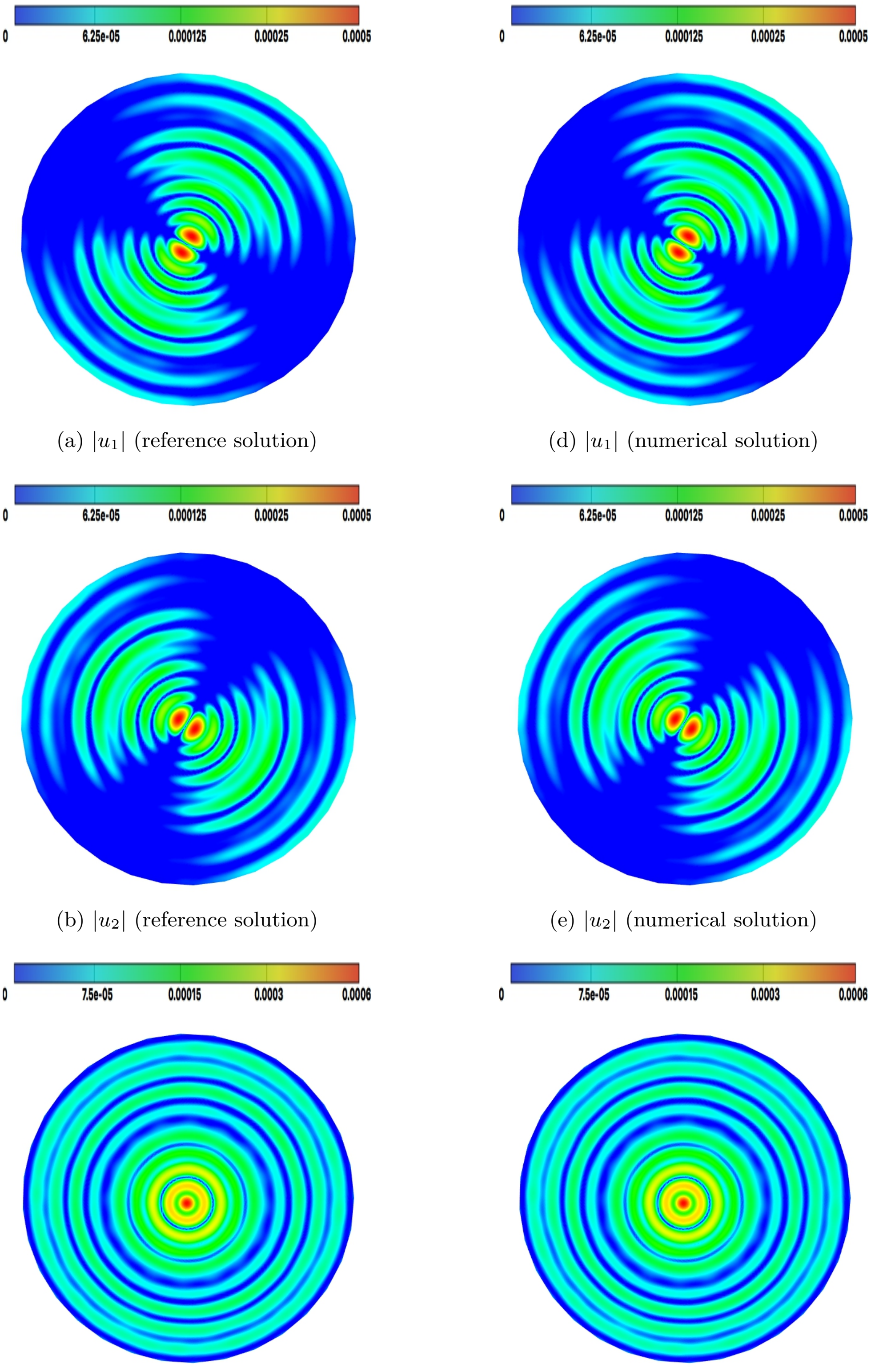
(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.11: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 5$.



(a) $|u_1|$ (reference solution)

(d) $|u_1|$ (numerical solution)

(b) $|u_2|$ (reference solution)

(e) $|u_2|$ (numerical solution)

(c) $|u_3|$ (reference solution)

(f) $|u_3|$ (numerical solution)

Figure 4.12: Green's displacement function of a homogeneous half-space: magnitude of the three components of $\mathbf{u}(\mathbf{x})$ on plane $x_3 = 6$.

4.5 Elastic Half-Space with a Semi-Spherical Alluvial Basin

For the validation of the $\mathcal{H}$-matrix based direct solver in the case of an half-space with a topographic irregularity, we consider the 3D model corresponding to a semi-spherical alluvial basin, depicted in Figure 4.13. The boundary of the domain includes the semi-spherical basin S of radius $R = 1$ and the part $\Gamma_{\mathrm{F}}(L)$ of the free-surface Γ_{F} bounded by a circle of radius $L = 5R$. The contribution of the free-surface $L \geq 5R$ is neglected. The model is excited by a vertical point load $\hat{\mathbf{t}}(\mathbf{x}) = \delta_{i3}\delta(\mathbf{x} - \mathbf{x}_0)$, applied at a point $\mathbf{x}_0 \in \Gamma_{\mathrm{F}}(L)$. The material properties are fixed to $\mu = \varrho = 1$ and $\nu = 1/3$.

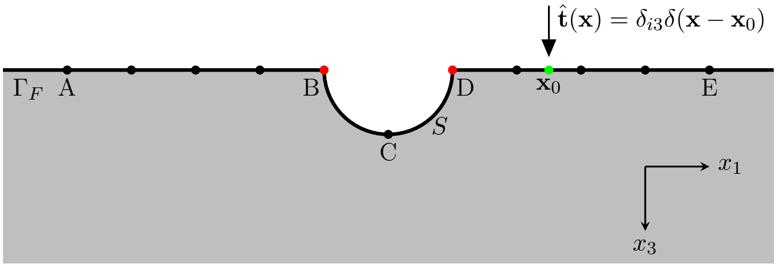

$\hat{\mathbf{t}}(\mathbf{x}) = \delta_{i3}\delta(\mathbf{x} - \mathbf{x}_0)$
Γ_F
A
B
D
$\mathbf{x}_0$
E
S
C
x_1
x_3

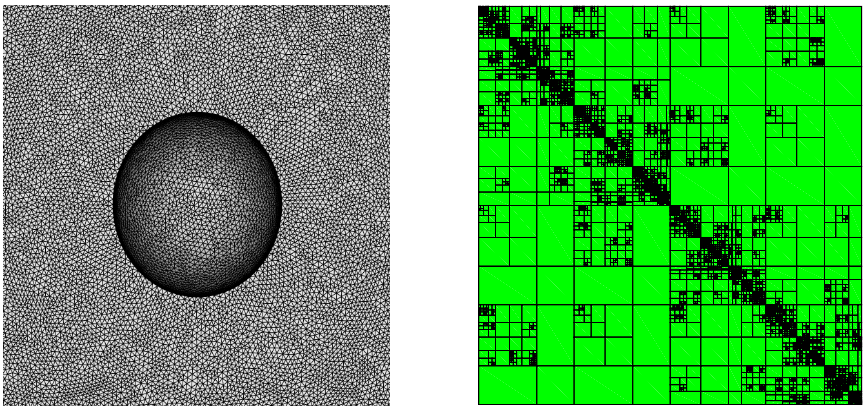
Figure 4.13: Forced vibration problem: elastic half-space with semi-spherical canyon.

We discretize $\Gamma_{\mathrm{F}}(L)$ with a density of $d_{\lambda_{\mathrm{S}}} = 10$ degrees of freedom per S-wavelength. This leads to a mesh with 73 595 triangular boundary elements and 37 079 collocation nodes. Consequently, the number of degrees of freedom of the problem is $N = 111\,237$ and the circular frequency is $\omega = 12.15$ (see Figure 4.14a). Since the geometry of the boundary represents the only difference with the problem treated in the previous Section, the BEM discretization leads to the same matrix equation, i.e.

$$\left(-\frac{1}{2}I + D\right)\tilde{\mathbf{u}} = S\tilde{\mathbf{t}}. \tag{4.25}$$

To assemble the hierarchical representations $D_{\mathcal{H}}$ and $S_{\mathcal{H}}$ of matrices D and S respectively, a hierarchical cluster tree is constructed with a minimum number of elements $N_{\mathrm{LEAF}} = 100$, resulting in 965 clusters and 11 cluster levels. The parameter $\eta = 3$ is used in the admissibility condition with, consequently, 14 305 block clusters in the block cluster tree. The $\mathcal{H}$-matrix structure of $D_{\mathcal{H}}$ and $S_{\mathcal{H}}$ is shown in Figure 4.14b.

For the approximation of the low-rank blocks, we propose a mixed approach: we use the vector ACA to compute the low-rank approximations of admissible blocks with cluster pair in different planes, while we use the randomized SVD to calculate the low-rank approximations of admissible blocks corresponding to cluster pair in the same plane. The threshold $\varepsilon_{\mathrm{ACA}} = 10^{-4}$ is used in the vector ACA algorithm and the same value is assigned to the parameter $\varepsilon_{\mathrm{rSVD}}$ in the randomized SVD algorithm. The truncated SVD recompression is applied to each admissible block with the parameter $\varepsilon_{\mathrm{SVD}} = 10^{-4}$. A tolerance $\varepsilon_{\mathrm{LU}} = 10^{-4}$ is used to perform the $\mathcal{H}$-matrix arithmetic during the $\mathcal{H}$-LU factorization.



(a) Mesh of the basin

(b) $\mathcal{H}$-matrix representation of matrices D and S

Figure 4.14: Excitation of an elastic half-space with a semi-spherical alluvial basin: mesh of the basin and $\mathcal{H}$-matrix representation of the BEM matrix ($N = 111\ 237$).

Since the analytical solution of the problem is unknown, in Figure 4.15 and 4.16 we compare the results obtained by the proposed $\mathcal{H}$-matrix based direct solver with the results obtained by the standard BEM. The three components of the displacement field $\mathbf{u}$ are plotted along line ABCDE (with points $A = (-5, 0, 0)$, B, C, D and $E = (5, 0, 0)$ defined in Figure 4.13), with a density of 10 points per S-wavelength. We observe that the results produced by the two approaches are in good agreement and allow us to conclude the level of accuracy achieved by the $\mathcal{H}$-matrix solution strategy is of the same order of the standard BEM.

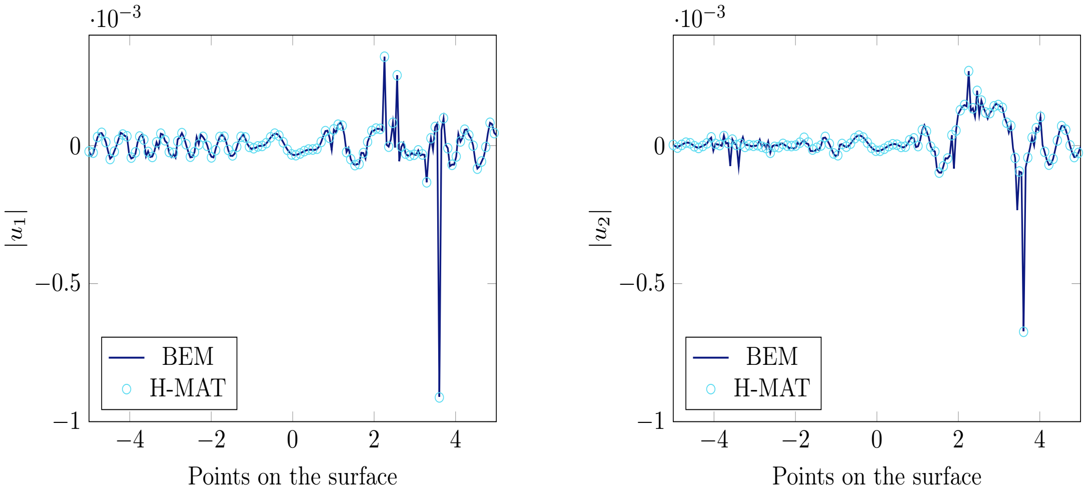


Figure 4.15: Excitation of an elastic half-space with a semi-spherical alluvial basin: comparison between the results obtained with the standard BEM and the proposed $\mathcal{H}$-matrix based direct solver.

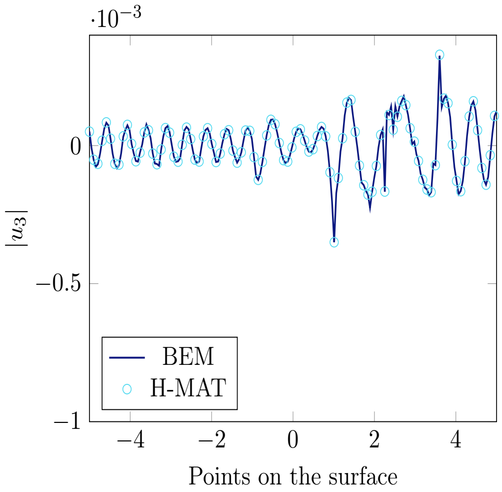


Figure 4.16: Excitation of an elastic half-space with a semi-spherical alluvial basin: comparison between the results obtained with the standard BEM and the proposed $\mathcal{H}$-matrix based direct solver.

The model just presented does not describe a realistic scenario, because its surface is not smooth in the points $B = (-1, 0, 0)$ and $D = (1, 0, 0)$, connecting the alluvial basin and the free-surface. This is the explication of the noisy events that we observe in the representation of the solution.

4.6 Conclusions

In this Chapter, we have extended the $\mathcal{H}$-matrix based Boundary Element Method to forced-vibration problems. Firstly, we have proposed a hierarchical multi-domain BEM formulation, based on single-region hierarchical BEM presented in Chapter 3. The proposed BEM-BEM coupling formulation is naturally suited to define $\mathcal{H}$-matrix based solution strategies, because the entire coefficient matrix is partitioned and decomposed using Schur component. The analysis of the response of a homogeneous medium to time-harmonic point load (Green's function of a homogeneous half-space) shows the numerical efficiency of the method and suggest that it is suitable to deal with realistic applications. In this context, we have presented the randomized Singular Value Decomposition as an alternative to the vectorial partially-pivoted Adaptive Cross Approximation, introduced in Chapter 2. For the validation of the $\mathcal{H}$-matrix based direct solver in the case of an half-space with a topographic irregularity, we have considered the 3D model corresponding to a semispherical alluvial basin. We have observed that the results produced by the proposed hierarchical BEM are in good agreement with the results provided by the standard BEM. In this way, we have concluded that the level of accuracy achieved by the $\mathcal{H}$-matrix solution strategy is of the same order of the standard BEM.

5 Conclusions and Directions for Future Work

Contents

5.1 Conclusions 124

5.2 Directions for Future Work 125

5.1 Conclusions

We have developed an innovative fast 3D Boundary Element Method based on $\mathcal{H}$-matrices. Taking advantages of published developments for Helmholtz, Laplace and Lamé equations, the $\mathcal{H}$-matrices have been successfully extended to elastodynamics in the frequency domain in Chapter 3. The generalization of the scalar-value Adaptive Cross Approximation (ACA) to the matrix-valued problems has provided low-rank approximations for matrix blocks corresponding to admissible cluster pairs, resulting in an improved computational efficiency compared to classical boundary element methods. Vectorial ACA yields the near-best-possible approximation. Moreover, by combining it with a subsequent application of the Singular Value Decomposition (SVD) the best-possible approximation can be obtained. The extension of ACA presents all the features of standard ACA, i.e. its black-box and self-controlling nature. Due to the purely algebraic nature of $\mathcal{H}$-matrices, approximations of the usual matrix operations (addition, multiplication, inversion,etc...) can be computed with logarithmic-linear complexity and allow to define fast solvers. In particular, the $\mathcal{H}$-matrix/vector product has been used to accelerate the classical matrix/vector product and, consequently, to reduce the time of each iteration of a GMRES based iterative solver, which is the most expensive task for this type of solution method. A deterioration of the convergence rate (number of iterations) has been observed for large problem sizes. No preconditioner was used in this work, in order to compare the proposed iterative solver with a Fast Multipole based iterative solver. In parallel, $\mathcal{H}$-matrix arithmetic has been used to approximate the factors of the LU decomposition, in order to reduce the numerical effort of a direct solver. Both the proposed solvers permit to reduce the computational burden in CPU time, for the simulation of wave propagation and allow to run models of size $N = \mathcal{O}(10^6)$ on an ordinary PC. The accuracy of the methods has been tested in the context of scattering of time-harmonic elastic P-waves by a unit sphere, comparing the results with exact and previously-published numerical solutions. In particular, for the direct solver an error estimate has been proposed to certify the quality of the solution, when the analytical solution is unknown.

Next, as the other aim of this thesis was to develop a fast direct solver to treat realistic wave propagation problems, the ability of the $\mathcal{H}$-matrix based direct solver to deal with forced-vibration problems has been investigated in Chapter 4. Since the soil can not be considered as a homogeneous medium but can be approximated by piece-wise homogeneous media, whose thickness are irregular due to several abrupt and gradual variations. The natural extension of the proposed $\mathcal{H}$-matrix BEM to multi-domain problems consists in forming the coefficient matrix simply by assembling the coefficient matrices of each sub-domain and the interface conditions between sub-domains, without eliminating any unknown variables on the interfaces. As proved in the case of potential problems (see [91]), this approach is efficient with the fast multipole method but it is naturally suited to define a $\mathcal{H}$-matrix based direct and iterative solution strategies, because the entire coefficient matrix is partitioned and decomposed using Schur component. Since the single-layer and the double-layer operators presented in the integral formulations relative to scattering problems, are the basis of the integral equations suitable to treat the new problems, the main difficulty of the extension of the solvers has been linked to the quality of the approximation provided by the vector ACA, when we evaluate the double-layer operator. A mixed approach has been proposed: vectorial ACA has been used to compute the low-rank approximation of admissible blocks with cluster pairs in different planes, while Randomized Singular Value Decomposition (rand SVD) algorithm has been used to approximate admissible blocks corresponding to cluster pair in the same plane. This technique, always in association with a standard SVD recompression, has been proved to be a very useful alternative to the vectorial ACA. The efficiency of the method to compute the response of an elastic half-space, with or without topographic irregularities, to a time-harmonic point load has been shown, comparing the results with analytical solution or standard BEM.

5.2 Directions for Future Work

This work was the first stage of the development of a fast direct solver for frequency-domain elastodynamics, using $\mathcal{H}$-matrices, at ENSTA-ParisTech in the laboratory POEMS. Before this thesis, no efficient BEM direct solver for large 3D elastodynamic problems was available. In view of the encouraging results obtained in this work, further research will be devoted to increase the capabilities and possibilities of this elastodynamic solver. Some possible directions for future work are now briefly discussed.

Viscoelasticity. In this work, only linear elastodynamics has been considered but the ideal model of an elastic soil is often insufficiently realistic. The introduction of damping, using a viscoelastic law, complicates the computation of the response to dynamic excitation but usually provides a much better description of the material, because it takes into account the transformation of mechanical energy into thermic energy, i.e. heat, closely related to such mechanisms as friction between particles or grains, molecular collisions or irreversible intercrystal heat flux. In Section 1.8 we have seen that viscoelasticity can be easily derived from elastodynamics. The classical method consists in introducing complex-valued elastic constants.

Anisotropy. Many kinds of rock and most sedimentary soil show an anisotropic behaviour, reflecting the history of the medium (deposition, deformation, consolidation, etc...). In elastodynamics, the main difference between isotropic and anisotropic materials is the fact that in an isotropic material, the speed of wave propagation is the same in all directions, whereas in anisotropic materials the wave speed varies. The extension of the $\mathcal{H}$-matrix based direct solver to anisotropic cases is not straightforward because an efficient way to estimate the numerical elastic anisotropic Green's functions (see [168]) has to be reformulated and developed. During the master thesis of Aurore Texier, realized at the laboraty POEMS (ENSTA-ParisTech), initial numerical investigations have been considered in order to ensure that the numerical computation is accurate and fast enough to be integrated into the $\mathcal{H}$-matrix based direct solver. The first results, obtained by using the Radom transform, are encouraging.

Time-domain response computation. The $\mathcal{H}$-matrix based direct solver can be used to deal with time-domain problems, via Fourier synthesis, taking advantage of the accelerated BEM at each sampling frequency. The solution for each frequency can been computed independently, i.e. without using previously-obtained solutions at lower frequencies. An alternative strategy consists in working directly in time-domain incorporating the full-space or layered half-space time-domain Green's function into the $\mathcal{H}$-matrix based direct solver. These fundamental solutions contain convolution integrals with respect to the time, that can be performed with the convolution quadrature method. This technique has been successfully used by Messner and Schanz [131] in order to define an $\mathcal{H}$-matrix based iterative solver.

Parallelization. The present implementation of the elastodynamic $\mathcal{H}$-matrix based direct solver has been done for single-processor platforms. The introduction of the $\mathcal{H}$-matrices has been shown in this thesis to enhance the capability of the standard BEM. Now, with the increasing performance of computers, the parallelization of the code would further extend the capabilities of the method in terms of BEM model size of frequency range. During his internship of Master 1, Shashank Kumar Anand worked on the parallelization of the construction of the $\mathcal{H}$-matrix representation of BEM matrix, obtaining a drastic reduction of the CPU time. Instead, the parallelization of $\mathcal{H}$-LU factorization is still a difficult task, because various stages of the algorithm link at least two sub-blocks. Recently, the idea of splitting the $\mathcal{H}$-LU factorization into single tasks and of defining corresponding dependencies to form a Direct Acyclic Graph (DAG), has been presented (see [112]). This task/DAG based algorithm is able to utilize parallel CPU much more efficiently compared to the recursive algorithm and, in particular, demonstrates an optimal parallel scaling behavior on many-core systems.

Preconditioning. For BEM models of size $N = \mathcal{O}(10^6)$ or more, the iteration count of an $\mathcal{H}$-matrix or Fast Multipole based iterative solver was found to be a major limiting factor. The definition of an effective preconditioner is crucial for developing an efficient iterative solver, able to compete with the proposed direct solver. Although finding appropriate preconditioners seems to be a purely algebraic problem at first glance, efficient preconditioners usually rely on the analytic background of the linear system. A very popular idea is approximations of the inverse, so called Sparse Approximate Inverses (SPAI), or of the LU factorization, so called Incomplete LU decomposition (ILU). To improve the construction of these preconditioners, one possibility is to use the $\mathcal{H}$-matrix arithmetic with a small threshold to compute an $\mathcal{H}$-SPAI or an $\mathcal{H}$-ILU. An alternative (see [69]) is a preconditioner based on low-rank compression of Schur complements, whose construction is inspired by standard nested-dissection and relies on the assumption that Schur complements can be approximated to high precision by Hierarchical-Block-Separable matrices. The preconditioner is built as an approximate $LDM^\top$ factorization (no knowledge of the matrix system in assembled form is required), whose inversion is fast to compute and can be applied as fast as well.

Coupling with other numerical methods. In this work, soil has been treated as a viscoelastic material, applying a linear elastic continuum model. This simplified approach may provide results of an acceptable quality for dry granular materials, which may adequately be described as a single-phase system, i.e. wave propagation only takes place as shear and dilatation in the grain skeleton. However, in several problems concerning saturated soil, i.e. liquefaction, it is necessary to model soil as a two-phase system, because the pore pressure in the fluid phase interacts with the stress carried by the solid phase. This results in a second P-wave in addition to P-wave that is identified in a single-phase elastic material. Poro-elastodynamic problems can be solved using a coupling BEM/FEM strategy, in order to use the adequateness of BEM to deal with unbounded media (see [156]) and the flexibility of FEM to deal with non-linear materials (see [177]). An interesting perspective of this work is to extend the numerical efficiency of $\mathcal{H}$-matrices to poro-elastodynamic FEM/BEM coupling, taking advantage of recent results in dynamic soil-structure interaction (see [52]).

Inverse Problem. In recent years inverse elastodynamic problems have received increasing attention due to a broad range of engineering necessities. Inverse models employ measurements to obtain a better understanding of the system behavior and to infer system parameters. In particular, 3D imaging of cavities embedded in a semi-infinite solid using elastic waves is a topic of intrinsic interest in a number of applications, ranging from non-destructive material testing to oil prospecting underground and object detection. In situations when detailed mapping of buried objects is required and only a few measurements can be made, the Boundary Element Method provides the most direct link between the surface measurements and the buried geometrical objects. Since inverse analysis usually require recursive or iterative computation of the forward model, the forward calculations have an especially huge effect on the computing time and storage requirement. For this reason, the developed $\mathcal{H}$-matrix based direct solver paves the way for the solution of this inverse problem.

Bibliography

[1] B.T. Aagaard, J.F. Hall, and T.H. Heaton. Characterization of near-source ground motions with earthquake simulations. *Earthquake Spectra*, 17(2):177–207, 2001.

[2] J.D. Achenbach. *Wave propagation in elastic solids.* North-Holland Publishing Company, 1975.

[3] A. Aimi, M. Diligenti, A. Frangi, and C. Guardasoni. A stable 3D energetic Galerkin BEM approach for wave propagation interior problems. *Engineering Analysis with Boundary Elements*, 36(3):1756–1765, 2012.

[4] A. Aimi, M. Diligenti, and F. Lunardini. Panel clustering method and restriction matrices for symmetric Galerkin BEM. *Numerical Algorithms*, 40:355–382, 2005.

[5] K. Aki and Richards P.G. *Quantitative Seismology: Theory and Methods.* W. H. Freeman and Company, San Francisco, California, 1980.

[6] L. Andersen. *Linear Elastodynamic Analysis.* Aalborg University, 2006.

[7] H. Aochi, T. Ubrich, A. Ducellier, F. Dupros, and D. Michea. Finite difference simulations of seismic wave propagation for understanding earthquake physics and predicting ground motions: Advances and challenges. *Journal of Physics: Conference Series*, 454(1):1–9, 2013.

[8] P.K. Banerjee and S.M. Mamoon. A fundamental solution due to a periodic point force in the interior of an elastic half-space. *Earthquake Engineering and Structural Dynamics*, 19:91–105, 1990.

[9] L. Banjai and W. Hackbusch. Hierarchical matrix techniques for low and high frequency Helmholtz problems. *IMA Journal of Numerical Analysis*, 28:46–79, 2008.

[10] L. Banjai and M. Schanz. Wave propagation problems treated with convolution quadrature and BEM. *Lecture Notes in Applied and Computational Mechanics*, 63:145–184, 2012.

[11] H. Bao, J. Bielak, O. Ghattas, L.F. Kallivokas, D.R. O'Hallaronb, J.R. Shewchukb, and J. Xu. Large-scale simulation of elastic wave propagation in heterogeneous media on parallel computers. *Computer Method in Applied Mechanics and Engineering*, 152:85–102, 1998.

[12] U. Basu and A.K. Chopra. Perfectly matched layers for time-harmonic elastodynamics of unbounded domains: theory and finite-element implementation. *Computer Methods in Applied Mechanics and Engineering*, 192:1337–1375, 2003.

[13] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86:565–589, 2000.

[14] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74:225–247, 2005.

[15] M. Bebendorf. *Hierarchical matrices: a means to efficiently solve elliptic boundary value problems*. Springer, 2008.

[16] M. Bebendorf and S. Kunis. Recompression techniques for Adaptive Cross Approximation. *Journal of Integral Equations and Applications*, 21(3):331–357, 2009.

[17] M. Bebendorf, C. Kuske, and R. Venn. Wideband nested cross approximation for Helmholtz problems. Technical Report SFB 611 Preprint, Universität Bonn, 2012.

[18] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.

[19] A. Ben-Menahem. A concise history of mainstream seismology: origins, legacy and prespectives. *Bulletin of the Seismological Society of America*, 85(4):1202–1225, 1995.

[20] I. Benedetti and M.H. Aliabadi. A fast hierarchical dual boundary element method for three-dimensional elastodynamic crack problems. *International Journal for Numerical Methods in Engineering*, 84(9):1038–1067, 2010.

[21] J. Bielak, J. Xu, and O. Ghattas. Earthquake ground motion and structural response in alluvional valleys. *Journal of Geotechnical and Geoenviromental Engineering*, 125:413–423, 1999.

[22] M. Bonnet. *Boundary Integral Equation Method for Solids and Fluids*. Wiley, 1999.

[23] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Numerische Mathematik*, 101:221–249, 2005.

[24] S. Börm, L. Grasedyck, and W. Hackbusch. Hierarchical matrices. Technical report, Max Planck Institute for Mathematics, 2003.

[25] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27:405–422, 2003.

[26] C.A. Brebbia and R. Butterfield. Formal equivalence of direct and indirect boundary element methods. *Applied Mathematical Modelling*, 2(2):132–134, 1978.

[27] C.A. Brebbia and J. Dominiguez. *Boundary Elements - An introductory course.* Southampton CMP.

[28] C.A. Brebbia, J.C.F. Telles, and L.C. Wrobel. *Boundary element techniques - theory and applications in engineering.* Springer-Verlag, 1984.

[29] H.D. Bui, B. Loret, and M. Bonnet. Régularisation des équations intégrales de l'élastostatique et de l'élastodynamique. *Comptes rendus de l'Academie des Sciences Serie II*, 300:633–636, 1985.

[30] J.M. Carcione. The wave equation in generalized coordinates. *Geophysics*, 59:1911–1919, 1994.

[31] J.M. Carcione and P.J. Wang. A Chebyshev collocation method for the wave equation in generalized coordinates. *Computational Fluid Dynamics Journal*, 2:269–290, 1993.

[32] W. Chai and D. Jiao. A complexity-reduced $\mathcal{H}$-matrix based direct integral equation solver with prescribed accuracy for large-scale electrodynamic analysis. *IEEE Antennas and Propagation Society International Symposium*, 2010:1–4, 2010.

[33] S. Chaillat. *Fast multipole method for 3D elastodynamic boundary integral equations. Application to seismic wave propagation.* PhD thesis, École Nationale des Ponts et Chaussées, 2008.

[34] S. Chaillat and G. Biros. FaIMS: a fast algorithm for the inverse medium problem with multiple frequencies and multiple sources for the scalar helmholtz equation. *Journal of Computational Physics*, 231:4403–4421, 2012.

[35] S. Chaillat and M. Bonnet. Recent advances on the fast multipole accelerated boundary element method for 3D time-harmonic elastodynamics. *Wave Motion*, 50:1090–1104, 2013.

[36] S. Chaillat and M. Bonnet. A new fast multipole formulation for the elastodynamic half-space Green's tensor. *Journal of Computational Physics*, 258:787–808, 2014.

[37] S. Chaillat, M. Bonnet, and J.F. Semblat. A multi-level fast multipole BEM for 3D elastodynamics in the frequency domain. *Computer Method in Applied Mechanics and Engineering*, 197:4233–4249, 2008.

[38] S. Chaillat, M. Darbas, and Le Louër. Approximate local Dirichlet-to-Neumann map for three-dimensional elastic waves. *Computer Methods in Applied Mechanics and Engineering*, 297:62–83, 2015.

[39] S. Chaillat, J.F. Semblat, and M. Bonnet. A preconditioned 3D multi-region fast multipole solver for seismic wave propagation in complex geometries. *Communications in Computational Physics*, 11:594–609, 2012.

[40] E. Chaljub. *Modélisation numérique de la propagation des ondes sismiques en géometrie spherique: application à la sismologie globale.* PhD thesis, Université Paris VII Denis Diderot, Paris, France, 2000.

[41] E. Chaljub, Y. Capdeville, and J.P. Vilotte. Solving elastodynamics in a fluid-soild heterogeneous sphere: a parallel spectral element approximation on non-conforming grids. *Journal of Computational Physics*, 187(2):457–491, 2003.

[42] E. Chaljub and B. Valette. Spectral-element modeling of three-dimensional wave propagation in a self-graviting earth with an arbitrary stratified outer core. *Geophysical Journal International*, 158:131–141, 2004.

[43] C.C. Chao. Dynamical response of an elastic half-space to tangential surface loadings. *Journal of Applied Mechanics*, 27:559–567, 1960.

[44] A.H.D. Cheng and D.T. Cheng. Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements*, 29(3):268–302, 2005.

[45] R.M. Christensen. *Theory of viscoelasticity.* Courier Dover Publications, 2nd edition, 2003.

[46] R. Clayton and B. Engquist. Absorbing boundary conditions for acoustic and elastic wave equations. *Bulletin of the Seismological Society of America*, 67(6):1529–1540, 1977.

[47] D. Clouteau and D. Aubry. Computational soil-structure interaction. In W.S Hall and G. Oliveto, editors, *Boundary element methods for soil-structure interaction*, pages 61–125. Kluwer Academic Publishers, 2004.

[48] G. Cohen, P. Joly, and N. Tordjman. Construction and analysis of higher-order finite elements with mass lumping for the wave equation. In Kleinman R, editor, *Proceedings of the Second International Conference on Mathematical and Numerical Aspects of Wave Propagation*, pages 152–160, Philadephia, PA, 1993. SIAM.

[49] F. Collino and C. Tsogka. Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*, 66(1):294–307, 2001.

[50] D. Colton and R. Kress. *Inverse acoustic and electromagnetoc scattering theory.* Springer, 1992.

[51] P. Coulier, S. François, G. Lombaert, and G. Degrande. Application of hierarchical matrices to boundary element methods for elastodynamics based on Green's functions for a horizontally layered half-space. *Engineering Analysis with Boundary Elements*, 37:1745–1758, 2013.

[52] P. Coulier, S. François, G. Lombaert, and G. Degrande. Coupled finite element-hierarchical boundary element methods for dynamic soil-structure interaction in the frequency domain. *International Journal for Numerical Methods in Engineering*, 97(7):505–530, 2014.

[53] T.A. Cruse. Numerical solutions in three-dimensional elastostatics. *International Journal of Solids Structures*, 5:1259–1274, 1969.

[54] T.A. Cruse. Recent advances in boundary element analysis methods. *Computer Method in Applied Mechanics and Engineering*, 62(3):227–244, 1987.

[55] T.A. Cruse and F.J. Rizzo. A direct formulation and numerical solution of the general transient elastodynamic problem. I. *Journal of Mathematical Analysis and Applications*, 22(1):244–259, 1968.

[56] P. Dangla, J.F. Semblat, H. Xiao, and N. Delépine. A simple and efficient regularization method for 3D BEM: application to frequency domain elastodynamics. *Bulletin of the Seismological Society of America*, 95:1916–1927, 2005.

[57] R. Dautray and J.L. Lions. *Mathematical analysis and numerical methods for Science and Technology*, volume 1. Springer-Verlag, 1990.

[58] J. de la Puente, J.P. Ampuero, and M. Käser. dynamic rupture modeling on unstructured meshes using a discontinuous Galerkin method. *Geophysics*, 114:B10302, 2009.

[59] J. de la Puente, M. Dumbser, M. Käser, and H. Igel. Discontinuous Galerkin methods for wave propagation in poroelastic media. *Geophysics*, 73(5):77–97, 2008.

[60] J. de la Puente, M. Käser, M. Dumbser, and H. Igel. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - IV Anisotropy. *Geophysical Journal International*, 169:1210–1228, 2007.

[61] J. Dominguez. Dynamic stiffness of rectangular foundations. Technical Report R78-20, Departement of Civil Engineering, MIT, Cambridge, MA, 1978.

[62] J. Dominguez. *Boundary Elements in Dynamics*. Computational Mechanics Publications, Southampton Boston, 1993.

[63] M. Dumbser and M. Käser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - II The three-dimensional isotropic case. *Geophysical Journal International*, 167:319–336, 2006.

[64] E. Faccioli, F. Maggio, R. Paolucci, and A. Quarteroni. 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method. *Journal of Seismology*, 1:237–251, 1997.

[65] G. Festa and J.P. Vilotte. The newmark scheme as velocity-stress time-staggering: an efficient PML implementation for spectral element simulations of elastodynamics. *Geophysical Journal International*, 161:789–812, 2005.

[66] T. Furumura, B.L.N. Kennett, and M. Furumura. Seismic wavefield calculation for laterally heterogeneous whole Earth models using the pseudospectral method. *Geophysical Journal International*, 135(3):845–860, 1998.

[67] T. Furumura, B.L.N. Kennett, and M. Furumura. Seismic wavefield calculation for laterally heterogeneous whole Earth models - II. The influence of the upper mantle heterogeneity. *Geophysical Journal International*, 139(3):623–644, 1999.

[68] K. Garatani, H. Nakamura, H. Okuda, and G. Yagawa. Large-scale parallel wave propagation analysis by GeoFEM. *Lecture Notes in Computer Science*, 1823:445–453, 2000.

[69] P. Gatto and J.S. Hesthaven. A preconditioner based on low-rank approximation of schur complements. *arXiv preprint arXiv:1508.07798*, 2015.

[70] R.J. Geller and N. Takeuchi. A new method for computing highly accurate DSM synthetic seismograms. *Geophysical Journal International*, 123:449–470, 1995.

[71] H. Goldstein. *Classical Mechanics, 2nd ed.* Addison-Wesley, Reading MA., 1980.

[72] S.A. Goreinov. Mosaic-skeleton approximations of matrices generated by asymptotically smooth and oscillatory kernels. pages 41–76, Moscow, 1999.

[73] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[74] E. Grasso, S. Chaillat, M. Bonnet, and J.F. Semblat. Application of the multi-level time-harmonic fast multipole bem to 3D visco-elastodynamics. *Engineering Analysis with Boundary Elements*, 36:744–758, 2012.

[75] R.W. Graves. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences. *Bulletin of the Seismological Society of America*, 86(4):1091–1106, 1996.

[76] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[77] M. Guiggiani and A. Gigante. A general algorithm for multidimensional Cauchy principal value integrals in the boundary element method. *Journal of Applied Mechanics*, 57:906–915, 1990.

[78] B.B. Guizina. *Seismic response of foundations and structures in multilayered media.* PhD thesis, University of Colorado, 1996.

[79] B.B. Guizina and R.Y.S. Pak. On the analysis of wave motions in a multi-layered solid. *Quarterly Journal of Mechanics and Applied Mathematics*, 54:13–37, 2001.

[80] H. Guo, J. Hu, H. Shao, and Z. Nie. Hierarchical matrices method and its application in electromagnetic integral equations. *International Journal of Antennas and Propagation*, 2012:1–9, 2012.

[81] W. Hackbusch. A sparse arithmetic based on $\mathcal{H}$-Matrices. Part I: Introduction to $\mathcal{H}$-Matrices. *Computing*, 62:89–108, 1999.

[82] W. Hackbusch. *$\mathcal{H}^2$-Matrices.* Springer, 2015.

[83] W. Hackbusch and B.N. Khoromskij. A sparse arithmetic based on $\mathcal{H}$-Matrices. Part II: Application to Multi-Dimensional Problems. *Computing*, 64:21–47, 2000.

[84] W. Hackbusch and B.N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic: general complexity estimates. *Journal of computational and applied mathematics*, 125:479–501, 2000.

[85] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the Boundary Element Method by Panel Clustering. *Numerische Mathematik*, 54:463–491, 1989.

[86] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randmness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[87] K. Hayami and S.A. Sauter. Application of the panel clustering method for 3D elastostatic problem. In *Boundary Elements XIX*, Southampton, 1997. Computational Mechanics Publications.

[88] J.L. Hess and A.M.O. Smith. *Calculation of potential flow about arbitrary bodies*, volume 8. Pergamon Press, New York, 1966.

[89] T.K. Hong and B. Kennett. A wavelet-based method for simulation of two-dimensional elastic wave propagation. *Geophysical Journal International*, 150:610–638, 2002.

[90] G.C. Hsiao. Boundary element methods - An overview. *Applied Numerical Mathematics*, 56(10-11):1356–69, 2006.

[91] S. Huang and Y.J. Liu. A new simple multidomain fast multipole boundary element method. *Computational Mechanics*, 58(3):533–548, 2016.

[92] T.J.R. Hughens. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis.* Prentice-Hall International, Englewood Cliffs, NJ, 1987.

[93] S.H. Hung and D. Forsyth. Modelling anisotropic wave propagation in oceanic inhomogeneous structures using the parallel multidomain pseudo-spectral method. *Geophysical Journal International*, 133(3):726–740, 1998.

[94] H. Igel. Wave propagation in three-dimensional spherical sections by the Chebyshev spectral method. *Geophysical Journal International*, 136:559–566, 1999.

[95] H. Igel, T. Nissen-Meyer, and G. Jahnke. Wave propagation in 3D spherical sections: Effects of subduction zones. *Physics of the Earth and Planetary Interiors*, 132:219–234, 2002.

[96] M. Käser, M. Dumbser, J. de la Puente, and H. Igel. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - III viscoelastic attenuation. *Geophysical Journal International*, 168:224–242, 2007.

[97] B.L.N. Kennett. Seismic wave propagation in stratified media. In *Advances in Applied Mechanics*, volume 21. Academic Press, 1981.

[98] D. Komatitsch. *Méthodes spectrales et éléments spectraux pour l'équation de l'élastodynamique 2D et 3D en milieu hétérogène*. PhD thesis, Institut de Physique du Globe, Paris, France, 1997.

[99] D. Komatitsch, Q. Liu, J. Tromp, P. Süss, C. Stidham, and J.H. Shaw. Simulations of ground motion in the Los Angeles Basin based upon the spectral-element method. *Bulletin of the Seismological Society of America*, 94:187–206, 2004.

[100] D. Komatitsch, J. Ritsema, and J. Tromp. The spectral-element method, Beowulf computing, and global seismology. *Science*, 298:1737–1742, 2002.

[101] D. Komatitsch and J. Tromp. Introduction to the spectral element method for 3D seismic wave propagation. *Geophysical Journal International*, 139:806–822, 1999.

[102] D. Komatitsch and J. Tromp. Spectral element simulations of global seismic wave propagation. I: Validation. *Geophysical Journal International*, 149:390–412, 2002.

[103] D. Komatitsch and J. Tromp. Spectral element simulations of global seismic wave propagation. II: 3D models, oceans, rotation and self-gravitation. *Geophysical Journal International*, 150:303–318, 2002.

[104] D. Komatitsch and J. Tromp. A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation. *Geophysical Journal International*, 154:146–153, 2003.

[105] D. Komatitsch, S. Tsuboi, J. Chen, and J. Tromp. A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth simulator. In *Proceeding of the ACM/IEEE Supercomputing SC'2003 Conference*, 2003.

[106] D. Komatitsch and J.P. Vilotte. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Seismological Society of America*, 88(2):368–392, 1998.

[107] D. Kosloff and E. Baysal. Forward modeling by the Fourier method. *Geophysics*, 47:1402–1412, 1982.

[108] D. Kosloff, D. Kessler, A.Q. Fiho, E. Tessmer, A. Behle, and R. Strahilevitz. Solution of the equations of dynamics elasticity by a Chebyshev spectral method. *Geophysics*, 55:748–754, 1990.

[109] D. Kosloff, M. Reshef, and D. Loewenthal. Elastic wave calculations by the Fourier method. *Bulletin of the Seismological Society of America*, 74:875–891, 1984.

[110] D. Kosloff and H. Tal-Ezer. A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction. *Journal of Computational Physics*, 104:457–469, 1993.

[111] S.L. Kramer. *Geotechnical earthquake engineering.* International Series in Civil Engineering and Engineering Mechanics. Prentice-Hall, 1996.

[112] R. Kriemann. $\mathcal{H}$-LU factorization on many-core systems. *Computing and Visualization in Science*, 16(3):105–117, 2013.

[113] G. Krishnasamy, F.J. Rizzo, and T.J. Rudolphi. Hypersingular boundary integral equations their occurrence, interpretation, regularization and computation. In P.K. Benerjee and S. Kobayashi, editors, *Developments in Boundary Element Methods*. Elsevier, 1992.

[114] H. Lamb. On the propagation of tremors over the surface of an elastic solid. *Proceedings of the royal society of London*, 203(1-42), 1904.

[115] S. Le Borne and L. Grasedyck. $\mathcal{H}$-preconditioners in convection-dominated problems. *SIAM Journal on Matrix Analysis and Applications*, 27(4):1172–1183, 2006.

[116] A.R. Levander. Fourth-order finite-difference P-SV seismograms. *Geophysics*, 53:1425–1436, 1988.

[117] L. Lin, J. Lu, and L. Ying. Fast construction of hierarchical matrix representation from matrix-vector multiplication. *Journal of Computational Physics*, 230:4071–4087, 2011.

[118] B. Lizé. *Résolution Directe Rapide pour les Éléments Finis de Frontière en Électromagnétisme et Acoustique: $\mathcal{H}$-Matrices. Parallélisme et Applications Industrielles.* PhD thesis, Université Paris 13, 2013.

[119] J.E. Luco and F.C.P. De Barros. On the three-dimensional seismic response of a class of cylindrical inclusions. In *Proceeding of the Sixth International Conference of Soil Dynamics and Earthquake*, pages 565–580, Bath, UK, 1993.

[120] J.E. Luco, H.L. Wong, and F.C.P. De Barros. Three-dimensional response of a cylindrical canyon in a layered half-space. *Earthquake Engineering and Structural Dynamics*, 19:799–817, 1990.

[121] F. Luzón, F.J. Sánchez-Sesma, J.L. Rodríguez-Zúñiga, A.M. Posadas, J.M. García, J. Martín, M.D. Romacho, and M. Navarro. Diffraction of P, S and Rayleigh waves by three-dimensional topographies. *Geophysical Journal International*, 129(3):571–578, 1997.

[122] J. Lysmer and L.A. Drake. A finite-difference method for seismology. In B.A. Bolt, editor, *Methods in Computational Physics*, volume 11, pages 181–216, New York, 1972. Academic Press.

[123] R. Madariaga. Dynamics of an expanding circular fault. *Bulletin of the Seismological Society of America*, 65:163–182, 1976.

[124] F. Maerten. Adaptive cross-approximation applied to the solution of system of equations and post-processing for 3D elastostatic problems using the boundary element method. *Engineering Analysis with Boundary Elements*, 34(5):483–491, 2010.

[125] L.E. Malvern. *Introduction to the mechanics of a continuous medium.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1969.

[126] W.J. Mansur. *A time-stepping technique to solve wave propagation problems using the boundary element method.* PhD thesis, University of Southampton, 1983.

[127] P.G. Martinisson, V. Rokhlin, and M. Tygert. A randomized algorithm for the approximation of matrices. Technical Report 1361, Department of Computer Science, Yale University, New Haven, CT, 2006.

[128] A. Maurel, V. Pagneux, F. Barra, and F. Lund. Interaction of a surface wave with a dislocation. *Physical Review B*, 75(22):224112(15), 2007.

[129] I. Mazzieri, C. Smerzini, P.F. Antonietti, F. Rapetti, M. Stupazzini, R. Paolucci, and A. Quarteroni. Non-conforming spectral approximations for elastic wave equation in heterogeneous media. In M. Papadrakakis, M. Fragiadakis, and V. Plevris, editors, *Eccomas Thematic Conference on Computational Methods in Structural Dynamics and Engineering*, Corfu, Greece, 25-28 May 2011.

[130] I. Mazzieri, M. Stupazzini, R. Guidotti, and C. Smerzini. SPEED-spectral elements in elastodynamics with discontinuous Galerkin: a non-conforming approach for 3D multi-scale problems. Technical Report 24/2013, MOX, Dipartimento di Matematica "F. Brioschi", Politecnico di Milano, Via Bonardi 9 - 20133 Milano, Italy, 2013.

[131] M. Messner and M. Schanz. An accelerated symmetric time-domain boundary element formulation for elasticity. *Engineering Analysis with Boundary Elements*, 34(11):944–955, 2010.

[132] A. Milazzo, I. Benedetti, and M.H. Aliabadi. Hierarchical fast BEM for anisotropic time-harmonic 3D elastodynamics. *Computers & Structures*, 96:9–24, 2012.

[133] G.F. Miller and H. Pursey. The field and radiation impendence of mechanical radiations on the free surface of a semi-infinite isotropic solid. *Proceedings of the royal society of London*, 223(1155):521–541, 1954.

[134] H.M. Mooney. Some numerical solutions for Lamb's problem. *Bulletin of the Seismological Society of America*, 64:473–491, 1974.

[135] G. Müller. Theory of elastic waves. Scientific Technical Report STR 07/03, Deutsches GeoForschungsZentrum GFZ, Postdam, 2007.

[136] O. Novotny. *Seismic Surface Waves*. Universidade Federal da Bahia - Centro de Pesquisa em Geofisica e Geologia, Salvador, Bahia, 1999.

[137] T. Ohminato and B.A. Chouet. A free-surface boundary condition for including 3D topography in finite difference method. *Bulletin of the Seismological Society of America*, 87:494–515, 1997.

[138] J. Ostrowski, Z. Andjelic, M. Bebendorf, B. Cranganu-Cretu, and J. Smajic. Fast BEM-solution of Laplace problems with $\mathcal{H}$-matrices and ACA. *IEEE Transactions on Magnatics*, 42(4):627–630, 2006.

[139] R.Y.S. Pak and B.B. Guzina. Seismic soil-structure interaction analysis by direct boundary element methods. *International Journal of Solids Structures*, 36:4743–4766, 1999.

[140] A.T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics*, 54:468–488, 1984.

[141] H. Pedersen, F.J. Sánchez-Sesma, and M. Campillo. Three-dimensional scattering by two-dimensional topographies. *Bulletin of the Seismological Society of America*, 84:1169–1183, 1994.

[142] C.L. Pekeris. The seismic surface pulse. *Proceedings of the National Academy of Sciences USA*, 41:469–480, 1955.

[143] A.V. Phan, V. Guduru, A. Salvadori, and L.J. Gray. Frequency domain analysis by the exponential window method and SGBEM for elastodynamics. *Computational Mechanics*, 48(5):615–630, 2011.

[144] E. Priolo, J.M. Carcione, and G. Seriani. Numerical simulation of interface waves by high-order spectral modeling techniques. *Journal of the Acoustical Society of America*, 95(2):681–693, 1994.

[145] A. Quarteroni, A. Tagliani, and E. Zampieri. Generalized Galerkin approximations of elastic waves with absorbing boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 163:323–341, 1998.

[146] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[147] P.G. Richards. Elementary solution to Lamb's problem for a point source and their relevance to three-dimensional studies of spontaneous crack propagation. *Bulletin of the Seismological Society of America*, 69:947–956, 1979.

[148] F.J. Rizzo. An integral equation approach to boundary value problems of classical elastostatics. *Quarterly Journal of Applied Mathematics*, 25:83–95, 1967.

[149] F.J. Rizzo, D.J. Shippy, and M. Rezayat. A boundary integral equation method for radiation and scattering. *International Journal for Numerical Methods in Engineering*, 21:115–129, 1985.

[150] J.O.A. Robertsson. A numerical free-surface condition for elastic/viscoelastic finite difference modeling in the presence of topography. *Geophysics*, 61:1921–1934, 1996.

[151] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187–207, 1985.

[152] Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM Journal on Scientific Computing*, 7:856–869, 1986.

[153] F.J. Sánchez-Sesma and M. Campillo. Diffraction of P, SV and Rayleigh waves by topographic features: a boundary integral formulation. *Bulletin of the Seismological Society of America*, 81(6):2234–2253, 1991.

[154] F.J. Sánchez-Sesma and M. Campillo. Topographic effects for incident P, SV and Rayleigh waves. *Tectonophysics*, 218:113–125, 1993.

[155] F.J. Sánchez-Sesma and F. Luzón. Seismic response of three-dimensional alluvion valley for incident P, S and Rayleigh waves: a boundary integral formulation. *Bulletin of the Seismological Society of America*, 85:269–284, 1995.

[156] M. Schanz. *Wave propagation in viscoelastic and poroelastic continua. A boundary element approach.* Lecture Notes in Applied and Computational Mechanics. 2001.

[157] M. Schanz and H. Antes. Application of the operational quadrature methods in time domain boundary element methods. *Meccanica*, 32(3):179–186, 1997.

[158] G. Seriani. 3D large-scale wave propagation modeling by a spectral element method on a Cray T3E multiprocessor. *Computer Methods in Applied Mechanics and Engineering*, 164:235–247, 1998.

[159] J. Shaeffer. Direct solve of electrically large integral equations for problem sizes to 1M unknowns. *IEEE Transactions on Antennas and Propagation*, 56:2306–2313, 2008.

[160] M. Stolper. Computing and compression of the boundary element matrices for the Helmholtz equation. *Journal of Numerical Mathematics*, 12(1):55–75, 2004.

[161] E. Tessmer. 3D seismic modelling of general material anisotropy in the presence of the free surface by a Chebyshev spectral method. *Geophysical Journal International*, 121:557–575, 1995.

[162] E. Tessmer, D. Kessler, D. Kosloff, and A. Behle. Multi-domain Chebyshev-Fourier method for the solution of the equations of motion of dynamic elasticity. *Journal of Computational Physics*, 100:355–363, 1992.

[163] E. Tessmer and D. Kosloff. 3D elastic modeling with surface topography by a Chebyshev spectral method. *Geophysics*, 59(3):464–473, 1994.

[164] T. Toshinawa and T. Ohmachi. Love wave propagation in a three-dimansional sedimentary basin. *Bulletin of the Seismological Society of America*, 82:1661–1677, 1992.

[165] J. Virieux. SH wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 49:1933–1942, 1984.

[166] J. Virieux. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51:889–901, 1986.

[167] J. Virieux, H. Calandra, and Plessix R.E. A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging. *Geophysical Prospecting*, 59:794–813, 2011.

[168] C.Y. Wang and J.D. Achenbach. Three-dimensional time-harmoinic elastodynamic Green's functions for anisotropic solids. In Royal Society, editor, *Proceedings: Mathematical and Physical Sciences*, volume 449, pages 441–458, 1995.

[169] Y. Wang and H. Takenaka. A multidomain approach of the Fourier pseudospectral method using discontinuous grid for elastic wave modeling. *Earth Planets Space*, 53:149–158, 2001.

[170] Y. Wang, H. Takenaka, and T. Furumura. Modelling seismic wave propagation in a two-dimensional cylindrical whole-Earth model using the pseudospectral method. *Geophysical Journal International*, 145:689–708, 2001.

[171] J.O. Watson. Boundary elements from 1960 to the present day. *Electronic Journal of Boundary Elements*, 1(1):34–46, 2003.

[172] L.T. Wheeler and E. Sternberg. Some theorems in classical elastodynamics. *Archive for Rational Mechanics Analysis*, 31(51-90), 1968.

[173] J. Xiao, W. Ye, Y. Cai, and J. Zhang. Precorrected FFT accelerated BEM for large-scale transient elastodynamic analysis using frequency-domain approach. *International Journal for Numerical Methods in Engineering*, 90(1):116–134, 2012.

[174] J. Xiao, W. Ye, and L. When. Efficiency improvement of the frequency-domain BEM for rapid transient elastodynamic analysis. *Computational Mechanics*, 52(4):903–912, 2013.

[175] K.I. Yoshida. *Application of fast multipole method to boundary integral equation method*. PhD thesis, Kyoto University, 2001.

[176] O.C. Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill, New York, 3 edition, 1977.

[177] O.C. Zienkiewicz and T. Shiomi. Dynamic behaviour of saturated porous media. The generalized biot formulation and its numerical solution. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8:71–96, 1984.

[178] D.W. Zingg. Comparison of high-accuracy finite-difference methods for linear wave propagation. *SIAM Journal on Scientific Computing*, 22(2):476–502, 2000.

[179] D.W. Zingg, H. Lomax, and H. Jurgens. High accuracy finite difference schemes for linear wave propagation. *SIAM Journal on Scientific Computing*, 17(2):328–346, 1996.

université PARIS-SACLAY
École doctorale de mathématiques Hadamard (EDMH)

Title: $\mathcal{H}$-matrix based Solvers for 3D Elastodynamic Boundary Integral Equations
Keywords: Boundary Element Method, $\mathcal{H}$-matrices, Adaptive Cross Approximation, Randomized Singular Value Decomposition, 3D Elastodynamics, Forced Vibration Problems.

Abstract: This thesis focuses on the theoretical and numerical study of fast methods to solve the equations of 3D elastodynamics in frequency-domain. We use the Boundary Element Method (BEM) as discretization technique, in association with the hierarchical matrices ($\mathcal{H}$-matrices) technique for the fast solution of the resulting linear system. The BEM is based on a boundary integral formulation which requires the discretization of the only domain boundaries. Thus, this method is well suited to treat seismic wave propagation problems. A major drawback of classical BEM is that it results in dense matrices, which leads to high memory requirement ($\mathcal{O}(N^2)$, if N is the number of degrees of freedom) and computational costs. Therefore, the simulation of realistic problems is limited by the number of degrees of freedom. Several fast BEMs have been developed to improve the computational efficiency. We propose a fast $\mathcal{H}$-matrix based direct BEM solver.

Titre : Solveurs fondés sur la méthode des $\mathcal{H}$-matrices pour les équations intégrales en élastodynamique 3D
Mots clés : Éléments Finis de Frontière, $\mathcal{H}$-matrices, Approximation Adaptive Croisée, Décomposition en Valeurs Singulières Aléatoire, Élastodynamique 3D, Problèmes de Vibrations Forcées.

Résumé : Cette thèse porte sur l'étude théorique et numérique des méthodes rapides pour résoudre les équations de l'élastodynamique 3D en domaine fréquentiel. La méthode repose sur l'utilisation des éléments finis de frontière (BEM) pour la discrétisation et sur les techniques de matrices hiérarchiques ($\mathcal{H}$-matrices) pour l'accélération de la résolution du système linéaire. La BEM, qui correspond à la résolution numérique des équations intégrales de frontière, présente l'avantage de ne nécessiter que la discrétisation de la frontière du domaine de calcul considéré. De plus, elle permet de simuler des milieux étendus en évitant la forte dispersion numérique associée à d'autres schémas. Cette méthode est donc bien adaptée pour le calcul de la propagation d'ondes sismiques. Les difficultés sont liées d'une part aux besoins de stockage en mémoire de l'ordre de N^2 (N étant le nombre de degrés de liberté) et d'autre part au coût de la résolution. La simulation de problèmes réalistes est donc limitée par le nombre de degrés de liberté que peut traiter sur une machine donnée. Afin de dépasser ces limites, des méthodes BEMs rapides ont été développées. Nous avons proposé un solveur direct pour le BEMs en utilisant une factorisation LU et un stockage hiérarchique.