Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel du 25 mai 2016

Préparée au sein de **du laboratoire Jean Kuntzmann et de l'Institut des Sciences de la Terre** et de l'école doctorale **l'Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Efficient preconditioning method for the CARP-CG iterative solver for the solution of the frequency-domain visco-elastic wave equation

Présentée par

Okba HAMITOU

Thèse dirigée par **Ludovic Métivier, Stéphane Labbé et Jean Virieux**

Thèse soutenue publiquement le 22 décembre 2016 ,
devant le jury composé de :

M, Luc Giraud	DR	Inria Bordaux	Président
M, Jan Hesthaven	Prof.	UPFL	Rapporteur
M, Stéphane Labbé	Prof.	Université Grenoble Alpes	Directeur de thèse
M, Tristan van Leeuwen	Assistant prof.	Université d'Utrecht	Examinateur
M, Ludovic Métivier	CR CNRS	Université Grenoble Alpes	Directeur de thèse
M, Jean Virieux	Prof.	Université Grenoble Alpes	Directeur de thèse

To my grandmother and my family.

Acknowledgments

I would like to express my sincere appreciation and acknowledgments to my supervisors Ludovic Métivier, Stéphane Labbé and Jean Virieux for giving me the opportunity to work on this thesis project. Thank you for encouraging my work and allowing me to grow as a research scientist. Thank you for your advice, your valuable time and your useful suggestions and critiques. These past three years have been full of challenges and stimulating discussions. Thank you for caring.

I would like to thank Romain Brossier for his insightful ideas which helped unlock many barriers.

My appreciation goes as well to Yang Li for providing the CARP-CG solver and also for his help and expertise using this solver. I have really appreciated your kindness and your patience at the beginning of my thesis and your quick responses to my emails near the end. I thank as well my colleagues within the SEISCOPE team Stéphane Beller, Clara Castellanos, François Lavoué, Borhan Tavakoli, Benjamin Pajot, Laure Combe, Alain Miniussi, Stéphane Garambois, Michel Dietrich and Stéphane Operto. Thank you for all the video-conferences we have shared and the fruitful discussions.

I would also like to express my sincere gratitude to the reviewers professor Luc Giraud and professor Jan Hesthaven and Dr. Tristan van Leeuwen.

I thank the SEISCOPE consortium and the sponsors for the financial support for this thesis, for providing excellent working conditions and the funding of numerous conferences and workshops. I would like to acknowledge the support the HPC resources of CIMENT and IDRIS.

I am very thankful as well to the administrative staff members, Stéphanie Mandjee, Magali Gardes and specially Jacques Pellet at ISTerre and Hélène Baum at the LJK for their help and for making administrative procedures so easy. It was always a pleasure to come see you and I hope you are enjoying retirement.

I thank my colleagues Hugo Pinard, Maureen Shinta Devi, Paul Wellington, Camille Crouzet, Wei Zhou, Isabella Masoni, Blandine Gardonio, Caroline Bissardon, Yacine Benjelloun, Eleanor Baker. Thank you for sharing so many interesting conversations, coffee breaks, soccer and ultimate games and more importantly thank you for making these past three years pleasant. I have enjoyed every moment of it thanks to you.

Finally, I thank my friends Mustafa, Kévin, Jessica et Amine for being there. Your presence was extremely valuable. I thank my family for their endless support and encouragements despite the distance. My thoughts go to my grandmother. This thesis is dedicated to them.

Résumé

La résolution de l'équation des ondes acoustiques et élastiques en 3D dans le domaine fréquentiel représente un enjeu majeur dans le cadre de l'inversion des formes d'ondes pour l'imagerie haute résolution de cibles crustales (Virieux and Operto, 2009). Après discrétisation, ce problème revient à résoudre un système linéaire à valeurs complexes, creux, de grande taille (plus de 10^9 degrés de liberté pour des cas d'application réalistes), non défini et mal conditionné. Les méthodes d'inversion sismique requièrent la solution de ce problème pour l'évaluation du problème direct pour un grand nombre de sources (plusieurs milliers voire dizaines de milliers). Dans l'approximation acoustique, les méthodes directes sont privilégiées. Cependant, le coût mémoire de ces méthodes les rendent aujourd'hui inutilisables pour résoudre les problèmes élastiques 3D. En raison de leur plus faible coût mémoire, les méthodes itératives pour les équations en fréquence peuvent être considérées pour l'élastodynamique. Cependant, une convergence rapide passe par des préconditionneurs adaptés pour les solveurs itératifs. Par ailleurs, les stratégies pour résoudre des systèmes linéaires avec des seconds membres multiples ne sont pas aussi efficaces que pour les méthodes directes. La modélisation dans le domaine temporelle quant à elle présente une importante complexité en coût de calcul et cette complexité croît linéairement avec le nombre de sources.

Dans cette thèse, l'approche utilisant un solveur itératif est considérée. Le solveur itératif CARP-CG introduit par Gordon and Gordon (2010a) est considéré. Cette méthode est basée sur la méthode de Kaczmarz qui transforme un système linéaire mal conditionné en un système hermitien, positif et qui peut être résolu en utilisant les méthodes du type gradient conjugué (CG). Dans des configurations de forts contrastes et hétérogénéités, ce solveur s'est révèlé être extrêmement robuste alors que les méthodes itératives standards basées sur les sous-espaces de Krylov telles que GMRES et BiCGSTAB nécessitent l'utilisation d'un préconditionneur pour converger (Li et al., 2015). Malgré les bonnes propriétés de la méthode CARP-CG, le nombre d'itérations nécessaires pour atteindre une précision suffisante reste néanmoins élevé. Je présente alors une stratégie de préconditionnement adaptée au problème de propagation des ondes et à la méthode CARP-CG. Ce préconditionneur est un inverse creux et approché d'un opérateur de propagation des ondes fortement amorti. Le calcul du préconditionneur est réalisé grâce un algorithme massivement parallèle pour les architectures à mémoire distribuée.

La méthode développée est appliquée à des cas d'étude réalistes. Les applications sont faites sur des modèles synthétiques 2D dans l'approximation visco-acoustique pour des fréquences allant jusqu'à 40 Hz puis dans l'approximation élastique pour des fréquences allant jusqu'à 20 Hz. Ces études montrent l'efficacité de la méthode CARP-CG munie de la stratégie de préconditionnement. Le nombre d'itérations est fortement réduit (jusqu'à un facteur 9) permettant d'améliorer considérablement la complexité de la méthode CARP-CG. Des gains en temps de calcul allant jusqu'à un facteur 3.5 sont ainsi obtenus. La méthode est ensuite appliquée à un cas 3D synthétique et réaliste dans l'approximation visco-élastique pour des fréquences allant de 1.25 Hz à 7.5 Hz. Des résultats encourageants sont obtenus. Munie du préconditioneur, la méthode CARP-CG permet de résoudre ces systèmes linéaires

deux fois plus rapidement.

La stratégie de préconditionnement implique la nécessité de plus grandes ressources en mémoire pour le solveur itératif; cependant, elles ne constituent pas une limitation pour la méthode et restent très négligeables devant celles requises par les solveurs directs. La principale limitation réside dans le temps de calcul qui demeure assez significatif. Cependant, cette méthode constitue un solveur compétitif comparé aux autres solveurs en temps et direct utilisés aujourd'hui dans le cadre de l'inversion des formes d'ondes.

Abstract

A robust and efficient wave modeling method is the cornerstone of high resolution seismic inversion methods such as the frequency-domain Full Waveform Inversion (Virieux and Operto, 2009). After discretization, frequency-domain wave modeling amounts to the solution of large (up to several billion of unknowns for realistic case studies), sparse, indefinite and ill-conditioned linear systems. Furthermore, seismic inversion methods require the solution of this problem for numerous sources (from several thousands up to tens of thousands). In the acoustic approximation, 3D real case studies can be handled efficiently using direct solvers. However because of their tremendous intrinsic memory requirements, they are not yet adapted to the solution of the 3D elastodynamics equations. Iterative solvers provide an alternative to direct solvers. However, they require a preconditioning strategy to ensure convergence for the frequency-domain wave equation. Besides, multiple right-hand sides linear systems are not treated as efficiently as direct solvers do.

In this thesis, we are interested in the use of a robust iterative solver adapted to the solution of these systems called CARP-CG (Gordon and Gordon, 2010a). The CARP-CG method has shown robust convergence properties for 2D and 3D elastic problems in highly heterogeneous media compared to standard Krylov methods such as GMRES or Bi-CGSTAB which require the use of a preconditioner to ensure convergence (Li et al., 2015). Despite the good convergence properties of CARP-CG, the latter still requires a large number of iterations to reach sufficient accuracy. I introduce an efficient preconditioning strategy adapted to the CARP-CG method and the frequency-domain wave problem. This preconditioner is computed as a sparse approximate inverse of a strongly damped wave propagation operator. The computation of the preconditioner is performed in a massively parallel algorithm for distributed memory architectures.

The efficiency of the preconditioner is evaluated on several case studies. First, applications are performed on realistic synthetic models in the 2D visco-acoustic approximation (up to 40 Hz) and the 2D visco-elastic approximation (up to 20 Hz). These studies show that the CARP-CG method together with the preconditioning strategy is robust and efficient. The number of iterations is significantly reduced (up to a factor 9) enabling a speedup in the computation time by a factor up to 3.5. Second, this method is investigated in the 3D elastic approximation on a realistic synthetic case study on the range of frequencies 1.25 to 7.5 Hz. Very encouraging results are obtained with a significant reduction in the number of iterations. A slow increase of the number of iterations with respect to the frequency is noted.

This preconditioning strategy adapted to the CARP-CG method implies larger memory requirements. However, this extra memory cost remains one order lower compared to direct solver memory requirement, and should be affordable on standard HPC facilities. The main bottleneck preventing from the possible use of this iterative solver for 3D elastic FWI remains the computation time for the wave equation solves.

Contents

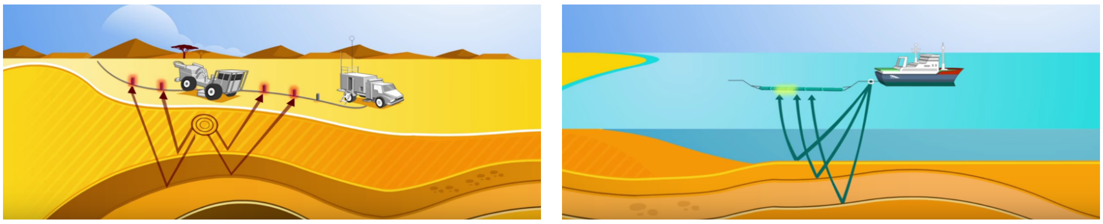
Introduction

Systems of linear equations are ubiquitous in science and engineering. Being able to solve large scale systems in a short amount of time is a mathematical and computational challenge. The subject of this work is the design of a preconditioning strategy for the frequency-domain wave equations adapted to the iterative solver CARP-CG. The reasons for this particular choice are discussed in this introduction. First, the research field which has motivated this work, namely the field of seismic imaging techniques, is described. Then, we provide arguments for considering iterative methods and we shall identify where one needs to make progress for considering these methods competitive with respect to other methods such as direct solvers or the time dependent techniques. The design, construction, analysis and testing of different strategies of preconditioning the frequency-domain wave equations are the main contributions of this work.

Imaging Earth's structure at the crustal scale

Seismic imaging is one of many remote sensing methods available to geophysicists to study and build quantitative images of the subsurface parameters. The inversion of seismic waves allows to recover certain properties of the geological structures. P-wave velocity, density, attenuation and anisotropy parameters can be reconstructed in the acoustic approximation. One may consider additional quantities as S-wave velocity and S-wave attenuation in the framework of the elastic inversion. At the crustal scale, seismic imaging methods help the understanding of many scientific questionings on the internal mechanisms of the Earth such as landslide, volcanic activity and seismology which require sharper and sharper imaging techniques. At the exploration scale, these methods are a crucial stake for the natural resource search and in particular for the oil and gas industries for purposes of exploration or exploitation of the hydrocarbon reservoirs. Besides the seek of natural resource aspects, as CO_2 and radioactive waste are confined underground, seismic inversion methods are used as well for locating, monitoring and the surveillance of these disposal sites.

In the context of seismic exploration, geophysicists use these seismic imaging tools to perform a veritable echography of the subsurface. Their investigations cover several tens of kilometers at the surface and rarely exceed ten kilometers at depth. They characterize the P-wave velocity, density, attenuation or anisotropy quantities underground. 3D surveys are set up by laying out source points and receiver points in a grid over the area to be surveyed. In an onshore seismic survey (see Figure 1a), a shock on the surface of the ground emits seismic waves. The shocks can be produced either by small explosive charges set at the surface of the ground or by large vehicles equipped with heavy plates called vibrating trucks which vibrate at a specific power and frequency. The seismic wave travels into the earth, is reflected by the subsurface formations and returns to the surface where it is recorded by receivers similar to microphones called geophones. In marine exploration, the shock is produced by



(a) Onshore (b) Offshore

Figure 1: Schematic of an onshore and offshore setup for a seismic survey (Total).

an air gun and hydrophones record the data which reflects to the surface (see Figure 1b). There are two classes of seismic sensors which are used to record the data. In onshore surveys, vector sensors are usually used. They record the particle displacement. A classic example of a vector sensor is the moving-coil geophone. They are directional sensors and thus can distinguish between horizontal and vertical motions. In offshore surveys, scalar sensors are used to measure only the magnitude of the disturbance and do not indicate the direction of the motion. The hydrophone is the classic example of scalar sensor used in the seismic industry to measure pressure variations.

Over the past decades, 3D seismic surveys have become increasingly denser. The area covered by these surveys exceeds easily several hundred square kilometers. The spacing between the source and receiver points is determined by the design and the aims of the survey. Often, source and receiver are 20 to 100 meters apart along the source line direction and 50 to 100 meters apart laterally to create parallel source lines. Such dense acquisitions provide finally several ten thousands of source and receivers points.

Seismic imaging techniques – Full Waveform Inversion FWI

Full waveform inversion is recognized to be a promising technique for the estimation of the surface parameters (Virieux and Operto, 2009). This method relies on the entire waveform information, i.e. phase and amplitude of both transmitted and reflected energy, to provide high resolution imaging results. FWI was first formulated as a least square optimization problem (Tarantola, 1984) which aims at minimizing the difference between the simulated data and the observed data $\mathbf{d}_{obs}$ in the sens of the L_2 norm. Consider the forward problem equation

$$F(\mathbf{m})\mathbf{u} = \phi, \tag{1}$$

where $\mathbf{m}$ denotes the model parameters, $F(\mathbf{m})$ denotes the linear forward problem operator which corresponds to wave equation discretization, ϕ is the source vector and $\mathbf{u}$ is the wavefield vector. The model parameter $\mathbf{m}$ could embrace any set of the medium parameters influencing the propagation of seismic waves in the subsurface. These notations are general and can be either in the time domain or in the frequency domain. The FWI problem can be expressed as

$$\min_{\mathbf{m}} \mathcal{C}(\mathbf{m}) = \frac{1}{2} \sum_{s=1}^{N_s} ||P_s \mathbf{u}_s(\mathbf{m}) - \mathbf{d}_s||_2^2, \tag{2}$$

where $\mathbf{u}_s(\mathbf{m})$ and $\mathbf{d}_s$ are respectively the solution of the forward problem (1) with respect to the source ϕ_s and the recorded data, N_s is the number of sources and P_s is a restriction operator which maps the wavefield $\mathbf{u}_s(\mathbf{m})$ to the receivers locations.

Although the cost function was originally formulated using the L_2 norm, the latter is sensitive to ambient noise naturally present in the subsurface giving rise to a lack of robustness of the L_2 norm. Alternative norms are have been investigated since then to remedy this problem. The L_1 norm is introduced by Tarantola (1987) showing a robustness towards the presence of noise in the data as well as hybrid approaches combining the L_1 and L_2 norm (Bube and Langan, 1997). Recently, the Wasserstein metric, related to the optimal transport, is introduced to measure the distance between the observed and simulated data (Métivier et al., 2016; **?**). This breakthrough helps mitigating another well known issue of FWI in the seismic imaging community which is the cycle skipping problem.

The expression of the gradient of the misfit function $\mathcal{C}$ is given by

$$\nabla\mathcal{C}(\mathbf{m}) = \mathfrak{Re}\left(\sum_{s=1}^{N_s} J_s^H P_s^H (P_s \mathbf{u}_s(\mathbf{m}) - \mathbf{d}_s)\right),$$

where J_s denotes the Jacobian matrix $\partial_{\mathbf{m}}\mathbf{u}_s(\mathbf{m})$, the symbol ^{H} denotes the complex conjugate transpose operator and $\mathfrak{Re}$ is the real part of a complex number. The Hessian operator is

$$H(\mathbf{m}) = \mathfrak{Re}\left(\sum_{s=1}^{N_s}\left(J_s^H P_s^H P_s J_s + \sum_{j=1}^{m}\left[P_s^H (P_s \mathbf{u}_s(\mathbf{m}) - \mathbf{d}_s)\right]_j H_{sj}\right)\right),$$

with

$$H_{sj} = \partial^2_{\mathbf{mm}}(\mathbf{u}_s)_j(\mathbf{m}),$$

where $m \in \mathbb{N}$ is the number of components of the model parameter $\mathbf{m}$. The Gaussian-Newton approximation of the Hessian is

$$\tilde{H}(\mathbf{m}) = \mathfrak{Re}\left(\sum_{s=1}^{N_s} J_s^H P_s^H P_s J_s\right)$$

Thanks to the gradient and the Hessian, the minimization problem (2) is solved using the update formula

$$\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} + \delta\mathbf{m}^{(k)},$$

where $\delta\mathbf{m}^{(k)}$ is the solution of

$$H^{(k)}\delta\mathbf{m}^{(k)} = -\nabla\mathcal{C}(\mathbf{m}^{(k)}),$$

or, in the Gauss-Newton approximation

$$\tilde{H}^{(k)}\delta\mathbf{m}^{(k)} = -\nabla\mathcal{C}(\mathbf{m}^{(k)}),$$

The explicit computation and storage of the the Jacobian and Hessian matrices is prohibitive for large scale problems in FWI applications. Diagonal approximations of the Hessian are more conventionally used (Pratt et al., 1998; Shin et al., 2001). However, the computation of the gradient can be performed efficiently and relies on using the classical adjoint-state method (Plessix, 2006a) where the ith component of the vector $\nabla\mathcal{C}(\mathbf{m})$ is computed through

$$(\nabla\mathcal{C}(\mathbf{m}))_i = \sum_{s=1}^{N_s} \mathfrak{Re}\left(\partial_{\mathbf{m}_i} F(\mathbf{m})\mathbf{u}_s(\mathbf{m}), \boldsymbol{\lambda}_{\boldsymbol{s}}(\mathbf{m})\right), \tag{3}$$

where $\boldsymbol{\lambda}_s(\mathbf{m})$, defined as the adjoint-state or the adjoint wavefield, is the solution of

$$F(\mathbf{m})^H \boldsymbol{\lambda}_s(\mathbf{m}) = P_s^H\left(\mathbf{d}_s - P_s\mathbf{u}_s\left(\mathbf{m}\right)\right), \tag{4}$$

which is referred to as the adjoint problem. In essence, the gradient is proportional to the correlation of the direct wavefield which is solution of the forward problem (1) and the adjoint wavefield which is solution of the adjoint problem (4). Therefore, a fundamental element of FWI is a full waveform modeling method which enables robust and fast computations of the direct and adjoint wavefields.

Time-domain versus frequency-domain FWI

As noted, FWI was first formulated in the time domain (Tarantola, 1984). In the time domain, the data are represented by temporal seismograms. The first applications of time-domain FWI suffered from the huge computational costs of this method which limits the possible number of iterations of the optimization problem. An implicit summation over all the time steps is performed in the computation of the gradient (3) which requires the storage in memory of the direct and adjoint wavefields on all the computation domain for all time steps and for all sources at a time. The computation of the gradient becomes quickly impossible due to the large number of time steps and the memory request it involves. In practice, the gradient is updated at each time step t_i. Once the adjoint wavefield is computed, the direct wavefield is recomputed from scratch. The contribution of the correlation for the time step t_i is added to the gradient resulting from the previous time steps. The re-computation of the direct wavefield is accelerated by storing samples of the direct wavefield at specific time steps. The computation of the direct wavefield is then performed starting from the closest previous stored value which is used as an initial condition. This method is referred to as checkpointing (Griewank, 1992; Symes, 2007). Another approach consists in the storage of the direct wavefields at all time steps but only on the boundaries of the computation domain. From these values, the direct wavefield is computed on all the domain by time-reversal (Clapp, 2008). However, the main limitation of such approach is the reverse propagation in media with attenuation which is a difficult framework for time-domain wave propagation as attenuation is naturally present in the subsurface.

In the early 1990s, Pratt and Worthington (1990) suggested a frequency-domain formulation of FWI. The temporal seismic data is transformed in the frequency domain by Fourier Transform yielding complex valued data. The frequency-domain FWI approach is equivalent to the time-domain approach when all the frequencies are inverted simultaneously. Sirgue and Pratt (2004) showed that frequency-domain FWI can be performed using a limited number of frequencies which is one of the most advantageous feature of this approach from a computational point of view. It is therefore feasible to store the direct and adjoint wavefields on all the computation domain for a limited number of frequencies. The implicit summation over all the time steps which is performed in the computation of the gradient (3) is therefore replaced by an implicit summation over the number of frequencies. The frequency-domain FWI was conferred a privileged position for seismic imaging ever since.

Frequency-domain versus time-domain forward problem

Frequency-domain FWI requires the computation of frequency-domain direct and adjoint wavefields in arbitrarily heterogeneous media and for a large number of sources. There are numerous methods for modeling such problems such as finite-difference methods (Virieux, 1986; Moczo, 1989), finite-element methods (Marfurt, 1984; Brossier et al., 2010) and finite-volume methods (Dormy and Tarantola, 1995; Brossier et al., 2008). Among these methods, the finite-difference method provides a good compromise between computational cost, accuracy, simplicity of implementation and the ability to describe heterogeneous media. Over the past decades, substantial efforts have been concentrated in including more realistic descriptions of the media in FWI such as density, anisotropy and attenuation. Most of the FWI applications are mainly based on the acoustic approximation of the wave propagation problem (Dessa et al., 2004; Operto et al., 2006; Plessix et al., 2012). Such approximation is advantageous due to the low numerical cost of solving the wave propagation problem. Furthermore, thanks to the enhancement of modern computing clusters, recent developments are evolving from 2D to 3D geometries which requires a very efficient forward problem solver and efficient strategies for handling large memory requests. Thus, the central criterion for a feasible and efficient frequency-domain FWI is a robust and efficient forward problem solver that is able to manage efficiently multiple sources. In the following discussion, the dimension of a 3D N^3 computational grid is denoted by N.

The numerical solutions of the wave equations can be obtained by considering the latter in the time domain. An explicit time-marching algorithm is commonly used. At each time step, the solution over the computation domain is estimated from the solutions at the previous time steps. The computation of the frequency-domain wavefields is obtained by a discrete Fourier (DFT) summation over time (Sirgue et al., 2008). The time-domain algorithm can be efficiently parallelized using domain decomposition methods. The time and memory complexities for the time-domain solver for 3D finite-difference problems are $O(N^4)$ and $O(N^3)$ respectively for one source modeling. The main limitation of this approach is the lack of strategies for accelerating multiple sources modeling and the computational cost of the DFT which is as expensive as the time-marching algorithm. Therefore, the time complexity scales linearly with the number of sources. Warner et al. (2013) performed 3D acoustic FWI using a time-domain solver combined with DFT on the Tommeliten Alpha field in the North Sea. The 3D survey covers a surface of 208 km^2 and is 4 km deep with 1,440 sources. This approach enables to extract frequencies up to 6.5 Hz which corresponds to a 321x261x81 discretization of the 3D domain. A sub-sampling of the data is carried out to drastically reduce the computational cost of the method. Only 80 sources among the 1,440 are used at each iteration of FWI.

Alternatively, the wave equations can be formulated straightforwardly in the frequency-domain yielding a sparse linear system with multiple right-hand sides to be solved. This linear system can be solved using a direct solver based on a LU decomposition. Parallelism is implemented through multifrontal and nested dissection techniques such as the MUMPS solver (George and Liu, 1981; Duff et al., 1986; Amestoy et al., 2000b). Once the parallel lower-upper factorization is performed, the solutions can be computed efficiently by backward and forward substitutions. With such approach, the time and memory complexities are $O(N^6)$ and $O(N^4)$ respectively which points out the main limitation of the direct solver approach that is the memory issue. This direct solver is used in the framework of the 3D acoustic frequency-domain FWI in anisotropic media by Operto et al. (2015) on the Valhall case study. The model covers a surface of 145 km^2 and is 4 km deep. The frequencies modeled reach 10 Hz yielding a linear system of size 17 million unknowns with more than 4,000 right hand sides.

An alternative approach for the solution of the frequency-domain wave equations relies on preconditioned iterative methods (Saad, 2003). Krylov subspace methods are generally used such as the generalized minimal residual method (Saad, 1986) or the biconjugate gradient stabilized method (Van der Vorst, 1992). These iterative methods start from an initial guess and compute successive approximations of the solution through a projection process. The central ingredient of any iterative method is an efficient preconditioning method. A preconditioner is an operator which transforms the original linear system into another with better spectral properties, i.e. location and clustering of the eigenvalues. The design of an efficient preconditioner for the frequency-domain wave equations is an active field of research. The impedance matrix is indefinite with positive and negative eigenvalues along the real axis and ill-conditioned. Made et al. (2000) have shown on the Helmholtz equation that by introducing a complex perturbation in the diagonal of the impedance matrix, the spectral properties of the preconditioned linear system are well improved by shifting and clustering the eigenvalues in the positive part of the real axis. Later, Erlangga et al. (2006a), Riyanti et al. (2006) and Plessix (2007) introduced the complex shifted Laplacian preconditioner where an imaginary term is added to the Laplacian of the Helmholtz equation which is equivalent to a damped version of the Helmholtz equation. The approximate solution of the damped wave equation is obtained by one cycle of multigrid. With such a preconditioner, Plessix (2007) have shown that the number of iterations increases linearly with respect to the frequency. The complex shifted Laplacian preconditioner is further improved by Erlangga and Nabben (2008b) who introduced a multigrid multi-level Krylov method based on a deflated preconditioner. This approach led to a number of iterations that is almost independent of the frequency. In this ideal case, the computational complexity of the iterative solver would be $O(N^3)$ while previous iterative methods have $O(N^4)$. Their memory complexity is $O(N^3)$. Plessix (2009) present a 3D frequency-domain acoustic FWI with the multigrid preconditioned iterative solver. Applications on the synthetic SEG/EAGE overthrust and salt case studies and a real data example show the ability to use the preconditioned iterative solver as a forward modeling tool in FWI at least up to 10 Hz with more than 1,300 sources on standard modern computers. At 7 Hz, the computational grid contains 307x103x103 points and the 15 iterations of the inversion at 7Hz are performed in 36 hours using 1369 cores (2.6 GHz AMD processors).

Unlike time-domain approaches, the use of block iterative solvers and projection-based methods should enable efficient solving of multiple right-hand side linear systems (O'Leary, 1980; Smith et al., 1989; Chan and Wan, 1997). Such techniques should theoretically provide a better numerical scheme than the time-domain solver.

Toward 3D frequency-domain elastic FWI: Challenges and bottlenecks

In the industry, challenging 3D problems involving large grid size of 1001^3 or larger are investigated (**?**). The high computational cost and memory requirements of the seismic wave modeling limits the feasibility of frequency-domain FWI. In the acoustic approximation, FWI can performed in the frequency-domain efficiently using a sparse parallel direct solver for numerous sources (Operto et al., 2015). However, in the elastic approximation, the size of the linear systems to be solved increase dramatically which prohibits the use of sparse direct solvers on the actual clusters. Although recent attempts have addressed 3D frequency-domain elastic wave modeling with direct solvers (Wang et al., 2012; Gosselin-Cliche and Giroux, 2014), challenging large scale problems involving large grid size seem to be still out of reach for direct solvers. Therefore, other approaches suggest the use of iterative solvers or time-domain solvers as they require far less memory than direct solvers.

Time-domain approaches have matured enough and are extensively used in seismic imaging techniques. However, no multiple right-hand side accelerations can be implemented and furthermore, the implementation of attenuation effects is more difficult and computationally expensive. Frequency-domain iterative solvers present similar theoretical complexities as time-domain schemes and allow to account for attenuation effects efficiently by means of complex velocity (Kolsky, 1956; Futterman, 1962). Until recently, iterative solvers have been questioned on their robustness in the case of very heterogeneous media as well as the feasibility of elastic wave modeling (Virieux et al., 2009). The development of optimal iterative solvers is still an active field of research (Plessix, 2007; Erlangga and Nabben, 2008a; Engquist et al., 2011; Poulson et al., 2013). Although they are successfully used in the framework of 2D and 3D frequency-domain acoustic FWI, the evolution towards 3D elastic is still an open question. A 3D elastic frequency-domain FWI application is carried out by Pan et al. (2015) using BiCGStab with the complex shifted Laplacian preconditioner. However, the size of the problem (hundred of meters per dimension) which is investigated is way far from realistic problem size or even medium-sized typical FWI applications.

Recently, the iterative solver CARP-CG introduced by Gordon and Gordon (2010a) has attracted a particular interest for the solution of the frequency-domain wave equation. It is used in the 3D frequency-domain acoustic waveform inversion by van Leeuwen and Herrmann (2014) together with a Block iterative method to solve multiple right-hand sides linear systems. Furthermore, its robustness is demonstrated as well by Li et al. (2015) on large and complex models in the 2D and 3D elastic approximations. The convergence is obtained with a number of iteration in $O(N^{\alpha})$ with $\alpha < 1$ in arbitrarily heterogeneous media.

Objectives of the thesis

The work carried out in this thesis is to further investigate the efficiency of this particular iterative solver CARP-CG thanks to a preconditioning method. The objective of this thesis is to develop a numerical method of the solution of the 3D frequency-domain viscoelastic wave equations.

Outline

To begin with, Chapter 1 introduces the methods for the simulation of the wave propagation in the subsurface. In the first part, the equations governing the propagations of waves in the subsurface are introduced. These equations are formulated in the time domain from which the frequency-domain equations are derived. Both acoustic and elastic approximations are considered. Simulating properly the propagations of seismic waves at the regional scale requires to use adequate boundary conditions. The medium is considered infinite in the horizontal direction and the vertical direction at depth. Therefore, absorbing and free-surface boundary conditions are introduced. The Perfectly Matched Layers of Bérenger (1994) are used for the absorbing boundary conditions. The method of Mittet (2002) is used for the free-surface. Finally, this work focuses on the finite-difference method for the discretization of the wave equations. The staggered-grid method for the wave equations (Madariaga, 1976; Virieux, 1984) is introduced together with the parsimonious approach (Luo and Schuster, 1990). In the time domain, the system of algebraic equations written in the velocity-stress formulation is obtained in the elastic approximation. In the acoustic approximation, the system of equations reduces to one equation involving the pressure wavefield. In the second part, the three different methods for the computation of the seismic waves in the frequency-domain are discussed: (1) the time-domain approach combined with a Discrete Fourier Transform, (2) the solution of the linear system resulting from the discretization of the frequency-domain wave equations using a direct solver or (3) an iterative solver. These methods are evaluated in the framework of seismic imaging techniques, i.e. for the solution of the forward problem with numerous sources or right-hand sides. An overview of the standard iterative solvers is presented as well as the preconditioning methods.

Chapter 2 describes the iterative solver CARP-CG in the first part. In the second part, I seek to improve the convergence properties of this iterative method. This is the main contribution of this work. The spectral properties of the preconditioned equations are evaluated in the 2D visco-acoustic and visco-elastic approximations. A method for the computation of a sparse approximation of the preconditioner is then suggested.

Chapter 3 is dedicated to the evaluation of the preconditioned iterative method on 2D realistic case studies. In the first part, the method is used to solve the 2D frequency-domain visco-acoustic wave equation on the BP2004 model (Billette and Brandsberg-Dahl, 2004). In the second part, it is applied in the 2D visco-elastic approximation on the Marmousi 2 model (Martin et al., 2006). The efficiency of the preconditioner as well as the scaling properties of the CARP-CG method are investigated.

Last, Chapter 4 is dedicated to a 3D realistic application in the visco-elastic approximation of the preconditioned iterative solver. The efficiency of the preconditioner is evaluated. The memory issue is addressed as well. A theoretical comparison with the time-domain solver on the computational and memory complexities is carried out.

This thesis ends with conclusions and a list of improvements which can be carried out for future developments.

All the numerical tests are performed on the Grenoble Alpes University High Performance Computing Center (CIMENT). The cluster is composed of nodes having 2 Intel Sandy Bridge processors and connected to a FDR Infiniband non-blocking low latency network. Each node has two CPUs Intel Xeon E5-2670 2.6 GHz octacore, that is, 16 computing cores per node. Nodes with 64 GB of RAM as well as nodes with 128 GB of RAM are available.

Chapter 1

Modeling the seismic wave propagation in the subsurface

Contents

The context in which this work has been carried out is related to high resolution seismic imaging methods such as Full Waveform Inversion (FWI) or Reverse Time Migration (RTM). These methods require to perform numerous seismic wave propagation simulations during the optimization procedure to build quantitative images of model parameters. Designing efficient tools to perform these simulations is thus crucial.

The Earth is a solid medium and, as a consequence, both compressive and shear waves are propagating in the subsurface. This is the elastic wave propagation framework, which is a challenging problem from a numerical simulation point of view. Often, we may consider simpler and, therefore, faster simulation strategies (in the sens of the computation time) by considering the Earth as a fluid: only compressive waves are propagating. This is the acoustic wave propagation framework. During my research investigations, I consider both acoustic and elastic wave propagations. I shall consider the FWI in the frequency domain. Modeling the wavefield in the frequency domain may be performed in three ways. The first strategy consists in solving the wave problem in the time domain and extract the frequencies needed thanks to the Discrete Fourier Transform (DFT). The two other strategies solve the wave problem in the frequency domain using either a direct or an iterative linear solver. We are mainly interested in the latter approach. In this chapter, I introduce the key elements allowing to describe this problematic.

First, a description of the system of equations governing the propagation of waves in the subsurface is introduced. These equations are presented in the time domain for completeness although I shall not consider them in my investigation. I shall deduce related expressions in the frequency domain where I shall consider both elastic and acoustic approximations. They need to be solved efficiently in any arbitrarily heterogeneous medium in the framework of the imaging method we are interested in. Absorbing boundary conditions and free surface boundary conditions are introduced to simulate properly the seismic wave propagation in the Earth. The discretization of the acoustic wave equation and of the system of elastic wave equations is performed using the staggered grid finite-difference method, leading to a discrete linear system to be solved. This linear system could be solved either by a direct or an iterative solver.

Direct solvers can provide efficient techniques for the 3D acoustic equation at the exploration scale. In the framework of FWI, Operto et al. (2015) use a parallel sparse direct solver to compute the solution of the 3D frequency-domain visco-acoustic wave equation taking into account vertical anisotropy. Simulations are performed on the Valhall real case study. The model covers a large surface of 145 km^2 and is 4 km deep. The frequencies modeled reach 10 Hz, yielding a linear system of size 17 million unknowns with more than 4000 right hand sides which requires 1.6 TB of memory for the storage of the LU factors. The massively parallel sparse direct solver of MUMPS (Amestoy et al., 2001, 2000a) is used in his study. This solver relies on the multifrontal method (Duff et al., 1986) allowing to distribute the factorization on parallel architectures. This solver is combined with a block-low rank approximation of the dense frontal matrices to reduce the memory demand and the computational costs (Weisbecker et al., 2013; Amestoy et al., 2015b) as well as a nested-dissection algorithm (George and Liu, 1981) based on a reordering strategy to reduce the filling during the factorization.

A recent attempt was performed by Wang et al. (2012) for the 3D elastic wave problem in the frequency domain using a direct solver. They use a hierarchically semi-separable (HSS) low rank matrix compression for the reduction of the computation and memory costs. The 3D frequency-domain elastic wave equations are solved on the SEAM3D model covering a 19 km^2 surface with a 3.3 km depth. This model is discretized with a compact finite-difference stencil according the shortest wavelength yielding a $201 \times 201 \times 151$ mesh at the frequency 10 Hz and a linear system of size 54.9 million unknowns which requires 2.2 TB of memory for the storage of the LU factors.

In addition, specific stencils are designed for 3D wave modeling in the acoustic and elastic approximation when direct solvers are used. Operto et al. (2014) use a 3D fourth-order mixed-grid finite-difference stencil of compact support for the 3D visco-acoustic wave modeling. A similar strategy is adopted by Gosselin-Cliche and Giroux (2014) for the 3D visco-elastic wave modeling. They use a second-order finite-difference staggered grid stencil of compact support with optimal coefficients. Such strategies allow to reduce the bandwidth of the impedance matrix and thus minimize the computational cost of the lower-upper decomposition of the impedance matrix in terms of memory demand and floating-point operations.

Despite these efforts, direct solvers are not yet adequate for the solution of the 3D visco-elastic wave equations in realistic size models. The size of the achieved simulations is still modest and does not correspond to the size of the problems which could be treated in the oil and gas industry, where the number of discretization points could exceed several thousands in each dimension.

As iterative method fully benefit from the sparsity of the linear system, they appear as an interesting tool for such large scale simulations. I shall discuss why they are still facing difficulties related to their convergence. Designing preconditioning strategies will be at the heart of this work.

1.1 Seismic wave equation

Simulating the propagation of seismic waves in the subsurface requires to account for the full complexity of the Earth model parameters. Therefore, the solution of the full 3D equations must be considered rather than their 2D approximations. Moreover, the upper crust of the Earth must be considered as an elastic anisotropic medium with strong heterogeneities. Therefore, the seismic waves are composed of compressional waves and shear waves which travel at different velocities. Furthermore, the surface of the medium presents often a complex topography (in opposition to a flat topography) and, because of the free surface condition, another type of waves appears which are located mainly along the free surface, namely surface waves. Although many textbooks exist on how to build up partial differential equations for the modeling of the wave propagation (Aki and Richards, 1980, 2002; Chapman, 2004), I shall remind the main principles and governing laws in this section for the sake of completeness.

1.1.1 Newton's law: conservation of momentum

The seismic wave propagation is governed by elastodynamics wave equations. These equations are derived from the Newton's law relating the acceleration to external forces. This relation can be expressed as

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sum_j \partial_j \sigma_{ij} + F_i, \tag{1.1}$$

where the density is denoted by ρ. The quantities u_i and σ_{ij} are respectively the particle displacement and the stress tensor with i, j denoting possible space directions, and F_i is the volumetric applied force.

The deformation of a continuum body can be linked to the strain tensor ϵ_{ij} through the rheological laws of mechanics. The particle displacement and the strain tensors are therefore linked through

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right). \tag{1.2}$$

Note that in the relation (1.2), an implicit summation over the repeated index k is assumed (Einstein summation convention). One can see that the definition (1.2) is non-linear with respect to the displacements. In the framework of the seismic wave propagation, we consider small perturbations approximation: only the first-order terms are kept, and upper order terms in u_i are neglected. The corresponding linearized constitutive law writes

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{1.3}$$

1.1.2 Linear stress/strain relation: Hooke's law

In the framework of the wave propagation, one may also consider the simple relation given by the generalized Hooke's law where the strain and stress tensors are linked linearly. It is defined as follows using the tensor of fourth-order c_{ijkl}

$$\sigma_{ij} = \sum_{k=1}^{3}\sum_{l=1}^{3} c_{ijkl}\epsilon_{kl}.$$

In a general fully anisotropic medium, the above formula provides a 81 stiffness element tensor. However, inherent symmetries of stress tensor

$$\sigma_{ij} = \sigma_{ji} \Rightarrow c_{ijkl} = c_{jikl},$$

reduces the number of independent stiffness elements from 81 to 54, and symmetries of the strain tensor

$$\epsilon_{kl} = \epsilon_{lk} \Rightarrow c_{ijkl} = c_{ijlk},$$

reduces the number of independent stiffness elements from 54 to 36. To further reduce the number of independent stiffness elements, one needs to account for energetic considerations (Reddy, 2002). The strain energy ψ is introduced

$$\psi = \frac{1}{2}E\epsilon^2. \tag{1.4}$$

The Young's modulus, denoted by E and also known as the elastic modulus, describes the elastic properties of the medium. It is expressed as the ratio of the stress by the strain. Using the above equation (1.4), the stress tensor can be expressed as

$$\sigma_{ij} = \frac{\partial \psi}{\partial \epsilon_{ij}} = c_{ijkl}\epsilon_{kl},$$

leading to

$$c_{ijkl} = \frac{\partial^2 \psi}{\partial \epsilon_{kl}\partial \epsilon_{ij}} = \frac{\partial^2 \psi}{\partial \epsilon_{ij}\partial \epsilon_{kl}} = c_{klij}.$$

Considering the symmetries of the stress tensor further reduces the number of independent stiffness elements to only 21 which can be recast as a 6 × 6 matrix, using the Voigt's notations (Voigt, 1889). Following these notations, the stress tensor may be stated as a 6-components vector through

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)^T = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy})^T,$$

where the upper script ^{T} is for transposition. The strain tensor is written as well

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6)^T = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}, 2\epsilon_{yz}, 2\epsilon_{xz}, 2\epsilon_{xy})^T.$$

Note the factor 2 in the strain vector definition which will require cautiousness when manipulating these entities as they do not follow vector properties. We end up with the following expression

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{pmatrix}.$$

Here, I keep the subscripts x, y, z instead of the index numbers for the stress and strain tensors to keep in mind spatial coordinates while I use them.

1.1.3 Isotropic media

I focus in this thesis on isotropic media. In this case, the properties of the medium remain the same with respect to the space directions. The stiffness tensor can therefore be reduced to only two independent physical parameters as in the following expression

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{pmatrix}, \tag{1.5}$$

where the Lamé parameters are denoted by symbols λ and μ. An isotropic medium can then be characterized by only two parameters where μ is also known as the shear modulus. The mechanics community usually employs the Young modulus E which measures the stiffness and the compressional properties of a body and the Poisson ratio ν related to the Lamé parameters through the equations

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \quad \nu = \frac{\lambda}{2(\lambda + \mu)}.$$

These coefficients better describe the rock physics during laboratory experiments where the measurement is performed through strains. For the formulation of the wave propagation equations, the Lamé coefficients are usually preferred.

1.2 Time-domain formulation

Following the above relations, one can derive the equations governing the wave propagation in the time domain. In order to reduce the computational cost and the memory requirements, the equations are usually written with respect to the particle displacement or the particle velocity by eliminating the stress components in the elastic approximation. Under the acoustic approximation, the particle velocity is eliminated from the system of equations yielding one equation involving the scalar pressure wavefield.

In the following, the wave equations are derived in the time domain, first in the second-order displacement-stress formulation, then in the first-order velocity-stress formulation.

1.2.1 Second-order isotropic elastic wave equations: displacement-stress formulation

From the combination of the equation of motion (1.1) and the Hooke's law (1.5) under the assumption of an isotropic medium, the elastic system of equations in an isotropic medium can be expressed in the displacement-stress formulation as

$$\left\{\begin{array}{rcl}
\rho\dfrac{\partial^2 u_x}{\partial t^2} &=& \dfrac{\partial \sigma_{xx}}{\partial x}+\dfrac{\partial \sigma_{xy}}{\partial y}+\dfrac{\partial \sigma_{xz}}{\partial z}+F_x,\\
\rho\dfrac{\partial^2 u_y}{\partial t^2} &=& \dfrac{\partial \sigma_{xy}}{\partial x}+\dfrac{\partial \sigma_{yy}}{\partial y}+\dfrac{\partial \sigma_{yz}}{\partial z}+F_y,\\
\rho\dfrac{\partial^2 u_z}{\partial t^2} &=& \dfrac{\partial \sigma_{xz}}{\partial x}+\dfrac{\partial \sigma_{yz}}{\partial y}+\dfrac{\partial \sigma_{zz}}{\partial z}+F_z,\\
\sigma_{xx} &=& (\lambda+2\mu)\dfrac{\partial u_x}{\partial x}+\lambda\dfrac{\partial u_y}{\partial y}+\lambda\dfrac{\partial u_z}{\partial z}+\sigma_{xx_0},\\
\sigma_{yy} &=& \lambda\dfrac{\partial u_x}{\partial x}+(\lambda+2\mu)\dfrac{\partial u_y}{\partial y}+\lambda\dfrac{\partial u_z}{\partial z}+\sigma_{yy_0},\\
\sigma_{zz} &=& \lambda\dfrac{\partial u_x}{\partial x}+\lambda\dfrac{\partial u_y}{\partial y}+(\lambda+2\mu)\dfrac{\partial u_z}{\partial z}+\sigma_{zz_0},\\
\sigma_{xy} &=& \mu\left(\dfrac{\partial u_x}{\partial y}+\dfrac{\partial u_y}{\partial x}\right)+\sigma_{xy_0},\\
\sigma_{yz} &=& \mu\left(\dfrac{\partial u_y}{\partial z}+\dfrac{\partial u_z}{\partial y}\right)+\sigma_{yz_0},\\
\sigma_{xz} &=& \mu\left(\dfrac{\partial u_x}{\partial z}+\dfrac{\partial u_z}{\partial x}\right)+\sigma_{xz_0},
\end{array}\right. \tag{1.6}$$

where the excitation terms σ_{ij_0} on the stress are added. The system of equations (1.6) can be written with respect to the particle displacement by eliminating the stress terms, leading to

$$\left\{\begin{array}{r}
\rho\partial_{tt}u_x-\partial_x\left((\lambda+2\mu)\partial_x u_x\right)-\partial_y\left(\mu\partial_y u_x\right)-\partial_z\left(\mu\partial_z u_x\right)-\partial_x\left(\lambda\partial_y u_y\right)-\partial_y\left(\mu\partial_x u_y\right)\\
-\partial_x\left(\lambda\partial_z u_z\right)-\partial_z\left(\mu\partial_x u_z\right)=F_x+\partial_x\sigma_{xx_0}+\partial_y\sigma_{xy_0}+\partial_z\sigma_{xz_0},\\
\rho\partial_{tt}u_y-\partial_x\left(\mu\partial_x u_y\right)-\partial_y\left((\lambda+2\mu)\partial_y u_y\right)-\partial_z\left(\mu\partial_z u_y\right)-\partial_y\left(\lambda\partial_x u_x\right)-\partial_x\left(\mu\partial_y u_x\right)\\
-\partial_y\left(\lambda\partial_z u_z\right)-\partial_z\left(\mu\partial_y u_z\right)=F_y+\partial_x\sigma_{xy_0}+\partial_y\sigma_{yy_0}+\partial_z\sigma_{yz_0},\\
\rho\partial_{tt}u_z-\partial_x\left(\mu\partial_x u_z\right)-\partial_y\left(\mu\partial_y u_z\right)-\partial_z\left((\lambda+2\mu)\partial_z u_z\right)-\partial_z\left(\lambda\partial_x u_x\right)-\partial_x\left(\mu\partial_z u_x\right)\\
-\partial_z\left(\lambda\partial_y u_y\right)-\partial_y\left(\mu\partial_z u_y\right)=F_z+\partial_x\sigma_{xz_0}+\partial_y\sigma_{yz_0}+\partial_z\sigma_{zz_0}.
\end{array}\right. \tag{1.7}$$

The solution of the system (1.7) is implemented in many widespread computer codes such as the SPICE Code Validation for seismic wave propagation and earthquake motion simulation developed by Moczo et al. (2005).

1.2.2 First-order isotropic elastic wave equations: velocity-stress formulation

The system of elastodynamic equations (1.6) can be recast into an hyperbolic system of first-order equations involving now the particle velocities $v_i = \partial u_i/\partial t$ and the stress components (Virieux, 1986). This velocity-stress formulation in an isotropic medium is expressed as

$$\left\{\begin{array}{lcl}
\rho\dfrac{\partial v_x}{\partial t} & = & \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{xy}}{\partial y} + \dfrac{\partial \sigma_{xz}}{\partial z} + F_x, \\
\rho\dfrac{\partial v_y}{\partial t} & = & \dfrac{\partial \sigma_{xy}}{\partial x} + \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \sigma_{yz}}{\partial z} + F_y, \\
\rho\dfrac{\partial v_z}{\partial t} & = & \dfrac{\partial \sigma_{xz}}{\partial x} + \dfrac{\partial \sigma_{yz}}{\partial y} + \dfrac{\partial \sigma_{zz}}{\partial z} + F_z, \\
\dfrac{\partial \sigma_{xx}}{\partial t} & = & (\lambda+2\mu)\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_y}{\partial y} + \lambda\dfrac{\partial v_z}{\partial z} + \dfrac{\partial \sigma_{xx_0}}{\partial t}, \\
\dfrac{\partial \sigma_{yy}}{\partial t} & = & \lambda\dfrac{\partial v_x}{\partial x} + (\lambda+2\mu)\dfrac{\partial v_y}{\partial y} + \lambda\dfrac{\partial v_z}{\partial z} + \dfrac{\partial \sigma_{yy_0}}{\partial t}, \\
\dfrac{\partial \sigma_{zz}}{\partial t} & = & \lambda\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_y}{\partial y} + (\lambda+2\mu)\dfrac{\partial v_z}{\partial z} + \dfrac{\partial \sigma_{zz_0}}{\partial t}, \\
\dfrac{\partial \sigma_{xy}}{\partial t} & = & \mu\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right) + \dfrac{\partial \sigma_{xy_0}}{\partial t}, \\
\dfrac{\partial \sigma_{yz}}{\partial t} & = & \mu\left(\dfrac{\partial v_y}{\partial z} + \dfrac{\partial v_z}{\partial y}\right) + \dfrac{\partial \sigma_{yz_0}}{\partial t}, \\
\dfrac{\partial \sigma_{xz}}{\partial t} & = & \mu\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + \dfrac{\partial \sigma_{xz_0}}{\partial t}.
\end{array}\right. \tag{1.8}$$

1.2.3 First-order isotropic acoustic wave equations: velocity-pressure formulation

The acoustic approximation is a particular case that can be derived from the elastodynamic equations by considering the Lamé coefficient μ equal to zero. Under the acoustic assumption, only compressional waves travel in the medium. Therefore, only the normal stress components $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$ are kept. The isotropic elastic wave equations (1.8) reduce to the acoustic wave equation

$$\left\{\begin{array}{lcl}
\rho\dfrac{\partial v_x}{\partial t} & = & \dfrac{\partial \sigma_{xx}}{\partial x} + F_x + \dfrac{\partial}{\partial y}\left(\dfrac{\partial \sigma_{xy_0}}{\partial t}\right) + \dfrac{\partial}{\partial z}\left(\dfrac{\partial \sigma_{xz_0}}{\partial t}\right), \\
\rho\dfrac{\partial v_y}{\partial t} & = & \dfrac{\partial \sigma_{yy}}{\partial y} + F_y + \dfrac{\partial}{\partial x}\left(\dfrac{\partial \sigma_{xy_0}}{\partial t}\right) + \dfrac{\partial}{\partial z}\left(\dfrac{\partial \sigma_{yz_0}}{\partial t}\right), \\
\rho\dfrac{\partial v_z}{\partial t} & = & \dfrac{\partial \sigma_{zz}}{\partial z} + F_z + \dfrac{\partial}{\partial x}\left(\dfrac{\partial \sigma_{xz_0}}{\partial t}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{\partial \sigma_{yz_0}}{\partial t}\right), \\
\dfrac{\partial \sigma_{xx}}{\partial t} & = & \lambda\left(\dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}\right) + \dfrac{\partial \sigma_{xx_0}}{\partial t}, \\
\dfrac{\partial \sigma_{yy}}{\partial t} & = & \lambda\left(\dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}\right) + \dfrac{\partial \sigma_{yy_0}}{\partial t}, \\
\dfrac{\partial \sigma_{zz}}{\partial t} & = & \lambda\left(\dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}\right) + \dfrac{\partial \sigma_{zz_0}}{\partial t}.
\end{array}\right. \tag{1.9}$$

The last three equations in the system (1.9) are the same. Therefore by introducing the pressure wavefield defined as

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}),$$

the first-order hyperbolic acoustic system is expressed as

$$\left\{ \begin{array}{lcl} \rho \dfrac{\partial v_x}{\partial t} & = & -\dfrac{\partial p}{\partial x} + F_x + \dfrac{\partial}{\partial y}\left(\dfrac{\partial \sigma_{xy_0}}{\partial t}\right) + \dfrac{\partial}{\partial z}\left(\dfrac{\partial \sigma_{xz_0}}{\partial t}\right), \\ \rho \dfrac{\partial v_y}{\partial t} & = & -\dfrac{\partial p}{\partial y} + F_y + \dfrac{\partial}{\partial x}\left(\dfrac{\partial \sigma_{xy_0}}{\partial t}\right) + \dfrac{\partial}{\partial z}\left(\dfrac{\partial \sigma_{yz_0}}{\partial t}\right), \\ \rho \dfrac{\partial v_z}{\partial t} & = & -\dfrac{\partial p}{\partial z} + F_z + \dfrac{\partial}{\partial x}\left(\dfrac{\partial \sigma_{xz_0}}{\partial t}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{\partial \sigma_{yz_0}}{\partial t}\right), \\ \dfrac{\partial p}{\partial t} & = & -\kappa\left(\dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}\right) + q, \end{array} \right. \tag{1.10}$$

where the excitation q is introduced in the last equation of the system (1.10) and is defined by

$$q = -\frac{1}{3}\left(\frac{\partial \sigma_{xx_0}}{\partial t} + \frac{\partial \sigma_{zz_0}}{\partial t} + \frac{\partial \sigma_{zz_0}}{\partial t}\right).$$

This excitation term is sometimes called volume velocity source. The latter is adapted to provide an explosive-type source (Brekhovskikh and Godin, 1990, 1992). In the case of fluid approximation, the Lamé coefficient λ is equal to the incompressibility coefficient κ. In this case, the P-wave velocity is sometimes called also the bulk velocity defined as

$$V_P = \sqrt{\kappa/\rho}. \tag{1.11}$$

1.2.4 Second-order isotropic acoustic wave equation: pressure formulation

The system of equations (1.10) can be reduced to one second-order acoustic equation involving only the pressure variable as

$$\frac{\partial^2 p}{\partial t^2} - \kappa\left(\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial p}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right)\right) = \frac{\partial q}{\partial t} - \kappa f - \kappa g, \tag{1.12}$$

where the excitation terms f and g are defined by

$$f = \frac{\partial}{\partial x}\left(\frac{1}{\rho}F_x\right) + \frac{\partial}{\partial y}\left(\frac{1}{\rho}F_y\right) + \frac{\partial}{\partial z}\left(\frac{1}{\rho}F_z\right),$$

$$\begin{aligned} g = \frac{\partial}{\partial x}\left\{\frac{1}{\rho}\left[\frac{\partial}{\partial y}\left(\frac{\partial \sigma_{xy_0}}{\partial t}\right) + \frac{\partial}{\partial z}\left(\frac{\partial \sigma_{xz_0}}{\partial t}\right)\right]\right\} + \frac{\partial}{\partial y}\left\{\frac{1}{\rho}\left[\frac{\partial}{\partial x}\left(\frac{\partial \sigma_{xy_0}}{\partial t}\right) + \frac{\partial}{\partial z}\left(\frac{\partial \sigma_{yz_0}}{\partial t}\right)\right]\right\} \\ + \frac{\partial}{\partial z}\left\{\frac{1}{\rho}\left[\frac{\partial}{\partial x}\left(\frac{\partial \sigma_{xz_0}}{\partial t}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \sigma_{yz_0}}{\partial t}\right)\right]\right\}. \end{aligned}$$

1.2.5 Types of waves

By introducing the compact vector notation

$$\mathbf{U} = \left(v_x, v_y, v_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}\right)^T, \tag{1.13}$$

the system of equation (1.8) can be written in a general matrix form using (1.13) as

$$\partial_t \mathbf{U} - A\partial_x \mathbf{U} - B\partial_y \mathbf{U} - C\partial_z \mathbf{U} = \mathbf{F}$$

where **F** is the source term written in matrix form as

$$\mathbf{F} = \left(F_x, F_y, F_z, \sigma_{xx_0}, \sigma_{yy_0}, \sigma_{zz_0}, \sigma_{xy_0}, \sigma_{yz_0}, \sigma_{xz_0}\right)^T,$$

and where A, B and C are the 9×9 matrices accounting for the medium properties. For the sake of simplicity, consider the system of equations (1.10) written in the 2D approximation

$$\partial_t \mathbf{U} - A\partial_x \mathbf{U} - B\partial_z \mathbf{U} = \mathbf{F}$$

with

$$\mathbf{U} = \left(v_x, v_z, \sigma_{xx}, \sigma_{zz}, \sigma_{xz}\right)^T, \quad \mathbf{F} = \left(F_x, F_z, \sigma_{xx_0}, \sigma_{zz_0}, \sigma_{xz_0}\right)^T.$$

The matrices A and B writes explicitly

$$A = \begin{pmatrix} 0 & 0 & 1/\rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\rho \\ \lambda + 2\mu & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 & 0 & 1/\rho \\ 0 & 0 & 0 & 1/\rho & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda + 2\mu & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 \end{pmatrix},$$

and their characteristic polynomials

$$\chi_A(x) = \det(A - xI), \quad \chi_B(x) = \det(B - xI),$$

can be easily calculated. The latter are given by

$$\chi_A(x) = \chi_B(x) = -x\left(x^2 - \frac{\lambda + 2\mu}{\rho}\right)\left(x^2 - \frac{\mu}{\rho}\right).$$

The eigenvalues of the matrices A and B are given by

$$0, \quad -\sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad -\sqrt{\frac{\mu}{\rho}}, \quad \sqrt{\frac{\mu}{\rho}}.$$

These eigenvalues denote the wave propagation velocities. Two waves, called compressional waves or P-waves, vibrate in the direction of the propagation and travel at the same velocity

$$V_P = \sqrt{\frac{\lambda + 2\mu}{\rho}},$$

but in opposite directions and two other waves called shear waves or S-waves, travel at the velocity

$$V_S = \sqrt{\frac{\mu}{\rho}},$$

but in opposite directions. The latter vibrate in the plane which is orthogonal to the direction of the propagation.

The particular case where the Poisson ratio ν is equal to 0.5 implies that the Lamé coefficient μ is equal to 0. It corresponds to the acoustic approximation. Therefore, only compressional waves travel in the acoustic medium as $V_S = 0$.

A particular attention is given to the zero eigenvalue. It correspond to a singular mode which should not be excited during the numerical resolution as it may lead to instabilities.

In the following, we shall derive the wave equations in the frequency domain. The computation of the solution of these equations is the main interest of this work.

1.3 Frequency-domain formulation

The wave equation, either in the elastic or the acoustic approximation, can be written in the frequency domain. For simplicity, the same notations are used for describing the unknowns in the time domain and the frequency domain as I shall consider only the frequency domain for my investigation. The frequency-domain unknowns are introduced through the Fourier transform with the convention

$$\mathfrak{F} : g(t) \longmapsto \mathfrak{F}(g)(\omega) = \int_{-\infty}^{+\infty} g(t) e^{-i\omega t} dt,$$

where we consider the angular frequency ω related to the frequency f (inverse of the period for an harmonic signal) through the relation $\omega = 2\pi f$. The pure imaginary number i is such that $i^2 = -1$. By integration by parts, one can find that the Fourier transform of the time derivative of a function g is given through the equation

$$\int_{-\infty}^{+\infty} \frac{dg(t)}{dt} e^{-i\omega t} dt = i\omega g(\omega).$$

In the following, I shall derive the frequency-domain wave equations when considering the first-order velocity-stress formulation, the second-order velocity formulation and finally the second-order pressure formulation. While composing these linear systems, we focus on the second-order hyperbolic systems. The discretization of these systems using compact derivation stencils reduces the memory request for storing field components, as well as the size of the corresponding linear system. In 3D, the system reduces from 3 to 1 equation in the acoustic approximation, and from 9 to 3 in the elastic approximation.

1.3.1 First-order and second-order frequency-domain elastic wave equations

The velocity-stress elastic wave equations in an isotropic medium (1.8) can be expressed in the frequency domain as

$$\left\{\begin{array}{lcl}
i\omega\rho v_x & = & \dfrac{\partial\sigma_{xx}}{\partial x} + \dfrac{\partial\sigma_{xy}}{\partial y} + \dfrac{\partial\sigma_{xz}}{\partial z} + F_x, \\
i\omega\rho v_y & = & \dfrac{\partial\sigma_{xy}}{\partial x} + \dfrac{\partial\sigma_{yy}}{\partial y} + \dfrac{\partial\sigma_{yz}}{\partial z} + F_y, \\
i\omega\rho v_z & = & \dfrac{\partial\sigma_{xz}}{\partial x} + \dfrac{\partial\sigma_{yz}}{\partial y} + \dfrac{\partial\sigma_{zz}}{\partial z} + F_z, \\
i\omega\sigma_{xx} & = & (\lambda+2\mu)\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_y}{\partial y} + \lambda\dfrac{\partial v_z}{\partial z} + i\omega\sigma_{xx_0}, \\
i\omega\sigma_{yy} & = & \lambda\dfrac{\partial v_x}{\partial x} + (\lambda+2\mu)\dfrac{\partial v_y}{\partial y} + \lambda\dfrac{\partial v_z}{\partial z} + i\omega\sigma_{yy_0}, \\
i\omega\sigma_{zz} & = & \lambda\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_y}{\partial y} + (\lambda+2\mu)\dfrac{\partial v_z}{\partial z} + i\omega\sigma_{zz_0}, \\
i\omega\sigma_{xy} & = & \mu\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right) + i\omega\sigma_{xy_0}, \\
i\omega\sigma_{yz} & = & \mu\left(\dfrac{\partial v_y}{\partial z} + \dfrac{\partial v_z}{\partial y}\right) + i\omega\sigma_{yz_0}, \\
i\omega\sigma_{xz} & = & \mu\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + i\omega\sigma_{xz_0}.
\end{array}\right. \tag{1.14}$$

We may consider as well the second-order form involving only velocities eliminating the stress from system (1.14) as it was done for the system (1.8), which yields

$$\left\{\begin{array}{r}
\omega^2\rho v_x + \partial_x\left((\lambda+2\mu)\partial_x v_x\right) + \partial_y\left(\mu\partial_y v_x\right) + \partial_z\left(\mu\partial_z v_x\right) + \partial_x\left(\lambda\partial_y v_y\right) + \partial_y\left(\mu\partial_x v_y\right) \\
+\partial_x\left(\lambda\partial_z v_z\right) + \partial_z\left(\mu\partial_x v_z\right) = -i\omega F_x + \partial_x\sigma_{xx_0} + \partial_y\sigma_{xy_0} + \partial_z\sigma_{xz_0}, \\[2ex]
\omega^2\rho v_y + \partial_x\left(\mu\partial_x v_y\right) + \partial_y\left((\lambda+2\mu)\partial_y v_y\right) + \partial_z\left(\mu\partial_z v_y\right) + \partial_y\left(\lambda\partial_x v_x\right) + \partial_x\left(\mu\partial_y v_x\right) \\
+\partial_y\left(\lambda\partial_z v_z\right) + \partial_z\left(\mu\partial_y v_z\right) = -i\omega F_y + \partial_x\sigma_{xy_0} + \partial_y\sigma_{yy_0} + \partial_z\sigma_{yz_0}, \\[2ex]
\omega^2\rho v_z + \partial_x\left(\mu\partial_x v_z\right) + \partial_y\left(\mu\partial_y v_z\right) + \partial_z\left((\lambda+2\mu)\partial_z v_z\right) + \partial_z\left(\lambda\partial_x v_x\right) + \partial_x\left(\mu\partial_z v_x\right) \\
+\partial_z\left(\lambda\partial_y v_y\right) + \partial_y\left(\mu\partial_z v_y\right) = -i\omega F_z + \partial_x\sigma_{xz_0} + \partial_y\sigma_{yz_0} + \partial_z\sigma_{zz_0}.
\end{array}\right. \tag{1.15}$$

This system of second-order hyperbolic equations is the one which will be solved using the numerical methods introduced later. The system of equations (1.15) can be derived as well from equations (1.7) by Fourier transform after a time derivation and assuming the medium is at rest at the initial time.

1.3.2 Second-order frequency-domain acoustic wave equation

In the same manner, the acoustic wave equation (1.12) expressed in the time domain can be written in the frequency domain. The second-order acoustic wave equation writes

$$\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial p}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial p}{\partial z}\right) + \frac{\omega^2}{\kappa}p = -i\frac{\omega}{\kappa}q + f + i\omega\tilde{g}, \tag{1.16}$$

where $\tilde{g}$ is the Fourier transform of g

$$\tilde{g} = \frac{\partial}{\partial x}\left[\frac{1}{\rho}\left(\frac{\partial\sigma_{xy_0}}{\partial y} + \frac{\partial\sigma_{xz_0}}{\partial z}\right)\right] + \frac{\partial}{\partial y}\left[\frac{1}{\rho}\left(\frac{\partial\sigma_{xy_0}}{\partial x} + \frac{\partial\sigma_{yz_0}}{\partial z}\right)\right] + \frac{\partial}{\partial z}\left[\frac{1}{\rho}\left(\frac{\partial\sigma_{xz_0}}{\partial x} + \frac{\partial\sigma_{yz_0}}{\partial y}\right)\right].$$

In the above equations (1.15) and (1.16), the medium properties are supposed to be heterogeneous, leading to mixed spatial derivatives operators $\partial_x \frac{1}{\rho} \partial_x$, $\partial_y \frac{1}{\rho} \partial_y$ and $\partial_z \frac{1}{\rho} \partial_z$. When considering homogeneous density, the frequency-domain acoustic equation (1.16) reduces to the well-known Helmholtz elliptic equation

$$\Delta p + k^2 p = \rho(-i\frac{\omega}{\kappa} q + f + i\omega \tilde{g}), \tag{1.17}$$

where the expression $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ denotes the Laplacian operator and where we have introduced the wavenumber $k = \omega / c$. When the velocity c is homogeneous, analytical solutions of (1.17) can be easily derived. Note that the P-wave velocity c and the incompressibility coefficient κ are linked by the relation (1.11).

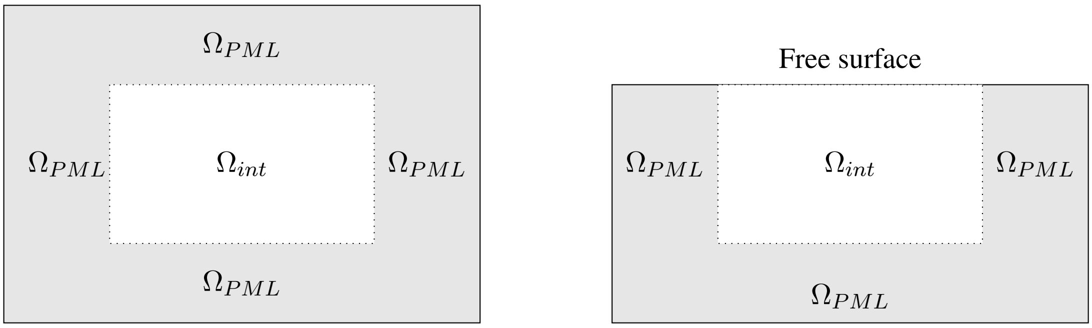
1.4 Boundary conditions

Simulating the propagation of seismic waves in the subsurface as defined by the time-domain (1.8), (1.12) or the frequency-domain (1.15),(1.16) equations requires the use of proper boundary conditions. At the exploration and regional scales the subsurface can be assimilated with a semi-infinite medium with a free-surface boundary condition at the air/subsurface interface. We thus need to use appropriate absorbing boundary conditions to mimic infinite medium in depth and in the horizontal directions when we consider numerical approximations to the solution of these equations.

1.4.1 Absorbing boundary conditions

Two general classes of methods allow to mimic wave propagation in infinite domain: absorbing boundary condition (ABC) (introduced by Clayton and Engquist (1977)) and absorbing boundary layers (introduced by Cerjan et al. (1985)).

The ABC method is based on paraxial approximations of the wave equation. This method assumes that the wave propagates in the direction perpendicular to the border. Only these waves are absorbed and therefore waves propagating in other directions (non-perpendicular to the border) will not be completely absorbed.



(a) PML (b) Free surface

Figure 1.1: Domain of interest Ω_{int} with PML layers Ω_{PML} (a) and free surface (b).

The second class of methods introduces fictitious layers at the boundaries of the domain. Two configurations are investigated in this work: the first one is presented in Figure (1.1a) where the interest domain with surrounded by absorbing layers and the second configuration is presented in Figure (1.1b) where the top boundary is replaced by a free surface. The latter is introduced in the following subsection 1.4.2. The ABC method was first promoted by Cerjan et al. (1985) for the second-order in time acoustic equation. The wave is artificially damped as it enters this area. However, this method generates reflections at the interface between the layer and the domain of interest. These reflections can be mitigated by choosing variable damping coefficient that smoothly grow from zero at the interface to the external boundary of the layer. However, this reduces the absorbing capability of the layer and requires to increase its size, and thus additional computations during the simulation.

The method I use in my investigation is the perfectly matched layers (PML) technique which Bérenger (1994) introduced first for the 2D and 3D Maxwell equations. The initial method is based on a splitting of the hyperbolic system and the introduction of smooth damping coefficients in the layer. An example of damping profile is presented in Figure (1.2). Theoretically the reflections are canceled but in practice, a small non-zero reflection coefficient remains. However, in many practical applications the amplitude of these undesired and artificial reflected waves remains very small. This method has become quickly a standard method in seismic applications.

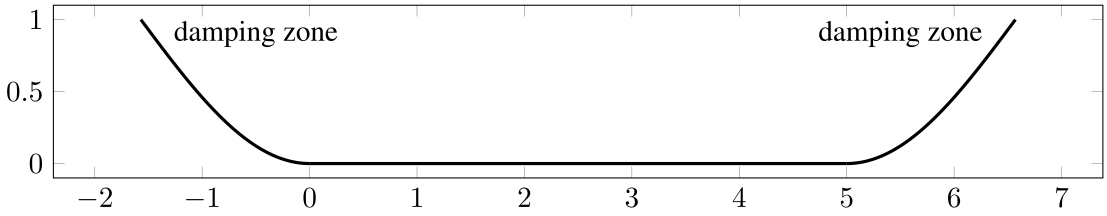


Figure 1.2: Example of damping profile along the horizontal direction.

Both time and frequency formulations could be considered but I shall focus on the frequency formulation. I shall describe how to consider this sponge zone for the frequency-domain acoustic wave equation (1.16) but similar expressions exist for the elastic cases. In order to write this equation with the PML boundary condition, one needs to consider first the system of equations (1.10) in the frequency domain and introduce the 1D space dependent damping functions $\gamma_x, \gamma_y, \gamma_z$. The damping function may have the form

$$\gamma_x(x) = C_{PML} \cos\left(\frac{\pi}{2}\frac{x}{L_x}\right),$$

where L_x is the width of the sponge layer and C_{PML} is a real value used to control the damping rate. The coefficient C_{PML} needs to be handled carefully in order to avoid reflections at the interface of the domain and the sponge layer. The pressure wavefield is split into three non-physical components

p_x, p_y, p_z such that $p = p_x + p_y + p_z$ and

$$\left\{ \begin{array}{rcl} i\omega v_x + \gamma_x v_x &=& b\dfrac{\partial p}{\partial x} + F_x + i\omega\left(\dfrac{\partial \sigma_{xy_0}}{\partial y} + \dfrac{\partial \sigma_{xz_0}}{\partial z}\right), \\ i\omega v_y + \gamma_y v_y &=& b\dfrac{\partial p}{\partial y} + F_y + i\omega\left(\dfrac{\partial \sigma_{xy_0}}{\partial x} + \dfrac{\partial \sigma_{yz_0}}{\partial z}\right), \\ i\omega v_z + \gamma_z v_z &=& b\dfrac{\partial p}{\partial z} + F_z + i\omega\left(\dfrac{\partial \sigma_{xz_0}}{\partial x} + \dfrac{\partial \sigma_{yz_0}}{\partial y}\right), \\ i\omega p_x + \gamma_x p_x &=& \kappa\dfrac{\partial v_x}{\partial x}, \\ i\omega p_y + \gamma_y p_y &=& \kappa\dfrac{\partial v_y}{\partial x}, \\ i\omega p_z + \gamma_z p_z &=& \kappa\dfrac{\partial v_z}{\partial x} + i\omega q. \end{array} \right. \tag{1.18}$$

Note the the source term on the pressure equations can applied equivalently to either of the last tree equations since it is additive. The system of equations (1.18) is written into the following one allowing to eliminate the particle velocity terms

$$\left\{ \begin{array}{rcl} i\omega v_x &=& \dfrac{b}{1 - i\gamma_x/\omega}\dfrac{\partial p}{\partial x} + \dfrac{1}{1 - i\gamma_x/\omega}F_x + \dfrac{i\omega}{1 - i\gamma_x/\omega}\left(\dfrac{\partial \sigma_{xy_0}}{\partial y} + \dfrac{\partial \sigma_{xz_0}}{\partial z}\right), \\ i\omega v_y &=& \dfrac{b}{1 - i\gamma_y/\omega}\dfrac{\partial p}{\partial y} + \dfrac{1}{1 - i\gamma_y/\omega}F_y + \dfrac{i\omega}{1 - i\gamma_y/\omega}\left(\dfrac{\partial \sigma_{xy_0}}{\partial x} + \dfrac{\partial \sigma_{yz_0}}{\partial z}\right), \\ i\omega v_z &=& \dfrac{b}{1 - i\gamma_z/\omega}\dfrac{\partial p}{\partial z} + \dfrac{1}{1 - i\gamma_z/\omega}F_z + \dfrac{i\omega}{1 - i\gamma_z/\omega}\left(\dfrac{\partial \sigma_{xz_0}}{\partial x} + \dfrac{\partial \sigma_{yz_0}}{\partial y}\right), \\ \dfrac{i\omega + \gamma_x}{\kappa}p_x &=& \dfrac{\partial v_x}{\partial x}, \\ \dfrac{i\omega + \gamma_y}{\kappa}p_y &=& \dfrac{\partial v_y}{\partial x}, \\ \dfrac{i\omega + \gamma_z}{\kappa}p_z &=& \dfrac{\partial v_z}{\partial x} + \dfrac{i\omega}{\kappa}q. \end{array} \right.$$

This system of equations is recast into

$$\frac{1}{b}\left(\frac{1}{\xi_x}\frac{\partial}{\partial x}\left(\frac{b}{\xi_x}\frac{\partial p}{\partial x}\right) + \frac{1}{\xi_y}\frac{\partial}{\partial y}\left(\frac{b}{\xi_y}\frac{\partial p}{\partial y}\right) + \frac{1}{\xi_z}\frac{\partial}{\partial z}\left(\frac{b}{\xi_z}\frac{\partial p}{\partial z}\right)\right) + k^2 p = \rho(-i\frac{\omega}{\kappa}q + f + i\omega\tilde{g}), \tag{1.19}$$

where the coefficients $\xi_j = 1 - i\gamma_j/\omega$ are introduced with j the space directions x, y, z. For the sake of simplicity, I shall denote by $s(\omega)$ the right-hand side of of equation (1.19).

1.4.2 Free surface boundary condition

Alternatively, the free-surface boundary condition can be used at the top boundary of the domain for realistic wave propagation modeling. The method used in this work is proposed by Mittet (2002). It is an efficient strategy based on the assumption that the elastic Hooke's tensor on the free surface can be taken similar to a transversely isotropic medium. The method is based on the first-order velocity-stress formulation of the elastic wave equations using staggered-grid modeling schemes. The density and the Lamé coefficient μ on the one hand and the Lamé coefficient λ on the other hand are given different treatments. The method uses an average of the internal value and the vacuum value for the density and the Lamé coefficient μ at the free-surface nodes. The Lamé coefficient λ is set to zero at the surface nodes. This approach has the advantage of simulating the free-surface without any additional grid points.

In this thesis, I shall be concerned only with wave equations formulated in the frequency domain. Namely, I will deal with the scalar 2D frequency-domain visco-acoustic wave equation (1.16) first and then with the 2D and 3D frequency-domain visco-elastic wave equations (1.15). I shall consider two configurations for the boundary conditions. PML are considered for all boundaries (Bérenger, 1994) in the first configuration. Then, the top boundary is replaced with a free surface boundary condition (Mittet, 2002) in the second configuration while PML are used on the lateral and the bottom boundaries.

1.5 Finite-difference discretization strategy

Any numerical method requires the continuous equations to be formulated in the discrete domain. Numerical discretization schemes are used to approximate in a stable, consistent and accurate way the continuous information of the problem. They can be described as a model reduction which transforms an infinite dimensional equation into a finite dimensional set of equations. Such reduction comes inherently with a loss of information due to the way of addressing the partial differential operators. I consider here one class of numerical discretization method: the finite-difference method. Therefore, this section focuses on several finite-difference approaches for the spatial discretization as I am not concerned with time-domain wave modeling. The reader may refer to the complete review of Moczo et al. (2007) for a deeper analysis on modeling techniques of seismic waves. I shall restrict myself to the second- and fourth-order staggered-grid stencils with optimal coefficients.

The finite-difference method is the most widespread method for time and space discretization. It is popular within the Earth science community for the simplicity of its implementation and its efficiency. The method presents low requirements in terms of computational cost and memory demand as the discrete problem is sparse. This method relies on the strong formulation of the problem. The main idea behind any finite-difference scheme is related to the derivative of the function u at a point x

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}, \tag{1.20}$$

and the fact that when h tends toward 0, the quotient on the right hand side of equation (1.20) provides a good approximation of the derivative u'. Arbitrary order approximations can be derived from the Taylor polynomial expansion.

Theorem 1.1. *Let I be an interval of $\mathbb{R}$, x a point of I and for a given integer n, $u : I \longrightarrow \mathbb{R}$ a function of class $\mathcal{C}^n$ on I. For any $h \in \mathbb{R}$ such that $x + h \in I$ we have*

$$\begin{aligned} u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2!}u^{(2)}(x) + \cdots \frac{h^n}{n!}u^{(n)}(x) + h^n\epsilon(h), \\ &= \sum_{k=1}^{n} \frac{h^k}{k!}u^{(k)}(x) + h^n\epsilon(h), \end{aligned}$$

where $\epsilon(h) \xrightarrow[\varepsilon \to 0]{} 0$. By convention, we assume $0! = 1$.

Definition 1.1. *The sum*

$$T_n(x) = \sum_{k=1}^{n} \frac{h^k}{k!} u^{(k)}(x),$$

is called the n^{th}-order Taylor polynomial of u.

The finite-difference approximation of u' at the point x can be derived from the Taylor polynomial (1.1) as in

- The forward difference $(D^+u)(x) = \dfrac{u(x+h) - u(x)}{h}$;
- The backward difference $(D^-u)(x) = \dfrac{u(x) - u(x-h)}{h}$;
- The centered difference $(Du)(x) = \dfrac{u(x+h) - u(x-h)}{2h}$.

It is easy to prove by Taylor expansion that

$$\begin{aligned}(D^+u)(x) - u'(x) &= O(h),\\ (D^-u)(x) - u'(x) &= O(h),\\ (Du)(x) - u'(x) &= O(h^2).\end{aligned}$$

Using the second-order Taylor polynomial, we have

$$\begin{aligned}\frac{u(x+h) - u(x)}{h} - u'(x) &= \frac{h}{2} u''(x) + h\epsilon(h),\\ \frac{u(x) - u(x-h)}{h} - u'(x) &= -\frac{h}{2} u''(x) + h\epsilon(h),\end{aligned}$$

which corresponds to two first-order approximations of $u'(x)$. The centered difference is obtained using the third degree Taylor polynomial and we obtain

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2} u''(x) + \frac{h^3}{3!} u^{(3)}(x) + h^3\epsilon(h), \tag{1.21}$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2} u''(x) - \frac{h^3}{3!} u^{(3)}(x) + h^3\epsilon(h). \tag{1.22}$$

The difference between the two above equations (1.21) and (1.22) gives the second-order approximation of $u'(x)$

$$\frac{u(x+h) - u(x-h)}{2h} - u'(x) = \frac{h^2}{3!} u^{(3)}(x) + h^2\epsilon(h).$$

The second-order approximation of the second derivative u'' can be expressed as well as

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2).$$

To better describe the finite-difference method, let us consider for instance the 2D Laplace problem with the Dirichlet boundary condition

$$\begin{aligned}\Delta u &= f, \quad \text{on} \quad \Omega =]0,1[\times]0,1[,\\ u &= 0, \quad \text{on} \quad \partial\Omega,\end{aligned} \tag{1.23}$$

where f is the source term. Given two integers $n, m \geq 2$, we construct the grid $\Omega_h = \{(x_i, y_j) \in \Omega\}$ (see Figure 1.3) by the tensor product of the two grids of $]0, 1[$ such that

$$x_i = (i-1)h_x, \quad i = 1, \cdots, n, \quad h_x = \frac{1}{n-1},$$
$$y_j = (j-1)h_y, \quad j = 1, \cdots, m, \quad h_y = \frac{1}{m-1}.$$

Let $h = \max\{h_x, h_y\}$ denote the size of the cell and $\Gamma_h = \{(x_i, y_j) \in \partial\Omega\}$ the boundary. The discrete function is defined by $u_h : \Omega_h \longrightarrow \mathbb{R}$ such that $u_h(x_i, y_j) = u(x_i, y_j)$. It is more convenient to use sub-index (i, j) for the discrete function which amounts to $u_{i,j} = u_h(x_i, y_j)$.

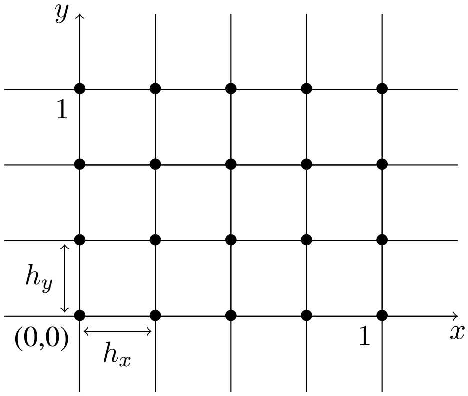


Figure 1.3: Finite-difference mesh

The problem (1.23) is therefore written on each node of the discretization grid as

$$\begin{aligned} \Delta u(x_i, y_j) &= f(x_i, y_j), && \text{in} \quad \Omega_h, \\ u(x_i, y_j) &= 0, && \text{on} \quad \Gamma_h. \end{aligned} \tag{1.24}$$

Assuming u is $\mathcal{C}^4$ and $h_x = h_y = h$, the Taylor series development of u at the point $(x_{i+1}, y_j) = (x_i + h, y_j)$ gives

$$u(x_{i+1}, y_j) = u(x_i, y_j) + h\frac{\partial u}{\partial x}(x_i, y_j) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j) + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i, y_j) + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4}(x_i, y_j) + o(h^4), \tag{1.25}$$

and similarly the Taylor series development of u at the point $(x_{i-1}, y_j) = (x_i - h, y_j)$ gives

$$u(x_{i-1}, y_j) = u(x_i, y_j) - h\frac{\partial u}{\partial x}(x_i, y_j) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x_i, y_j) - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i, y_j) + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4}(x_i, y_j) + o(h^4). \tag{1.26}$$

The sum of these two developments (1.25) and (1.26) gives us the second-order finite-difference approximation of the second-order derivative of u with respect to x

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \quad = \quad \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \quad - \quad \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i, y_j) \; + \; o(h^2), \tag{1.27}$$

where the term

$$\frac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)}{h^2}$$

represents the finite-difference approximation of the second-order derivative and

$$\tau(x_i,y_j,h)=\frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(x_i,y_j)+o(h^2)$$

is the truncation error. The expression (1.27) simplifies into

$$\frac{\partial^2 u}{\partial x^2}(x_i,y_i)=\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+o(h^2). \tag{1.28}$$

The same approach is adopted to calculate the second-order derivative finite-difference approximation with respect to y. Using (1.28), one can write the finite-difference discretization of the problem (1.23) as

$$\begin{aligned} \frac{u_{i+1,j}+u_{i,j+1}-4u_{i,j}+u_{i-1,j}+u_{i,j-1}}{h^2} &= f_{i,j}, \quad 1\le i\le N_x, \quad 1\le j\le N_y, \\ u_{i,j} &= 0, \quad \text{on the border nodes.} \end{aligned} \tag{1.29}$$

If the unknowns are sequentially indexed column-wise and if the solution on the grid points located at the column i of the mesh is denoted by $u_i=\left(u_{i,1},u_{i,2},\dots,u_{i,N_y}\right)^T$ then the equation (1.29) is formulated as

$$\begin{aligned} \frac{1}{h^2}\left(u_{i+1}+Au_i+u_{i-1}\right) &= f_i, \quad 1\le i\le N_x, \\ u_{i,j} &= 0, \quad \text{on the border nodes,} \end{aligned} \tag{1.30}$$

where A is the tridiagonal matrix of size N_y

$$A=\begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix},$$

corresponding to the discretization of the problem in the vertical direction. Therefore, the entire system can be written in matrix form as

$$\frac{1}{h^2}\begin{bmatrix} A & I & 0 & & \\ I & A & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & A & I \\ & & & I & A \end{bmatrix}\cdot\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N_x} \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N_x} \end{bmatrix}. \tag{1.31}$$

The solution of the discrete Laplace problem (1.29) is finally computed through the solution of the linear system (1.31).

1.5.1 Fourth-order staggered finite-difference scheme for the 2D acoustic wave equation

Consider the 2D acoustic wave equation formulated as a first-order hyperbolic system (Virieux, 1984) using the physical quantities defined by the pressure and the particle velocity. The 3D equations (1.18) which are introduced in subsection 1.4 are recast in 2D. Perfectly matched layers (Bérenger, 1994) are implemented at the edges of the domain to absorb the outgoing wave.

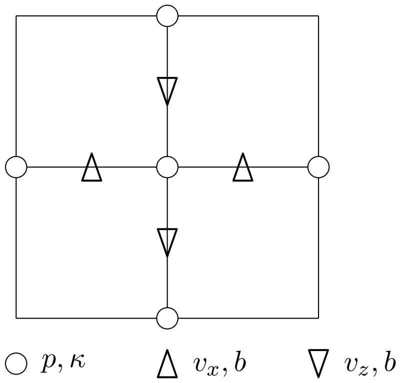


Figure 1.4: Illustration of the staggered-grid finite-difference scheme for the 2D frequency-domain acoustic wave equation. The pressure wavefield is defined on the regular grid (i, j) labeled by ○. The density is defined at intermediate positions at $(i \pm 1/2, j)$ and $(i, j \pm 1/2)$ labeled by ▽ and △.

Following the staggered-grid approach where the finite-difference scheme is presented in Figure 1.4, the system of equations (1.18) is discretized using the fourth-order centered finite-differences (Levander, 1988)

$$\begin{aligned}
\left[\frac{\partial v_x}{\partial x}\right]^h_{i,j} &= \frac{1}{h}\left[\tfrac{9}{8}\left((v_x)_{i+1/2,j} - (v_x)_{i-1/2,j}\right) - \tfrac{1}{24}\left((v_x)_{i+3/2,j} - (v_x)_{i-3/2,j}\right)\right], \\
\left[\frac{\partial v_z}{\partial z}\right]^h_{i,j} &= \frac{1}{h}\left[\tfrac{9}{8}\left((v_z)_{i,j+1/2} - (v_z)_{i,j-1/2}\right) - \tfrac{1}{24}\left((v_z)_{i,j+3/2} - (v_z)_{i,j-3/2}\right)\right], \\
\left[\frac{\partial p}{\partial x}\right]^h_{i+1/2,j} &= \frac{1}{h}\left[\tfrac{9}{8}\left(p_{i+1,j} - p_{i,j}\right) - \tfrac{1}{24}\left(p_{i+2,j} - p_{i-1,j}\right)\right], \\
\left[\frac{\partial p}{\partial z}\right]^h_{i,j+1/2} &= \frac{1}{h}\left[\tfrac{9}{8}\left(p_{i,j+1} - p_{i,j}\right) - \tfrac{1}{24}\left(p_{i,j+2} - p_{i,j-1}\right)\right],
\end{aligned} \tag{1.32}$$

where h is the discretization step in the uniform grid.

The particle velocities v_x and v_z are eliminated in the discretized equations. This is the so-called parsimonious approach (Luo and Schuster, 1990). After the summation of the two remaining discrete equations, the scalar equation for the pressure wavefield p is obtained

$$
\begin{aligned}
\frac{-\omega^2}{\kappa_{i,j}} p_{i,j} = \quad & \frac{1}{(\xi_x)_{i,j}} \frac{1}{h^2} \left\{ \frac{9}{8} \left[\frac{b_{i+1/2,j}}{(\xi_x)_{i+1/2,j}} \left(\frac{9}{8}(p_{i+1,j} - p_{i,j}) - \frac{1}{24}(p_{i+2,j} - p_{i-1,j}) \right) \right. \right. \\
& - \frac{b_{i-1/2,j}}{(\xi_x)_{i-1/2,j}} \left(\frac{9}{8}(p_{i,j} - p_{i-1,j}) - \frac{1}{24}(p_{i+1,j} - p_{i-2,j}) \right) \Bigg] \\
& - \frac{1}{24} \left[\frac{b_{i+3/2,j}}{(\xi_x)_{i+3/2,j}} \left(\frac{9}{8}(p_{i+2,j} - p_{i+1,j}) - \frac{1}{24}(p_{i+3,j} - p_{i,j}) \right) \right. \\
& \left. \left. - \frac{b_{i-3/2,j}}{(\xi_x)_{i-3/2,j}} \left(\frac{9}{8}(p_{i-1,j} - p_{i-2,j}) - \frac{1}{24}(p_{i,j} - p_{i-3,j}) \right) \right] \right\} \\
& + \frac{1}{(\xi_z)_{i,j}} \frac{1}{h^2} \left\{ \frac{9}{8} \left[\frac{b_{i,j+1/2}}{(\xi_z)_{i,j+1/2}} \left(\frac{9}{8}(p_{i,j+1} - p_{i,j}) - \frac{1}{24}(p_{i,j+2} - p_{i,j-1}) \right) \right) \right. \right. \\
& - \frac{b_{i,j-1/2}}{(\xi_z)_{i,j-1/2}} \left(\frac{9}{8}(p_{i,j} - p_{i,j-1}) - \frac{1}{24}(p_{i,j+1} - p_{i,j-2}) \right) \Bigg] \\
& - \frac{1}{24} \left[\frac{b_{i,j+3/2}}{(\xi_z)_{i,j+3/2}} \left(\frac{9}{8}(p_{i,j+2} - p_{i,j+1}) - \frac{1}{24}(p_{i,j+3} - p_{i,j}) \right) \right. \\
& \left. \left. - \frac{b_{i,j-3/2}}{(\xi_z)_{i,j-3/2}} \left(\frac{9}{8}(p_{i,j-1} - p_{i,j-2}) - \frac{1}{24}(p_{i,j} - p_{i,j-3}) \right) \right] \right\} + s_{i,j}.
\end{aligned}
\tag{1.33}
$$

Equation (1.33) can be written in a compact form with respect to the different nodes (i, j) such that

$$
\begin{aligned}
C_1\, p_{i,j} + \quad & C_2\, p_{i-1,j} + C_3\, p_{i+1,j} + C_4\, p_{i,j+1} + C_5\, p_{i,j-1} + C_6\, p_{i-2,j} + C_7\, p_{i+2,j} \\
& + C_8\, p_{i,j+2} + C_9\, p_{i,j-2} + C_{10}\, p_{i-3,j} + C_{11}\, p_{i+3,j} + C_{12}\, p_{i,j+3} + C_{13}\, p_{i,j-3} = s_{i,j},
\end{aligned}
\tag{1.34}
$$

where the coefficients C_k for $k = 1, \cdots, 13$ are easily determined by developing equation (1.33). The corresponding fourth-order stencil is presented in Figure 1.5.

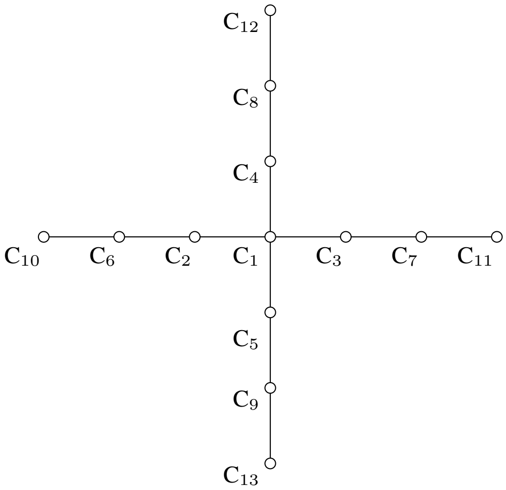


Figure 1.5: Fourth-order finite-difference stencil for the 2D frequency-domain acoustic wave equation.

By denoting N_x and N_z the number of discretization points along the horizontal and vertical directions, the following linear system of size $n = N_x N_z$ is obtained

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

with

$$A \in \mathcal{G}l_n(\mathbb{C}), \quad \mathbf{x} \in \mathbb{C}^n, \quad \mathbf{b} \in \mathbb{C}^n, \quad n \in \mathbb{N}$$

where A is a 13-diagonal complex large-scale sparse matrix. To obtain the minimum bandwidth, the following indexation method is used following the smallest dimension of the domain (the vertical direction in practice for seismic imaging applications)

$$k = (i-1)N_z + j, \quad i = 1, \ldots, N_x, \quad j = 1, \ldots, N_z.$$

The bandwidth of the matrix A is therefore equal to $6N_z + 6$. In seismic imaging applications, the ratio N_x/N_z is approximately up to ten. Therefore, choosing N_z for the indexation method allows to significantly reduce the bandwidth of the matrix A.

1.5.2 Fourth-order staggered-grid finite-difference scheme for the 2D elastic wave equations

Consider the 2D frequency-domain velocity-stress elastic system of equations for the heterogeneous isotropic media

$$\begin{cases} -i\omega\rho(x,z)\, v_x(x,z) = & (\partial_x \sigma_{xx}(x,z) + \partial_z \sigma_{xz}(x,z)) + f_x(x,z), \\ -i\omega\rho(x,z)\, v_z(x,z) = & (\partial_x \sigma_{xz}(x,z) + \partial_z \sigma_{zz}(x,z)) + f_z(x,z), \\ -i\omega\, \sigma_{xx}(x,z) & = (\lambda + 2\mu)(x,z)\, \partial_x v_x(x,z) + \lambda(x,z)\, \partial_z v_z(x,z) - i\omega s_1(x,z), \\ -i\omega\, \sigma_{zz}(x,z) & = (\lambda + 2\mu)(x,z)\, \partial_z v_z(x,z) + \lambda(x,z)\, \partial_x v_x(x,z) - i\omega s_2(x,z), \\ -i\omega\, \sigma_{xz}(x,z) & = \mu(x,z)\, (\partial_z v_x(x,z) + \partial_x v_z(x,z)). \end{cases} \tag{1.35}$$

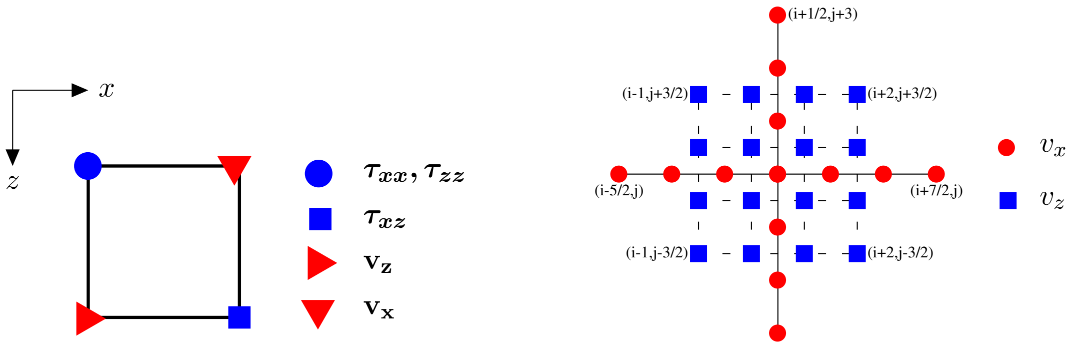
The particle velocities are denoted by v_x and v_z and the stress tensor components by σ_{xx}, σ_{zz} and σ_{xz}. The density is denoted by ρ and $\lambda,\ \mu$ are the Lamé parameters. (f_x, f_z) is the vector representing the external forces and (s_1, s_2) are the increments of the normal stresses which are related to the explosive source s.

The system of equations (1.35) is discretized using the fourth-order staggered-grid finite-difference method (Levander, 1988). The staggered-grid stencil for the elastodynamic system is illustrated in Figure 1.6a where the locations of the velocity terms and the the stress components are shown on the two grids shifted by the half space step. The fourth-order discretization scheme associated with this staggered-grid is presented in Figure 1.6b. The spatial derivatives are given by

$$\begin{aligned} \left[\frac{\partial v_x}{\partial x}\right]^h_{i,j} &= \frac{1}{h}\left[c_1\left((v_x)_{i+1/2,j} - (v_x)_{i-1/2,j}\right) + c_2\left((v_x)_{i+3/2,j} - (v_x)_{i-3/2,j}\right)\right], \\ \left[\frac{\partial v_z}{\partial z}\right]^h_{i,j} &= \frac{1}{h}\left[c_1\left((v_z)_{i,j+1/2} - (v_z)_{i,j-1/2}\right) + c_2\left((v_z)_{i,j+3/2} - (v_z)_{i,j-3/2}\right)\right], \end{aligned}$$

and

$$\begin{aligned} \left[\frac{\partial \sigma_{xx}}{\partial x}\right]^h_{i+1/2,j} &= \frac{1}{h}\left[c_1\left((\sigma_{xx})_{i+1,j} - (\sigma_{xx})_{i-1,j}\right) + c_2\left((\sigma_{xx})_{i+2,j} - (\sigma_{xx})_{i-2,j}\right)\right], \\ \left[\frac{\partial \sigma_{zz}}{\partial z}\right]^h_{i,j+1/2} &= \frac{1}{h}\left[c_1\left((\sigma_{zz})_{i,j+1} - (\sigma_{zz})_{i,j-1}\right) + c_2\left((\sigma_{zz})_{i,j+2} - (\sigma_{zz})_{i,j-2}\right)\right], \end{aligned}$$



(a) Staggered-grid stencil (b) Fourth-order scheme

Figure 1.6: 2D staggered-grid finite-difference stencil (a) and the 29-point fourth-order staggered-grid finite-difference scheme used to approximate the partial differential operators (b). In Figure a, blue marks represent stress positions and red marks represent velocity positions. The buoyancy is located on the same grid points as the particle velocity. The Lamé coefficient λ is located on the same grid as the normal stress components. The Lamé coefficient μ is located on the same grid as the stress component τ_{xz}.

where c_1 and c_2 are the fourth-order coefficients which are equal to $c_1 = 9/8$ and $c_2 = -1/24$ conventionally. The antilumped mass strategy is applied to decrease the numerical dispersion. Take for instance the horizontal particle velocity v_x. The mass term (ρv_x) is approximated by

$$\begin{aligned}
(\rho v_x)_{i+1/2,j} = a_1\ &(\rho v_x)_{i+1/2,j} \\
&+ a_2 \left[(\rho v_x)_{i+3/2,j} + (\rho v_x)_{i-1/2,j} + (\rho v_x)_{i+1/2,j+1} + (\rho v_x)_{i+1/2,j-1}\right] \\
&+ a_3 \left[(\rho v_x)_{i+5/2,j} + (\rho v_x)_{i-3/2,j} + (\rho v_x)_{i+1/2,j+2} + (\rho v_x)_{i+1/2,j-2}\right] \\
&+ a_4 \left[(\rho v_x)_{i+7/2,j} + (\rho v_x)_{i-5/2,j} + (\rho v_x)_{i+1/2,j+3} + (\rho v_x)_{i+1/2,j-3}\right],
\end{aligned}$$

where a_1, a_2, a_3 and a_4 are the weighting coefficients. An optimization problem is solved to compute the optimal coefficient values following the work of Li et al. (2016). These coefficients are used in the following chapter for our applications. They are presented in table (1.1) together with the conventional coefficient values.

	a_1	a_2	a_3	a_4	c_1	c_2
Conventional	1	0	0	0	$9/8$	$-1/24$
Optimized	0.81872	5.32025E − 2	−2.49E − 2	4.38444E − 3	1.12169	−4.92201E − 2

Table 1.1: Optimal coefficients for the 2D fourth-order staggered-grid finite-difference scheme.

Following the parsimonious approach proposed by Luo and Schuster (1990), the system of equations (1.35) is first discretized and the stress terms are eliminated to derive a finite-difference scheme associated with the fourth-order elastic wave equations based on the particle velocities only. This strategy gives more accurate finite-difference approximations of the partial derivative operator and less numerical dispersion. Using the optimal coefficients, the discretization of the models is performed using 5.4 grid points per minimum shear wavelength.

By denoting N_x and N_z the number of discretization points along the horizontal and vertical directions, the following linear system of size $n = 2N_xN_z$ is obtained

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

with

$$A \in \mathcal{G}l_n(\mathbb{C}), \quad \mathbf{x} \in \mathbb{C}^n, \quad \mathbf{b} \in \mathbb{C}^n, \quad n \in \mathbb{N}$$

where A is a 29-diagonal complex large-scale sparse matrix. To obtain the minimum bandwidth, the grid-points are ordered in the z, x manner, i.e.,

$$\underbrace{v_{1,1}, v_{1,2}, \cdots, v_{1,n_z}}, \underbrace{v_{2,1}, \cdots, v_{2,n_z}}, \cdots, \underbrace{v_{n_x,1}, \cdots, v_{n_x,n_z}},$$

where v is for instance the horizontal particle velocity field. The following indexation method is used following the smallest dimension of the domain (the vertical direction in practice for seismic imaging applications)

$$k = 2(i-1)N_z + 2(j-1) + 1, \quad i = 1, \ldots, N_x, \quad j = 1, \ldots, N_z.$$

The bandwidth of the matrix A is therefore equal to $12N_z + 14$. Therefore, choosing N_z instead of N_x reduces the bandwidth of the matrix A.

1.5.3 Fourth-order staggered-grid finite-difference scheme for the 3D elastic wave equations

Consider the 3D frequency-domain velocity-stress elastic system of equations for the heterogeneous isotropic media (1.14) which is reminded below

$$\left\{ \begin{array}{lcl} i\omega\rho(\mathbf{x})v_x & = & \partial_x\tau_{xx} + \partial_y\tau_{xy} + \partial_z\tau_{xz} + f_x, \\ i\omega\rho(\mathbf{x})v_y & = & \partial_x\tau_{xy} + \partial_y\tau_{yy} + \partial_z\tau_{yz} + f_y, \\ i\omega\rho(\mathbf{x})v_z & = & \partial_x\tau_{xz} + \partial_y\tau_{yz} + \partial_z\tau_{zz} + f_z, \\ i\omega\tau_{xx} & = & \nu(\mathbf{x})\partial_x v_x + \lambda(\mathbf{x})\partial_y v_y + \lambda(\mathbf{x})\partial_z v_z + i\omega s_1, \\ i\omega\tau_{yy} & = & \lambda(\mathbf{x})\partial_x v_x + \nu(\mathbf{x})\partial_y v_y + \lambda(\mathbf{x})\partial_z v_z + i\omega s_2, \\ i\omega\tau_{zz} & = & \lambda(\mathbf{x})\partial_x v_x + \lambda(\mathbf{x})\partial_y v_y + \nu(\mathbf{x})\partial_z v_z + i\omega s_3, \\ i\omega\tau_{yz} & = & \mu(\mathbf{x})(\partial_y v_z + \partial_z v_y), \\ i\omega\tau_{xz} & = & \mu(\mathbf{x})(\partial_x v_z + \partial_z v_x), \\ i\omega\tau_{xy} & = & \mu(\mathbf{x})(\partial_x v_y + \partial_y v_x), \end{array} \right. \tag{1.36}$$

where the vector $\mathbf{x} = (x, y, z)$ denotes the space position and space-dependency of the model parameters, the particle velocity components are denoted by v_x, v_y and v_z and the stress tensor components by σ_{xx}, σ_{yy} σ_{zz}, σ_{yz}, σ_{xz} and σ_{xy}. The density is denoted by ρ and λ, μ are the Lamé parameters. The P-wave modulus $\nu = \lambda + 2\mu$ is introduced. The vector (f_x, f_y, f_z) represents the external forces and (s_1, s_2, s_3) are the increments of the normal stresses which are related to the explosive source.

Perfectly Matched Layer (Bérenger, 1994) are used at the boundaries of the 3D domain. Following the work of Li et al. (2016), the fourth-order staggered-grid finite-difference scheme (Levander, 1988) is used to discretize the system of equations (1.36) with the PMLs. The staggered-grid stencil is presented in Figure 1.7(a).

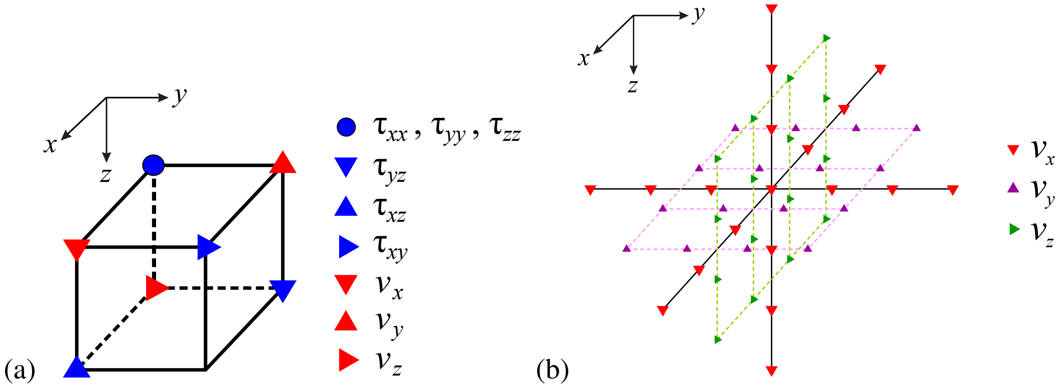


Figure 1.7: 3D staggered-grid finite-difference stencil (a). The normal stress components and the Lamé parameters $\lambda,\ \lambda+2\mu$ are defined on the reference grid (i,j,k). The particle velocity components are defined on the staggered-grid. The density and the Lamé parameter μ are interpolated on the staggered-grid. Grid points involved with the mass acceleration $(\rho v_x)_{i+1/2,j,k}$ averaging (b). The particle velocity component $(v_x)_{i+1/2,j,k}$ is located at the center.

The finite-difference approximation of the spatial derivatives (take the first x-derivative for instance) is given by

$$\left[\frac{\partial v_x}{\partial x}\right]^{\Delta}_{i,j,k} = \frac{1}{\Delta}\left[c_1\left((v_x)_{i+1/2,j,k} - (v_x)_{i-1/2,j,k}\right) + c_2\left((v_x)_{i+3/2,j,k} - (v_x)_{i-3/2,j,k}\right)\right]. \tag{1.37}$$

To decrease the numerical dispersion and the required number of grid points per wavelength, an antilumped mass strategy is used. The mass term (take for instance ρv_x) is distributed over the 19 grid points involved in the stencil (see Figure 1.7(b)). The mass acceleration term is approximated as

$$\begin{aligned}(\rho v_x)_{i+1/2,j,k} = {}& a_1\,(\rho v_x)_{i+1/2,j,k}\\ &+ a_2\,\big[(\rho v_x)_{i+3/2,j,k} + (\rho v_x)_{i-1/2,j,k} + (\rho v_x)_{i+1/2,j+1,k}\\ &\quad + (\rho v_x)_{i+1/2,j-1,k} + (\rho v_x)_{i+1/2,j,k+1} + (\rho v_x)_{i+1/2,j,k-1}\big]\\ &+ a_3\,\big[(\rho v_x)_{i+5/2,j,k} + (\rho v_x)_{i-3/2,j,k} + (\rho v_x)_{i+1/2,j+2,k}\\ &\quad + (\rho v_x)_{i+1/2,j-2,k} + (\rho v_x)_{i+1/2,j,k+2} + (\rho v_x)_{i+1/2,j,k-2}\big]\\ &+ a_4\,\big[(\rho v_x)_{i+7/2,j,k} + (\rho v_x)_{i-5/2,j,k} + (\rho v_x)_{i+1/2,j+3,k}\\ &\quad + (\rho v_x)_{i+1/2,j-3,k} + (\rho v_x)_{i+1/2,j,k+3} + (\rho v_x)_{i+1/2,j,k-3}\big],\end{aligned} \tag{1.38}$$

where a_1, a_2, a_3 and a_4 are the weighting coefficients which are determined by solving an optimization problem that minimizes the misfit between the normalized phase velocities and the unity. The resulting coefficients are presented in table (1.2) together with the standard fourth-order coefficients.

Following the parsimonious approach (Luo and Schuster, 1990), the system of equations (1.36) with PMLs is first discretized using (1.37) and (1.38), then the stress tensor components are eliminated by substitution from this discretized system. There only remains the three discretized equations involving the particle velocity components v_x, v_y and v_z which can be written in matrix form as

$$A\mathbf{x} = \mathbf{b},$$

	a_1	a_2	a_3	a_4	c_1	c_2
Conventional	1	0	0	0	9/8	-1/24
Optimized	0.926585	6.99990E-2	-3.28692E-2	6.28252E-3	1.25274	-5.45995E-2

Table 1.2: 3D Optimal coefficients for the fourth-order staggered-grid finite-difference stencil. The coefficients are obtained by minimizing the misfit between the normalized phase velocity and unity for the 3D fourth-order staggered-grid finite-difference stencil. The standard fourth-order coefficients are presented in the first row of the table as well.

where A is the so-called impedance matrix, $\mathbf{x}$ is the particle velocity vector and $\mathbf{b}$ is the source vector. The grid points are ordered in the z, y, x order (from the smallest to the largest dimension of the model which is usually in seismic applications z, y, x) such that

$$\mathbf{x} = \left[\underbrace{v_{1,1,1}, v_{1,1,2}, \cdots, v_{1,1,n_z}}, \underbrace{v_{1,2,1}, \cdots, v_{1,2,n_z}}, \cdots, \underbrace{v_{1,n_y,1}, \cdots, v_{1,n_y,n_z}}, \cdots, \underbrace{v_{n_x,n_y,1}, \cdots, v_{n_x,n_y,n_z}} \right],$$

where n_x, n_y and n_z are the number of grid points per dimension. This ordering corresponds to the following indexation method

$$k_x = 3(i_3 - 1)n_y n_z + 3(i_2 - 1)n_z + 3(i_1 - 1) + 1,$$

$$k_y = 3(i_3 - 1)n_y n_z + 3(i_2 - 1)n_z + 3(i_1 - 1) + 2,$$

$$k_z = 3(i_3 - 1)n_y n_z + 3(i_2 - 1)n_z + 3(i_1 - 1) + 3,$$

where k_x is used to access v_x, k_y is used to access v_y and k_z is used to access v_z for $i_1 = 1, \ldots, n_z$, $i_2 = 1, \ldots, n_y$ and $i_3 = 1, \ldots, n_x$. The bandwidth of the matrix A is therefore minimal and is given by $18 n_y n_z + 1$.

The combination of these three strategies (staggered-grid, parsimonious approach and second/fourth-order discretization stencil) provides a discretization scheme adapted to the elastodynamics equations. The formulation into higher order discretization is straightforward. From an implementation point of view, the resulting computational domain is a two dimensional rectangular grid or a three dimensional rectangular cuboid grid where the core of the computation is composed of the interactions of each grid with its immediate neighbors. Such meshes should be partitioned among the processors with the main objectives of minimal communication, understanding the minimum total communication volume, and equal load balance, i.e., at best equal number of grid points per processor. The mesh partitioning problem can be formulated as a standard graph partitioning problem where the mesh points are represented by the vertices of the graph and the edges of the graph correspond to an interaction between the mesh points. There are a number of software libraries (Karypis and Kumar, 1999; Boman et al., 2012) implementing graph partitioning methods where the objective is to minimize the number of interacting pairs of mesh points. Using the finite-difference discretization method, meshes are efficiently obtained using a Cartesian partitioning. As the interior mesh points require an equal number of interactions, load balance can be achieved by partitioning equally the number of mesh points among the processors. Assuming a 2D geometry for instance, a mesh of size $N_x \times N_z$ is partitioned among the $p \times q$ processors by equally diving the x and z coordinates by, respectively $p - 1$ and $q - 1$. Thus the finite-difference method is naturally well designed for parallel computing without paying the extra cost of graph partitioning methods.

1.6 Frequency-domain wave modeling : three strategies

The computation of the solution of the wave propagation problem in the frequency domain can be performed following three main approaches. The first strategy addresses the problem in the time domain and extracts the needed frequencies during the simulation using a Discrete Fourier Transform (DFT). The second and third approaches consider the problem formulated in the frequency domain. The linear system resulting from the discretization of the latter equations can be solved either with a direct method or an iterative method.

In the following, these three approaches are discussed. We shall give a special interest to the computational complexities and memory requirements. For the sake of simplicity, we consider a 3D problem of size N^3 with N the average number of discretization points in one dimension. The number of sources is denoted by N_s which is in the order of N^2 for 3D FWI applications considering a dense surface acquisition framework. The number of time steps for time-domain wave modeling is denoted by N_t. Finally the number of iterations performed by an iterative method to solve the frequency-domain wave equation is denoted by N_{iter}. Such comparisons have already been addressed by Plessix (2006b) and Pajot et al. (2014).

1.6.1 Time-domain wave modeling with DFT

The time-domain approach was promoted by Sirgue et al. (2008). A second-order leap-frog scheme for the time derivative and a 4^{th} or 8^{th} order staggered-grid finite-difference scheme for the spatial discretization are commonly used. The frequencies are extracted on the fly using the DFT in the loop over the time.

The evaluation of the wavefield for one time step has a complexity of $\mathcal{O}(N^3)$: updating the wavefield at a given time step with a leap-frog scheme requires to sweep the entire domain. Front tracking technique can mitigate this cost, but this yields only marginal improvement we do not consider here. As the number of time steps N_t is constrained by the CFL condition (Courant et al., 1928) and the recording time which depends on the size of the problem, a fair approximation of the complexity is $N_t = \mathcal{O}(N)$. Note that the recording time should be long enough to ensure the DFT has accurately converged. Therefore, the computational complexity of the time-domain wave modeling for one source has a complexity of $\mathcal{O}(N^4)$.

Finally, the computational complexity of the time-domain approach increases linearly with the number of sources. Therefore, this linear approach gives an overall computational cost in the order of $\mathcal{O}(N_s N_t N^3) = \mathcal{O}(N^6)$. The memory requirements of the time-domain are in the order of $\mathcal{O}(N^3)$ for one source modeling.

In 2D FWI applications, the discrete Fourier transform computes on the fly the frequency wavefield by discrete integration with an affordable memory and computational cost. In the 3D, It is computed as well on the fly but the cost cannot be ignored totally anymore. This step has a complexity of $\mathcal{O}(N^4)$ and is roughly as expensive as one time-domain wave modeling. The reader may refer to Brossier et al. (2014a) for a further analysis on the real computational complexities of the different steps required for time-domain FWI computations.

1.6.2 Frequency-domain wave modeling using a direct method

The second approach consists in considering the frequency-domain wave equation. The discretization of this equation is performed using a compact finite-difference stencil described in the previous section. This approach gives a large, sparse, banded and ill-conditioned linear system

$$A(\mathbf{x},\omega)\mathbf{u}(\mathbf{x},\omega) = \mathbf{b}(\mathbf{x},\omega), \tag{1.39}$$

Direct methods are robust and allow the computation of accurate up to machine precision solutions. Standard direct methods are based on a lower-upper (LU) factorization of the matrix $A = LU$. Once the factorization is performed, the solution is calculated by forward substitution $L\mathbf{y} = \mathbf{b}$ and backward substitution $U\mathbf{u} = \mathbf{y}$. The LU factorization of the matrix A produces dense triangular matrices L and U due to the fill-in effect (non-zero coefficients in the bandwidth are added), making the LU factorization memory demanding. This is the main reason why direct solvers are not used for large size problems. The computational cost of the LU factorization step is in $\mathcal{O}(N^6)$ whereas the cost of the forward/backward substitution operations is in $\mathcal{O}(N^4)$ for the solution of one right hand side. Therefore, considering the solution of the N^2 sources has a computational complexity of $\mathcal{O}(N^6)$. The memory complexity is in $\mathcal{O}(N^4)$. This is due to the storage of the LU factors and it is the main limitations for these methods.

For practical applications, a multifrontal parallel direct solver can be used. The multifrontal solver has been introduced by Duff and Reid (1983). This method allows to use several independent fronts during the LU factorization at the same time. The multifrontal technique is the key ingredient for enabling parallel computing when using direct solvers. Amestoy et al. (2015a) and Weisbecker et al. (2013) combine this algorithm with a block-low rank representation and a nested-dissection algorithm (George and Liu, 1981). These strategies allow to reduce the computational cost and the memory demand due to the storage of the dense frontal matrices by minimizing the fill-in during the factorization. This direct solver is used for the solution of the 3D visco-acoustic frequency-domain wave problem in vertical transversely isotropic media by Operto et al. (2014). This modeling engine is later used for the solution of the forward problem in the 3D acoustic FWI performed by Operto et al. (2015) on the Valhall real case study. The model covers a large surface of 145 km^2 and is 4 km deep. The frequencies modeled reach 10 Hz yielding a linear system of size 17 million unknowns with more than 4000 right hand sides. Another approach is promoted by Wang et al. (2012) where the hierarchically semi-separable (HSS) structures are used to approximate the dense frontal matrices with block compressed forms.

1.6.3 Frequency-domain wave modeling using an iterative method

Iterative methods based on Krylov subspace for the solution of non hermitian linear systems provide an alternative to direct methods for the solution of the frequency-domain wave equations. These methods are considered as direct as well as they offer the opportunity to compute exact solution up to machine precision, however approximate solutions can be computed by limiting the number iterations. Given an initial iterate $\mathbf{u}^{(0)}$, an approximate solution is computed by iteratively updating the iterate $\mathbf{u}^{(i)}$. The method stops once the solution reaches a satisfying accuracy. Each iteration performs only basic linear algebra operations (sparse matrix vector products, scalar products and vector updates). These methods have a computational complexity in $\mathcal{O}(N_{iter}N^3)$ for the solution of one right-hand side. Thus for a number of sources in the order of N^2 the overall computational cost of the iterative method is in $\mathcal{O}(N_{iter}N^5)$. These methods require the storage of few vectors of size N^3 for the computation of the solution. An exception is made of the iterative solver GMRES (Saad, 1986) where the Krylov space of size mN^3 needs to be allocated with m an integer ranging from few tens to few hundreds. Thus, the memory demand is in the order of $\mathcal{O}(N^3)$. Iterative methods often require the use of preconditioners. The latter are used to ensure the fast convergence. Therefore, if the number of iterations is reduced to the order of N or lower, iterative methods become competitive with the time-domain approach (Table 1.3).

Most of the recent developments on preconditioning methods allowed to reduce the number of iterations to $N_{iter} = \mathcal{O}(N)$ which makes the overall computational cost in $\mathcal{O}(N^6)$. More importantly, multiple right-hand side accelerations may be combined with iterative solvers. These methods mainly rely on the block conjugate gradient (CG) method (O'Leary, 1980) and projection methods introduced by Smith et al. (1989) and Chan and Ng (1999). An application to the 3D frequency-domain seismic inversion is performed in the acoustic approximation using an Krylov based iterative solver combined with the block CG method in (van Leeuwen and Herrmann, 2014) giving significant reduction in the computational cost for the solution of numerous right-hand sides.

	Computational complexity	Memory complexity
Time-domain + DFT	$\mathcal{O}(N_\text{s} \times N_\text{t} \times N^3) = \mathcal{O}(N^6)$	$\mathcal{O}(N^3)$
Freq-domain + Direct	$\mathcal{O}(N^6)$	$\mathcal{O}(N^4)$
Freq-domain + Iterative	$\mathcal{O}(N_\text{iter} \times N_\text{s} \times N^3) = \mathcal{O}(N_\text{iter} \times N^5)$	$\mathcal{O}(N^3)$

Table 1.3: Theoretical computational and memory complexities of the three modeling approaches in the frequency-domain for a 3D problem of size N^3.

Discussion

Current FWI studies demonstrate the ability of using direct solvers in the 3D acoustic approximation for reasonable target size (Operto et al., 2015) solving up to 17 million unknowns linear system with several thousands of right-hand sides. However, the high memory requirements of these methods, due to the fill-in associated with the matrix factorization, put to the test today's computing architectures when facing larger size problems associated with realistic size 3D elastic simulations. Pioneering work by Wang et al. (2012) shows that a hierarchical semi-separable (HSS) low rank matrix can be used to reduce the memory and computation costs. However, the size of the application they present is still modest (201x201x151 grid points). Challenging large scale 3D elastodynamics problems involving larger grid size seem to be still out of reach for direct solvers.

In the perspective of performing 3D frequency-domain elastodynamics FWI, two strategies could thus be adopted. The first consists in solving the elastic wave equations in the time domain together with the DFT approach. This strategy is appealing because of its straightforward implementation, however, three limitations can be identified. First, no Multiple Right-Hand Sides (MRHS) acceleration can be implemented. Second, the integration time is controlled by a CFL condition depending on the highest P-wave-velocity and the smallest discretization step which is governed by the smallest wavelength. Finally, the integration time should be long enough for the DFT to be sufficiently accurate. For elastodynamics simulations in media with low S-wave velocity, this could lead to severe restrictions on the time step. A drawback of time-domain methods is the more difficult and computationally expensive implementation of attenuation effects.

These limitations lead us to rather investigate the possibility of solving the elastodynamics equations using iterative solvers. These algorithms fully benefit from the sparsity of the linear system as they only require the computation of scalar products and sparse matrix vector products, and therefore present significantly lower memory requirements in comparison with direct solvers as it has been shown earlier. However, these methods require efficient preconditioning strategies to ensure the convergence. In the acoustic approximation, the preconditioner based on the shifted Laplacian operator has been successfully applied with the BiConjugate Gradient Stabilized method (BiCGStab) and the Generalized Minimal Residual (GMRES) method together with a multigrid approach for the solution of the 2D and 3D acoustic wave equation in (Erlangga and Nabben, 2008c) and as a forward problem solver in the 2D acoustic FWI by Plessix and Perkins (2009). More recently, studies and applications in the elastic approximation are attempted. Rizzuti and Mulder (2016) investigated theoretically the 2D frequency-domain elastic wave equations together with the shifted Laplace preconditioner on the elastic Marmousi 2 model. A 3D FWI application in the elastic approximation is performed by Pan et al. (2015) using BiCGStab and the shifted Laplace preconditioner. A problem of the size of a couple hundred meters in each dimension is studied. These applications show mainly the feasibility of the method. Realistic size problems are still a challenge.

In the following section, I will focus my investigations on iterative methods which are at the center of this thesis. I shall review the classic iterative methods used for the solution of the frequency-domain wave equations. This will inevitably lead us to investigate preconditioning strategies.

1.7 Iterative methods and preconditioning for the solution of the frequency-domain wave equation

Iterative methods refer to a wide class of techniques that use a set of successive approximations to compute accurate solutions of a linear system at each iteration. The range of iterative methods we are particularly interested in this investigation are the methods based on Krylov subspace and designed for non-hermitian linear systems. Most of these methods are considered as direct in the sense that they allow the computation of exact solutions up to machine precision when the number of iterations is equal to the size of the problem. However, approximate solutions can be computed by using a convergence threshold and thus limiting the number of iterations.

Consider the linear system

$$A\mathbf{x} = \mathbf{b}, \tag{1.40}$$

where $A \in \mathcal{G}l_n(\mathbb{K})$ is an invertible matrix of size n with coefficients in $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$ and $\mathbf{x}$ and $\mathbf{b}$ are two vectors in $\mathbb{K}^n$ denoting respectively the solution and the right hand side vectors.

The idea behind Krylov subspace iterative methods is to start from an initial guess $\mathbf{x}^{(0)}$ and to generate a sequence of approximate solutions, denoted by $\mathbf{x}^{(m)}$, which converges to the exact solution such that

$$\mathbf{x}^{(m)} \in \mathbf{x}^{(0)} + \mathcal{K}_m(A, \mathbf{r}^{(0)}).$$

The vector $\mathbf{r}^{(m)}$ is called the residual vector and is equal to $\mathbf{r}^{(m)} = \mathbf{b} - A\mathbf{x}^{(m)}$ and $\mathcal{K}_m(A, \mathbf{r}^{(0)})$ is defined as the Krylov subspace which is generated by A and $\mathbf{r}^{(0)}$

$$\mathcal{K}_m(A, \mathbf{r}^{(0)}) = \text{span}\left\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, \ldots, A^{m-1}\mathbf{r}^{(0)}\right\}.$$

The Conjugate Gradient (CG) method (Hestenes and Stiefel, 1952) is the archetype of Krylov subspace based methods. Even though this method is used for the solution of hermitian positive definite linear systems, it is the starting point of most Krylov based iterative method and therefore it is included in widely used iterative methods presented below. In the following, we review the standard iterative methods for the solution of, first, hermitian positive definite linear systems and second non-hermitian positive semi-definite linear systems.

1.7.1 Iterative methods for the solution of linear systems

CG for the solution of hermitian positive definite linear systems

The CG method (Hestenes and Stiefel, 1952) is used for the solution of linear systems where the matrix A is hermitian positive definite. The problem (1.40) can be stated equivalently as the minimization problem of the quadratic function (see Nocedal and Wright (2006) for instance)

$$\min_{\mathbf{x}\in\mathbb{K}^n} \phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^H A\mathbf{x} - \mathbf{b}^H\mathbf{x}, \tag{1.41}$$

for which the minimum is reached when

$$\nabla\phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = \mathbf{0}$$

Therefore, problems (1.40) and (1.41) have the same solution. The quadratic function ϕ can be minimized in n steps by successively minimizing it along the directions $\mathbf{p}^{(i)}$ which are conjugate with respect to the hermitian positive definite matrix A

$$(\mathbf{p}^{(i)}, \mathbf{p}^{(j)})_A = \mathbf{p}^{(i)^H} A\mathbf{p}^{(j)} = 0, \quad \text{for all } i \neq j. \tag{1.42}$$

Given a starting point $\mathbf{x}^{(0)}$ and a set of A-conjugate directions $\{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(i)}\}$, the solution vector $\mathbf{x}^{(i+1)}$ is generated through

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)},$$

and the residual $\mathbf{r}^{(i+1)}$ is computed using

$$\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i A\mathbf{p}^{(i)}, \tag{1.43}$$

where α_i is the scalar minimizer of the quadratic function ϕ given by

$$\alpha_i = \frac{(\mathbf{r}^{(i)}, \mathbf{r}^{(i)})}{(\mathbf{p}^{(i)}, \mathbf{p}^{(i)})_A}.$$

In the conjugate gradient method, two successive search directions are required and each direction $\mathbf{p}^{(i)}$ is chosen to be a linear combination of the residual $\mathbf{r}^{(i)}$ and the previous direction $\mathbf{p}^{(i-1)}$

$$\mathbf{p}^{(i)} = \mathbf{r}^{(i)} + \beta_{i-1}\mathbf{p}^{(i-1)}, \tag{1.44}$$

where the scalar β_i is determined by the A-conjugacy condition of the directions vectors $\mathbf{p}^{(i-1)}$ and $\mathbf{p}^{(i)}$. Therefore, the following expression of β_i is obtained

$$\beta_i = \frac{(\mathbf{r}^{(i)}, \mathbf{p}^{(i-1)})_A}{(\mathbf{p}^{(i-1)}, \mathbf{p}^{(i-1)})_A}.$$

From (1.43), we have that $-\alpha_i A\mathbf{p}^{(i)} = \mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}$ and using (1.42) and (1.44) one can simplify the expression of β_i as

$$\beta_i = \frac{(\mathbf{r}^{(i)}, \mathbf{r}^{(i)})}{(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i-1)})}.$$

The CG algorithm can be summarized as follows (Algorithm 1.1)

Algorithm 1.1 Conjugate Gradient method (CG)

```
Given an initial guess $\mathbf{x}^{(0)}$,
Compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$,
Set $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$,
for $i = 0, \dots$ until convergence do
    $\alpha_i = (\mathbf{r}^{(i)}, \mathbf{r}^{(i)})/(A\mathbf{p}^{(i)}, \mathbf{p}^{(i)})$,
    $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)}$,
    $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i A\mathbf{p}^{(i)}$,
    $\beta_i = (\mathbf{r}^{(i+1)}, \mathbf{r}^{(i+1)})/(\mathbf{r}^{(i)}, \mathbf{r}^{(i)})$,
    $\mathbf{p}^{(i+1)} = \mathbf{r}^{(i+1)} + \beta_i \mathbf{p}^{(i)}$,
endfor
```

Computational and memory costs The CG method as presented in Algorithm 1.1 uses two coupled term recurrences: one for the residual update using a search direction vector and one updating the search director with newly computed residual and the solution vector naturally. It has the nice property of requiring one matrix vector product, two inner products and three vector updates per iteration.

Convergence Defining the condition number of a matrix $A \in \mathcal{G}l_n(\mathbb{K})$ by

$$\mathtt{cond}(A) = \|A\|_2 \|A^{-1}\|_2,$$

or in the case where the matrix A is hermitian, through the eigenvalues of the matrix A, $|\lambda_{min}| = |\lambda_1| \leqslant \cdots \leqslant |\lambda_n| = |\lambda_{max}|$

$$\mathtt{cond}(A) = \frac{|\lambda_{max}|}{|\lambda_{min}|},$$

it can be shown that at the i^{th} iteration of the conjugate gradient method, the error of the approximate solution can be bounded by

$$\|\mathbf{x}^{(i)} - \mathbf{x}^\star\|_A \leq 2 \left(\frac{\sqrt{\mathtt{cond}(A)} - 1}{\sqrt{\mathtt{cond}(A)} + 1} \right)^i \|\mathbf{x}^{(0)} - \mathbf{x}^\star\|_A.$$

where $\mathbf{x}^\star$ is the exact solution of the linear system (1.40). Therefore, the number of iterations to reach a relative reduction of ϵ in the error $\|\mathbf{x}^{(i)} - \mathbf{x}^\star\|$ is proportional to $\sqrt{\mathtt{cond}(A)}$ (Golub, 1996). Thus the CG method converges faster when the matrix A has a small condition number.

Iterative methods for the solution of non-hermitian linear systems

When the matrix A is no longer hermitian, one needs to turn to other iterative methods for the solution of these systems. In the following, we are interested in the CG method applied to the normal equations (CGNR) (Saad, 2003), the Generalized Minimal Residual (GMRES) method (Saad, 1986) and the Bi-Conjugate Gradient (Bi-CGSTAB) method (Van der Vorst, 1992).

CG on the Normal equations: CGNR, CGNE

When the matrix A is not hermitian, one can solve the so-called normal equations system defined by

$$A^H A\mathbf{x} = A^H\mathbf{b}, \tag{1.45}$$

using the conjugate gradient method, where the matrix $A^H A$ is hermitian

$$(A^H A)^H = A^H A, \tag{1.46}$$

positive definite as for $\mathbf{u} \neq 0$ and assuming $A \in \mathcal{G}l_n(\mathbb{C})$

$$\mathbf{u}^H A^H A\mathbf{u} = (A\mathbf{u}, A\mathbf{u}) = \|A\mathbf{u}\|^2 > 0. \tag{1.47}$$

Alternatively, one can solve the following system by setting $\mathbf{x} = A^H\mathbf{u}$

$$AA^H\mathbf{u} = \mathbf{b}. \tag{1.48}$$

The solution of the system (1.45) with the CG method is then labeled CGNR for Normal Residual, while the solution of (1.48) is labeled CGNE for Normal Error (Saad, 2003). In the following, the properties of the CGNR method are discussed as the CGNE method is very similar. Algorithm 1.1 applied on the normal equations (1.45) gives the CGNR in Algorithm 1.2.

Algorithm 1.2 Conjugate Gradient method for Normal Residuals (CGNR)

```
1:  Given an initial guess $\mathbf{x}^{(0)}$,
2:  Compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$,
3:  Compute $\mathbf{z}^{(0)} = A^H\mathbf{r}^{(0)}$,
4:  Set $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$,
5:  for $i = 0, \dots$ until convergence do
6:      $\alpha_i = (\mathbf{z}^{(i)}, \mathbf{z}^{(i)})/(A\mathbf{p}^{(i)}, A\mathbf{p}^{(i)})$,
7:      $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha_i\mathbf{p}^{(i)}$,
8:      $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha_i A\mathbf{p}^{(i)}$,
9:      $\mathbf{z}^{(i+1)} = A^H\mathbf{r}^{(i+1)}$,
10:     $\beta_i = (\mathbf{z}^{(i+1)}, \mathbf{z}^{(i+1)})/(\mathbf{z}^{(i)}, \mathbf{z}^{(i)})$,
11:     $\mathbf{p}^{(i+1)} = \mathbf{z}^{(i+1)} + \beta_i\mathbf{p}^{(i)}$,
12: endfor
```

Computational and memory costs The CGNR method requires the storage of 5 vectors as one more vector is required in addition to the CG algorithm. Two matrix vector products are required where one involves the conjugate transpose matrix vector product as shown at lines 3 and 9 of Algorithm 1.2.

Convergence Solving the normal equations using the CG method can be an inefficient approach when the matrix A is poorly conditioned $\texttt{cond}(A) >> 1$. Indeed, the condition number of the matrix $A^H A$ is

$$\texttt{cond}(A^H A) = \texttt{cond}(A)^2.$$

Therefore, the solution of non-hermitian linear system should preferably be computed using other iterative methods when the matrix A is poorly conditioned.

The Generalized Minimal Residual GMRES method

The GMRES method was proposed by Saad (1986) for the solution of large, sparse and non hermitian linear systems. A set of convergent solutions $\{\mathbf{x}^{(m)}\}$ are calculated as

$$\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \boldsymbol{\delta},$$

where $\mathbf{x}^{(0)}$ is an initial guess and δ is a vector in $\mathcal{K}_m(A, \mathbf{r}^{(0)})$. The GMRES method transforms the Krylov subspace $\mathcal{K}_m(A, \mathbf{r}^{(0)})$ into an orthonormal basis $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(m)}\}$

$$\mathcal{K}_m(A, \mathbf{v}^{(1)}) = \text{span}\{\mathbf{v}^{(1)}, A\mathbf{v}^{(1)}, \ldots, A^{m-1}\mathbf{v}^{(1)}\},$$

with $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\|\mathbf{r}^{(0)}\|_2$ using the modified Arnoldi-Gram-Schmidt process of orthogonalization (Algorithm 1.3).

Algorithm 1.3 Modified Arnoldi-Gram-Schmidt process

```
$\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\|\mathbf{r}^{(0)}\|_2$;
for $i = 1, \ldots, m$ do
    $\mathbf{w} = A\mathbf{v}^{(i)}$;
    for $k = 1, \ldots, i$ do
        $h_{k,i} = (\mathbf{w}, \mathbf{v}^{(i)}$;
        $\mathbf{w} = \mathbf{w} - h_{k,i}\mathbf{v}^{(k)}$;
    endfor
    $h_{i+1,i} = \|\mathbf{w}\|_2$;
    $\mathbf{v}^{(i+1)} = \mathbf{w}/h_{i+1,i}$;
endfor
```

The modified Arnoldi-Gram-Schmidt process allows to compute the upper Hessenberg matrix $H_m = (h_{i,j})$ of size $m \times m$. The coefficients $h_{i,j}$ are computed in Algorithm 1.3. Defining the matrix V_m by

$$V_m = \begin{bmatrix} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \cdots & \mathbf{v}^{(m)} \end{bmatrix},$$

this algorithm can be written in matrix form which leads to the following relation

$$AV_m = V_m H_m + h_{m+1,m}\mathbf{v}^{(m+1)}\mathbf{e}_m^T,$$

where $\mathbf{e}_m$ is the unitary vector of which the m^{th} component is equal to 1, or

$$AV_m = V_{m+1}\bar{H}_m,$$

where the matrix $\bar{H}_m$ is defined as the matrix H_m but with an additional line with one non-zero coefficient $h_{m+1,m}$

$$\bar{H}_m = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \dots & h_{2,m-1} & h_{2,m} \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \dots & 0 & h_{m+1,m} \end{pmatrix}.$$

The following relation can be deduced

$$V_m^H A V_m = H_m$$

The correction vector $\boldsymbol{\delta} \in \mathcal{K}_m(A, \mathbf{r}^{(0)})$ can therefore be written as

$$\boldsymbol{\delta} = V_m \mathbf{y},$$

where $\mathbf{y}$ is a vector of size m. Therefore the approximate solution is computed in the form

$$\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + V_m \mathbf{y}.$$

Thus, the GMRES method aims at building the solution $\mathbf{x}^{(m)}$ which minimizes the l_2-norm of the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$. At the iteration m, the minimization problem can be formulated as

Find $\mathbf{x}^{(m)} \in \mathcal{K}_m(A, \mathbf{r}^{(0)})$ *such that*

$$\min_{\mathbf{x} \in \mathbf{x}^{(0)} + \mathcal{K}_m(A, \mathbf{r}^{(0)})} \|\mathbf{b} - A\mathbf{x}\|_2 = \|\mathbf{b} - A\mathbf{x}^{(m)})\|_2. \tag{1.49}$$

Therefore, we have

$$\begin{aligned} \mathbf{r}^{(m)} &= \mathbf{b} - A\mathbf{x}^{(m)} \\ &= \mathbf{b} - A(\mathbf{x}^{(0)} + V_m \mathbf{y}) \\ &= (\mathbf{b} - A\mathbf{x}^{(0)}) - AV_m \mathbf{y} \\ &= \mathbf{r}^{(0)} - AV_m \mathbf{y}, \end{aligned}$$

which implies that

$$\begin{aligned} \mathbf{r}^{(m)} &= \beta \mathbf{v}^{(1)} - V_{m+1} \bar{H}_m \mathbf{y} \\ &= V_{m+1}(\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}), \end{aligned}$$

where $\beta = \|\mathbf{r}^{(0)}\|_2$. As the matrix V_{m+1} is orthonormal , then

$$\|V_{m+1}(\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y})\|_2 = \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2,$$

Therefore, the minimization problem (1.49) can be written as

$$\min_{\mathbf{y} \in \mathbb{K}^m} \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2 = \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}^{(m)}\|_2.$$

The computation of $\mathbf{y}^{(m)}$ amounts to solve a least square problem of size $(m+1) \times m$. The solution at iteration m is reconstructed by

$$\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + V_m \mathbf{y}^{(m)}.$$

The GMRES method is summarized in Algorithm 1.4 in the restarted version.

Remark 1.1. *"Full" GMRES is equivalent to the restarted version when $m = n$. Because the "full" GMRES method may require a large number of iterations to reach convergence and therefore a prohibitive amount of memory, the restarted GMRES method is used in practice to limit the memory cost of the method in particular for large size linear systems. The size m of the Krylov subspace is however set empirically. Restarted GMRES often leads to slow convergence or even stagnation. This poor convergence properties is caused by the loss of information when the iterations are restarted from scratch.*

Algorithm 1.4 Generalized Minimal Residual GMRES method (restarted version)

```
1:  Given an initial guess $\mathbf{x}^{(0)}$,
2:  Compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$,
3:  Compute $\beta = \|\mathbf{r}^{(0)}\|_2$,
4:  Compute $\mathbf{v}^{(1)} = \mathbf{r}^{(0)}/\beta$,
5:  for $i = 1, \ldots, m$ do
6:      $\boldsymbol{\omega}^{(i)} = A\mathbf{v}^{(i)}$,
7:      for $j = 1, \ldots, i$ do
8:          $h_{j,i} = (\boldsymbol{\omega}^{(i)}, \mathbf{v}^{(j)})$,
9:          $\boldsymbol{\omega}^{(i)} = \boldsymbol{\omega}^{(i)} - h_{j,i}\mathbf{v}^{(j)}$,
10:     endfor
11:     $h_{i+1,i} = \|\boldsymbol{\omega}^{(i)}\|_2$,
12:     $\mathbf{v}^{(i)} = \boldsymbol{\omega}^{(i)}/h_{i+1,i}$,
13: endfor
14: Compute $\mathbf{y}^{(m)}$ the minimizer of $\|\beta\mathbf{e}_1 - H_m\mathbf{y}\|_2$ and
15: Form the solution $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + V_m\mathbf{y}^{(m)}$,
16: If no convergence, set $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(m)}$ and restart at line 3,
```

Note that in the GMRES algorithm, if $h_{i+1,i} = 0$ then $\mathbf{x}^{(i)}$ is the solution of $A\mathbf{x} = \mathbf{b}$.

Convergence In exact arithmetic, which is not the case in numerical computing environment, full GMRES will always converge in at most n iterations. This convergence is monotonic since $\|\mathbf{r}_{i+1}\| \leq \|\mathbf{r}_i\|$, $\mathbf{r}_i$ is minimized over $\mathcal{K}_n$. In order to analyze the convergence, let P_m be the space of all polynomials of degree lower or equal to m and σ be the spectrum of the matrix A. Suppose A is diagonalizable so that $A = XDX^{-1}$ and

$$\epsilon^{(k)} = \min_{\substack{p \in P_k \\ p(0)=1}} \max_{\lambda_i \in \sigma} |p(\lambda_i)|.$$

Then the residual norm at the ith iteration of GMRES satisfies

$$\|\mathbf{r}_{i+1}\| \leq \mathtt{cond}(X)\epsilon^{(i)}\|\mathbf{r}_0\|.$$

When A is positive real with symmetric part M, then

$$\|\mathbf{r}_i\| \leq (1 - \frac{\alpha}{\beta})^{(i/2)}\|\mathbf{r}_0\|,$$

with $\alpha = (\lambda_{min}(M))^2$ and $\beta = \lambda_{max}(A^T A)$.

When A is no longer positive real, assume that there are ν eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_\nu$ of A with non-positive real parts and let the other eigenvalues be enclosed in a circle centered at C with $C > 0$ and having radius R with $R > C$. Then

$$\epsilon^{(k)} \leq \left(\frac{R}{C}\right)^{k-\nu} \max_{j=\nu+1,n} \prod_{i=1}^{\nu} \frac{|\lambda_i - \lambda_j|}{|\lambda_i|} \leq \left(\frac{D}{d}\right)^{\nu} \left(\frac{R}{C}\right)^{k-\nu},$$

where

$$D = \max_{\substack{i=1,\nu \\ j=\nu+1,n}} |\lambda_i - \lambda_j|, \quad d = \min_{i=1,\nu} |\lambda_i|.$$

Note that the eigenvalue distribution is much more important that the condition number of A which is the main criterion for rapid convergence of the conjugate gradient method. The rate of convergence of the GMRES method depends on the distribution of the eigenvalues of A in the complex plane. In general, for fast convergence, the eigenvalues need to be clustered away from the origin.

Computational and memory costs The main disadvantage of GMRES is that the size m of the Kylov subspace increases with the number of iterations which is prohibitive in terms of memory. Therefore, a restarted version of GMRES is proposed in (Saad, 1986). It allows to control the maximum size m of the search space (see Algorithm 1.4). The total cost of computing $\mathbf{x}^{(m)}$ by GMRES can be divided in two parts

- The computation of the Arnoldi vectors $\mathbf{v}^{(j+1)}$ for $j = 1, \ldots, m$. The jth step in this procedure requires $2(j+1)n$ multiplications and one matrix vector product, assuming that the vectors $\mathbf{v}^{(i)}$ are not normalized but that their norms are only computed and saved. The last step requires $(m+1)n$ multiplications. Therefore, the total number of multiplications for this part is approximately $m(m+1)n$ plus m matrix vector products.
- The computation of the approximate solution $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + V_m\mathbf{y}^{(m)}$ requires mn multiplications.

Therefore, the k-th iteration of GMRES requires $k(k+2)n$ multiplications and one matrix vector products. The memory request of this method is $(k+2)n$ as the vectors $\mathbf{v}^{(i)}$ for $i = 1, \ldots, k$, the approximate solution $\mathbf{x}^{(k)}$ and the vector $A\mathbf{v}^{(k)}$ are needed to be stored keeping in mind that the restarted GMRES is preferred in order to limit the size of the Krylov subspace.

The Bi-Conjugate Gradient Stabilized Bi-CGSTAB

Another strategy for the solution of indefinite non-hermitian systems is the the Bi-CGSTAB method. This method is a fast and smoothly converging variant of BiConjugate Gradient method (BiCG) (Fletcher, 1976) and the Conjugate Gradient Squared (Sonneveld, 1989) for the solution of large sparse indefinite non-hermitian linear systems. Implicitly, the algorithm solves not only the original system $A\mathbf{x} = \mathbf{b}$ but also the dual linear system $A^H\mathbf{x}^\star = \mathbf{b}^\star$. Starting from an initial approximation $\mathbf{x}^{(0)}$ and a residual $\tilde{\mathbf{r}}^{(0)}$, the Bi-CGSTAB algorithm produces iteratively sequences of residuals $\mathbf{r}^{(k)}$ and search directions $\mathbf{p}^{(k)}$ such that

$$\tilde{\mathbf{r}}^{(k)} = \psi_k(A)\phi_k(A)\mathbf{r}^{(k)}$$

where ψ_k is a certain polynomial (Bi-CG polynomial) of degree k of which the goal is to stabilize the convergence behaviour of the algorithm defined by

$$\psi_k(A) = (I - \omega_1 A)(I - \omega_2 A) \ldots (I - \omega_k A)$$

with a suitable coefficient ω_k helping to stabilize the convergence of the algorithm. The polynomial ϕ_k is defined by the recurrence

$$\begin{aligned}\phi_k(A) &= \phi_{k-1}(A) - \alpha_k \pi_{k-1}(A),\\ \pi_k(A) &= \phi_k(A) - \beta k + 1\phi_{k-1}(A),\end{aligned}$$

where α_k and β_k are constants to be chosen. From the above recurrences and the definition of ψ_k, it follows that

$$\psi_k(A)\phi_k(A) = (I - \omega_k A)(\psi_{k-1}(A)\phi_{k-1}(A) - \alpha_k A\psi_{k-1}(A)\pi_{k-1}(A),$$

where the product $\psi_k(A)\pi_k(A)$ can be derived from the above recurrences as

$$\psi_k(A)\pi_k(A) = \psi_k(A)\phi_k(A) + \beta_{k+1}(I - \omega_k A)\psi_{k-1}(A)\phi_{k-1}(A).$$

The Bi-CGSTAB defines the search direction vector $\mathbf{p}^{(k)}$ such that

$$\mathbf{p}^{(k+1)} = \psi_k(A)\pi_k(A)\mathbf{r}^{(0)},$$

which writes in vector form as

$$\begin{aligned}\mathbf{r}^{(k+1)} &= (I - \omega_k A)(\mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)}),\\ \mathbf{p}^{(k+1)} &= \mathbf{r}^{(k+1)} + \beta_k(I - \omega_k A)\mathbf{p}^{(k)}.\end{aligned}$$

To derive the recurrence for $\mathbf{x}^{(k)}$, the vector $\mathbf{s}^{(k)}$ is introduced such that

$$\mathbf{s}^{(k)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)},$$

therefore, the recurrence for $\mathbf{r}^{(k)}$ can written as

$$\mathbf{r}^{(k+1)} = (I - \omega_k A)\mathbf{s}^{(k)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)} - \omega_k A\mathbf{s}^{(k)},$$

which yields

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)} + \omega_k A\mathbf{s}^{(k)}.$$

The optimal value for ω_k is chosen to minimize the l_2-norm of the vector

$$(I - \omega_k A)\psi_k(A)\phi_{k+1}(A)\mathbf{r}^{(0)}.$$

It is given by

$$\omega_k = \frac{(A\mathbf{s}^{(k)}, \mathbf{s}^{(k)})}{(A\mathbf{s}^{(k)}, A\mathbf{s}^{(k)})}.$$

The scalar α_k and β_k are chosen so that the residuals and the search directions satisfy biorthogonality and biconjugacy conditions

$$\begin{aligned}(\tilde{\mathbf{r}}^{(i)}, \mathbf{r}^{(j)}) &= 0,\\ (\mathbf{p}^{(i)}, \mathbf{p}^{(j)}) &= 0,\end{aligned}$$

for $i \neq j$. They are given by

$$\alpha_k = \frac{(\mathbf{r}^{(k)}, \tilde{\mathbf{r}}^{(0)})}{(A\mathbf{p}^{(k)}, \tilde{\mathbf{r}}^{(0)})},$$

$$\beta_k = \frac{(\mathbf{r}^{(k+1)}, \tilde{\mathbf{r}}^{(0)})}{(\mathbf{r}^{(k)}, \tilde{\mathbf{r}}^{(0)})} \times \frac{\alpha_k}{\omega_k}.$$

After putting together the above relations, the Bi-CGSTAB algorithm due to Van der Vorst (1992) is given in Algorithm 1.5.

Algorithm 1.5 Bi-Conjugate Gradient Stabilized method (Bi-CGSTAB)

```
Given an initial guess $\mathbf{x}^{(0)}$,
Compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$,
Choose $\tilde{\mathbf{r}}$ such that $(\mathbf{r}^{(0)}, \tilde{\mathbf{r}}) \neq 0$ ,
Set $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$,
for $i = 0, \dots$ until convergence do
    $\alpha_i = (\mathbf{r}^{(i)}, \tilde{\mathbf{r}})/(A\mathbf{p}^{(i)}, \tilde{\mathbf{r}})$,
    $\mathbf{s}^{(i)} = \mathbf{r}^{(i)} - \alpha_i A\mathbf{p}^{(i)}$,
    $\omega_i = (A\mathbf{s}^{(i)}, \mathbf{s}^{(i)})/(A\mathbf{s}^{(i)}, A\mathbf{s}^{(i)})$,
    $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)} + \omega_i \mathbf{s}^{(i)}$,
    $\mathbf{r}^{(i+1)} = \mathbf{s}^{(i)} - \omega_i A\mathbf{s}^{(i)}$,
    $\beta_i = (\mathbf{r}^{(i+1)}, \tilde{\mathbf{r}})/(\mathbf{r}^{(i)}, \tilde{\mathbf{r}}) \times \alpha_i/\omega_i$,
    $\mathbf{p}^{(i+1)} = \mathbf{r}^{(i+1)} + \beta_i \left(\mathbf{p}^{(i)} - \omega_i A\mathbf{p}^{(i)}\right)$,
endfor
```

Convergence Few theoretical results are known about the convergence of Bi-CGSTAB method. Convergence is obtained for $2n$ iterations. For hermitian positive definite systems, this method provides the same results as CG, but for twice the cost per iteration as the dual linear system $A^H\mathbf{x}^\star = \mathbf{b}^\star$ is solved as well. For non-hermitian matrices, it has been shown in (Van der Vorst, 1992), (Sleijpen and Fokkema, 1993) and (Saad, 2003) that in phases of the process where there is significant reduction of the norm of the residual, the method is comparable to full GMRES in terms of numbers of iterations. In practice this is often confirmed, but it is also observed that the convergence behavior may be irregular and the method may even break down in particular for indefinite matrices.

Computational and memory costs Algorithm 1.5 allows to summarize the operations performed per iteration and the total vectors needed. Thus, the method requires 2 matrix vector products, 5 inner products and 6 vector updates. The method requires the storage of 7 vectors including the solution vector.

Summary

Efficient solution of non-hermitian indefinite and ill-conditioned linear system is largely related to an adequate choice of the iterative method.

Method	Inner product	Vector update	Matrix vector product
CG	2	3	1
CGNR	2	3	2
GMRES	$k+1$	$k+1$	1
Bi-CGSTAB	5	6	2

Table 1.4: Summary of the operations required at the iteration k.

Fortunately, operations which are performed in iterative methods as presented in Table 1.4 can be executed at a very high efficiency on today's parallel computers. Table 1.5 shows the memory requirements of the iterative methods.

Method	Storage requirements
CG	$4n$
CGNR	$5n$
GMRES	$(k+2)n$
Bi-CGSTAB	$7n$

Table 1.5: Summary of the memory required at the iteration k with n denoting the size of the matrix

1.7.2 Difficulties to use an iterative solver for the solution of the Helmholtz equation

The frequency-domain wave equation is known to be one challenging problem for iterative methods. Even when it is reduced to its simplest form where the density is assumed homogeneous giving the well-known Helmholtz equation, and when the wavenumber $k = \omega/c$ ($\omega = 2\pi f$ is the angular frequency and c is the P-wave velocity) is constant, the impedance matrix is indefinite which leads to severe convergence problem of iterative solvers. The Helmholtz equation is difficult to solve for mainly two reasons (Ernst and Gander, 2012). First, the solutions of the homogeneous Helmholtz equation oscillate on a scale of $1/k$. Therefore the total number of degrees of freedom n increases proportionally to k^d where d is the space dimension ($d = 2$ or 3). This growth of n leads to very large matrices and hence to large computational costs. Second, the matrix resulting from the discretization of the Helmholtz equation is indefinite. This means that the eigenvalues of the impedance matrix have both negative and positive real parts with extremely small amplitude as the wavenumber increases. This yields an ill-conditioned impedance matrix difficult to invert for iterative solvers.

Consider the 2D frequency-domain acoustic wave equation equation (1.33). Perfectly matched layers (Bérenger, 1994) are used at the boundary of domain with $N_{PML} = 10$ grid points in each layer. The damping coefficient in the PMLs is set to $C_{PML} = 1500$. The equation (1.33) is discretized using a fourth-order staggered-grid finite-difference method with 5 grid points per wavelength yielding a sparse impedance matrix A of size n.

We compute the spectrum of the impedance matrix A based on the domain Ω of size 12 km ×6 km. The P-wave velocity is assumed constant ($c = 1500$ m/s). The spectrum of the impedance matrix A is presented in Figure 1.8. It shows the shift in the real part of the eigenvalues due to the zero-th term in k^2 giving eigenvalues with positive and negative real parts.

For the numerical solution using the iterative methods CGNR, GMRES and Bi-CGSTAB, we consider the heterogeneous medium given by the BP2004 model (Billette and Brandsberg-Dahl, 2004) of size 67 km × 12 km (a fully detailed description of the model is given in Chapter 3 Section 3.1.1). The model is discretized on 48×270 grid for $f = 1$ Hz and 209×1113 grid for $f = 5$ Hz. The stopping criterion

$$\frac{\|A\mathbf{x}^{(k)} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2} \leq 10^{-4},$$

is applied using the true residual.

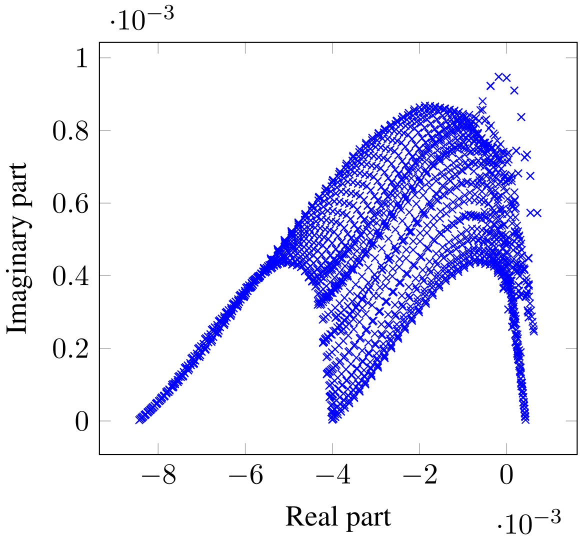
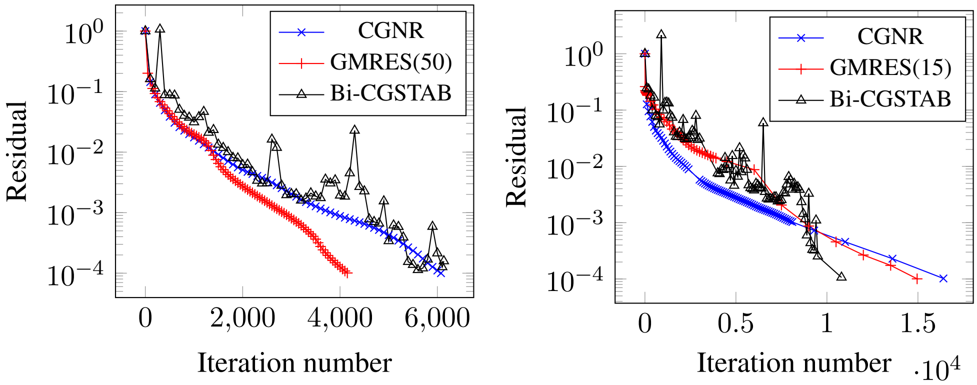


Figure 1.8: Spectrum of the impedance matrix A on a homogeneous medium.



(a) $f = 1$ Hz.

(b) $f = 5$ Hz.

Figure 1.9: Performances of CGNR, CGMRES and Bi-CGSTAB on the Helmholtz problem in the heterogeneous medium BP2004.

Figure 1.9 shows the number of iterations performed by CGNR, GMRES and Bi-CGSTAB for the solution of the heterogeneous Helmholtz problem for various frequencies. The three iterative solvers perform more than $10,000$ iterations to converge which a large number for a relatively small discrete problem. The CGNR method is robust however it requires a large number of iterations to converge. Because it solves the normal equations, the rate of convergence is proportional to the square of the condition number of the matrix A. Both GMRES and Bi-CGSTAB suffer from a slow convergence as the size of the problem increases. Bi-CGSTAB is more sensitive to the spectrum of the matrix. As the size of the linear system increases, the convergence becomes increasingly non-smooth underlining a loss of accuracy in the residual update.

Therefore, preconditioning strategies become mandatory to improve the convergence of iterative solvers. These techniques are introduced in the following section.

1.7.3 Preconditioning strategies for iterative methods

The convergence rate of iterative methods based on Krylov subspace is directly linked to the spectrum of the matrix A. For conjugate gradient methods, this convergence is quantified by the condition number of the matrix A which is denoted by $\texttt{cond}(A)$. When the condition number of the matrix A is small, solving the linear system (1.40) results in a convergence with a low number of iterations. On the contrary, an ill-conditioned matrix has a relatively high condition number and a CG iterative method applied to this linear system requires a large number of iterations to converge. In addition, and particularly for GMRES, rapid convergence is obtained when the eigenvalues of preconditioned matrix are clustered away from the origin. Therefore, a preconditioning strategy is required to enable a fast convergence as it improves the eigenvalue distribution of the matrix to be solved.

Principle

Preconditioning means transforming the original system into another system "easier" to solve. General preconditioning strategies for solving the linear system (1.40) consist in defining the invertible matrices P_l and P_r such that

$$P_l A P_r \mathbf{y} = P_l \mathbf{b}, \quad \mathbf{x} = P_r \mathbf{y}. \tag{1.50}$$

The condition number of the matrix $P_l A P_r$ should be closer to 1 than the condition number of A or in some cases, P_l and P_r help clustering the eigenvalues of the matrix $P_l A P_r$ away from the origin. Preconditioning should then enable fast convergence of iterative solvers on the system (1.50). Standard strategies for non-hermitian systems consists for instance in choosing $P_l = A^{-1} + E, \quad P_r = I$ (left preconditioning), or equivalently $P_l = I, \quad P_r = A^{-1} + E$ (right preconditioning) with E a residual or error matrix of the approximation.

Ideally , one can choose P_l or P_r as

$$P_l = A^{-1} \text{ or } P_r = A^{-1},$$

the convergence is then obtained in one iteration and we would have

$$\texttt{cond}(P_l A) = \texttt{cond}(A P_r) = \texttt{cond}(I) = 1.$$

However, the matrix A^{-1} is not available in practice either for computational or memory issues (or both). Therefore, preconditioning strategies consists in computing a preconditioner P which approximates at "best" the matrix A^{-1} such that

- the computational cost of applying the preconditioner on a vector is in the same order of $A\mathbf{x}$,
- the cost for computing and storing the preconditioner is affordable,
- and $P\mathbf{x}$ is a good approximation of $A^{-1}\mathbf{x}$.

Standard preconditioning strategies for the wave problem

Preconditioning strategies for the frequency-domain wave equation have been intensively studied over the past decades. Ernst and Gander (2012) propose an overview of the iterative methods together with the preconditioning strategies which I detail later in this section, for the solution of Helmholtz-like problems. This review explains why the Helmholtz problem is difficult to solve by iterative methods and addresses the theoretical aspects of the problem. GMRES and Bi-CGSTAB are promoted as the best iterative methods for this problem. However they require an efficient preconditioning method as they fail to converge for large and/or complex heterogeneous problems. Many iterative methods combined with specific preconditioners have been designed for solving the frequency-domain wave equations. Most of the recent developments of the preconditioning strategies focus on the acoustic frequency-domain wave equation, particularly on the Helmholtz equation which considers a homogeneous density. Therefore, consider the latter equation

$$-\left(\Delta + k^2\right) u = f, \tag{1.51}$$

with the suitable boundary conditions. The discretization of the Helmholtz problem (1.51) using finite-difference methods yields a sparse linear system

$$A\mathbf{x} = \mathbf{b}. \tag{1.52}$$

The matrix A is sparse, large dependently on the size of the model and the wave numbers modeled, complex-valued and non hermitian because of the absorbing boundary conditions often used. This matrix remains positive definite as long as k^2 is smaller than the first eigenvalue of the discrete Laplacian operator. However, in geophysical applications, the wavenumber becomes rapidly large which gives to the matrix A both positive and negative real part eigenvalues. This makes the matrix A indefinite. Therefore, iterative methods alone are not effective methods for the solution of this type of problem.

In this section, I shall focus on the different preconditioning methods which have been discussed recently. I shall distinguish them into two categories. The first category of preconditioners is called *algebraic*. They do not require a prior knowledge of the discrete operator A. The second category is called *operator-based* as it attempts to produce a preconditioning operator which approximates from a "physics" point of view the operator A.

Algebraic preconditioners Among the most reliable algebraic preconditioning strategies is the Incomplete LU factorization of the matrix A introduced by Saad (2003). This factorization performs a decomposition of the form $A = LU - R$ where L and U are respectively a lower and an upper matrix which have the same non-zero pattern as the lower and upper parts of the matrix A and R is a residual matrix. The basic method is called the ILU(0) where the matrices L and U have the same sparsity pattern as the matrix A. Further developments of the ILU preconditioner were about introducing a control on the level of fill p and/or a threshold τ to improve its quality. This strategy allows the preconditioner to be more accurate with a control on the fill-in effect (Saad, 1994). Therefore, the action of the ILU preconditioner on a vector $\mathbf{x}$ can be known by performing a forward and backward substitution. This kind of preconditioner may be described as *implicit* as its action on a vector requires the solution of a linear system. However, such preconditioner suffer from instabilities during the Gauss elimination process. Null pivots appear and cause possible breakdown of the factorization. This preconditioner is used by Osei-Kuffuor and Saad (2010) on the Helmholtz problem. The incomplete factorization is performed on the matrix $A + i\alpha I$ where a small complex perturbation on the diagonal of the matrix is introduced. Such strategy improves the quality of the preconditioner and more importantly acts as a tool for safeguarding the stability of the incomplete factorization procedure.

Among the algebraic preconditioner, one can recall the approximate inverse (AINV) preconditioner. This technique introduced by Benzi and Tuma (1998) is based on an incomplete inverse triangular factorization of the matrix A. It computes an incomplete factorization of the form $W^H AZ = D$ using a bi-conjugation algorithm, where W and Z are upper triangular matrices, and D is a diagonal matrix

$$W^H AZ = D,$$

thus

$$(W^H AZ)^{-1} = Z^{-1}A^{-1}W^{-H} = D^{-1},$$

and

$$A^{-1} = ZD^{-1}W^H.$$

The matrices W and Z are computed column-wise by performing an approximate A-bi-orthogonalization of the Gram-Schmidt kind. In fact, as $A = LDU$ is an LDU factorization of the matrix A, then W and Z are respectively the inverse of the matrices L and U. To obtain a sparse preconditioner, a dropping rule may be applied as for the ILU preconditioner at each iteration of the preconditioner computation process. This straightforward implementation of the bi-conjugation algorithm is refereed to DDS (Dynamic Data Structures) by Benzi and Tuma (1998). Because most of the fill-in in the factors Z and W appears in the late steps of the factorization, it is difficult to minimize the computational cost and the high proportion of non-floating-point operations. Therefore, an alternative implementation called SDS (Static Data Structures) is preferred. Non-zero coefficients are computed according to the sparsity pattern chosen. Both implementations are completely equivalent, however the SDS implementation allows to compute separately the factors Z and W but at the cost of increasing the number of floating-point operations. It appears to be well suited to parallel implementation. The action of such a preconditioner on a vector is known by performing matrix vector products of the matrices W, Z and D^{-1}. Therefore, such preconditioner is advantageous as its action can be easily parallelized. This kind of preconditioner may be described as *explicit*.

One last preconditioning strategy we may consider is the algebraic domain decomposition method based on a hybrid direct-iterative solver. The computational domain is decomposed into sub-domains. The points which are located in the interior of each sub-domain are called the interior points. The points which are located at the interfaces between the sub-domains are called interface points (Smith et al., 1996). A sparse direct solver is used to solve the local linear systems defined on each sub-domain allowing to compute the solution on the interior points. An iterative method is used to solve the reduced system, the so-called Schur complement system, for the interface points (Saad, 2003). A detailed presentation of this method is given by Haidar (2008) and Sourbier et al. (2011). Denote by n_p the number of sub-domains in one direction. For complexity comparison, it is assumed the same number of sub-domains in the three space directions. Therefore, the total number of sub-domains is n_p^3 and the dimension of the cubic sub-domains is N/n_p. The theoretical memory used to perform the local lower-upper factorizations for all sub-domain is $\mathcal{O}(n_p^3(N/n_p)^4) = \mathcal{O}(N^4/n_p)$. The dimension of the dense Schur complement matrices is $(N/n_p)^2$. Thus, the total memory complexity which required for the storage of the n_p^3 local Schur complements is $\mathcal{O}(n_p^3(N/n_p)^4) = \mathcal{O}(N^4/n_p)$ which is equal to the memory complexity of the LU factorizations. Thus, the memory complexity of the hybrid solver is $\mathcal{O}(N^4/n_p)$. The computational complexity of the local LU factorizations for all sub-domain is $\mathcal{O}(n_p^3(N/n_p)^6) = \mathcal{O}(N^6/n_p^3)$ and the computational complexity for the solution of the Schur complement systems using the iterative part of the hybrid solver is $\mathcal{O}(n_p^3 N_{iter}(N/n_p)^4) = \mathcal{O}(N_{iter}N^4/n_p)$. Therefore, assuming that $\mathcal{O}(n_p) = \mathcal{O}(N)$ and $\mathcal{O}(N_{iter}) = \mathcal{O}(N)$, the computational complexity of the hybrid solver is $\mathcal{O}(N^4)$ and its memory complexity is $\mathcal{O}(N^3)$ for one source modeling. These computational and memory costs are the same as that of iterative solvers. However, one need to keep in mind that a large number of cores $\mathcal{O}(N^3)$ is required to achieve such performances.

Operator-based preconditioners The second class of preconditioners is known as operator-based preconditioners. The first attempts to precondition the Helmholtz operator $(\Delta + k^2)$ was using the inverse of the Laplacian operator Δ as introduced by Bayliss et al. (1983). They propose to solve the Helmholtz problem using the CGNR method where the iterations are preconditioned with the inverse of the discrete Laplacian operator. This inverse is not computed explicitly but its action is evaluated by applying sweeps of Symmetric Successive Over Relaxation (SSOR) iterations (Saad, 2003). Bayliss et al. (1983) indicate that performing more sweeps is less efficient because of the additional cost of each sweep and the results obtained with one SSOR sweep present significant acceleration even when the wavenumber is not small.

Laird and Giles (2002) define the preconditioner based on a modified Helmholtz operator $(\Delta - k^2)$. In the definition of the preconditioner, Dirichlet boundary conditions are used despite the use the radiation boundary condition in the original problem and the sign of the mass matrix term is changed. Therefore, the preconditioner is symmetric positive definite. The preconditioner is applied using one V-cycle of multigrid. To minimize the cost of the application of the preconditioner, only one pre- and post-smoothing Jacobi iteration is used within the multigrid cycle. This method is fully described by Aruliah and Ascher (2003) and Mulder (2006). GMRES is used for small size systems and CGNR for larger size systems.

Made et al. (2000) introduce first the idea of preconditioning the Helmholtz operator using the same operator but with a complex shift on the Laplace operator which amounts to use the operator $(\Delta + i\epsilon + k^2)$ as a preconditioner with ϵ a small real perturbation. GMRES is used for the solution of the 2D Helmholtz problem and the preconditioner is computed using the incomplete LU factorization (ILU) strategy with different levels of filling.

Erlangga et al. (2006b) introduce the complex shifted Laplace preconditioner. This preconditioner is based on the operator $(\Delta - ik^2)$. The shifted Laplace operator is used to precondition the Bi-CGSTAB iterations. It is computed using the ILU strategy described in the previous part. The results are compared with the results obtained when the preconditioner is applied with a multigrid approach using the same iterative solver. The experiments are performed on a constant wavenumber model, a wedge model and finally on the Marmousi problem. Hence, both strategies are effective under the condition that the computation of the ILU preconditioner is allowed with some fill-in.

Ultimately, several investigations have been performed in this direction. The preconditioning strategy has been addressed in a more general fashion through the operator $(\Delta + (\beta_1 + i\beta_2)k^2)$ by Erlangga et al. (2004) and Plessix (2006b) with proofs for defining the optimal shift. The values $\beta_1 = 1$ and $\beta_2 = 0.5$ have shown the optimal results.

Recently, another type of preconditioner was introduced. The sweeping preconditioner and moving PML sweeping preconditioner are introduced in (Engquist and Ying, 2011) and (Poulson et al., 2013) for both constant and non-constant wavenumber. This method is based on the computation a reasonably accurate approximate factorization of the shifted Helmholtz operator by multifrontal approach and hierarchical partitioning of the domain layer by layer. The approach of this method is to reduce the dimension of the problem by one order of magnitude allowing the feasibility of the factorization. GMRES is used to solve the 2D and 3D acoustic wave problem.

Up to now, the combination of iterative methods together with preconditioning strategies allows to speed up the convergence on the Helmholtz equation. As most of these methods are based on multigrid approaches, it is well-known that the shift has to be large enough for these methods to be effective to invert the shifted operator, but not too large to still be a good preconditioner for the original Helmholtz problem as it has been pointed out in (Ernst and Gander, 2012) and (Gander et al., 2015). The number of iterations still increases with respect to the frequency modeled yielding a complexity of $N_{iter} = \mathcal{O}(N)$ where N is the average number of grid point per dimension. The ideal case is having a convergence that is independent of the size from the problem. This results is nearly obtained by Poulson et al. (2013) for the 2D and 3D Helmholtz equation using the sweeping PML preconditioner combined with GMRES.

1.8 Conclusion

In this chapter, I was interested in describing the main ingredients governing this study. I introduced the equations governing the acoustic and elastic wave propagation and the standard discretization method I use in this thesis. These equations are discretized using the finite-difference method with the appropriate schemes. Namely, the staggered-grid finite-difference scheme combined with the parsimonious approach is used.

Iterative method and preconditioning strategies are discussed for the solution of the frequency-domain wave equation. Most of the recent developments were done in the acoustic approximation and particularly using the Helmholtz equation which supposes a constant density. Some attempts for the resolution of the 3D frequency-domain elastic wave equations are carried out but the size of the problems are not realistic. Among the standard iterative methods, the Conjugate Gradient Minimum Norm (CGMN) iterative solver introduced by Björck and Elfving (1979) and efficiently parallelized into the Component-Averaged Row Projections Conjugate Gradient method (CARP-CG) method by Gordon and Gordon (2010a) has been successfully used for the resolution of the frequency-domain acoustic wave equation in the 3D FWI framework (van Leeuwen and Herrmann, 2014) as well as in the 2D and 3D elastic approximations for the frequency-domain seismic wave simulation by Li et al. (2015). In the latter case, CARP-CG reveals to be extremely robust. The convergence is obtained without preconditioning with $N_{iter} = \mathcal{O}(N^{\alpha})$ with $\alpha < 1$ in media with strong contrasts, presenting high Poisson's ratio, and with a free surface boundary condition responsible for the generation of surface waves. In addition, iterative methods benefit from multiple right hand side accelerations such as the block CG methods introduced by O'Leary (1980) and the projection methods of Chan and Wan (1997) leading to a significant reduction in the number of iterations. Therefore, the CARP-CG method can be computationally enhanced with a proper preconditioning strategy. This method becomes particularly appealing for FWI applications in the 3D elastic approximation where other strategies fail or require too much resources. However, the number of iterations required to converge is still large in some configurations.

The linear system resulting from the discretization of the frequency-domain wave equations and its resolution using the iterative method CARP-CG are at the heart of this thesis. I am interested into pushing further this method by suggesting a preconditioning strategy which is adapted. Based on the recent developments, this preconditioner is based on the formulation of this same problem in a strongly damped medium which is equivalent to the shifted Laplace preconditioner already discussed. The method for the evaluation of the preconditioner and its application with CARP-CG method will be detailed in the following chapters.

Chapter 2

The CARP-CG method and preconditioning strategies

Contents

A robust and efficient solver for the frequency-domain elastic wave problem is an essential element for any seismic imaging technique such as the reverse time migration (RTM) and the full waveform inversion (FWI) methods. The discretization of the frequency-domain elastic wave equations (1.14) or (1.15) governing the elastic wave propagation in the frequency-domain yields a large sparse indefinite and ill-conditioned impedance matrix. Moreover, the conditioning of such matrix is worsened by accounting for the heterogeneities of the subsurface (P and S-wave velocity and density fields) as well as the nature of the boundary condition on top. The free-surface boundary induces the propagation of surface waves which makes the linear system even harder to solve using iterative solvers.

The CARP-CG method is a Krylov sub-space iterative method designed for the solution of large sparse and non-hermitian linear systems on parallel architectures (Gordon and Gordon, 2010b). This method is based on the combination of the Kaczmarz iterations and the conjugate gradient method. CARP-CG can be considered as a parallel version of the CGMN method introduced by Björck and Elfving (1979). By introducing the parallelization of the Kaczmarz iterations through the CARP method (Gordon and Gordon, 2005), the CARP-CG method can also be seen as a conjugate gradient acceleration of CARP. CARP-CG has been successfully applied for solving the 2D and 3D Helmholtz equation in heterogeneous media (Gordon and Gordon, 2013; van Leeuwen and Herrmann, 2014). Gordon and Gordon (2013) show that the CARP-CG method outperforms any iterative solver or multigrid method combined with the shifted Laplacian preconditioner at all wave numbers in term of computation time. Recently, the CARP-CG method was used for solving the 2D and 3D frequency-domain elastic wave equations in strongly heterogeneous media (Li et al., 2015).

However, in the 3D elastic approximation, considering CARP-CG as a solver for the forward problem in the framework of seismic imaging methods requires to further improve the convergence properties of this method. The number of iterations and the computation time performed by the CARP-CG method for the solution of the frequency-domain elastic wave equation still needs to be reduced. Thus preconditioning techniques have to be used to enhance the performances of the CARP-CG method. However, unlike standard iterative methods such as CGNR, GMRES or BiCGSTAB, the CARP-CG algorithm is based on a reformulation of the linear system to be solved. Therefore, designing a preconditioner adapted to this method is not straightforward.

In this chapter, I shall give an overview of the CARP-CG method and study the possible preconditioning strategies which can be combined with it. In the first part, the CARP-CG method is introduced through the Kaczmarz iterations and the conjugate gradient method. The numerical complexity and the theoretical convergence properties are presented. A spectral analysis of this method based on the Helmholtz equation is presented as well. In the second part, preconditioning strategies are investigated for the CARP-CG method. In the third and fourth parts, spectral studies are carried out on the preconditioned problems in the 2D acoustic and elastic approximations respectively.

2.1 The CARP-CG method

The purpose of this section is to introduce the CARP-CG method. This method is based on the acceleration of the Kaczmarz iterations using the conjugate gradient method. I start this overview by introducing the Kaczmarz method. The latter is combined with the conjugate gradient method yielding the sequential CGMN method. The parallelization of the Kaczmarz iterations is performed by the CARP method. Finally, the CARP-CG method is the CG acceleration of CARP. It can be considered as well as the parallelization of CGMN. In the Figure 2.1, the links between the above iterative methods (Kaczmarz, CARP, CGMN and CARP-CG) are presented.

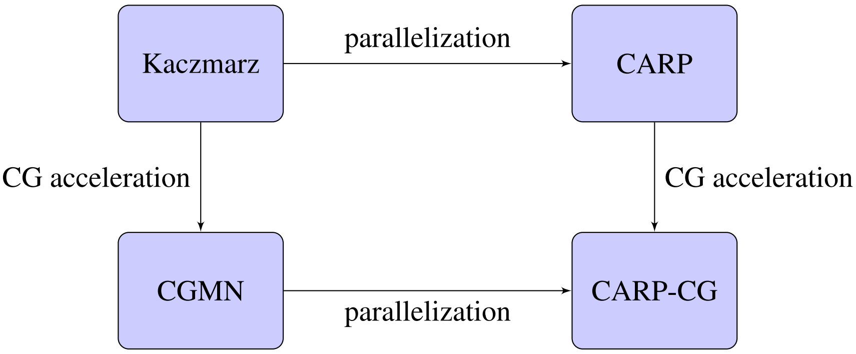


Figure 2.1: Generalized diagram showing the links between the iterative methods Kaczmarz, CGMN, CARP and CARP-CG.

2.1.1 The Kaczmarz method

The Kaczmarz method is one of the first iterative method for computing the solution of a sparse linear system (Kaczmarz, 1937). Consider a linear system of size $n \in \mathbb{N} \setminus \{0\}$

$$A\mathbf{x} = \mathbf{b}, \tag{2.1}$$

with

$$A \in Gl_n(\mathbb{K}), \quad \mathbf{x} \in \mathbb{K}^n, \quad \mathbf{b} \in \mathbb{K}^n,$$

where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Given an initial guess $\mathbf{x}^{(0)}$, at each iteration, the approximate solution $\mathbf{x}^{(k)}$ is updated by performing an orthogonal projection of the current iterate towards the hyperplane defined by one equation of the linear system (2.1). These projections are denoted by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}\right)}{\|\mathbf{a}_{i\bullet}\|_2^2} \mathbf{a}_{i\bullet}^H, \quad i = k+1\,[n], \tag{2.2}$$

where $(\cdot, \cdot)$ denotes the canonical scalar product in $\mathbb{K}$ and $\|\mathbf{x}\|_2^2 = (\mathbf{x}, \mathbf{x})$ is the norm induced by the canonical scalar product. Consider for instance a linear system of size 2 given by

$$\left\{ \begin{array}{rcl} a_{11}\,x_1 + a_{12}\,x_2 & = & b_1, \\ a_{21}\,x_1 + a_{22}\,x_2 & = & b_2. \end{array} \right. \tag{2.3}$$

The coefficients of the linear system (2.3) are set to $(a_{11}, a_{12}, a_{21}, a_{22}) = (-2, 1, 0, 1)$ and $(b_1, b_2) = (1, 1)$. Each equation of the system (2.3) is represented by a line in Figure 2.2. Applying the Kaczmarz method for the solution of the system (2.3) with a given initial guess $\mathbf{x}^{(0)}$, the set of approximate solutions $\{\mathbf{x}^{(0)},\ \mathbf{x}^{(1)},\ \mathbf{x}^{(2)},\ \cdots\}$ is calculated following the orthogonal projections given in the equation (2.2).

Consider the case where the hyperplanes of the linear system (2.3) are orthogonal. Note that the resulting matrix A

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

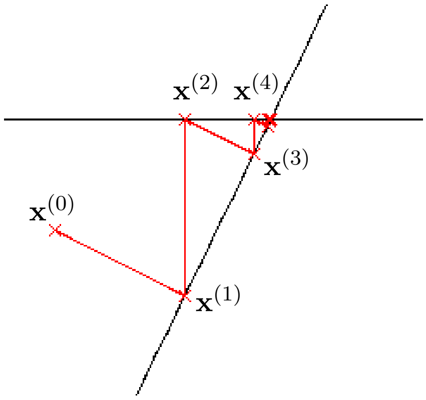


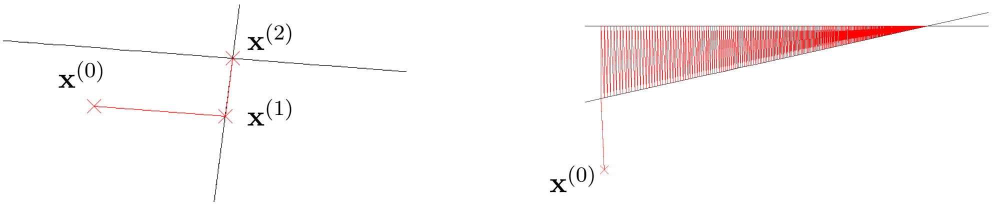
Figure 2.2: Successive approximate solutions computed by the Kaczmarz method on system (2.3) with $(a_{11}, a_{12}, a_{21}, a_{22}) = (-2, 1, 0, 1)$ and $(b_1, b_2) = (1, 1)$.

is orthogonal, i.e. $(\mathbf{a}_{i\bullet}, \mathbf{a}_{j\bullet}) = 0$ if $i \neq j$. In such case, the Kaczmarz method converges in 2 iterations as it is the shown in Figure 2.3a where the matrix A is given by

$$A = \begin{pmatrix} -1 & 0.1 \\ 0.1 & 1 \end{pmatrix}.$$

On the other hand, when the hyperplanes of the linear system (2.3) are nearly collinear, the Kaczmarz method performs a large number of iterations to converge towards the solution. This is shown in Figure 2.3b where the matrix A is chosen as

$$A = \begin{pmatrix} -1 & 0.1 \\ 0 & 1 \end{pmatrix}.$$



(a) Case 1

(b) Case 2

Figure 2.3: Successive approximate solutions computed by the Kaczmarz method on system (2.3) with $(a_{11}, a_{12}, a_{21}, a_{22}) = (-1, 0.1, 0.1, 1, 1)$ for (a) denoting an orthogonal matrix A and $(a_{11}, a_{12}, a_{21}, a_{22}) = (-0.1, 1, 0, 1)$ for (b) denoting a matrix A with nearly collinear row vectors and $(b_1, b_2) = (1, 1)$.

Remark 2.1. *The computational cost of the Kaczmarz method can be reduced assuming the rows of the matrix A have been previously normalized by dividing each equation by the L_2-norm of its coefficients*

$$DA\mathbf{x} = D\mathbf{b}, \tag{2.4}$$

where $D \in Gl_n(\mathbb{R})$ is a diagonal matrix such that

$$d_{ii} = \frac{1}{\|\mathbf{a}_{i\bullet}\|_2}, \quad i = 1, \cdots, n, \quad \textit{and} \quad d_{ij} = 0, \quad i \neq j.$$

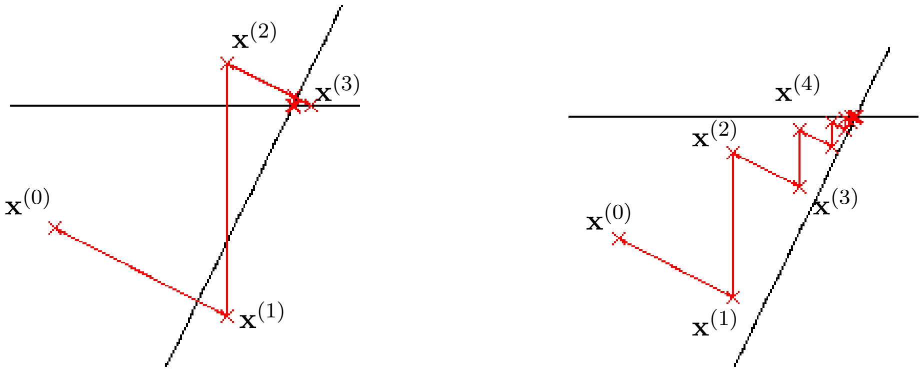
This normalization step is called Geometric Row Scaling (GRS) and can be considered as a first preconditioning step for the linear system (2.1) *(Gordon and Gordon, 2010c). Throughout the rest of this thesis, the linear system* (2.1) *is considered as already normalized through equation* (2.4). *The iterations* (2.2) *are then rewritten as*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}\right)\right) \mathbf{a}_{i\bullet}^H, \quad i = k+1\,[n]\,. \tag{2.5}$$

A relaxation parameter λ $(0 < \lambda < 2)$ can be introduced in the Kaczmarz method, yielding the iterations

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}\right)\right) \mathbf{a}_{i\bullet}^H, \quad i = k+1\,[n]\,. \tag{2.6}$$

When the relaxation parameter is greater than 1, the projections are performed *over* the hyperplanes defined by the row equations whereas when $\lambda < 1$, these projections are performed *under* the hyperplanes (see figure 2.4). Depending on the hyperplanes, the relaxation parameter gives either an acceleration of the convergence or on the opposite a slow down of the convergence. Unfortunately, in most cases the optimal value for the relaxation parameter needs to be estimated empirically. Theoretical results are derived for the estimation of the optimal value of the relaxation parameter λ



(a) $\lambda > 1$ (b) $\lambda < 1$

Figure 2.4: Relaxed Kaczmarz method with $\lambda > 1$ (a) and $\lambda < 1$ (b).

$$\lambda_{opt} = \frac{2}{1 + \sqrt{1 - \rho(G)}}, \tag{2.7}$$

where G is the iteration matrix defined by $G = (D - L)^{-1}U$ with D, L and U the diagonal, lower and upper parts of the matrix A and $\rho(G)$ the spectral radius of the matrix G (Saad, 2003). Expression (2.7) shows the difficulty to estimate the optimal value of the relaxation parameter as one needs to access the spectral radius of the matrix G.

Remark 2.2. *Note that in addition to dividing the ith equation by $\|\mathbf{a}_{i\bullet}\|_2^2$, we can also multiply it by $\sqrt{\lambda}$ and thus save some additional multiplications during the Kaczmarz iterations. The matrix $\tilde{D} \in Gl_n(\mathbb{R})$ is such that*

$$\tilde{D}A\mathbf{x} = \tilde{D}\mathbf{b},$$

with

$$\tilde{d}_{ii} = \frac{\sqrt{\lambda}}{\|\mathbf{a}_{i\bullet}\|_2}, \quad i = 1, \cdots, n, \quad \text{and} \quad \tilde{d}_{ij} = 0, \quad i \neq j.$$

The relaxed version of the Kaczmarz algorithm is equivalent to the Successive Over Relaxation (SOR) method applied to the normal equations (Saad, 2003)

$$AA^H\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = A^H\mathbf{y}, \tag{2.8}$$

with $\mathbf{y} \in \mathbb{K}^n$. The SOR method is based on the recursion

$$(D + \lambda E)\mathbf{x}^{(k+1)} = [(1-\lambda)D - \lambda F]\,\mathbf{x}^{(k)} + \lambda\mathbf{b},$$

which can be rewritten as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda D^{-1}\left[\mathbf{b} - E\mathbf{x}^{(k+1)} - (F + D)\mathbf{x}^{(k)}\right], \tag{2.9}$$

where $A = D + E + F$ in which D is the diagonal of A, E and F are respectively its strict lower and upper parts. The ith inner step of the above iteration k of equation (2.9) is given by

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\lambda}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right], \quad i = k+1\,[n]\,. \tag{2.10}$$

Given the initial guess $\mathbf{x}^{(0)} \in \mathbb{K}^n$, the first iteration of equation (2.10) writes

$$\begin{aligned} x_1^{(1)} &= x_1^{(0)} + \frac{\lambda}{a_{11}}\left[b_1 - \sum_{j=1}^{n} a_{1j}x_j^{(0)}\right], \\ &= x_1^{(0)} + \frac{\lambda}{a_{11}}\left[b_1 - (A\mathbf{x}^{(0)})_1\right], \\ &= x_1^{(0)} + \frac{\lambda}{a_{11}}\left[b_1 - (\mathbf{a}_{1\bullet}, \mathbf{x}^{(0)})\right], \end{aligned}$$

with $(A\mathbf{x}^{(k)})_j$ denoting the jth coefficient of the vector $A\mathbf{x}^{(k)}$. At the iteration $k+1$, only the ith ($i = k+1\,[n]$) coefficient of the vector $\mathbf{x}^{(k+1)}$ is updated, therefore the coefficients $x_j^{(k+1)}$ for $j = 1, \ldots, i-1$ remain unchanged from the previous iteration k

$$x_j^{(k+1)} = x_j^{(k)}, \quad j = 1, \ldots, i-1, \quad i = k+1\,[n]\,.$$

Thus, at the iteration $k+1$ we have

$$\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} = \sum_{j=1}^{i-1} a_{ij}x_j^{(k)}. \tag{2.11}$$

Using the above equation (2.11), the iteration given in the equation (2.10) becomes

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\lambda}{a_{ii}} \left[b_i - (\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}) \right], \quad i = k+1\,[n],$$

or in vector form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\lambda}{a_{ii}} \left[b_i - (\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}) \right] \mathbf{e}_i, \quad i = k+1\,[n].$$

When applied to the normal equations (2.8), the previous iterations write

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \frac{\lambda}{a_{i\bullet} a_{i\bullet}^H} \left[b_i - (\mathbf{a}_{i\bullet}, A^H \mathbf{y}^{(k)}) \right] \mathbf{e}_i, \quad i = k+1\,[n]. \tag{2.12}$$

The above equation (2.12) is multiplied by A^H on the left and by noting that $\|a_{i\bullet}\|_2^2 = a_{i\bullet} a_{i\bullet}^H$, the following equation is obtained

$$A^H \mathbf{y}^{(k+1)} = A^H \mathbf{y}^{(k)} + \frac{\lambda}{\|a_{i\bullet}\|_2^2} \left[b_i - (\mathbf{a}_{i\bullet}, A^H \mathbf{y}^{(k)}) \right] A^H \mathbf{e}_i, \quad i = k+1\,[n],$$

and using the change of variable $\mathbf{x} = A^H \mathbf{y}$, the Kaczmarz iterations (2.2) are retrieved

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\lambda}{\|a_{i\bullet}\|_2^2} \left[b_i - (\mathbf{a}_{i\bullet}, \mathbf{x}^{(k)}) \right] \mathbf{a}_{i\bullet}^H, \quad i = k+1\,[n],$$

In the following, a forward sweep is defined as a sequence of projections from the first to the last row of the matrix A

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \lambda \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(i-1)} \right) \right) \mathbf{a}_{i\bullet}^H, \quad i = 1, \cdots, n. \tag{2.13}$$

A Kaczmarz forward sweep can be defined as well as follows by extending its application to non square linear systems.

Definition 2.1. *Let A be a given $m \times n$ matrix in $\mathbb{K}$, $\mathbf{b}$ a given vector of $\mathbb{K}^m$, $\mathbf{x}$ a vector in $\mathbb{K}^n$ and $\lambda \in I_\lambda =]0, 2[$ the real-valued relaxation parameter, a Kaczmarz forward sweep is defined as the function* $\mathbf{FKACZ}$

$$\begin{array}{llll} \mathbf{FKACZ}: & \mathcal{M}_{m,n}(\mathbb{K}), \mathbb{K}^m, I_\lambda, \mathbb{K}^n & \longrightarrow & \mathbb{K}^n \\ & A, \mathbf{b}, \lambda, \mathbf{x} & \longmapsto & \mathbf{y}, \end{array}$$

which takes $A, \mathbf{b}, \lambda$ and $\mathbf{x}$ and computes $\mathbf{y} \in \mathbb{K}^n$ following the projections

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \lambda \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(i-1)} \right) \right) \mathbf{a}_{i\bullet}^H, \quad i = 1, \cdots, m,$$

using $\mathbf{x}$ as an initial iterate. The matrix A and the right-hand side $\mathbf{b}$ are assumed to be previously normalized through the GRS operation (2.4).

A backward sweep is similar to a forward sweep, but traversing the equations in the reverse order

$$\mathbf{x}^{(n-i+1)} = \mathbf{x}^{(n-i)} + \lambda \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(n-i)} \right) \right) \mathbf{a}_{i\bullet}^H, \quad i = n, \cdots, 1, \tag{2.14}$$

A Kaczmarz backward sweep can be defined as well as follows by extending its application to non square linear systems.

Definition 2.2. *Let A be a given $m \times n$ matrix in $\mathbb{K}$, $\mathbf{b}$ a given vector of $\mathbb{K}^m$, $\mathbf{x}$ a vector in $\mathbb{K}^n$ and $\lambda \in I_\lambda =]0,2[$ the real-valued relaxation parameter, a Kaczmarz backward sweep is defined as the function* $\mathbf{BKACZ}$

$$\begin{aligned} \mathbf{BKACZ}: \quad & \mathcal{M}_{m,n}(\mathbb{K}), \mathbb{K}^m, I_\lambda, \mathbb{K}^n && \longrightarrow \mathbb{K}^n \\ & A, \mathbf{b}, \lambda, \mathbf{x} && \longmapsto \mathbf{y}, \end{aligned}$$

which takes $A, \mathbf{b}, \lambda$ and $\mathbf{x}$ and computes $\mathbf{y} \in \mathbb{K}^n$ following the projections

$$\mathbf{x}^{(n-i+1)} = \mathbf{x}^{(n-i)} + \lambda \left(b_i - \left(\mathbf{a}_{i\bullet}, \mathbf{x}^{(n-i)} \right) \right) \mathbf{a}_{i\bullet}^H, \quad i = m, \cdots, 1,$$

using $\mathbf{x}$ as an initial iterate. The matrix A and the right-hand side $\mathbf{b}$ are assumed to be previously normalized through the GRS operation (2.4).

A double sweep is defined as a forward sweep followed by a backward sweep. It can be defined as the function $\mathbf{DKACZ}$ which is introduced below.

Definition 2.3. *Let A be a given $m \times n$ matrix in $\mathbb{K}$, $\mathbf{b}$ a given vector of $\mathbb{K}^m$, $\mathbf{x}$ a vector in $\mathbb{K}^n$ and $\lambda \in I_\lambda =]0,2[$ the real-valued relaxation parameter, a Kaczmarz double sweep is defined as the function $\mathbf{BKACZ}$ which takes $A, \mathbf{b}, \lambda$ and $\mathbf{x}$ and computes $\mathbf{y} \in \mathbb{K}^n$ such that*

$$\begin{aligned} \mathbf{DKACZ}: \quad & \mathcal{M}_{m,n}(\mathbb{K}), \mathbb{K}^m, I_\lambda, \mathbb{K}^n && \longrightarrow \mathbb{K}^n \\ & A, \mathbf{b}, \lambda, \mathbf{x} && \longmapsto \mathbf{y} = \mathbf{BKACZ}(A, \mathbf{b}, \lambda, \mathbf{FKACZ}(A, \mathbf{b}, \lambda, \mathbf{x})). \end{aligned}$$

The matrix A and the right-hand side $\mathbf{b}$ are assumed to be previously normalized through the GRS operation (2.4).

Performing successive Kaczmarz double sweep operations is equivalent to the Symmetric Successive Over Relaxation (SSOR) method applied to the normal equations (2.8). Symmetric SOR steps consist of the SOR step followed by a backward SOR step (Saad, 2003)

$$\begin{aligned} (D + \lambda E)\mathbf{x}^{(k+1/2)} &= \left[(1-\lambda)D - \lambda F\right] \mathbf{x}^{(k)} + \lambda \mathbf{b}, \\ (D + \lambda F)\mathbf{x}^{(k+1)} &= \left[(1-\lambda)D - \lambda E\right] \mathbf{x}^{(k+1/2)} + \lambda \mathbf{b}. \end{aligned}$$

The backward SOR inner iteration steps can be written as

$$x_i^{(k+1)} = x_i^{(k+1/2)} + \frac{\lambda}{a_{ii}} \left[b_i - \sum_{j=i}^{n} a_{ij} x_j^{(k+1)} - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1/2)} \right], \quad i = n - k\,[n+1]. \tag{2.15}$$

Starting from row n, the backward SOR inner iteration writes

$$\begin{aligned} x_n^{(1)} &= x_n^{(1/2)} + \frac{\lambda}{a_{nn}} \left[b_n - \sum_{j=i}^{n} a_{nj} x_j^{(1/2)} \right], \\ &= x_n^{(1/2)} + \frac{\lambda}{a_{nn}} \left[b_n - (A\mathbf{x}^{(1/2)})_n \right], \\ &= x_n^{(1/2)} + \frac{\lambda}{a_{nn}} \left[b_n - (\mathbf{a}_{n\bullet}, \mathbf{x}^{(1/2)}) \right]. \end{aligned}$$

The coefficients of the $\mathbf{x}^{(k+1/2)}$ are updated from the last coefficient n to the i-th one in a reverse order, therefore the coefficients $x_j^{(k+1/2)}$ for $j = i, \ldots, n$ remain the same from the iteration $k + 1/2$ to the iteration $k + 1$

$$x_j^{(k+1/2)} = x_j^{(k)}, \quad j = i, \ldots, n, \quad i = k + 1\,[n]\,.$$

Thus, at the iteration $k + 1$ we have

$$\sum_{j=i}^{n} a_{ij} x_j^{(k+1)} = \sum_{j=i}^{n} a_{ij} x_j^{(k+1/2)}. \tag{2.16}$$

Using the above equation (2.16), the backward SOR iteration (2.15) can be expressed as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k+1/2)} + \frac{\lambda}{a_{ii}} \left[b_i - (\mathbf{a}_{i\bullet}, \mathbf{x}^{(k+1/2)}) \right] \mathbf{e}_i, \quad i = n - k\,[n+1]\,. \tag{2.17}$$

Using the above equation (2.17), it is easy to check that the backward SOR method (2.15) applied to the normal equations (2.8) is equivalent to the backward Kaczmarz sweeps.

2.1.2 The CGMN method

The CGMN method introduced by Björck and Elfving (1979) allows to accelerate the Kaczmarz iterations using the conjugate gradient method. The linear system (2.1) is transformed into a positive, semi-definite and hermitian linear system through the Kaczmarz iterations performed in double sweeps. The latter is solved with the CG method.

The Kaczmarz projection on the ith row of the linear system (2.1) can be formulated in matrix form using the following definition.

Definition 2.4. *Given the linear system* (2.1)*, the Kaczmarz ith row projection operation is defined by*

$$\mathbf{x}^{(i)} = Q_i(A)\mathbf{x}^{(i-1)} + R_i(A)\mathbf{b}.$$

where the matrices $Q_i(A)$ and $R_i(A)$ are given by

$$\begin{aligned} Q_i(A) &= I - \lambda A \mathbf{e}_i \mathbf{e}_i^H A^H = I - \lambda \mathbf{a}_{i\bullet} \mathbf{a}_{i\bullet}^H, \\ R_i(A) &= \lambda A \mathbf{e}_i \mathbf{e}_i^T = \lambda \mathbf{a}_{i\bullet} \mathbf{e}_i^T. \end{aligned}$$

The matrix I is the identity and λ is the relaxation parameter introduced by the Kaczmarz method. By construction, the matrices $Q_i(A)$ for $i = 1, \ldots, n$ are hermitian.

Using the definition 2.4, the double sweep operations can be expressed as

$$\mathbf{y} = Q(A)\mathbf{x} + R(A)\mathbf{b}, \quad \mathbf{y} \in \mathbb{K}^n,$$

where

$$Q(A) = Q_1(A) \ldots Q_n(A) Q_n(A) \ldots Q_1(A),$$

and

$$R(A) = \prod_{j=1}^{n} Q_j(A) \left(\sum_{i=1}^{n} \left(\prod_{j=n}^{i+1} Q_j(A) \right) R_i(A) \right) + \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} Q_j(A) \right) R_i(A).$$

Remark 2.3. *The matrix $Q(A)$ is hermitian as a product of hermitian matrices $Q_i(A)$ for $i = 1, \dots, n$.*

With the previous notations, the Kaczmarz method applied with double sweeps can be written as

$$\mathbf{x}^{(k+1)} = Q(A)\mathbf{x}^{(k)} + R(A)\mathbf{b}. \tag{2.18}$$

At convergence, i.e. when $\mathbf{x}^{(k)} = \mathbf{x}^\star = A^{-1}\mathbf{b}$, the equation (2.18) becomes

$$\mathbf{x}^\star = Q(A)\mathbf{x}^\star + R(A)\mathbf{b},$$

and therefore, the solution $\mathbf{x}^\star$ of the problem (2.1) is the solution as well of

$$(I - Q(A))\mathbf{x} = R(A)\mathbf{b}, \tag{2.19}$$

which is equivalent to considering the problem (2.18) as a fixed-point iteration problem.

Theorem 2.1. *The matrix $I - Q(A)$ is hermitian positive semi-definite.*

Proof. By construction, the matrix $Q(A)$ is hermitian (see remark 2.3). We have immediately that $I - Q(A)$ is hermitian. We prove that $I - Q(A)$ is positive semi-definite. We define the matrix $P_i(A)$ for $i = 1, \dots, n$ by

$$P_i(A) = A\mathbf{e}_i\mathbf{e}_i^H A^H,$$

is the orthogonal projector onto $\mathbf{a}_{i\bullet}$. Indeed, for $i = 1, \dots, n$

$$\begin{aligned} P_i^2(A) &= A\mathbf{e}_i\mathbf{e}_i^H A^H A\mathbf{e}_i\mathbf{e}_i^H A^H, \\ &= A\mathbf{e}_i(A\mathbf{e}_i)^H(A\mathbf{e}_i)\mathbf{e}_i^H A^H. \end{aligned}$$

With A previously normalized (see remark 2.1), we have

$$(A\mathbf{e}_i)^H(A\mathbf{e}_i) = \|A\mathbf{e}_i\|_2^2 = 1,$$

it follows that

$$\begin{aligned} P_i^2(A) &= A\mathbf{e}_i\mathbf{e}_i^H A^H, \\ &= P_i(A). \end{aligned}$$

Therefore, $P_i(A)$ has its eigenvalues in $\{0, 1\}$. It follows that the eigenvalues of the matrix $Q_i(A) = I - \lambda P_i(A)$ are 1 and $1 - \lambda$ for $i = 1, \dots, n$. Thus, for $0 < \lambda < 2$, we have

$$\rho(Q_i(A)) \leq 1, \quad i = 1, \dots, n \tag{2.20}$$

where $\rho(M)$ denotes the spectral radius of the matrix M. In the following, the definition of the matrix norm is introduced.

Definition 2.5. *Given a square complex or real matrix A in $\mathcal{M}_n(\mathbb{K})$, a matrix norm $||A||$ is a non-negative real number associated with A having the properties*

- *for $A \neq 0_{n,n}$, $||A|| > 0$,*
- *$||A|| = 0$ if and only if $A = 0_{n,n}$,*

- $||kA|| = |k|\ ||A||$ *for any scalar* $k \in \mathbb{K}$,
- $||A + B|| \leq ||A|| + ||B||$ *for any matrix* $B \in \mathcal{M}_n(\mathbb{K})$,
- $||AB|| \leq ||A||\ ||B||$ *for any matrix* $B \in \mathcal{M}_n(\mathbb{K})$,
- *for* A *hermitian and using the* L_2*-norm* $||\cdot||_2$, *we have* $\rho(A) = ||A||_2$.

Thus, using the above definition 2.5 and equation (2.20), we have

$$\rho(Q) \leq 1.$$

The spectrum of the matrix Q is contained in $[-1, 1]$ and therefore, the spectrum of $I - Q(A)$ is contained in $[0, 2]$. □

Thus, the Conjugate Gradient method can be applied straightforwardly to the system (2.19). This method is called CGMN and is presented in Algorithm 2.1.

Algorithm 2.1 CGMN

```
$\mathbf{x}^{(0)} \in \mathbb{K}^n$
$\mathbf{r}^{(0)} = R(A)\mathbf{b} - (I - Q(A))\mathbf{x}^{(0)} = \mathbf{DKACZ}(\mathbf{A}, \mathbf{b}, \lambda, \mathbf{x}^{(0)}) - \mathbf{x}^{(0)}$
                                        ▷ Relaxed Kaczmarz double sweep operations
$\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$
for $i = 1, ...,$ until convergence do
    $\mathbf{q}^{(i)} = (I - Q(A))\mathbf{p}^{(i)} = \mathbf{p}^{(i)} - \mathbf{DKACZ}(\mathbf{A}, \mathbf{0_n}, \lambda, \mathbf{p}^{(i)})$
                                        ▷ Relaxed Kaczmarz double sweep operations with $\mathbf{b} = \mathbf{0}_n$
    $\alpha = \left(\mathbf{r}^{(i)}, \mathbf{r}^{(i)}\right) / \left(\mathbf{p}^{(i)}, \mathbf{q}^{(i)}\right)$
    $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha\mathbf{p}^{(i)}$
    $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha\mathbf{q}^{(i)}$
    $\beta = \left(\mathbf{r}^{(i+1)}, \mathbf{r}^{(i+1)}\right) / \left(\mathbf{r}^{(i)}, \mathbf{r}^{(i)}\right)$
    $\mathbf{p}^{(i+1)} = \mathbf{r}^{(i+1)} + \beta\mathbf{p}^{(i)}$
endfor
```

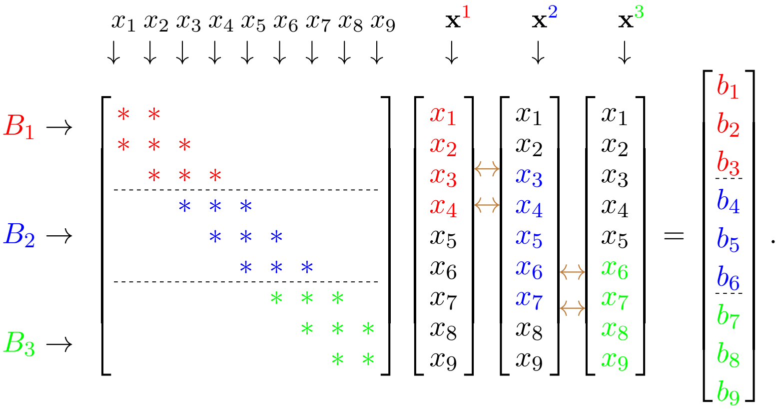
2.1.3 The CARP method

As the double sweep operations are intrinsically sequential, Gordon and Gordon (2005) suggest to break the Kaczmarz iterations performed in double sweeps using a block parallel method called the component-averaged row projections (CARP) method. This is the main motivation of the CARP method. This method takes inspiration from domain decomposition methods. The computation domain is partitioned into slices, which is equivalent to dividing the linear system into blocks of equations. Each processor is assigned with a block of equations and performs the sequence of sweeps within the block's equations. After each sweep (forward or backward) the local solutions which are computed by each processor are merged together by averaging to form the next iterate. The block decomposition and the averaging operation are presented in this paragraph.

Consider the linear system (2.1) which is divided into $t \in \mathbb{N} \setminus \{0\}$ blocks. In practice, t is the number cores used to perform the computations. This partition is denoted by $\mathscr{B} = \{B_1, \ldots, B_t\}$ where B_q denotes the block q of the linear system. One or more unknowns can be shared by the blocks. For

$1 \leqslant j \leqslant n$, we denote the set of indexes of row blocks which share the unknown x_j by I_j. The size of I_j is denoted by $s_j = |I_j|$.

For instance, take a simple example which is illustrated by a linear system of size 9 divided into 3 blocks. The blocks are denoted by $\mathscr{B} = \{B_1, B_2, B_3\}$ and the unknowns by the vector $(x_1, x_2, \dots, x_9)$. The coefficients of the right hand side are denoted by $(b_1, b_2, \dots, b_9)$.



The solution vector $\mathbf{x}$ is duplicated for each block yielding the local solution vectors $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$. One can see that a sweep into the block B_1 acts on the unknowns x_1, x_2, x_3, x_4 of the local solution vector $\mathbf{x}^1$ and leaves the remaining unknowns x_5, x_6, x_7, x_8 and x_9 unchanged. Similarly a sweep in the block B_2 acts only on the unknowns x_3, x_4, x_5, x_6, x_7 of the local solution vector $\mathbf{x}^2$ and a sweep into the block B_3 acts only on the unknowns x_6, x_7, x_8, x_9 of the local solution vector $\mathbf{x}^3$. Therefore, the sweeps in the blocks B_1 and B_2 act on the same unknowns x_3, x_4 held by $\mathbf{x}^1$ and x_3, x_4 held by $\mathbf{x}^2$. Similarly, the unknowns x_6, x_7 held by $\mathbf{x}^2$ and x_6, x_7 held by $\mathbf{x}^3$ are shared by the blocks B_2 and B_3.

Remark 2.4. *In practice, the solution vector* $\mathbf{x}$ *is not duplicated entirely. It is partitioned following the block division of linear system and with respect to the smallest and largest column indexes of the coefficients of the block matrix. For instance, following the simple example which is described above, the local solution vectors are given by*

$$\mathbf{x}^1 = (x_1, x_2, x_3, x_4), \quad \mathbf{x}^2 = (x_3, x_4, x_5, x_6, x_7), \quad \mathbf{x}^3 = (x_6, x_7, x_8, x_9).$$

Therefore, the sets I_j for $j = 1, \dots, 9$ can be defined as

$$I_1 = I_2 = \{1\},\ I_3 = I_4 = \{1, 2\},\ I_5 = \{2\},\ I_6 = I_7 = \{2, 3\},\ I_8 = I_9 = \{3\},$$

and consequently the size of I_j for $j = 1, \dots, 9$ denoted by s_j are given by

$$s_1 = s_2 = 1,\ s_3 = s_4 = 2,\ s_5 = 1,\ s_6 = s_7 = 2,\ s_8 = s_9 = 1.$$

The following definition explains the averaging operation performed in the CARP method

Definition 2.6. *For* $t \in \mathbb{N} \setminus \{0\}$ *let* $\mathscr{B} = \{B_1, \dots, B_t\}$ *be the decomposition of the linear system* (2.1) *as described above. The component-averaged operator relative to* $\mathscr{B}$ *is a mapping*

$$\mathrm{CA}_{\mathscr{B}} : (\mathbb{K}^n)^t \to \mathbb{K}^n$$

such that for $\left(\mathbf{y}^1,\ldots,\mathbf{y}^t\right) \in \left(\mathbb{K}^n\right)^t$, $\mathrm{CA}_{\mathscr{B}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^t\right)$ *is the vector in* $\mathbb{K}^n$ *whose* j^{th} *coefficient is given by*

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^t\right)_j = \frac{1}{s_j}\sum_{q\in I_j} y_j^q, \tag{2.21}$$

where y_j^q *is the* j^{th} *coefficient of* $\mathbf{y}^q$ *for* $1 \leqslant q \leqslant t$.

The application of the Component-averaged operator to the simple case of size 9 gives

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_1 = x_1,\ \mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_2 = x_2,$$

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_3 = \frac{x_3+x_3}{2},\ \mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_4 = \frac{x_4+x_4}{2},$$

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_5 = x_5,$$

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_6 = \frac{x_6+x_6}{2},\ \mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_7 = \frac{x_7+x_7}{2},$$

$$\mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_8 = x_8,\ \mathrm{CA}_{\mathscr{B}}\left(\mathbf{x}^1,\mathbf{x}^2,\mathbf{x}^3\right)_9 = x_9,$$

which after simplification writes in matrix form as

$$\left[\begin{array}{ccccccccc} * & * & & & & & & & \\ * & * & * & & & & & & \\ & * & * & * & & & & & \\ \hdashline & & * & * & * & & & & \\ & & & * & * & * & & & \\ & & & & * & * & * & & \\ \hdashline & & & & & * & * & * & \\ & & & & & & * & * & * \\ & & & & & & & * & * \end{array}\right] \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8\\ x_9 \end{bmatrix} \begin{matrix} \\ \\ \leftrightarrow\\ \leftrightarrow\\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8\\ x_9 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \leftrightarrow\\ \leftrightarrow\\ \\ \\ \end{matrix} \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8\\ x_9 \end{bmatrix} \overset{\underline{\mathrm{CA}}}{\longrightarrow} \left[\begin{array}{c} x_1\\ x_2\\ \mathrm{CA}(\mathbf{x}^1,\mathbf{x}^2)_3\\ \hdashline \mathrm{CA}(\mathbf{x}^1,\mathbf{x}^2)_4\\ x_5\\ \mathrm{CA}(\mathbf{x}^2,\mathbf{x}^3)_6\\ \hdashline \mathrm{CA}(\mathbf{x}^2,\mathbf{x}^3)_7\\ x_8\\ x_9 \end{array}\right] \rightarrow \left[\begin{array}{c} x_1\\ x_2\\ (x_3+x_3)/2\\ \hdashline (x_4+x_4)/2\\ x_5\\ (x_6+x_6)/2\\ \hdashline (x_7+x_7)/2\\ x_8\\ x_9 \end{array}\right].$$

The linear system (2.1) is transformed into a system of equations in some superspace $\mathbb{K}^s$ with

$$s = \sum_{i=1}^{n} s_i,$$

where the *averaging* equations are added to the initial system. The equations belonging to the different blocks do not have common variables anymore. Thus performing the Kaczmarz sweeps in parallel in $\mathbb{K}^s$ is equivalent to performing the Kaczmarz sweeps in serial in $\mathbb{K}^n$.

Consider a vector $\mathbf{x}$ in $\mathbb{K}^n$ where a number r, $1 \leq r \leq n$, of its coefficients are shared by two or more blocks. These common coefficients are denoted by $x_1,\ldots,x_r$. Therefore,

$$s_1,\cdots,s_r \geq 2,$$

and

$$s_{r+1} = \cdots = s_n = 1.$$

The expansion mapping $E : \mathbb{K}^n \longrightarrow \mathbb{K}^s$ allows to expend the vector $\mathbf{x}$ from $\mathbb{K}^n$ into the superspace $\mathbb{K}^s$ such that

$$E(x_1, \ldots, x_n) = (y_{1,1}, \cdots, y_{1,s_1}, \ldots, y_{r,1}, \cdots, y_{r,s_r}, y_{r+1}, \cdots, y_n),$$

where for $j = 1, \cdots, r$

$$y_{j,1}, \ldots, y_{j,s_j} = x_j,$$

and for $j = r+1, \cdots, n$

$$y_j = x_j.$$

The inverse of the expansion mapping E is defined as

$$\begin{array}{rll} E^{-1} : & \mathbb{K}^s & \longrightarrow \mathbb{K}^n \\ & \mathbf{y} & \longmapsto \mathbf{x}, \end{array}$$

such that

$$x_j = \begin{cases} y_{j,1}, & \text{for } 1 \leq j \leq r, \\ y_j, & \text{if } r < j \leq n. \end{cases}$$

Thus, the index set of the blocks containing a nonzero coefficient x_j denoted by I_j has s_j elements by definition. It can be written as

$$I_j = \{i_{j,1}, i_{j,2}, \cdots, i_{j,s_j}\}.$$

Therefore, for each $1 \leq j \leq n$ and $1 \leq l \leq s_j$, the variables $y_{j,l}$ represent a copy of x_j in the block $B_{i_{j,l}}$ following the mapping E.

Let $B'_1, B'_2, \cdots, B'_t$ denote the blocks of equations derived from $B_1, \cdots, B_t$ using the above replacements and $\mathscr{B}' = \bigcup_{q=1}^{t} B'_q$. This set of equation is defined in $\mathbb{K}^s$. The system of equations resulting from $\mathscr{B}'$ is denoted by

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

Performing CARP in $\mathbb{K}^s$ is presented in the following Algorithm 2.2

Algorithm 2.2 CARP in $\mathbb{K}^s$

```
$\mathbf{x}^0 = (x_1^0, x_2,^0 \cdots, x_n^0) \in \mathbb{K}^n$,
$\mathbf{y}^0 = E(\mathbf{x}^0) \in \mathbb{K}^s$,
for $q = 1, ..., t$ (independently for each bloc $B'_q$) do
    for $k = 1, ...,$ until convergence do
        $\mathbf{z}^q = \mathbf{FKACZ}(A'^q, \mathbf{b}'^q, \lambda, \mathbf{y}^k)$,
        For $1 \leq j \leq r$, set $y_{j,1}^k, \ldots, y_{j,s_j}^k = \frac{1}{s_j} \sum_{l=1}^{s_j} z_{j,l}^{q_{j,l}}$ and for $r < j \leq n$, set $y_j^k = z_{q_j,1}^{q_j}$,
    endfor
endfor
Output: $\mathbf{x} = E^{-1}(\mathbf{y}^{k+1}) \in \mathbb{K}^n$
```

Other than the initial and final steps involving the expansion mapping E and its inverse, Algorithm 2.2 can be executed iteratively in parallel in $\mathbb{K}^s$ since the blocks B'_q dot not have common variables.

Using the component-averaged operator definition 2.6, the CARP method is easily formulated in $\mathbb{K}^n$ using the forward sweep definition (2.1). One forward sweep of CARP can be defined as the function $\mathbf{FCARP}$ which takes the block decomposition $\mathscr{B}$, a vector $\mathbf{x} \in \mathbb{K}^n$ and computes $\mathbf{y} \in \mathbb{K}^n$ such that

Algorithm 2.3 $\mathbf{y} = \mathbf{FCARP}(\mathscr{B}, \lambda, \mathbf{x})$

```
Let $\mathbf{z}^t \in \mathbb{K}^n$ be local vectors for $i = 1, \dots, t$,
for $q = 1, ..., t$ (independently for each bloc $B_q$) do
    $\mathbf{z}^q = \mathbf{FKACZ}(A^q, \mathbf{b}^q, \lambda, \mathbf{x})$,
endfor
$\mathbf{y} = \mathrm{CA}_{\mathscr{B}}(\mathbf{z}^1, \dots, \mathbf{z}^t)$.
```

Using the above definition of the forward sweep $\mathbf{FCARP}$ performed in parallel (see Algorithm 2.3), the CARP method performed in several sweeps is presented in Algorithm 2.4.

Algorithm 2.4 CARP in forward sweeps

```
1: $\mathbf{x}^{(0)} \in \mathbb{K}^n$
2: for $k = 0, 1, ...,$ until convergence do
3:     $\mathbf{x}^{(k+1)} = \mathbf{FCARP}(\mathscr{B}, \lambda, \mathbf{x}^{(k)})$,
4: endfor
```

Each block q in $\mathscr{B}$ of the linear system (2.1) is associated with the system of equations of size $n_q \times n$ with $n_q = \frac{n}{t}$

$$A^q \mathbf{x}^q = \mathbf{b}^q,$$

where

$$A^q \in M_{n_q,n}(\mathbb{K}), \quad \mathbf{x}^q \in \mathbb{K}^n, \quad \mathbf{b}^q \in \mathbb{K}^{n_q}.$$

The relaxed Kaczmarz method is therefore parallelized by performing independently the forward sweeps within each block q of $\mathscr{B}$. This is indicated at line 3 in Algorithm 2.4. After each sweep the component-averaged $\mathrm{CA}_{\mathscr{B}}$ operator is applied to compute the next iterate $\mathbf{x}^{(k+1)}$.

The CARP method is introduced in double sweeps as well. The CARP double sweep steps consist in

1. perform one forward sweep of CARP,
2. apply the Component-Averaged operator,
3. perform one backward sweep of CARP,
4. apply the Component-Averaged operator.

One backward sweep of CARP can be defined as the function $\mathbf{BCARP}$ which takes the block decomposition $\mathscr{B}$, a vector $\mathbf{x} \in \mathbb{K}^n$ and computes $\mathbf{y} \in \mathbb{K}^n$ following Algorithm 2.5.

A backward sweep of CARP consists simply in applying the CARP sweep by performing the row projections in reverse order from the last row to the first row. Therefore, the double sweep operation in

Algorithm 2.5 $\mathbf{y} = \mathbf{BCARP}(\mathscr{B}, \lambda, \mathbf{x})$

```
Let $\mathbf{z}^t \in \mathbb{K}^n$ be local vectors for $i = 1, \dots, t$,
for $q = 1, ..., t$ (independently for each bloc $B_q$) do
    $\mathbf{z}^q = \mathbf{BKACZ}(A^q, \mathbf{b}^q, \lambda, \mathbf{x})$,
endfor
$\mathbf{y} = \mathrm{CA}_{\mathscr{B}}(\mathbf{z}^1, \dots, \mathbf{z}^t)$.
```

CARP can be defined as the function $\mathbf{BCARP}$ which takes the block decomposition $\mathscr{B}$, the relaxation parameter $\lambda \in I_\lambda =]0, 2[$, a vector $\mathbf{x} \in \mathbb{K}^n$ and computes $\mathbf{y} \in \mathbb{K}^n$ such that

$$\mathbf{y} = \mathbf{DCARP}(\mathscr{B}, \lambda, \mathbf{x}) = \mathbf{BCARP}(\mathscr{B}, \lambda, \mathbf{FCARP}(\mathscr{B}, \lambda, \mathbf{x})). \tag{2.22}$$

The CARP method performed in double sweeps is presented in Algorithm 2.6.

Algorithm 2.6 CARP in double sweeps

```
$\mathbf{x}^{(0)} \in \mathbb{K}^n$
for $k = 0, 1, ...,$ until convergence do
    $\mathbf{x}^{(k+1)} = \mathbf{DCARP}(\mathscr{B}, \lambda, \mathbf{x}^{(k)})$
endfor
```

2.1.4 The CARP-CG method

We have seen that the sequential Kaczmarz sweeps are the starting point of all the above iterative methods. The Kaczmarz method is accelerated by conjugate gradient method and by performing the Kaczmarz projection in double sweeps. The resulting method is sequential and is called CGMN. Alternatively, the Kaczmarz method is parallelized into the CARP method. The combination of the CARP method performed in double sweeps and the conjugate gradient yields the CARP-CG method (Gordon and Gordon, 2010b). CARP is easily formulated in the double sweep form in Algorithm 2.6.

Therefore, the CARP-CG method is written straightforwardly by replacing the sequential Kaczmarz iterations performed in double sweeps in CGMN (see lines 2 and 6 in Algorithm 2.1) by the block parallel version which is presented in equation (2.22). The CARP-CG method is presented in Algorithm 2.7.

2.1.5 Computational and memory costs

Consider the linear system (2.1) with A a sparse matrix with s nonzero coefficients per row. A double sweep requires $4sn$ operations (projections on sparse rows), with s being in $O(1)$. This is the computational cost required for two sparse matrix-vector products. Using the double sweep operation, the complexity of the matrix-vector product $(I - Q(A))\mathbf{x}$ is thus in $O(n)$, despite the matrix $I - Q(A)$ is dense.

Therefore, the computational cost of one iteration of CARP-CG in terms of matrix-vector products is equal to the cost of one iteration of Bi-CGSTAB or CGNR, and is twice the cost of one iteration of GMRES. As the CARP-CG method relies on the conjugate gradient algorithm, it performs the same

Algorithm 2.7 CARP-CG

```
$\mathbf{x}^{(0)} \in \mathbb{K}^n$,
Given $\mathscr{B}$ the block decomposition of linear system,
$\mathbf{r}^{(0)} = \mathbf{DCARP}(\mathscr{B}, \lambda, \mathbf{x}^{(0)})$,
$\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$,
for $i = 1, ...,$ until convergence do
    $\mathbf{q}^{(i)} = \mathbf{DCARP}(\mathscr{B}, \lambda, \mathbf{p}^{(i)})$        ▷ Apply $\mathbf{DCARP}$ with $\mathbf{b} = \mathbf{0}_n$
    $\alpha = \left(\mathbf{r}^{(i)}, \mathbf{r}^{(i)}\right) / \left(\mathbf{p}^{(i)}, \mathbf{q}^{(i)}\right)$
    $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha \mathbf{p}^{(i)}$
    $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \alpha \mathbf{q}^{(i)}$
    $\beta = \left(\mathbf{r}^{(i+1)}, \mathbf{r}^{(i+1)}\right) / \left(\mathbf{r}^{(i)}, \mathbf{r}^{(i)}\right)$
    $\mathbf{p}^{(i+1)} = \mathbf{r}^{(i+1)} + \beta \mathbf{p}^{(i)}$
endfor
```

vector updates and inner products for the computation of the solution, the residual and the direction vectors. In practice, the Kaczmarz double sweep operation updates the current iterate and as the conjugate gradient algorithm requires the knowledge of two successive iterates, therefore, one extra vector is used in the CARP-CG algorithm to store a *copy* of the current iterate.

2.1.6 Spectral analysis of the matrix $I - Q(A)$ for the Helmholtz equation

Consider the 2D Helmholtz equation (1.17) which is recast in the 2D approximation on the square domain of size 900 m. The source term is defined as a Dirac delta function

$$s(x, z) = \delta(x - x_0)\delta(z - z_0),$$

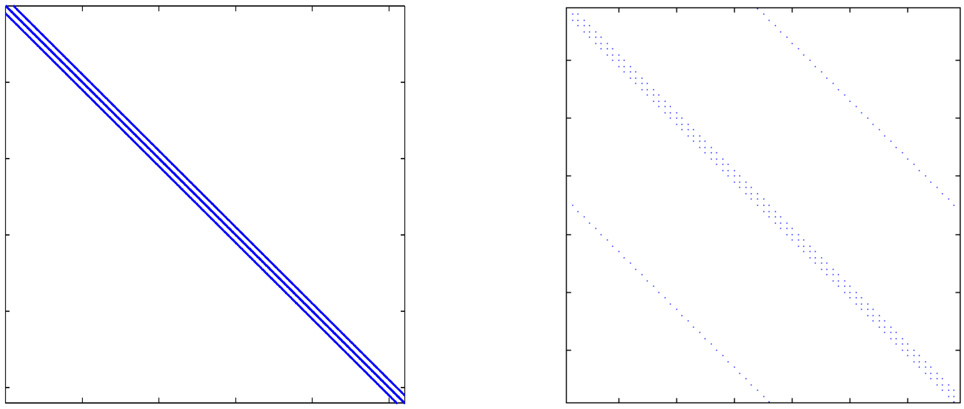
which symbolizes a source point located at $x_0 = 450$ m and $z_0 = 90$ m. Perfectly Matched Layers (PML) are used at all the boundaries to mimic wave propagation in infinite domain. The discretization of this problem is performed using the second-order staggered-grid finite-difference method (see Chapter 1 Section 1.5). The minimum 10 grid points per wavelength are used for the discretization to ensure the accuracy of the discretization scheme. The P-wave velocity is taken as constant $c = 1500$ m/s. The frequency is set to $f = 5$ Hz. $N_{PML} = 10$ grid points are used in each PML layer. The damping coefficient in the PMLs is set to $C_{PML} = 1500$. This yields the linear system

$$A\mathbf{x} = \mathbf{b}. \tag{2.23}$$

where A is a sparse complex-valued indefinite and ill-conditioned impedance matrix of size 2601 with 12901 non-zero values, the right-hand side $\mathbf{b}$ represents the Dirac source term and $\mathbf{x}$ is the pressure wavefield.

Remark 2.5. *We may refer to this simple "toy" problem as Problem 2.1.6. We shall use it later for the study of the preconditioning effects on the matrix A.*

The sparsity pattern of the matrix A is presented in Figure 2.5a. A zoom on the diagonal of the matrix is performed in the Figure 2.5b.



(a) Sparsity pattern of A (b) Zoom on the diagonal of A

Figure 2.5: Sparsity pattern of the matrix A resulting from the discretization of the 2D Helmholtz equation using the second-order staggered-grid finite-difference method (a) and zoom on the diagonal of the matrix A (b).

The geometric row scaling (2.1) is applied to A. The CGMN method is applied to the system (2.23) using the relaxation parameter $\lambda = 1$. In section 2.1.2, we have seen that the linear system (2.23) is transformed using the Kaczmarz iterations performed in double sweeps into

$$(I - Q(A))\mathbf{x} = R(A)\mathbf{b},$$

where the matrix $I - Q(A)$ is hermitian positive semi-definite.

Remark 2.6. *As the matrix $I - Q(A)$ is computed explicitly for the eigenvalues computation, the size of the domain is reduced to 360 m in order to allow the computations in a reasonable amount of time.*

In Figure 2.6, we show the convergence histories for the solution of this problem using the iterative methods CGMN, CGNR, GMRES and BiCGSTAB. Restarted GMRES(m) is used with $m = 15$. The relative residual

$$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2}{\|\mathbf{b}\|_2},$$

is plotted as a function of the iteration number on a semi-log axis.

Clearly, the CGMN method exhibits a better rate of convergence compared to the other iterative solvers. Table 2.1 presents the condition numbers of the matrices A, $A^H A$ and $I - Q(A)$ and it is interesting to note that the matrix $I - Q(A)$ associated with the linear system solved by CGMN is far better conditioned than $A^H A$ which is associated with the linear system solved by CGNR. This is due to the updates performed on the approximation after each Kaczmarz projection whereas in CGNR the approximation is updated after a whole matrix-vector product. A factor 15 is noted.

	A	AA^H	$I - Q(A)$
cond	102.8	10572.7	671.6

Table 2.1: Condition number of the matrices A, AA^H and $I - Q(A)$ on the homogeneous Helmholtz problem 2.1.6.

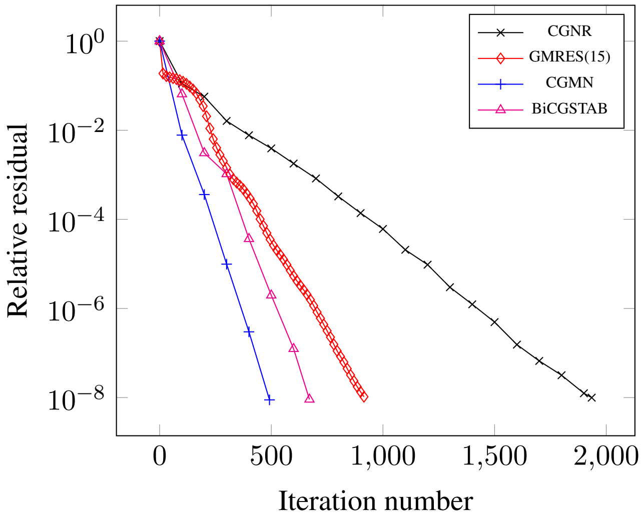
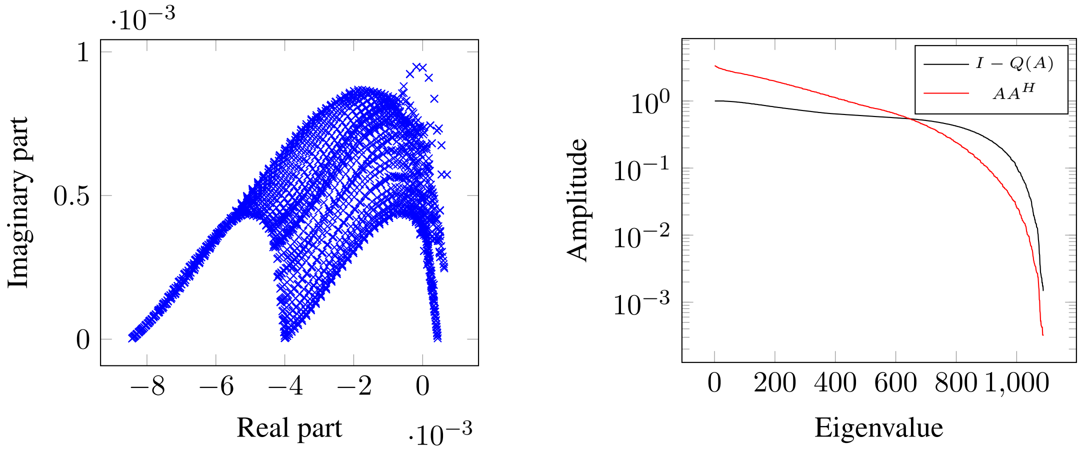


Figure 2.6: Convergence histories for the homogeneous Helmholtz problem 2.1.6 using CGMN, CGNR, GMRES(15) and BiCGSTAB.

It is interesting as well to note that CGMN outperforms GMRES despite the condition number of the matrix $I - Q(A)$ is worse than the one of the matrix A. Indeed, for indefinite matrices with eigenvalues on both sides of the origin. In Figure 2.7a, the eigenvalues of the matrix A lie close to the origin and GMRES is known to have difficulties to converge in such configuration. The spectra of $I - Q(A)$ and AA^H are plotted in Figure 2.7b. The eigenvalues of the matrix $I - Q(A)$ are located between 0 and 2 (see proof of Theorem 2.1) and seems more clustered around 1. The spectrum of the matrix AA^H is less flattened and presents more extreme values compared to the spectrum of $I - Q(A)$ which denotes a larger condition number. As the convergence of the CG method is sensitive to the condition number of the matrix, it requires a larger number of iterations to reach convergence.



(a) Eigenvalues of A.

(b) Eigenvalues of $I - Q(A)$ and AA^H.

Figure 2.7: Eigenvalues of the matrices A, AA^H and $I - Q(A)$ for the homogeneous Helmholtz problem 2.1.6.

2.2 Preconditioning strategy for the CARP-CG method

The purpose of this section is to propose a preconditioning strategy to further enhance the performances of the CARP-CG method by reducing the number of iterations to reach the solution. This preconditioning strategy is dedicated to the frequency-domain wave equations. Unlike standard iterative methods such as CGNR, GMRES or BiCGSTAB, the CARP-CG algorithm is based on a reformulation of the linear system to be solved. Therefore, designing a preconditioner adapted to this method is not straightforward. In the first part, this question is addressed by investigating the different possibilities. Furthermore, the preconditioner which we consider in this work is introduced. This preconditioner is based on the damped frequency-domain wave equation approach. This approach relies on the computation of a sparse approximate inverse preconditioner. The properties of this preconditioner are investigated in this chapter, in the sections 2.4 and 2.3 on the 2D frequency-domain acoustic and elastic wave equations.

2.2.1 How to precondition the CARP-CG method ?

Designing a preconditioning strategy for the CARP-CG method is not an easy task because of the Kaczmarz iterations. Consider the double sweep operation

$$\mathbf{y} = (I - Q(A))\mathbf{x},$$

where for a given vector $\mathbf{x} \in \mathbb{K}^n$ the vector $\mathbf{y} \in \mathbb{K}^n$ is computed.

The first approach is use a preconditioner which preserves the hermitian property of the matrix $I - Q(A)$ which amounts to perform the operation

$$\mathbf{y} = P^H(I - Q(A))P\mathbf{z}, \quad \mathbf{x} = P\mathbf{z}.$$

However, designing a preconditioner for the matrix $I - Q(A)$ cannot be done for the simple reason that this matrix is dense and not formed explicitly and therefore one cannot have access to its coefficients. Consequently, the standard preconditioning techniques which have been discusses in Chapter 1 Section 1.7.3, cannot be used since they require access to the matrix coefficients and as they are designed for sparse matrices as well. For these reasons, both types of preconditioners ("explicit" and "implicit") are out of reach for this first approach.

The second approach to precondition the CARP-CG method is to apply the latter to the system

$$AP\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P\mathbf{y}. \tag{2.24}$$

This strategy is supported by the fact that if the matrix A is unitary, then applying the CARP-CG method amounts to solve a diagonal system.

Theorem 2.2. *If $A \in \mathcal{G}l_n(\mathbb{K})$ is unitary, i.e., $A^H A = AA^H = I$, then $Q(A) = (1-\lambda)^2 I$ and $I - Q(A) = \lambda(2-\lambda) I$ with λ the relaxation parameter.*

Proof. We assume that $A \in \mathcal{G}l_n(\mathbb{K})$ is a unitary matrix $\left(A^H A = AA^H = I\right)$. Therefore, $(\mathbf{a}_{i\bullet})_{i=1,\cdots,n}$ forms an orthonormal basis in $\mathbb{K}^n$. Thus, any vector $\mathbf{x} \in \mathbb{K}^n$ can be decomposed in this basis such that

$$\forall \mathbf{x} \in \mathbb{K}^n, \quad \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{a}_{i\bullet}, \quad \text{with} \quad (\alpha_i)_{i=1,\cdots,n} \in \mathbb{K}^n. \tag{2.25}$$

Following the definition (2.4) of the matrix $Q_i(A)$ and using (2.25), the matrix-vector product of $Q_i(A)$ with $\mathbf{x}$ is given by

$$Q_i(A)\mathbf{x} = \left(I - \lambda \mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\right)\mathbf{x} = \mathbf{x} - \lambda \mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\mathbf{x} = \mathbf{x} - \lambda\alpha_i\mathbf{a}_{i\bullet}.$$

Moreover, for $1 \leq i, j \leq n$ and $i \neq j$

$$\begin{aligned} Q_i(A)Q_j(A)\mathbf{x} &= \left(I - \lambda \mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\right)\left(\mathbf{x} - \lambda\alpha_j\mathbf{a}_{j\bullet}\right), \\ &= \mathbf{x} - \lambda\alpha_j\mathbf{a}_{j\bullet} - \lambda\mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\mathbf{x} - \lambda^2\alpha_j\mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\mathbf{a}_{j\bullet}, \end{aligned}$$

as the $\left(\mathbf{a}_{i\bullet}\right)_{i=1,\cdots,n}$ are an orthonormal basis of $\mathbb{K}^n$, thus

$$Q_i(A)Q_j(A)\mathbf{x} = \mathbf{x} - \lambda\left(\alpha_i\mathbf{a}_{i\bullet} + \alpha_j\mathbf{a}_{j\bullet}\right).$$

Consequently, the expression of the Kaczmarz forward sweep reduces to

$$Q_1(A)\cdots Q_n(A)\mathbf{x} = \mathbf{x} - \lambda\sum_{i=1}^{n}\alpha_i\mathbf{a}_{i\bullet} = \mathbf{x} - \lambda\mathbf{x} = (1-\lambda)\,\mathbf{x}.$$

Similarly, we have for a Kaczmarz backward sweep

$$Q_n(A)\cdots Q_1(A)\mathbf{x} = (1-\lambda)\,\mathbf{x},$$

and finally

$$Q(A)\mathbf{x} = (1-\lambda)^2\,\mathbf{x}, \quad \text{and} \quad I - Q(A) = \lambda\,(2-\lambda)\,I.$$

□

Thus, applying the CARP-CG method to the system (2.24) should give a nearly diagonal system.

One possible way to implement this approach is to design a preconditioner based on the matrix A and perform the iterations

$$\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} + \lambda\left(b_i - \left(\mathbf{a}_{i\bullet}, P\mathbf{y}^{(i-1)}\right)\right)\mathbf{a}_{i\bullet}^H, \quad i = 1, \cdots, n, \tag{2.26}$$

with $\mathbf{x}^{(i)} = P\mathbf{y}^{(i)}$. However, one can easily see that one Kaczmarz row projection requires the computation of the matrix vector product $P\mathbf{y}^{(i-1)}$. Assuming the preconditioner sparse and its application has a complexity $\mathcal{O}(n)$, the complexity of the Kaczmarz double sweep operation is $\mathcal{O}(n^2)$ which cannot be worth considering. On alternative is to rewrite the iterations (2.26) as

$$\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} + \lambda\left(b_i - \left(\mathbf{ap}_{i\bullet}, \mathbf{y}^{(i-1)}\right)\right)\mathbf{ap}_{i\bullet}^H, \quad i = 1, \cdots, n,$$

with $\mathbf{x}^{(i)} = P\mathbf{y}^{(i)}$ and where $\mathbf{ap}_{i\bullet}$ denotes the ith row of the matrix AP. The explicit computation of the matrix-matrix product AP is thus required. Therefore, one needs to be pay a close attention to the fill-in effect induced by the matrix-matrix product. Thus the preconditioner needs to be sparse and consequently, “implicit” type preconditioners such as ILU are discarded as the matrix $\tilde{U}^{-1}\tilde{L}^{-1}$ is dense. In our applications, the matrix A is sparse and banded because of the finite-difference discretization method. Therefore, the preconditioner P needs to have the same properties in order to keep the matrix AP sparse and banded as well.

Without completely discarding the first approach, one can obtain simplifications if the matrix P is assumed to be unitary. In this case, the following theorem can be derived.

Theorem 2.3. *If $P \in \mathcal{G}l_n(\mathbb{K})$ is unitary, i.e., $P^H P = PP^H = I$, then*

$$P^H \left(I - Q(A)\right) P = I - Q(AP).$$

Proof. Let $P \in \mathcal{G}l_n(\mathbb{K})$ be a unitary matrix. Following the definition (2.4) of the matrix $Q_i(A)$, let $Q_i(AP)$ be the ith sweeping matrix resulting from the application of the Kaczmarz row projection to the preconditioned system (2.24)

$$\begin{aligned}
Q_i(AP) &= I - \lambda(\mathbf{ap})_{i\bullet}(\mathbf{ap})_{i\bullet}^H, \\
&= P^H P - \lambda P^H \mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H P, \\
&= P^H \left(I - \lambda \mathbf{a}_{i\bullet}\mathbf{a}_{i\bullet}^H\right) P, \\
&= P^H Q_i(A) P.
\end{aligned}$$

Thus

$$\begin{aligned}
Q(AP) &= Q_1(AP)\cdots Q_n(AP)Q_n(AP)\cdots Q_1(AP), \\
&= P^H Q_1(A)P \cdots P^H Q_n(A)PP^H Q_n(A)P \cdots P^H Q_1(A)P, \\
&= P^H Q_1(A)\cdots Q_n(A)Q_n(A)\cdots Q_1(A)P, \\
Q(AP) &= P^H Q(A)P.
\end{aligned}$$

It results that

$$I - Q(AP) = P^H(I - Q(A))P.$$

□

The Theorem 2.3 shows two interesting results. First, one can circumvent the requirement of accessing the entries of the matrix $I - Q(A)$ for the computation of the preconditioner. This preconditioner can be computed out of the matrix A under the assumption that P is unitary. Therefore, such preconditioner can be computed using standard methods (explicit or implicit approaches) such as AINV or ILU applied to the matrix A. Furthermore, this preconditioner is applied only two times: before and after the double sweep operation. Second, this approach can be seen as a simplification of the second preconditioning strategy which consists in applying the CARP-CG method to the preconditioned system (2.24). This theorem shows that one can approximate the second preconditioning strategy by applying the preconditioner only two times (before and after the double sweep operation) but under the assumption that P is unitary.

Although this assumption may seem unrealistic, one can set up a framework in which it should be "weakly" satisfied. By "weakly", we suggest that in the case where A is a banded and diagonal dominant matrix, its inverse presents an exponential decay of the amplitude of the off-diagonal entries as it is pointed out by Demko et al. (1984). For both preconditioning approaches which are described above, the preconditioner P needs to be computed on a diagonal dominant matrix A. In the context of seismic wave modeling, the matrix A is only banded because of the finite-difference method. The matrix A is made diagonal dominant by considering the wave propagation in a strongly damped medium. This is achieved by using a complex velocity such as the Kolsky model (Kolsky, 1956). In the acoustic approximation, the complex-valued P-wave velocity $\tilde{c}$ is introduced by

$$\tilde{c}(\mathbf{x}) = c(\mathbf{x})\left(1 - \frac{i}{2q_{att}}\right), \tag{2.27}$$

where i is the complex number such that $i^2 = -1$ and q_{att} is the quality factor which controls the attenuation in the medium. This strategy justifies the need compute P from G which is a damped version of A.

In the following, I consider the two preconditioning strategies. We recall that the first strategy consists in solving the preconditioned system

$$P_1^H(I - Q(A))P_1\mathbf{y} = P_1^H R(A)\mathbf{b}, \quad \mathbf{x} = P_1\mathbf{y},$$

and the second strategy where the preconditioned linear system

$$AP_2\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P_2\mathbf{y},$$

is solved using the CARP-CG with a prior computation of the matrix AP_2. In both cases, the preconditioners P_1 and P_2 are computed from G. P_1 shall be computed using the ILUT technique and P_2 using the AINV technique. I shall illustrate the efficiency of these two strategies on the Helmholtz equation using the Marmousi P-wave velocity model.

2.2.2 The ILUT preconditioner for the frequency-domain wave equation

In this part, I describe the method for the computation of a preconditioner for the frequency-domain wave equation which adapted to the first preconditioning strategy. Therefore, the solution of the linear system

$$P_1^H(I - Q(A))P_1\mathbf{y} = P_1^H R(A)\mathbf{b}, \quad \mathbf{x} = P_1\mathbf{y},$$

is investigated where P_1 is computed using the ILUT strategy. The ILUT method belongs to a broader class of widely used preconditioners based on incomplete factorizations (ILU) (Saad, 2003). A factorization is called incomplete if during the process certain fill-in coefficients are dropped, i.e. set to zero. This incomplete factorization gives the lower and upper triangular sparse factors $\tilde{L}$ and $\tilde{U}$ such that

$$A = \tilde{L}\tilde{U} + E,$$

with E a residual or error matrix of the factorization. The ILUT(τ, p) uses a dual dropping strategy which aims at controlling the computational cost and the memory requirements during the factorization and the application of the preconditioner. The drop tolerance τ reduces the computational cost, while the parameter p controls the memory required to store the sparse factors $\tilde{L}$ and $\tilde{U}$, by controlling the number of coefficients kept per row. During the factorization process, the ILUT(τ, p) algorithm drops any entries whose magnitude is smaller than the threshold τ. From the remaining non-zero entries, the p largest coefficients in L and the p largest coefficients in U in addition to the diagonal coefficient are kept. From the L and U factors, we can define the preconditioner

$$P = (\tilde{L}\tilde{U})^{-1} = \tilde{U}^{-1}\tilde{L}^{-1}.$$

In this context, P is not formed explicitly. Indeed, even if $\tilde{L}$ and $\tilde{U}$ are kept sparse through the incomplete factorization process, their inverse are dense matrices. However, the action of P on a vector can be evaluated by solving one upper and one lower triangular sparse systems by forward and backward substitutions. The cost of these operations is $2(p + 1)n$ for sparse triangular matrices with $p + 1$ nonzero values by row. The complexity of applying this preconditioner is then in $O(n)$ as $p << n$. The resulting algorithm is described in Algorithm 2.8.

Algorithm 2.8 ILUT(τ, p)

```
for $i = 1 \dots n$ do
    $\mathbf{w} = \mathbf{a}_{i\bullet}$
    for $k = 1 \dots i-1$ and $w_k \neq 0$ do
        $w_k = w_k / a_{kk}$
        Apply dropping on $w_k$ using the threshold $\tau$
        if $w_k \neq 0$ then
            $w = w - w_k u_{k\bullet}$
        endif
    endfor
    Apply dropping on $w$: keep the $p$ largest entries
    $l_{ij} = w_j$,    for $j = 1, \cdots, i-1$
    $u_{ij} = w_j$,    for $j = i, \cdots, n$
endfor
```

2.2.3 The AINV preconditioner for the frequency-domain wave equation

In this part, I describe a method for the computation of a sparse approximate inverse preconditioner for the frequency-domain wave equation and adapted to the second preconditioning strategy which consists in solving the linear system

$$AP_2\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P_2\mathbf{y},$$

using the CARP-CG. Therefore the design of a preconditioner needs to satisfy the following two requirements

- First, the preconditioner P_2 needs to be computed explicitly as an approximate inverse of A in order to allow the explicit computation of the matrix AP_2.
- Second, the preconditioner P_2 needs to be sparse and banded in order to reduce the fill-in induced by the matrix-matrix product AP_2.

Following these two requirements, standard implicit preconditioners such as the ILU type are discarded. The reason is after performing the incomplete factorization yielding the sparse factors $\tilde{L}$ and $\tilde{U}$, the preconditioner resulting from the computation of $\tilde{U}^{-1}\tilde{L}^{-1}$ gives a dense matrix. Therefore, the combination of the CARP-CG method with such preconditioner is not possible.

A more attractive solution is to consider the Approximate Inverse (AINV) preconditioner (Benzi and Tuma, 1998). The latter belongs to the class of algebraic "explicit" preconditioners in the sense that the action of the preconditioner on a vector is know by a matrix-vector product. This method attempts to approximate A^{-1}, which is dense, with a sparse matrix by performing a sparse incomplete triangular factorization of the matrix A^{-1}.

The computation of the preconditioner is based on an algorithm which computes two sets of vectors $\{\mathbf{z}_i\}_{i=1\cdots n}$, $\{\mathbf{w}_i\}_{i=1\cdots n}$, which are A-biconjugate

$$\mathbf{z}_i A \mathbf{w}_j = 0 \Longleftrightarrow i \neq j.$$

The exact factorization yields two upper triangular matrices Z and W such that

$$W^H A Z = D = \begin{pmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_n \end{pmatrix},$$

where

$$Z = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_n], \quad W = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_n], \quad \text{and} \quad p_i = \mathbf{w}_i A \mathbf{z}_i \neq 0.$$

It follows that

$$A^{-1} = Z D^{-1} W^H.$$

The matrices Z and W whose columns are A-biconjugate can be computed by means of a bi-conjugation process applied to the columns of any non-singular matrices $Z^{(0)}$ and $W^{(0)}$. A convenient choice is to use

$$Z^{(0)} = W^{(0)} = I.$$

For simplicity, only the Z factor is considered. The same procedure is applied for the matrix W by considering the matrix A^H. The resulting algorithm is described in Algorithm 2.9.

Algorithm 2.9 AINV

```
1:  $\mathbf{z}_1^{(0)} = \mathbf{e}_1$
2:  $a_1^{(0)} = a_{11}$
3:  for $i = 2 \dots n$ do
4:      $\mathbf{z}_i^{(0)} = \mathbf{e}_i$
5:      for $j = 1 \dots i-1$ do
6:          $a_i^{(j-1)} = \mathbf{a}_{j\bullet}\mathbf{z}_i^{(j-1)}$
7:          $\mathbf{z}_i^{(j)} = \mathbf{z}_i^{(j-1)} - \frac{a_i^{(j-1)}}{a_j^{(j-1)}}\mathbf{z}_j^{(j-1)}$
8:      endfor
9:      $a_i^{(i-1)} = \mathbf{a}_{i\bullet}\mathbf{z}_i^{(i-1)}$
10: endfor
```

The computation of a sparse approximation of the factors Z, W and D can be performed following two strategies inspired from the ILUT(τ, p) preconditioner.

- The first strategy consists in performing the linear combination in the internal loop indicated at line 5 in Algorithm 2.9 using only the last ν vectors. Therefore, the internal loop at line 5 in Algorithm 2.9 becomes

Algorithm 2.10 Reduced internal loop

```
for $j = i - \nu \ldots i - 1$ do
    $a_i^{(j-1)} = A_{j\bullet}\mathbf{z}_i^{(j-1)}$
    $\mathbf{z}_i^{(j)} = \mathbf{z}_i^{(j-1)} - \frac{a_i^{(j-1)}}{a_j^{(j-1)}}\mathbf{z}_j^{(j-1)}$
    $a_i^{(i-1)} = A_{i\bullet}\mathbf{z}_i^{(i-1)}$
endfor
```

- The second strategy is to discard the small value coefficients of the vectors $\mathbf{z}_i$ and $\mathbf{w}_i$ using a threshold τ.

Algorithm 2.11 Dropping strategy

```
for $j = 1 \ldots i - 1$ do
    $a_i^{(j-1)} = A_{j\bullet}\mathbf{z}_i^{(j-1)}$
    $\mathbf{z}_i^{(j)} = \mathbf{z}_i^{(j-1)} - \frac{a_i^{(j-1)}}{a_j^{(j-1)}}\mathbf{z}_j^{(j-1)}$
endfor
if $|\mathbf{z}_i^{(j)}| < \tau$ for $i = 1 \ldots n$ then
    $\mathbf{z}_i^{(j)} = 0$
endif
$a_i^{(i-1)} = A_{i\bullet}\mathbf{z}_i^{(i-1)}$
```

Therefore, the computation of the sparse approximate factors which we denote by $\tilde{Z}$, $\tilde{W}$ and $\tilde{D}$, gives the desired sparse preconditioner

$$P_2 = \tilde{Z}\tilde{D}^{-1}\tilde{W}^H, \tag{2.28}$$

These two strategies for making the preconditioner P_2 sparse are supported by the fact that the matrix A is a banded. Moreover, the matrix A is made diagonal dominant by introducing a strong damping in the wave equation. Therefore, its inverse has an exponential decay of the amplitude of the off-diagonal coefficients (Demko et al., 1984).

2.2.4 Numerical illustration of the ILUT and AINV preconditioners for the frequency-domain wave equation using CGMN

In this part, I briefly show an application of the ILUT and the AINV preconditioners for the solution of the Helmholtz equation on the heterogeneous 2D Marmousi 2 P-wave velocity model using the CGMN method.

2.2.4.1 The Marmousi 2 model

The Marmousi model (Versteeg and Grau, 1991) was created by the *Institut Français du Pétrole* (IFP) in 1988. The geometry of this model is based on a profile through the North Quenguela trough in the Cuanza basin. The geometry and velocity model were created to produce complex seismic data which require advanced processing techniques to obtain a correct earth image at this time. The model was used for the workshop on practical aspects of seismic data inversion at the 1990 EAGE meeting in Copenhagen, where different groups (contractors, universities, and oil companies) applied their proffered imaging tools on this data set.

Since its inception in 1990, it has become a standard benchmark dataset for seismic imaging techniques. The model contains 158 horizontally layered horizons. A series of normal faults and resulting tilted blocks complicates the model towards its center.

The Marmousi model is precisely 9192 m long and 2904 m deep. It has been upgraded to the Marmousi 2 model for elastic applications (Martin et al., 2006) without taking away any of qualities that have made it so useful. A shear wave velocity model is then added to the dataset. The model length has been extended from 9.2 km to 17 km. A water layer of 500 meters has also been added on top. The Marmousi 2 velocity is given in a 2801×13601 grid with a space step length $h = 1.25$ m. For this application, I use the P-wave velocity model of Marmousi 2 which is presented in Figure 2.8.

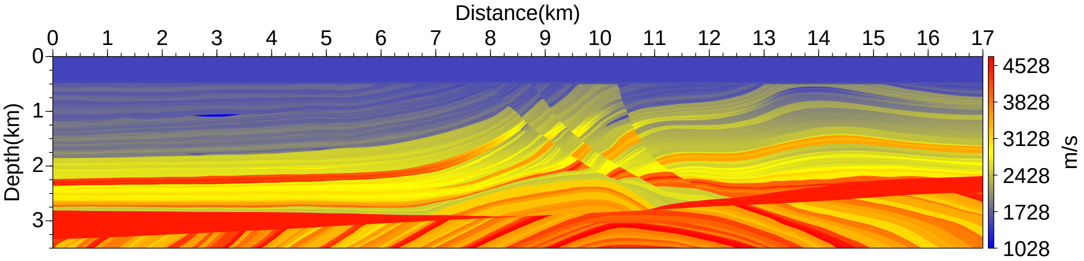


Figure 2.8: Marmousi 2: P-wave velocity model

2.2.4.2 The 2D Helmholtz equation

Consider the 2D Helmholtz equation (1.17) which is recast in the 2D approximation. An explosive source s modeled by a Delta function at the position (x_0, z_0) m

$$\mathbf{s}(x, z) = \delta(x - x_0)\delta(z - z_0).$$

Perfectly Matched Layers (PML) are used at the boundaries to mimic infinite domain wave propagation. The discretization of this problem is performed using the second-order staggered-grid finite-difference method (see Chapter 1 Section 1.5). Attenuation is accounted for using the Kolsky model (Kolsky, 1956) through the relation (2.27).

2.2.4.3 Numerical results

Several experiments are performed to illustrate the performance of the CGMN method for solving the 2D Helmholtz equation on the Marmousi 2 model together with the preconditioning strategies. For each experiment, the starting point $\mathbf{x}^{(0)}$ is set to $\mathbf{0}$. The relaxation parameter used for the Kaczmarz projections is set to 1.2 which is the optimal value selected from several experiments. The stopping criterion is set to 10^{-4} on the relative residual

$$\frac{||A\mathbf{x}^{(k)} - \mathbf{b}||}{||\mathbf{b}||} < 10^{-4}.$$

The explosive Dirac source is located at the middle top of the domain at $x_0 = 8500$ m and $z_0 = 100$ m.

The Marmousi 2 model, presented in Figure 2.8 is given in a 2801 × 13601 grid points with $d = 1.25$ meter space step length. This model is filtered and then sampled for several frequencies $f = \{5, 10, 15, 20\}$. We use 10 grid points per wavelength to ensure the accuracy of the second-order finite-difference scheme. $N_{PML} = 10$ grid points are used in each PML. The damping coefficient inside the PML is set to $c_{PML} = 1500$. Table 2.2 summarizes the size of the models used for each frequency. Representative pressure wavefields solution of this problem are presented at the end of this section at the frequencies 5 and 20 Hz in Figures 2.11 and 2.12.

f	h	N_z	N_x	N_u
5	30	145	709	120285
10	15	289	1417	444033
15	10	433	2125	971685
20	7.5	577	2833	1703241

Table 2.2: Discritization of the Marmousi 2 model (N_x, N_z) following the second-order finite-difference scheme and number of unknowns N_u. 10 grid points per minimum wavelength are used.

CGMN In Figure 2.9, a complexity analysis of the CGMN method is presented. The number of iterations is plotted as a function of the mean size of one dimension

$$N = \sqrt{N_x N_z}$$

on a log-log scale. The black solid line represents the linear increase of the computational complexity $N_{iter} = N$. The solid red curve indicates the dependence between N_{iter} and N. A standard linear regression, indicated by the dotted line, is performed on the solid red curve.

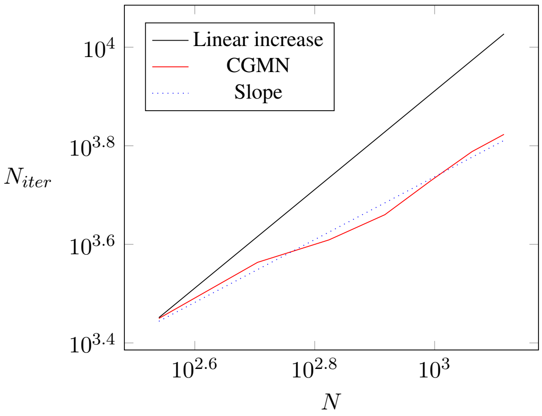


Figure 2.9: Complexity analysis of the CGMN method for the frequency domain wave propagation problem in the 2D heterogeneous acoustic model Marmousi 2. The number of iterations are plotted as a function of the average number of grid points by dimension N on a log-log scale (——). The blue line (⋯⋯) represents the line regression.

The complexity of the CGMN method is approximately given by

$$N_{iter} = 50\ N^{0.68}.$$

Therefore, following the analysis presented in Chapter 1 Section 1.6, the total computational complexity of the CGMN method for 2D modeling with one source is

$$\mathcal{O}(N_{iter} \times N^2) = \mathcal{O}(N^{2.68}).$$

This sub-linear complexity of the number of iterations with respect to N indicates the good performances of the CGMN method. This complexity is an improvement compared to $\mathcal{O}(N^3)$ performed by the time-domain modeling strategy.

In Figure 2.10, the relative residuals of the CGMN method are plotted as a function of the iteration number for each frequency described in Table 2.2. The rate of convergence is very fast until the norm of the relative residual reaches 10^{-2}. From then on, the rate of convergence decreases. The number of iterations and the computation time performed to solve the linear systems are summarized in Table 2.3.

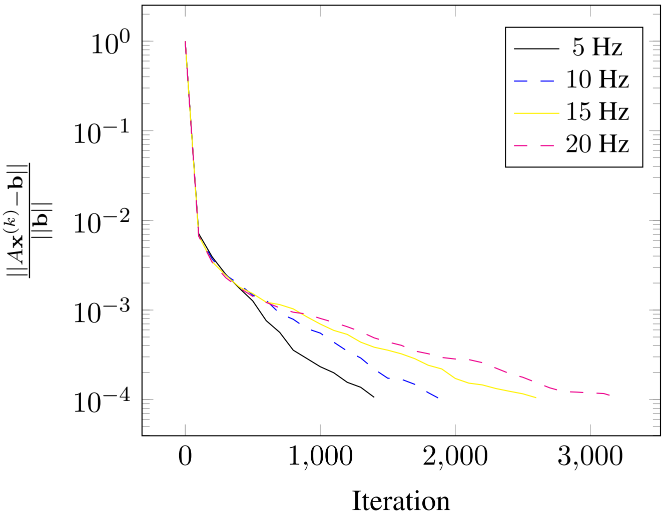


Figure 2.10: Convergence histories of the CGMN method for the frequency-domain acoustic wave simulations using the Marmousi 2 P-wave velocity model. The relative residuals are plotted as a function of the iteration number during the CGMN runs.

f	N_u	N_{iter}^{CGMN}	Time (s)
5	120285	2764	44.3
10	444033	4038	263.5
15	971685	5407	778.6
20	1703241	6878	1786.9

Table 2.3: Number of iterations performed by the CGMN method applied to the Helmholtz problem using the Marmousi 2 P-wave velocity model.

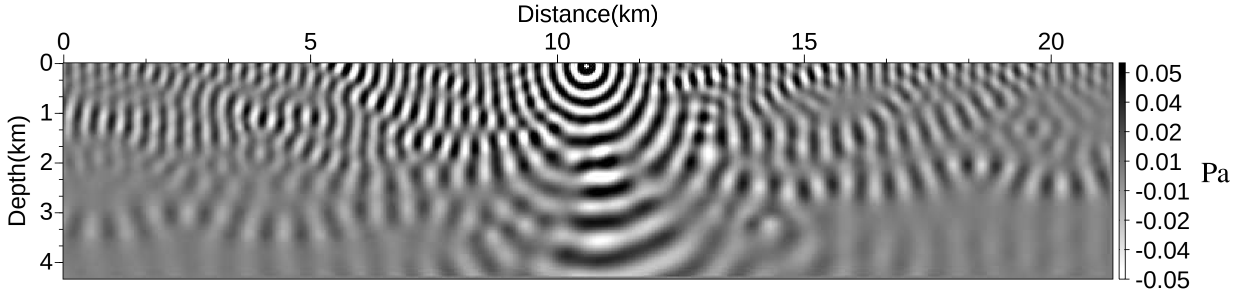


Figure 2.11: Marmousi 2 acoustic pressure wavefield (real part) solution at 5 Hz.

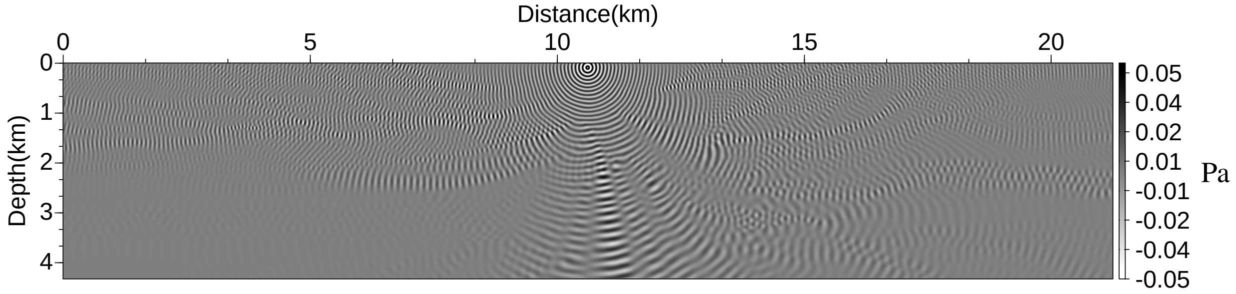


Figure 2.12: Marmousi 2 acoustic pressure wavefield (real part) solution at 20 Hz.

CGMN with ILUT and AINV preconditioners In this paragraph, the results of the CGMN method with the preconditioning techniques ILUT and AINV are presented. Both preconditioners are computed from the damped version of the matrix A. Damping is introduced in the medium by tuning the quality factor q_{att}. The ILUT preconditioner is computed with $p = 10$ and $\tau = 10^{-10}$. As discussed in the previous part, the AINV preconditioner is made sparse following two dropping strategies. The factorization process is performed by reducing the internal loop to the bandwidth $\nu = N_z$. A threshold $\tau = 10^{-2}$ is applied allowing to keep the preconditioner sparse with only 9 nonzero coefficients per columns.

In the experiment (2.13), the influence of the attenuation on the quality of the preconditioner for the $f = 10$ Hz problem is investigated. The experiment is performed with the ILUT preconditioner but similar behavior is obtained with the AINV preconditioner. The CGMN method performs 4038 iterations to solve the Helmholtz equation applied to the Marmousi 2 P-wave velocity model for the frequency $f = 10$ Hz. Figure 2.13 shows how the CGMN method performs when the ILUT preconditioner is computed based on a damped version of A, which is denoted by G. When a small damping is applied, the quality of the preconditioner is very poor. The preconditioned method performs even more iterations than CGMN. However, by introducing a stronger damping, the preconditioned method becomes efficient. The minimum number of iterations $N_{iter} = 1205$ performed by the ILUT-preconditioned CGMN method is reached for $q_{att} = 1.1$. Such behavior is due to the fact that the preconditioner needs to be computed based on a diagonal dominant matrix. If the damping is not strong enough, the sparse approximation of the ILUT preconditioner does not compute a good a approximation of the G^{-1}. In the following, both preconditioner are computed based on the damped version of A using $q_{att} = 1.1$.

Figure 2.14 shows the convergence of the relative residual for the given frequencies. For each frequency, the comparison between the convergence histories of CGMN, CGMN with ILUT and CGMN with AINV shows the AINV preconditioner reduces at best the number of iterations. A “super” convergence is observed compared to CGMN.

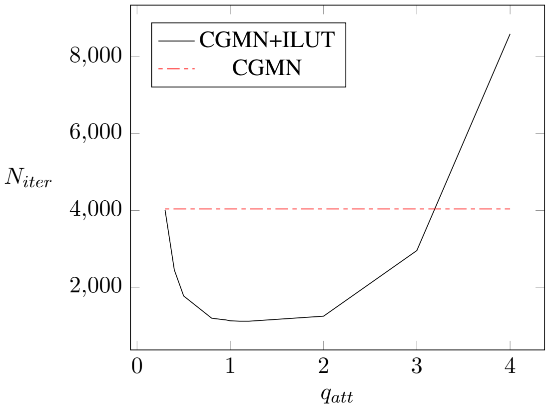


Figure 2.13: Effect of the quality factor on the quality of the preconditioner. The number of iterations N_{iter} performed by the preconditioned CGMN method is plotted as a function the quality factor q_{att} (——). The Helmholtz problem at the frequency $f = 10$ Hz is considered for this experiment. The ILUT preconditioner is computed with $p = 10$, $\tau = 10^{-10}$ by varying the quality factor q_{att}. The number of iterations performed by the CGMN method without preconditioning (-·-·) is $N_{iter} = 4038$.

Table (2.4) summarizes the number of iterations and the computation time in seconds performed by the CGMN method and the preconditioned CGMN method using the ILUT and AINV preconditioners. The number of iterations is reduced by a factor going from 3.6 to 6.0 when using the ILUT preconditioner. The computation time is reduced by a factor 1.5 up to 2.2.

The AINV preconditioner clearly performs better than the ILUT preconditioner for the reduction in the number of iterations. A factor 8.6 up to 9.2 is obtained. A reduction in the computation time by a factor 3.6 up to 3.9 is obtained.

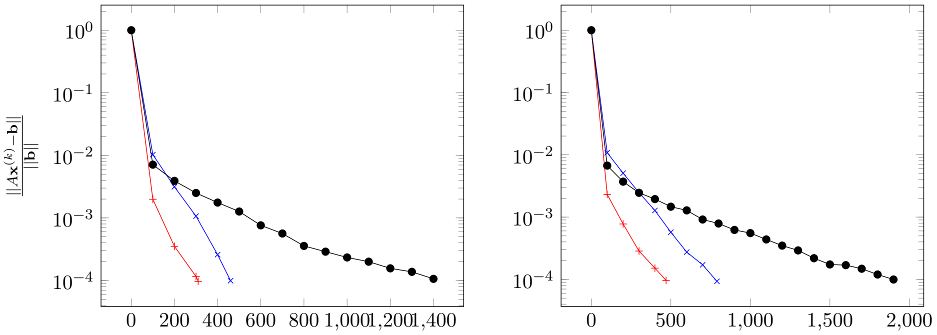
f	N_u	N_{iter}^{CGMN}	Time (s)	N_{iter}^{ILUT}	Time (s)	N_{iter}^{AINV}	Time (s)
5	120285	2764	44.3	460	20.1	310	12.3
10	444033	4038	263.5	790	134.1	470	70.8
15	971685	5407	778.6	1350	500.2	620	215.1
20	1703241	6878	1786.9	1800	1189.2	750	449.7

Table 2.4: Number of iterations and computation time performed by the preconditioned CGMN method for the solution of the 2D Helmholtz problem using the Marmousi 2 P-wave velocity model. CGMN is preconditioned with the ILUT and the AINV preconditioners. N_{iter}^{CGMN} denotes the number of iterations performed by the CGMN method without preconditioning. N_{iter}^{ILUT} denotes the number of iterations performed by the CGMN method using the ILUT preconditioner. N_{iter}^{AINV} denotes the number of iterations performed by the CGMN method using the AINV preconditioner. The computation time is presented in the column “Time” in seconds.

A complexity analysis of the CGMN method using the ILUT and AINV preconditioners is presented in Figure 2.15. For the ILUT preconditioner, we have

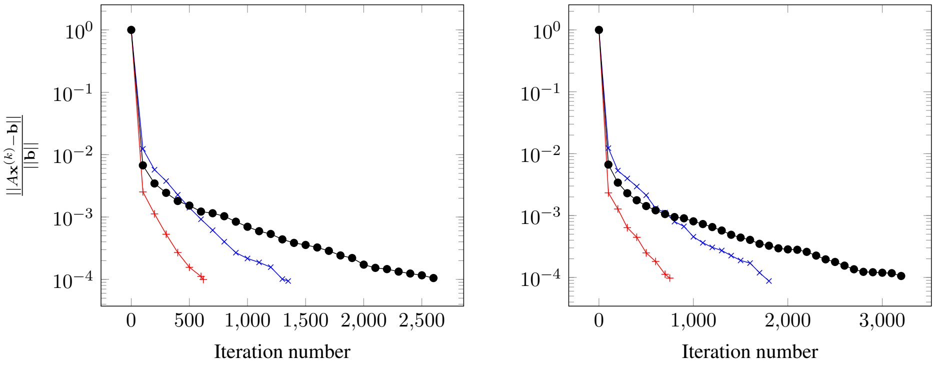
$$N_{iter}^{ILUT} = 0.9\, N^{1.04}.$$

Despite the gain in computation time, the complexity of the CGMN seems worsen for high frequencies.



(a) 5 Hz

(b) 10 Hz



(c) 15 Hz

(d) 20 Hz

Figure 2.14: Convergence histories of the CGMN method for the resolution of the Helmholtz problem using the 2D Marmousi 2 P-wave velocity model. CGMN without preconditioned is plotted with (—●—), with the ILUT preconditioner (—×—) and with the AINV preconditioner (—+—). The ILUT preconditioner is computed with $p = 10$, $\tau = 10^{-10}$ and $q_{att} = 1.1$. The AINV preconditioner is computed on the same damped matrix as for the ILUT preconditioner. The threshold $\tau = 10^{-2}$ so that the preconditioner only has 9 non-zero values per column.

A complexity analysis of the CGMN method using the AINV preconditioner gives

$$N_{iter}^{AINV} = 6.2\ N^{0.66}.$$

The latter clearly indicates the reduction in the complexity of the preconditioned method. Both coefficients α and β as in $\alpha\ N^{\beta}$ are reduced. In comparison with the CGMN method without preconditioning, the exponent is sensitively the same but the constant multiplying the complexity is drastically reduced by a factor 8.1.

The above experiments show that the second preconditioning strategy provides clearly better results. Therefore, the approach which consists in solving the linear system

$$AP_2\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P_2\mathbf{y},$$

using the CGMN or CARP-CG method shall be preferred in the following.

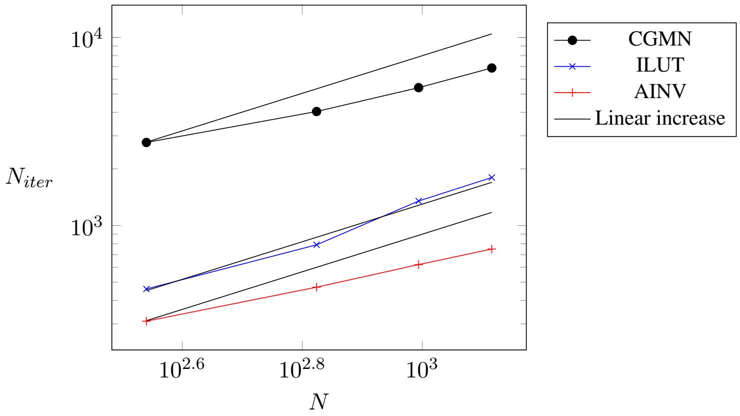


Figure 2.15: Complexity analysis of the CGMN method with and without preconditioning (ILUT, AINV) for the frequency domain wave propagation problem in the 2D heterogeneous acoustic model Marmousi 2.

2.2.5 A revisited sparse approximate inverse preconditioner for the frequency-domain wave equation

The AINV addresses the problem of computing the preconditioner from an algebraic point of view. Because the operator A is well-known, one can consider the problem from an operator point of view, namely as a frequency-domain wave propagation problem in a strongly damped medium. We denote by G this frequency-domain wave propagation operator in a strongly damped medium. The kth column vector of G^{-1}, denoted by $\mathbf{G}^{-1}_{\bullet k}$, is the solution of the system

$$G\mathbf{x} = \mathbf{e}_k, \tag{2.29}$$

where $\mathbf{e}_k$ is a vector of size n with a single nonzero coefficient equal to 1 at the index k. Assuming for instance a 2D approximation, let (i_k, j_k) be the couple of indexes representing the spatial position associated with the index k, the column vector $\mathbf{G}^{-1}_{\bullet,k}$ can be identified with the wavefield generated by a Dirac source located at the position (i_k, j_k) in a strongly attenuating medium. Each column of P can thus be sparsely approximated by restraining $\mathbf{G}^{-1}_{\bullet,k}$ to few discretization points around (i_k, j_k) (see Figure 2.16).

Therefore, discarding the small values of $\mathbf{G}^{-1}_{k,\bullet}$ and taking into account the exponential decay of the off-diagonal entries is equivalent to say that physical points far from the source position are ignored. Thus, by introducing a strong attenuation in the media and by considering only the immediate source point neighbors, one can build a good approximation of the wavefield $\mathbf{G}^{-1}_{k,\bullet}$

$$\mathbf{P}_{k,\bullet} = \mathbf{G}^{-1}_{k,\bullet} + \mathbf{E}_{k,\bullet}, \tag{2.30}$$

where E is an error matrix due to the approximation.

In practice, the preconditioner P is computed efficiently by solving the n local frequency-domain wave propagation problems in a strongly damped small medium defined by the sparsity pattern chosen. Perfectly Matched Layers (Bérenger, 1994) surround this local domain to absorb the outgoing waves. The sparsity pattern of the preconditioner can be controlled efficiently. Using this method, the computation of the preconditioner presents low memory requirements as it only requires the solution of small frequency-domain wave propagation problems. In addition it is highly scalable as each column of the

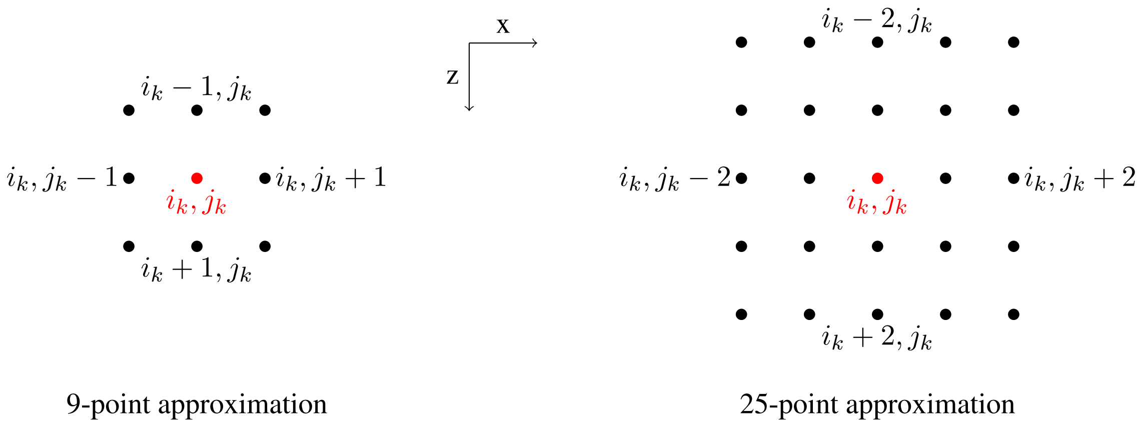


Figure 2.16: Sparse approximation patterns of the wavefield using a 9-point or 25-point approximation around the source position (i_k, j_k).

preconditioner can be computed independently. Finally this approach ensures P is composed of only few non-zero diagonals. The matrix AP remains sparse and banded. It is computed efficiently in $O(n)$.

The complex shifted Laplacian preconditioner has attracted considerable attention for the speed up of the convergence of iterative methods for the solution of the 2D and 3D Helmholtz equation. It has been intensively studied over the past decade by Erlangga et al. (2003), Riyanti et al. (2006), Plessix (2007), Osei-Kuffor and Saad (2013) and more recently by Gander et al. (2015). In this work, we extend this strategy equivalent to preconditioning by the strongly damped frequency-domain wave operator to the general case for the 2D and 3D frequency domain elastic wave equations.

2.3 Spectral study of the preconditioned Helmholtz equation

In the following section, we restrict the study to the Helmholtz equation where the density is assumed homogeneous. In this case, the eigenvalues and the eigenvectors can be derived analytically. First, some properties of the matrix resulting from the discretization of this equation are discussed. Then, the damped frequency-domain wave preconditioner is introduced and the preconditioned problem is examined on the toy problem 2.1.6.

2.3.1 The Helmholtz equation

Consider the 2D Helmholtz equation (1.17) which is recast in the 2D approximation in the domain Ω. Attenuation is introduced in the Helmholtz operator using a complex shift on the zero-th-order term k^2. This complex shift is defined by

$$z_\nu = (1 - i\nu),$$

where $\nu \in \mathbb{R}^+$. Therefore the Helmholtz operator with attenuation writes

$$\mathcal{H}_\nu := -\Delta - z_\nu k^2. \tag{2.31}$$

The Helmholtz equation (2.31) may be defined with the boundary conditions

$$\mathbf{p} = 0, \quad \text{on } \partial\Omega, \tag{2.32}$$

for the homogeneous Dirichlet condition, or

$$\frac{\partial \mathbf{p}}{\partial \eta} = ik\mathbf{p}, \quad \text{on } \partial\Omega,$$

which represents radiating boundaries where η is the outward normal unit vector with respect to the boundary. This condition is also known as the first kind Sommerfeld boundary condition. It simulates the seismic wave propagation in infinite domains. In practice, Perfectly Matched Layer of Bérenger (1994) are used.

Remark 2.7. *In this study, I shall restrain the investigations to the homogeneous Dirichlet boundary condition* (2.32). *However, I shall present numerical results for PMLs.*

The eigenvalues $\lambda_{-\Delta}$ of the negative Laplacian problem are solution of the problem

$$-\Delta\mathbf{v} = \lambda_{-\Delta}\mathbf{v} \tag{2.33}$$

where $\mathbf{v}$ is the eigenvector related to the eigenvalue $\lambda_{-\Delta}$. The solution of the eigenvalue problem (2.33) is obtained using eigenvectors of the form

$$\mathbf{v}_{k_1,k_2}(x,z) = \sin(k_1 x)\cos(k_2 z) \tag{2.34}$$

where $k_1 = j_1\pi$, $j_1 \in \mathbb{N} \setminus \{0\}$ and $k_2 = j_2\pi$, $j_2 \in \mathbb{N} \setminus \{0\}$ satisfying the homogeneous Dirichlet boundary condition (2.32). By substituting the eigenvectors (2.34) in equation (2.33), the eigenvalues of the negative Laplacian are obtained

$$\begin{aligned} -\Delta\mathbf{v}_{k_1,k_2} &= k_1^2\mathbf{v}_{k_1,k_2} + k_2^2\mathbf{v}_{k_1,k_2}, \\ &= (k_1^2 + k_2^2)\mathbf{v}_{k_1,k_2}, \end{aligned}$$

which gives

$$\lambda_{-\Delta} = k_1^2 + k_2^2,$$

with $k_1 = j_1\pi, j_1 \in \mathbb{N} \setminus \{0\}$ and $k_2 = j_2\pi, j_2 \in \mathbb{N} \setminus \{0\}$.

Thus, the eigenvalues $\lambda_{\mathcal{H}_\nu}$ of the Helmholtz operator (2.31) write

$$\lambda_{\mathcal{H}_\nu} = (k_1^2 + k_2^2) - z_\nu k^2, \quad z_\nu = 1 - i\nu, \quad \nu \in \mathbb{R}^+, \tag{2.35}$$

with $k_1 = j_1\pi, j_1 \in \mathbb{N} \setminus \{0\}$ and $k_2 = j_2\pi, j_2 \in \mathbb{N} \setminus \{0\}$. Thus, for large wavenumbers k, the real part of the eigenvalues of the Helmholtz operator change sign which shows its indefiniteness. The imaginary part of the eigenvalues is shifted by a constant value equal to ν.

The discretization of the negative Laplacian operator $-\Delta$ using the second-order finite-difference method together with the homogeneous Dirichlet boundary condition (2.32) yields the real symmetric positive definite matrix $-L$ where

$$L = \frac{1}{h^2}\begin{bmatrix} L' & I & 0 & & \\ I & L' & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & L' & I \\ & & & I & L' \end{bmatrix},$$

and where the matrix L' is given by

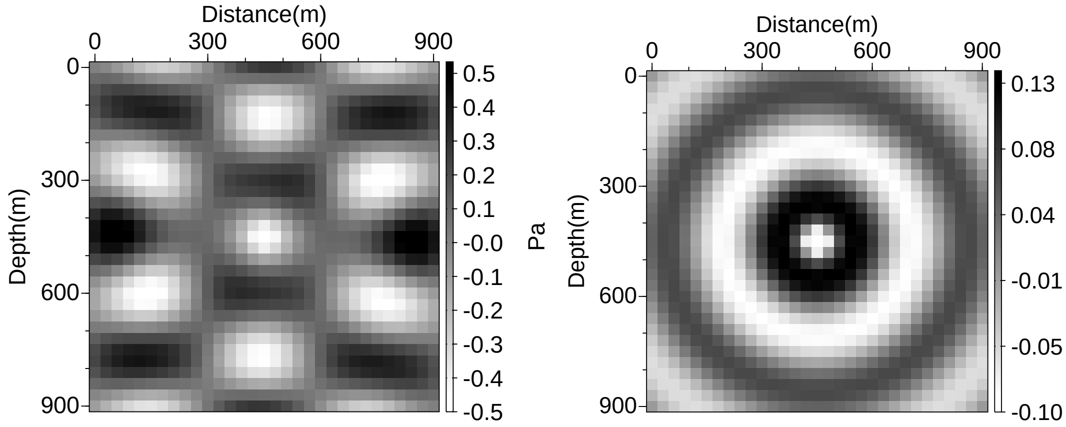
$$L' = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix}.$$

Therefore, the discretization of the Helmholtz equation (2.31) together with the homogeneous Dirichlet boundary condition (2.32) using the second-order finite-difference method gives a sparse linear system which can be written in the matrix form as

$$A\mathbf{x} = (-L - z_\nu k^2 I)\mathbf{x} = \mathbf{b} \tag{2.36}$$

where $\mathbf{x}$ is the pressure wavefield and $\mathbf{b}$ is the right-hand side representing the source term.

The pressure wavefield solution of the toy problem 2.1.6 with homogeneous Dirichlet and PML boundary conditions are presented in Figures 2.17 and 2.18.



(a) Homogeneous Dirichlet boundary

(b) PML boundary condition

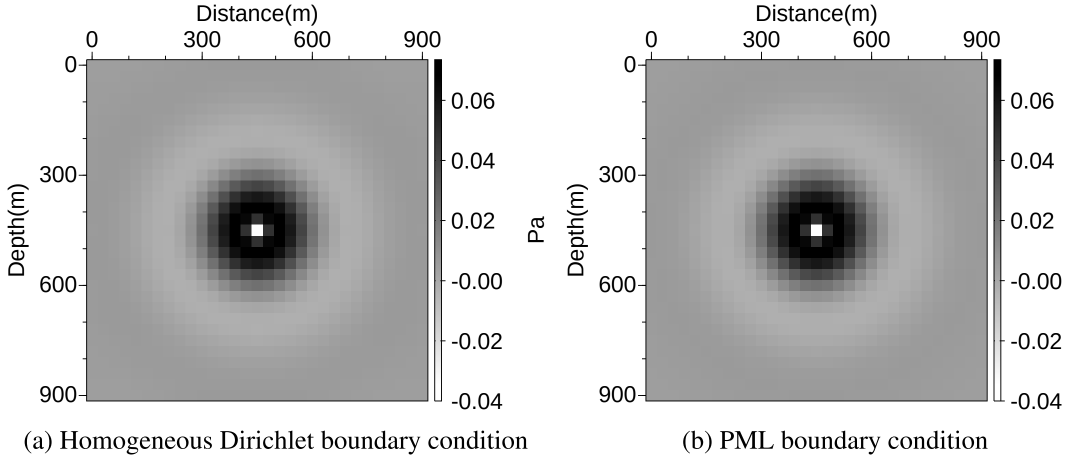
Figure 2.17: Pressure wavefields (real parts) solution of the 2D Helmholtz equation on the toy problem 2.1.6 in a medium without attenuation using the second-order staggered-grid finite difference discretization with homogeneous Dirichlet boundary condition (a) and PMLs (b)

Figure 2.17 presents the wavefields computed in a medium without damping. A strong damping is introduced in the medium ($q_{att} = 1.0$) in Figure 2.18. The wavefields are very similar for homogeneous Dirichlet and PML boundary condition. No reflections are observed at the boundary of the medium due to the very weak amplitude of the wavefield in the case of the Dirichlet boundary configuration.

2.3.2 Spectral properties of the damped frequency-domain wave preconditioner

The damped frequency-domain wave preconditioner is introduced in 2D as

$$(-\Delta - zk^2)\mathbf{p}(x,z,\omega) = \mathbf{s}(x,z,\omega), \quad \mathcal{P}_z := -\Delta - zk^2, \tag{2.37}$$



(a) Homogeneous Dirichlet boundary condition (b) PML boundary condition

Figure 2.18: Pressure wavefields (real parts) solution of the 2D Helmholtz equation on the toy problem 2.1.6 in a medium with a strong attenuation using the second-order staggered-grid finite difference discretization with homogeneous Dirichlet boundary condition (a) and PMLs (b)

where z is a complex number defined by

$$z = (\alpha - i\beta),$$

with α and β real positive values. The operator $\mathcal{P}_z$ is defined the same way as the Helmholtz operator using the homogeneous Dirichlet boundary condition (2.32). Following the equations (2.34) and (2.35), the eigenvalues of the damped Helmholtz operator $\mathcal{P}_z$ are given by

$$\lambda_{\mathcal{P}_z} = \lambda_{-\Delta} - zk^2, \quad z = \alpha - i\beta,$$

with $\lambda_{-\Delta} = k_1^2 + k_2^2$, $k_1 = j_1\pi$, $j_1 \in \mathbb{N} \setminus \{0\}$ and $k_2 = j_2\pi$, $j_2 \in \mathbb{N} \setminus \{0\}$ the eigenvalues of the negative Laplacian operator.

The Helmholtz operator $\mathcal{H}_\nu$ (2.31) is preconditioned with $\mathcal{P}_z$, which gives

$$\left(\mathcal{P}_z^{-1}\mathcal{H}_\nu\right) \mathbf{p}(x, z, \omega) = \mathcal{P}_z^{-1}\mathbf{s}(x, z, \omega). \tag{2.38}$$

The eigenvalues θ of the operator $\mathcal{P}_z^{-1}\mathcal{H}_\nu$ are given by

$$\theta = \frac{\lambda_{-\Delta} - z_\nu k^2}{\lambda_{-\Delta} - zk^2}. \tag{2.39}$$

Note that it is assumed that $zk^2 \neq \lambda_{-\Delta}$, otherwise the preconditioner $\mathcal{P}_z$ is singular (resonance case). Such case is preferably avoided. Following the work of Erlangga (2005), we have the theorem

Theorem 2.4. *Let the 2D Helmholtz problem* (2.31) *with the homogeneous Dirichlet boundary condition be preconditioned by $\mathcal{P}_z$ and assume that resonance does not occur. The spectrum of the preconditioned operator $\mathcal{P}_z^{-1}\mathcal{H}_\nu$ has the following properties:*

(i) All eigenvalues lie on a circle with center c and radius r given by

$$c = \frac{z_\nu - \bar{z}}{z - \bar{z}}, \quad r = \left| \frac{z - z_\nu}{z - \bar{z}} \right| .$$

(ii) This circle is independent of the wavenumber k.

Proof. Let θ be the eigenvalues of $\mathcal{P}_z^{-1}\mathcal{H}_\nu$ given in equation (2.39). We have

$$\begin{aligned}
|\theta - c| &= \left| \frac{\lambda_{-\Delta} - z_\nu k^2}{\lambda_{-\Delta} - zk^2} - \frac{z_\nu - \bar{z}}{z - \bar{z}} \right|, \\
&= \left| \frac{(\lambda_{-\Delta} - z_\nu k^2)(z - \bar{z}) - (\lambda_{-\Delta} - zk^2)(z_\nu - \bar{z})}{(\lambda_{-\Delta} - zk^2)(z - \bar{z})} \right|, \\
&= \left| \frac{\lambda_{-\Delta}(z - z_\nu) - \bar{z}k^2(z - z_\nu)}{(\lambda_{-\Delta} - zk^2)(z - \bar{z})} \right|, \\
&= \left| \frac{z - z_\nu}{z - \bar{z}} \right| \left| \frac{\lambda_{-\Delta} - \bar{z}k^2}{\lambda_{-\Delta} - zk^2} \right|, \\
&= r \left| \frac{\lambda_{-\Delta} - \bar{z}k^2}{\lambda_{-\Delta} - zk^2} \right|.
\end{aligned}$$

By expanding the expression $\lambda_{-\Delta} - zk^2$ and $\lambda_{-\Delta} - \bar{z}k^2$, we have

$$\lambda_{-\Delta} - zk^2 = (\lambda_{-\Delta} - \alpha k^2) + i\beta k^2,$$

and

$$\lambda_{-\Delta} - \bar{z}k^2 = (\lambda_{-\Delta} - \alpha k^2) - i\beta k^2,$$

and it is easy to see that

$$| \lambda_{-\Delta} - zk^2 | = | \lambda_{-\Delta} - \bar{z}k^2 | .$$

Thus, the property (i) of theorem 2.4 is retrieved

$$|\theta - c| = r.$$

Furthermore, the center c and the radius r of the circle are independent of the wavenumber k, which shows the property (ii) of theorem 2.4. ☐

Remark 2.8. *The origin is not enclosed by the circle defined in theorem* (2.4)*. This is true if the distance of the center c to the origin is larger than the radius which is equivalent to*

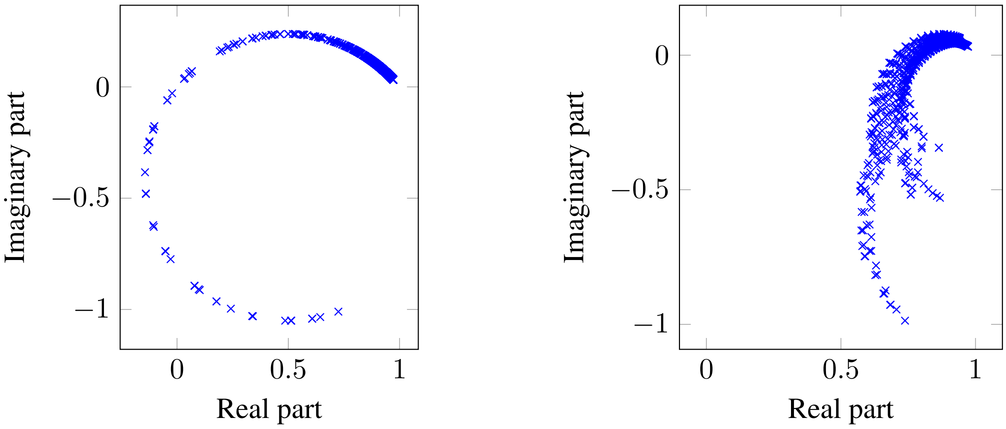
$$\begin{aligned}
| z_\nu - \bar{z} | > | z - z_\nu | &\iff (1-\alpha)^2 + (\nu + \beta)^2 > (1-\alpha)^2 + (\nu - \beta)^2, \\
&\iff \nu\beta > -\nu\beta, \\
&\iff \nu\beta > 0,
\end{aligned}$$

which is clearly the case as ν and β are positive real numbers.

The operator $\mathcal{P}_z$ is discretized using the second-order staggered-grid finite-difference method. The matrix G is obtained and its inverse G^{-1} is computed. In the Figure 2.19, the spectrum of the matrix AG^{-1} is shown in the case of the toy problem 2.1.6 with homogeneous Dirichlet boundary condition and PMLs. The preconditioner is computed with the value of damping $q_{att} = 1.0$.

The spectra of the preconditioned matrices are improved and this result corroborates theorem (2.4). The eigenvalues are closely gathered around the value 1. In the case of the homogeneous Dirichlet configuration, the condition number of the linear system is reduced

$$\texttt{cond}(A) = 343.3, \quad \texttt{cond}(AG^{-1}) = 18.9. \tag{2.40}$$



(a) Homogeneous Dirichlet boundary condition (b) PML boundary condition

Figure 2.19: Eigenvalue distribution of the matrix AG^{-1} with homogeneous Dirichlet boundary condition (a) and PML (b). The matrix G results from the discretization of the 2D Helmholtz operator in a strongly damped medium with $q_{att} = 1.0$.

In the case of the PML configuration, the condition number of the linear system is improved as well

$$\texttt{cond}(A) = 107.4, \quad \texttt{cond}(AG^{-1}) = 6.5. \tag{2.41}$$

This situation is very favorable when using iterative solvers. First, the indefinite matrix A is transformed into a definite one AG^{-1} by moving the eigenvalues toward the positive real half space and away from the origin. This criterion is suitable for GMRES type iterative solvers. Second, the spectrum of the preconditioned matrix is gathered around the value 1 which is crucial for conjugate gradient type iterative solvers which the convergence rate depends on the condition number of the matrix. Thus, the convergence rate of iterative solvers is improved and they converge using a small number of iterations.

2.3.3 Sparse approximation of the damped frequency-domain wave preconditioner

For our applications, a sparse approximation of the preconditioner is required. Following the wavefield approximation approach in the strongly damped medium, the discretization points immediately close to the source are kept. The sparse preconditioner is denoted by P. In the following, I show the influence of increasing the approximation of the wavefield on the spectrum of the matrix AP.

Naturally, the spectrum of the matrix AP is damaged compared to using the dense preconditioner G^{-1} as it is shown in the Figures 2.20 and 2.21. But, the eigenvalues are still gathered around the value 1. In the homogeneous Dirichlet configuration, the condition number of the matrix AP is equal to

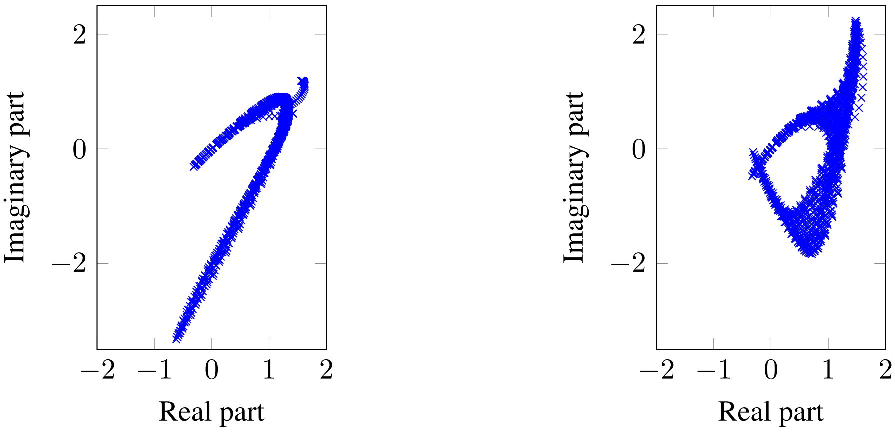
$$\texttt{cond}(AP) = 143.4,$$

for the 9-point approximation, and

$$\texttt{cond}(AP) = 96.5,$$

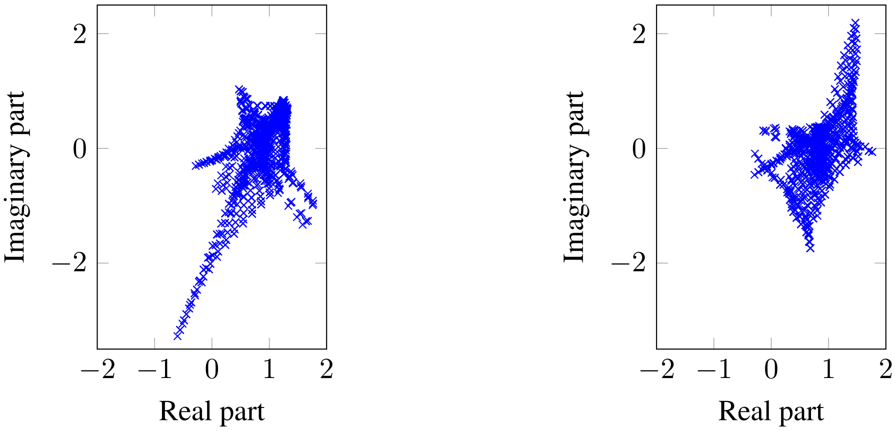
for the 25-point approximation, which has to be confronted with the condition number of the matrices A and AG^{-1}

$$\mathrm{cond}(A) = 343.3, \quad \mathrm{cond}(AG^{-1}) = 18.9.$$



(a) 9-point approximation (b) 25-point approximation

Figure 2.20: Eigenvalue distribution of the preconditioned system AP with homogeneous Dirichlet boundary condition using the 9-point approximation (a) and the 25-point approximation (b).



(a) 9-point approximation (b) 25-point approximation

Figure 2.21: Eigenvalue distribution of the preconditioned system AP with PML boundary condition using the 9-point approximation (a) and the 25-point approximation (b).

In the PML configuration, the condition number of the matrix AP is equal to

$$\texttt{cond}(AP) = 69.5,$$

for the 9-point approximation, and

$$\texttt{cond}(AP) = 43.7,$$

for the 25-point approximation, which has to be confronted as well with the condition number of the matrices A and AG^{-1}

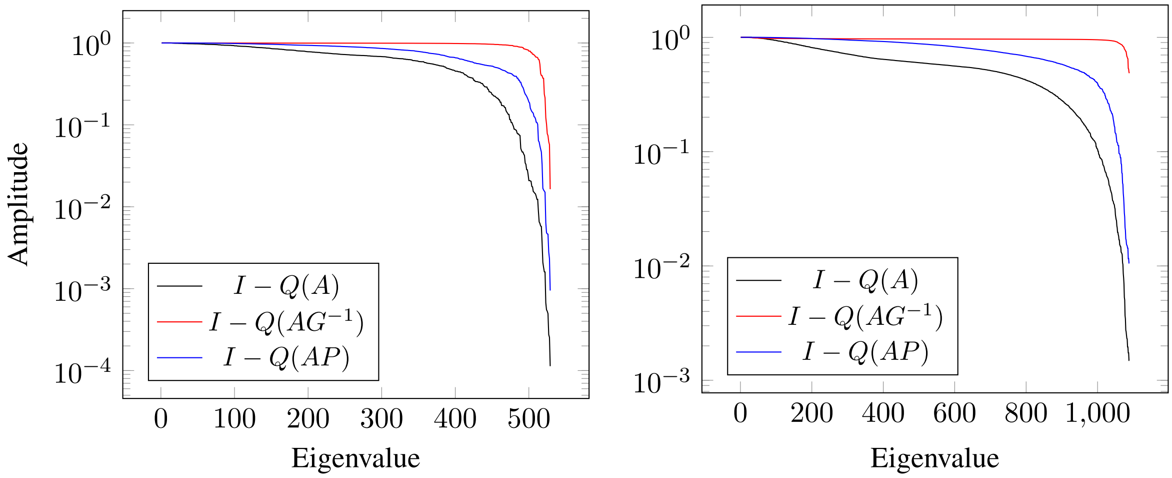
$$\texttt{cond}(A) = 107.4, \quad \texttt{cond}(AG^{-1}) = 6.5.$$

However, one has to pay attention when performing the sparse approximation. Some eigenvalues of the preconditioned system might be shifted to the negative real half space and closer to the origin as well, making the matrix AP indefinite. Therefore, the amount of damping needs to be tuned in order to compensate the loss of definiteness. Conjugate gradient based iterative solver should not be influenced by this effect since these methods are more sensitive to the condition number of the preconditioned matrix.

Related work The way the preconditioner is sparsely approximated in this work meets the strategy adopted by Poulson et al. (2013) with the parallel sweeping preconditioner. Indeed, here, the wavefield computed in each column of the preconditioner in the strongly damped medium is truncated using PMLs. This is justified by fact that far from the source point the amplitude of the wavefield is almost equal to zero. Through his moving PML, Poulson et al. (2013) approximate the Schur complements resulting from the factorization using auxiliary smaller problems truncated with artificial PML boundary conditions. Since, the inverse of the Schur complement is a restricted space of the Green's function defined on the grid points located at the interface of the sub-domains, an approximation can be made with another restricted smaller space of the Green's function by moving the PMLs.

2.3.4 Spectral analysis of the matrices $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$

In this section, I focus on the spectral properties of the matrices $I - Q(A)$ and $I - Q(AG^{-1})$ and $I - Q(AP)$ which represent the double sweeping matrices on A, AG^{-1} and AP where the preconditioner G^{-1} is build on the damped medium using a quality factor $q_{att} = 1$ without any sparse approximation. The matrix P denotes the sparse approximation of G^{-1} using the 9-point sparsity pattern. The spectra of the matrices $I - Q(AG^{-1})$ and $I - Q(AP)$ are plotted in Figure 2.22. On the same figure, they are compared to the spectrum of the matrix $I - Q(A)$. On the Figure 2.22a, the computation is performed using the homogeneous Dirichlet boundary condition and on the Figure 2.22b, PMLs are used.



(a) Homogeneous Dirichlet boundary condition

(b) PML

Figure 2.22: Eigenvalue distribution of the matrices $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$ with homogeneous Dirichlet boundary condition (a) and PML (b). The preconditioner G^{-1} is build on the damped medium using $q_{att} = 1$ without any sparse approximation and P is the sparse approximation of G^{-1} using the 9-point sparsity pattern.

Ideally, when the dense preconditioner is used (red curves), the spectrum of the matrix $I - Q(AG^{-1})$ is significantly improved compared to the spectrum of the matrix $I - Q(A)$ (black curves). The eigenvalues are clustered around the value 1. The spectrum is improved as well when the sparse approximation, i.e. P, is used (blue curves).

The condition numbers of the matrices A, AG^{-1}, AP, A^HA, $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$ are summarized in Table 2.5. Both dense and sparse preconditioners improve the conditioning of the matrix A. The matrix $I - Q(AG^{-1})$ has clearly a better conditioning compared to the matrix associated with the normal equations A^HA. Despite the sparse approximation applied to the preconditioner, the matrix $I - Q(AP)$ still remains very well conditioned.

	A	AG^{-1}	AP	A^HA	$I - Q(A)$	$I - Q(AG^{-1})$	$I - Q(AP)$
Dirichlet	343.3	18.9	143.4	117898.1	8835.4	61.0	1053.8
PML	107.4	6.5	69.5	11543.5	671.6	2.1	94.4

Table 2.5: Condition numbers of the matrices A, AG^{-1}, AP, A^HA, $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$ with homogeneous Dirichlet and PMLs configurations.

The application of the CARP-CG method on the preconditioned system AP gives very promising results in terms of condition number and localization of the eigenvalues for the frequency-domain acoustic wave equation. In the following seciton, I shall carry out this analysis in the 2D elastic approximation.

2.4 Spectral study of the preconditioned frequency-domain elastic wave equations

Because the damped frequency-domain wave preconditioner has shown good properties for preconditioning the Helmholtz equation, in the following it is applied to the frequency-domain elastic wave equations. The same approach used for the spectral study of the Helmholtz equation is carried out. The spectral properties of the preconditioned system in the elastic approximation using the velocity formulation are studied. First, some properties of the matrix resulting from the discretization of the frequency-domain elastic wave equations are discussed. Then, the damped frequency-domain wave preconditioner is introduced and the preconditioned problem is examined. I shall retain this study to the 2D approximation only. The numerical experiments are performed on a simple toy problem which I describe in the following.

Consider the 2D domain $\Omega = 3\times 2$ km^2 with constant P-wave and S-wave velocities $V_P = 1500\ m/s$ and $V_S = 1000\ m/s$ corresponding to a constant Poisson's ratio $\xi = 0.1$. The 2D frequency-domain elastic wave equations are discretized using the fourth-order staggered-grid finite-difference method for the frequency $f = 5$ Hz and using 5.4 points per wavelength. The discrete domain of 71 × 141 grid points is obtained. The domain is surrounded by PMLs of width 500 m corresponding to one wavelength of fasted traveling wave. The damping coefficient in the PML is set to $C_{PML} = 700$. A vertical force source is located at $x_s = 1500$ and $z_s = 1000$ m. For the computation of the eigenvalues, the size of the domain is reduced to 600 × 600 m^2 to allow the computations in a reasonable amount of time.

2.4.1 The frequency-domain elastic wave equations

Consider now the frequency-domain elastic wave equations in the 2D approximation. The model parameters, namely the density ρ and the Lamé parameters λ and μ are supposed homogeneous so that they do not step in the partial derivatives. The medium is supposed isotropic and viscous. Attenuation is introduced in the medium using a complex shift on the zero-th-order term ω^2. This complex shift is defined by the coefficient

$$z_\nu = 1 - i\nu.$$

In practice, attenuation is accounted for using the Kolsky model (Kolsky, 1956). The complex-valued P-wave and S-wave velocities are introduced as in the previous section such that

$$V_p^{(q_{att})} = V_p\left(1 - \frac{i}{2q_{att}}\right), \quad V_S^{(q_{att})} = V_S\left(1 - \frac{i}{2q_{att}}\right),$$

with q_{att} the quality factor.

Remark 2.9. *Note that different values of the quality factor q_{att} could be used to define attenuation for P-waves and S-waves. Here, we use the same value for both types of waves.*

The problem is considered on a 2D domain Ω and bounded by $\partial\Omega$. The frequency-domain elastic wave equations in such configuration write

$$\left\{\begin{aligned} z_\nu\omega^2\rho v_x + (\lambda+2\mu)\partial_{xx}v_x + \mu\partial_{zz}v_x + \lambda\partial_{xz}v_z + \mu\partial_{zx}v_z &= -i\omega F_x, \\ z_\nu\omega^2\rho v_z + \mu\partial_{xx}v_z + (\lambda+2\mu)\partial_{zz}v_z + \lambda\partial_{zx}v_x + \mu\partial_{xz}v_x &= -i\omega F_z. \end{aligned}\right. \quad \text{in} \quad \Omega \qquad (2.42)$$

We are not concerned with the source terms, the full expression which involves excitations on the stress terms is not provided. Note that the differential operators $\partial_i\partial_j$ are noted as ∂_{ij} to simplify notations. The homogeneous Dirichlet boundary conditions are used in this study

$$\mathbf{v} = \mathbf{0}, \quad \text{on} \quad \partial\Omega. \qquad (2.43)$$

Remark 2.10. *I shall present as well numerical results using PMLs for completeness.*

Using the following operator

$$\mathcal{L} := \begin{bmatrix} (\lambda+2\mu)\partial_{xx} + \mu\partial_{zz} & \lambda\partial_{xz} + \mu\partial_{zx} \\ \lambda\partial_{zx} + \mu\partial_{xz} & \mu\partial_{xx} + (\lambda+2\mu)\partial_{zz} \end{bmatrix},$$

and the vector notations

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_z \end{bmatrix}, \quad \mathbf{f} = -i\omega\begin{bmatrix} F_x \\ F_z \end{bmatrix},$$

the system of equation (2.42) together with the homogeneous Dirichlet boundary condition (2.43) can be written in matrix form as

$$(\mathcal{L} + z_\nu\omega^2\rho I)\mathbf{v} = \mathbf{f}, \quad \mathcal{E}_\nu := \mathcal{L} + z_\nu\omega^2\rho I. \qquad (2.44)$$

Let $\Lambda_{\mathcal{L}}$ be the eigenvalues of the operator $\mathcal{L}$. The eigenvectors $\mathbf{u}$ given by

$$\mathbf{u} = \begin{bmatrix} \cos(k_1\pi x)\sin(k_2\pi z) \\ \sin(k_1\pi x)\cos(k_2\pi z) \end{bmatrix},$$

for $k_1, k_2 \in \mathbb{N} \setminus \{0\}$ satisfy the eigenvalue problem $\mathcal{L}\mathbf{u} = \Lambda_{\mathcal{L}}\mathbf{u}$. One can easily derive the eigenvalues $\Lambda_{\mathcal{L}}$ of the operator $\mathcal{L}$ which are given by

$$\Lambda_{\mathcal{L}}(1) = -\pi^2 \left((\lambda + 2\mu)k_1^2 + \mu k_2^2 + (\lambda + \mu)k_1 k_2\right),$$

$$\Lambda_{\mathcal{L}}(2) = -\pi^2 \left(\mu k_1^2 + (\lambda + 2\mu)k_2^2 + (\lambda + \mu)k_1 k_2\right),$$

for $k_1, k_2 \in \mathbb{N} \setminus \{0\}$. Thus, the eigenvalues of $\Lambda_{\mathcal{E}_\nu}$ of the operator $\mathcal{E}_\nu$ write

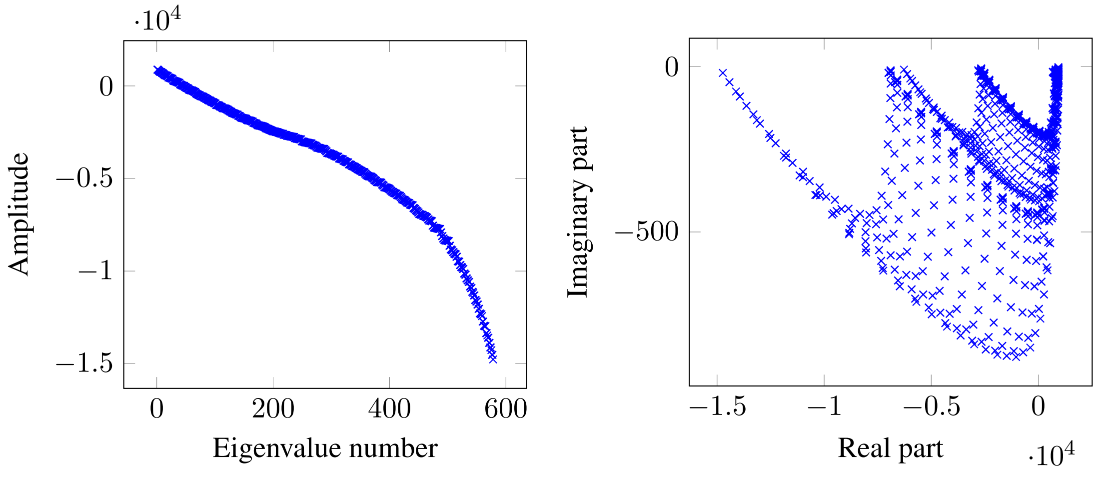
$$\Lambda_{\mathcal{E}_\nu} = \begin{bmatrix} \Lambda_{\mathcal{L}}(1) + z_\nu \omega^2 \rho & 0 \\ 0 & \Lambda_{\mathcal{L}}(2) + z_\nu \omega^2 \rho \end{bmatrix}. \tag{2.45}$$

Similarly to the Helmholtz operator, for large wavenumber, the real part of the eigenvalues of the operator $\mathcal{E}_\nu$ change sign which shows its indefiniteness.

The problem (2.44) is discretized using the fourth-order staggered-grid finite-difference method for the numerical simulations. The following sparse linear system is obtained

$$A\mathbf{x} = (L + z_\nu \omega^2 \rho I)\mathbf{x} = \mathbf{b}, \tag{2.46}$$

where L is the discretization of the partial derivative operator $\mathcal{L}$ with the homogeneous Dirichlet boundary condition (2.43). Considering the toy problem (2.4) and by setting $\nu = 0$ which amounts to a frequency-domain elastic wave propagation in a medium without attenuation, the eigenvalues of the matrix A are presented in Figure 2.23. The spectrum of the matrix A is presented as well with the PML boundary condition.

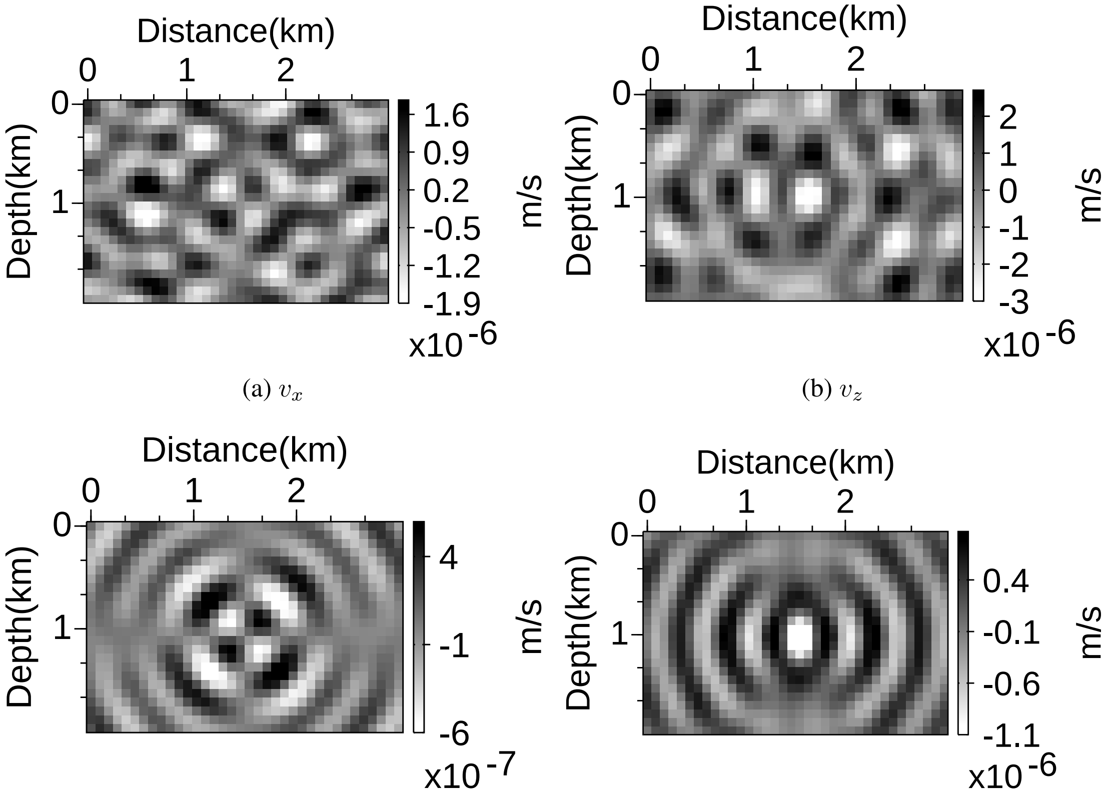


(a) Homogeneous Dirichlet boundary condition

(b) PML boundary condition

Figure 2.23: Spectrum of the impedance matrix A resulting from the discretization of 2D frequency-domain elastic wave equations on the toy problem 2.4 with homogeneous Dirichlet boundary condition (a) and PMLs (b).

It shows indeed the negative real part of its eigenvalues relative to L and some small positive ones as it has been discussed earlier. Finally, the solution of the linear system for a vertical force source located in the middle of the domain is presented in Figure 2.24.



(a) v_x (b) v_z

(c) v_x (d) v_z

Figure 2.24: Solutions of the 2D frequency-domain elastic wave problem on the toy problem 2.4 for a vertical force source located at $x_s = 1500$ m $z_s = 1000$ m with homogeneous Dirichlet boundary condition (a,b) and PML (c,d).

2.4.2 Spectral properties of the damped frequency-domain wave preconditioner

The damped frequency-domain wave preconditioner is introduced for the 2D frequency-domain elastic equations by

$$(\mathcal{L} + z\omega^2\rho I)\mathbf{v} = \mathbf{f}, \quad \mathcal{P}_z := \mathcal{L} + z\omega^2\rho I, \tag{2.47}$$

where z is complex number defined by

$$z = (\alpha - i\beta),$$

with α and β real positive numbers. The operator $\mathcal{P}_z$ is defined the same way as the elastic wave operator $\mathcal{E}_\nu$ using the homogeneous Dirichlet boundary condition. Following the equation (2.45), the eigenvalues of the operator $\mathcal{P}_z$ are given by

$$\Lambda_{\mathcal{P}_z} = \begin{bmatrix} \Lambda_{\mathcal{L}}(1) + z\omega^2\rho & 0 \\ 0 & \Lambda_{\mathcal{L}}(2) + z\omega^2\rho \end{bmatrix}.$$

The elastic wave operator $\mathcal{E}_\nu$ is preconditioned with $\mathcal{P}_z$ which gives

$$\mathcal{P}_z^{-1}\mathcal{E}_\nu\mathbf{v} = \mathcal{P}_z^{-1}\mathbf{f},$$

and the eigenvalues Γ of the operator $\mathcal{P}_z^{-1}\mathcal{E}_\nu$ are given by

$$\Gamma(1) = \frac{\mathcal{L}(1) + z_\nu\omega^2\rho}{\mathcal{L}(1) + z\omega^2\rho}, \quad \Gamma(2) = \frac{\mathcal{L}(2) + z_\nu\omega^2\rho}{\mathcal{L}(2) + z\omega^2\rho}.$$

Note that it is assumed that $\mathcal{L}(1) \neq -z\omega^2\rho$ and $\mathcal{L}(2) \neq -z\omega^2\rho$. In this case the preconditioner is singular and such case is avoided.

Theorem (2.4) applies here. For the 2D frequency domain elastic problem (2.44) with homogeneous Dirichlet boundary condition preconditioned by $\mathcal{P}_z$, the eigenvalues of the resulting operator $\mathcal{P}_z^{-1}\mathcal{E}_\nu$ lie on a circle with center c and radius r given by

$$c = \frac{z_\nu - \bar{z}}{z - \bar{z}}, \quad r = \left| \frac{z - z_\nu}{z - \bar{z}} \right| .$$

This circle is independent of the wavenumber k. The proof (2.3.2) can be carried out as well in the elastic case.

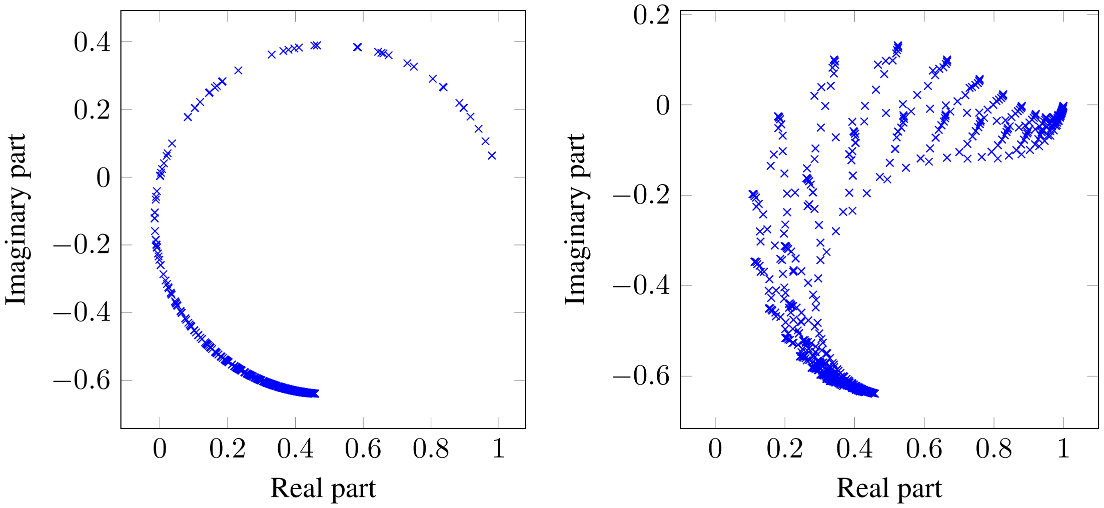
The system of equations (2.44) and (2.47) together with the homogeneous Dirichlet boundary equation (2.43) are discretized using the fourth-order staggered-grid finite-difference method which gives the sparse linear systems

$$A\mathbf{x} = \mathbf{b},$$

and

$$AG^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = G^{-1}\mathbf{y}.$$

The preconditioner is computed using the quality factor $q_{att} = 1.0$ to account for a strong damping. In the Figure 2.25, the spectrum of the matrix AG^{-1} in the homogeneous Dirichlet and PML configurations are presented. These results corroborate the theoretical results obtained earlier.



(a) Homogeneous Dirichlet boundary condition

(b) PML boundary condition

Figure 2.25: Eigenvalue distribution of the preconditioned matrix AG^{-1} with homogeneous Dirichlet boundary condition (a) and PML (b). The preconditioner is computed on a strongly damped medium with $q_{att} = 1.0$.

Table 2.6 summarizes the conditioner number of the matrices A and AP^{-1} in the two configurations: homogeneous Dirichlet and PMLs.

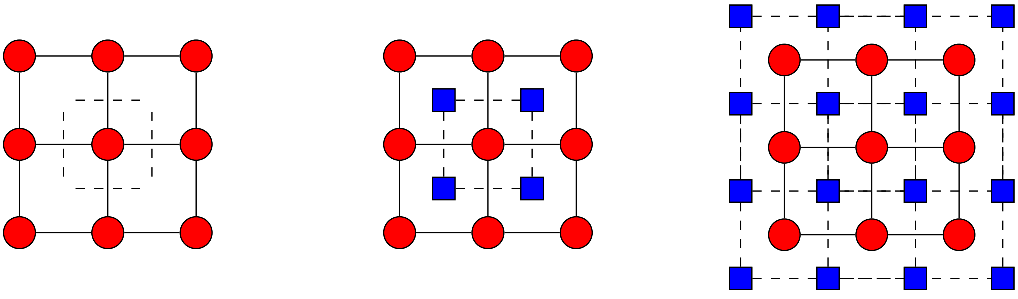
As expected, the spectrum of the matrix AG^{-1} is improved. Such preconditioner improves as well its condition number in both configurations.

	A	AG^{-1}
Dirichlet	4906.9	314.5
PML	138.8	10.4

Table 2.6: Condition number of the matrices A and AG^{-1} with homogeneous Dirichlet and PMLs configurations.

2.4.3 Sparse approximation of the damped frequency-domain wave preconditioner

In the 2D elastic approximation, two velocity wavefields are computed on a staggered grid. Therefore, selecting a sparsity pattern is not as straightforward as for the frequency-domain acoustic problem where only one pressure wavefield is computed. I carry out in the paragraph a study on the influence of the preconditioner sparsity pattern. Consider the toy problem 2.4. I investigate several sparsity patterns which can be adopted for the computation of the preconditioner. In the following, the matrix P denote the sparsely approximated preconditioner from the matrix G^{-1}.



(a) $9/0$ (b) $9/4$ (c) $9/16$

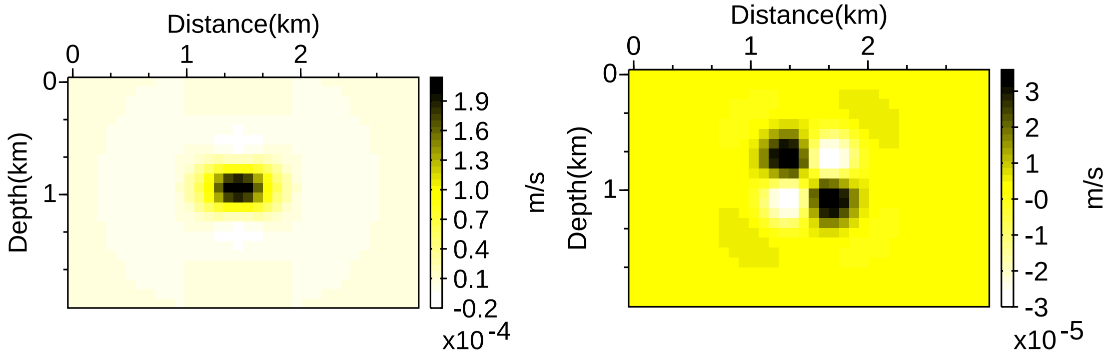
Figure 2.26: Sparsity patterns for the preconditioner in the 2D elastic approximation. For a horizontal force source, red circles denote the horizontal particle velocity v_x and blue square denote the vertical particle velocity v_z.

In Figure 2.26, the sparsity patterns $9/0$, $9/4$ and $9/16$ are presented. Following the solution of the 2D frequency-domain elastic wave problem in a strongly damped medium which can be seen in Figure 2.27, one can keep the 9 discretization points for v_x (red circles in Figure 2.26a) and either discard v_z as in the sparsity pattern $9/0$ or keep 4 or 16 discretization points as in the sparsity patterns $9/4$ and $9/16$ (blue squares in Figures 2.26b and 2.26c). In Figure 2.27c, the kinetic energy is presented. It shows that most of the energy is located at the 9 discretization points surrounding the source. Such observation suggests to consider preferably the sparsity pattern $9/0$.

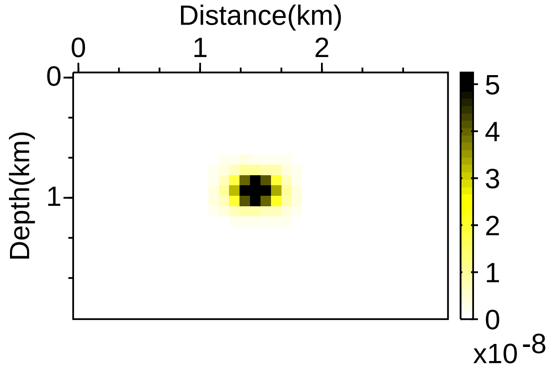
Moreover, such sparsity patterns can be enlarged with respect to the number of discretization points around the source. We refer to this operation as the sparsity approximation of a given sparsity pattern. In Figure 2.28, the sparsity approximation is enlarged from 1 point around the source to 2 for the sparsity pattern $9/4$.

In Table 2.7, by considering the kinetic energy $|v_x|^2 + |v_z|^2$, the different sparsity patterns cover most of the kinetic energy in the medium. The relative error for the kinetic energy

$$\text{Relative energy} = \frac{||\text{Exact energy} - \text{Sparse energy}||}{||\text{Exact energy}||},$$

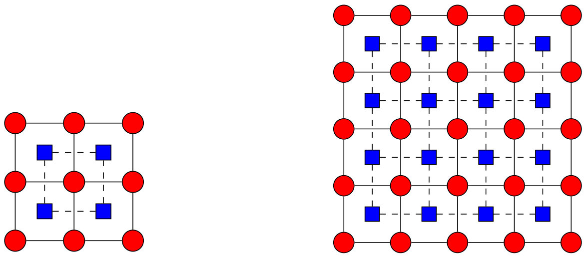


(a) v_x (b) v_z



(c) Energy $|v_x|^2 + |v_z|^2$

Figure 2.27: Horizontal (a) and vertical (b) particle velocity wavefields solution of the 2D frequency-domain elastic wave equations for a horizontal force source in a strongly damped homogeneous medium (Problem 2.4) with homogeneous Dirichlet boundary condition. Kinetic energy distribution is presented in figure (c).



(b) $25/16$

(a) $9/4$

Figure 2.28: Increasing the approximation of the sparsity pattern $9/4$ for the preconditioner in the 2D elastic approximation: one point around the source (a) and two points around the source (b).

remains small in particular for the sparsity pattern $9/0$ where the vertical particle velocity v_z is discarded.

These considerations of the kinetic energy are corroborated to the condition number of the matrices AP. The condition number of the matrix AP is approximately the same for all the sparsity pattern. In

Sparsity pattern	$9/0$		$9/4$		$9/16$	
Approximation	1	2	1	2	1	2
Kinetic energy $(\times 10^{-7})$	4.85	4.88	4.90	4.96	4.92	4.96
Relative error $(\%)$	14.4	5.7	14.3	5.2	13.5	4.8
Exact kinetic energy $(\times 10^{-7})$			4.97			

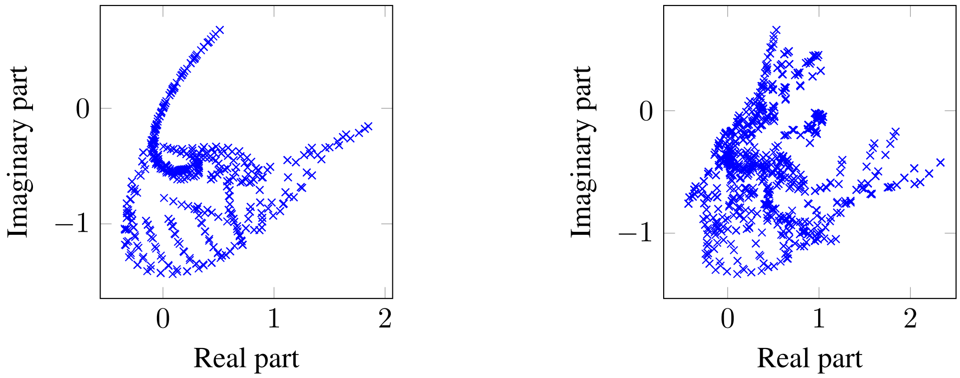
Table 2.7: Kinetic energy $|v_x|^2 + |v_z|^2$ with respect to the sparsity pattern chosen and the number of discretization points around the source and comparison with the exact solution.

the case where the homogeneous Dirichlet boundary condition is used, it is equal to

$$\operatorname{cond}(AP) = 900.4,$$

and with the PMLs, it is equal to

$$\operatorname{cond}(AP) = 34.5.$$



(a) Homogeneous Dirichlet boundary condition

(b) PML boundary condition

Figure 2.29: Eigenvalue distribution of the preconditioned matrix AP with homogeneous Dirichlet boundary condition (a) and PML (b). The preconditioner is computed on a strongly damped medium with $q_{att} = 1.0$ using the sparsity pattern $9/0$.

The spectrum of the matrix AP with the preconditioner computed using the sparsity pattern $9/0$ is presented in Figure 2.29. Compared to the spectrum of the matrix A (see Figure 2.23), the spectrum of the matrix AP is improved in both configurations (Dirichlet and PMLs). However, some eigenvalues have their real parts in both positive and negative half-planes. This matrix is therefore still indefinite but the eigenvalues are far from the origin and well clustered. Furthermore, the number of iterations performed by the CGMN method on the matrix AP is significantly reduced. This reduction is all the more improved when the sparsity pattern is denser ($9/0$, $9/4$, $9/16$). Table 2.8 shows the effect of the sparsity approximation on the number of iterations performed by CGMN on the toy problem (2.4) with PML at the boundaries.

Furthermore, one can expect that the quality of the preconditioner is improved by considering increasingly larger sparsity pattern. The number of iterations decreases accordingly. The same observation is made by considering each sparsity pattern at a time and by increasing the approximation (number

Sparsity pattern	N_{iter}	η
without preconditioning	285	1
$9/0$	66	2.7
$9/4$	59	3.1
$9/16$	54	4.3
Full	32	$>> 1$

Table 2.8: Number of iterations performed by CGMN with respect to the sparsity pattern of the preconditioner when solving the 2D frequency-domain elastic wave equations on the toy problem (2.4) with PMLs at the boundaries. η denotes the ratio $nnz(AP)/nnz(A)$ which is the increase in the memory request.

of discretization points around the source) as shown in Table 2.9. However, one needs to keep in mind the impact of such large sparsity pattern on the memory request denoted by the ratio

$$\eta = \frac{nnz(AP)}{nnz(A)}.$$

Indeed, for 3D applications, we are very concerned with the memory requirements. Therefore, I shall consider the sparsity pattern $9/0$ which induces the least memory increase.

Approx.	N_{iter}	η
1 (9/0)	66	2.7
2 (25/0)	58	4.9

(a) $9/0$

Approx.	N_{iter}	η
1 (9/4)	59	3.1
2 (25/16)	40	5.5

(b) $9/4$

Approx.	N_{iter}	η
1 (9/16)	54	4.3
2 (25/36)	39	6.9

(c) $9/16$

Table 2.9: Number of iterations performed by CGMN with respect to the increase of the number of discretization points in the sparsity pattern of the preconditioner when solving 2D frequency-domain elastic wave equations on the toy problem (2.4) with PMLs at the boundaries. η is the ratio $nnz(AP)/nnz(A)$ denoting the increase in the memory request.

In Tables 2.8 and 2.9, the memory increase is presented for each sparsity pattern. We observe that the memory request increases quickly when large approximations are used which is prohibitive considering the future 3D applications. Finally, Table 2.10 shows the summary of the condition number

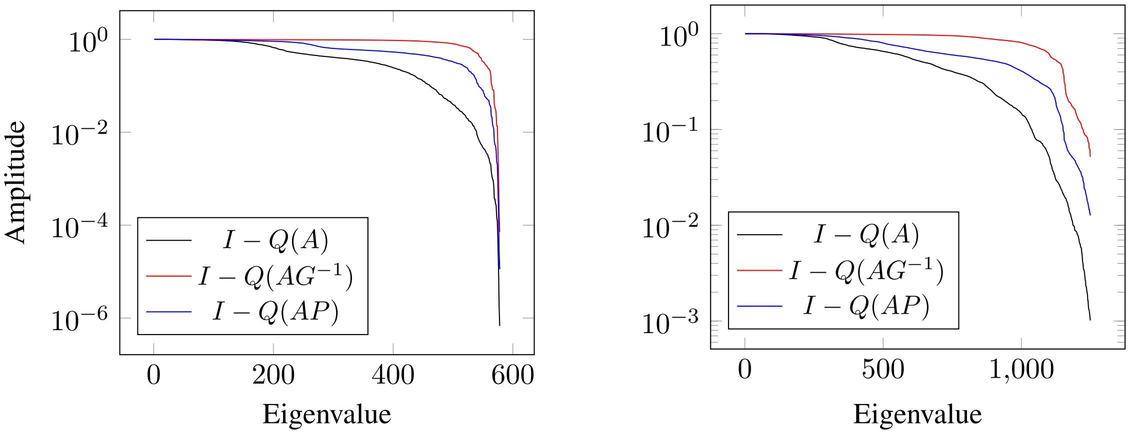
	A	AG^{-1}	AP
Dirichlet	4906.9	314.5	900.4
PML	138.8	10.4	34.5

Table 2.10: Condition number of the matrices A, AG^{-1} and AP with homogeneous Dirichlet and PML configurations. The sparse preconditioner P is computed using the sparsity pattern $9/0$ from the fully dense preconditioner G^{-1}.

of the matrices A, AG^{-1} and AP when homogeneous Dirichlet boundary condition and PMLs are used. These results show the efficiency of the dense and sparse preconditioners.

2.4.4 Spectral analysis of the matrices $I - Q(AG^{-1})$ and $I - Q(AP)$

In this section, I focus on the spectral properties of the matrices $I-Q(A)$, $I-Q(AG^{-1})$ and $I-Q(AP)$ where the preconditioner G^{-1} is build on the damped medium using $q_{att} = 1$ without any sparse approximation. Furthermore, P denotes the sparse approximate inverse preconditioner which is a sparse approximation of G^{-1} using the $9/0$ point sparsity pattern. The spectrum of the matrices $I-Q(AG^{-1})$ (red curve) and $I-Q(AP)$ (blue curve) are presented in Figure 2.30. They are compared to the spectrum of the matrix $I - Q(A)$ (black curve). On the Figure 2.30a, the 2D frequency-domain elastic wave equations are defined using the homogeneous Dirichlet boundary condition and on the Figure 2.30b, PMLs are used.



(a) Homogeneous Dirichlet boundary condition

(b) PML

Figure 2.30: Eigenvalue distribution of the matrices $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$ with homogeneous Dirichlet boundary condition (a) and PML (b). The preconditioner G^{-1} is computed upon the strongly damped medium using $q_{att} = 1$ without any sparse approximation and P is the sparse approximation of G^{-1} using the 9-point sparsity pattern.

Ideally, when the full preconditioner G^{-1} is used (red curves), the spectrum of the matrix $I - Q(AG^{-1})$ is significantly improved compared to the spectrum of the matrix $I - Q(A)$ (black curves). The eigenvalues are clustered around 1. When the sparse approximation is introduced (blue curves), the spectrum of the matrix $I - Q(AP)$ remains well clustered around 1 despite the drastic sparse approximation.

The condition number of the matrices A, AG^{-1}, AP, A^HA, $I - Q(A)$, $I - Q(AG^{-1})$ and $I - Q(AP)$ are summarized in Table 2.11.

	A	AG^{-1}	AP	A^HA	$I-Q(A)$	$I-Q(AG^{-1})$	$I-Q(AP)$
Dirichlet	4906.9	314.5	900.4	$2.4\ 10^7$	$1.4\ 10^6$	1053.8	14037.7
PML	138.8	10.4	34.5	$1.9\ 10^4$	983.6	19.1	78.7

Table 2.11: Condition number of the matrices A, AG^{-1}, AP, A^HA, $I - Q(A)$, $I - Q(AG^{-1})$ and $I-Q(AP)$ with homogeneous Dirichlet and PMLs configurations in the 2D elastic approximation. The preconditioner P is computed using the $9/0$-point sparsity pattern from the fully dense preconditioner G^{-1}.

Both dense and sparse preconditioners improve the conditioning of the matrix A. The matrix $I - Q(AG^{-1})$ has clearly a better conditioning compared to the matrix associated with the normal equations $A^H A$. Despite the sparse approximation on the preconditioner, the matrix $I - Q(A\tilde{P}^{-1})$ still remains very well conditioned.

2.5 Conclusion

In this chapter, the CARP-CG method is introduced. This method solves robustly the frequency-domain acoustic and elastic wave equations in both 2D and 3D approximations in complex media and for large scale problems. We are mainly interested in improving the spectral properties of the frequency-domain wave equations in both acoustic and elastic approximations and more importantly, investigate how a preconditioning strategy can be adapted to the CARP-CG method when applied to the frequency-domain wave equations.

Two preconditioning strategies are introduced for the frequency-domain wave equations and the CARP-CG method. The preconditioning strategy which is retained is based on applying the CARP-CG method to the preconditioned linear system

$$AP\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P\mathbf{y},$$

where P is a sparse approximate inverse which the computation is based on a version of A in a strongly damped medium. I presented the damped frequency-domain wave preconditioner in the 2D approximation for the acoustic and elastic wave equations. Using artificial damping, a complex diagonal shift is introduced in the preconditioner allowing the improvement of the conditioning of the overall matrix AP. The eigenvalues are localized in a closed-off area away from the origin assuming that the preconditioner is computed exactly for this theoretical study. Thanks to the strong damping, this preconditioner is sparsely approximated by selecting few discretization points around the source area. Through this sparse approximation, the preconditioner remains efficient on these small problems. The performances of the CARP-CG method are improved with the use of such preconditioner. However, one needs to pay close attention of the sparsity pattern of the preconditioner as its use with CARP-CG may lead to severe memory costs.

In practice, the preconditioner is computed efficiently column-wise by solving the n small linear systems corresponding to the frequency-domain wave problems in the small domain defined by the approximation. Perfectly Matched Layers are used to absorb the outgoing waves and truncate the domain of computation. Therefore, the computation of the precondition is embarrassingly parallel as each column can be computed independently and presents low memory requirements. More importantly, the sparsity pattern of the preconditioner can be controlled easily with respect to the number of discretization points accounted for in the approximation and thus one has a hold over the memory request for the computation of the matrix AP. Finally, the computation of the preconditioned system can be performed in low complexity $O(n)$ thanks to the well-known sparsity pattern of the matrices A and P and their sparsity. Therefore, for the frequency-domain acoustic wave equation, the 9-point sparsity pattern is adopted for the upcoming numerical experiments. For the 2D elastic approximation, the $9/0$ sparsity pattern is used.

However, some questions rise: indeed, the preconditioning strategy is proven effective in small homogeneous cases. What would one expect when this preconditioner is used on larger and heterogeneous problems? Moreover, what would be the effect of another kind of boundary condition such as the free-surface. Regarding the computation of the preconditioner and in the perspective of 3D applications, to what extent the computation of such preconditioner can be affordable? Could the computational cost of the preconditioner be reduced under the assumption of homogeneous media? These are the questions we investigate in the next two chapters.

Chapter 3

Application to 2D realistic cases

Contents

Seismic imaging of crustal structures is one of the main challenges of geophysical exploration at different scales for oil exploration and crustal investigations. At the exploration scale, heterogeneous properties of the media are encountered with strong contrasts in both density and velocity parameters. So far in this work, problems only in homogeneous media were treated in the analysis and the numerical tests. In this chapter, 2D typical problems often investigated in seismic imaging applications are investigated. I shall pay a particular attention to complexity issues as we are looking for extensions in 3D which is mandatory in seismic imaging.

The 2D frequency-domain acoustic wave equation is considered in the first part. The sequential CGMN method is used for the solution of this problem. Several numerical experiments are performed to investigate the convergence of this method. The complexity and convergence histories of the CGMN method are investigates at for several frequencies. These experiments are performed on the BP2004 model (Billette and Brandsberg-Dahl, 2004). This model is 67 km long and 12 km deep and provides heterogeneous P-wave velocity and density model with strong heterogeneities. Furthermore, the convergence properties of the preconditioned CGMN method are investigated as well. The preconditioner is computed using the 9-point sparsity pattern introduced in the previous chapter. Therefore, the preconditioner is sparse with only 9 nonzero coefficients per column and it is computed upon the same media but with a strong damping.

In the second part, the CARP-CG method is used to solve the 2D frequency-domain elastic wave equations. The Marmousi 2 elastic model (Martin et al., 2006) is used with its P-wave and S-wave velocity models and its density model. The results using different frequencies are presented to show the complexity of the CARP-CG method with respect to the size of the problem. Furthermore, experiments with several processors show its scaling properties. The convergence properties of the CARP-CG method are then compared with those of the preconditioned CARP-CG with the sparse approximate inverse preconditioner built on the same media but with a strong damping. The $9/0$ sparsity pattern is used, ensuring a sparse preconditioner.

3.1 Numerical applications in the acoustic approximation

In this section, the 2D frequency-domain acoustic wave equation is solved on the heterogeneous density and velocity BP2004 model. I consider the CGMN method as the resulting linear systems are of relatively small size for these 2D problems. The CARP-CG method will be used in the elastic approximation where the size of these linear systems requires a parallel strategy even when considering 2D problems. The convergence properties of the CGMN method with and without preconditioning are finally compared.

3.1.1 Presentation of the BP2004 model

The numerical experiments are performed using the 67 km long and 12 km deep BP2004 model provided by Billette and Brandsberg-Dahl (2004). The velocity and density fields are presented in Figure 3.1.

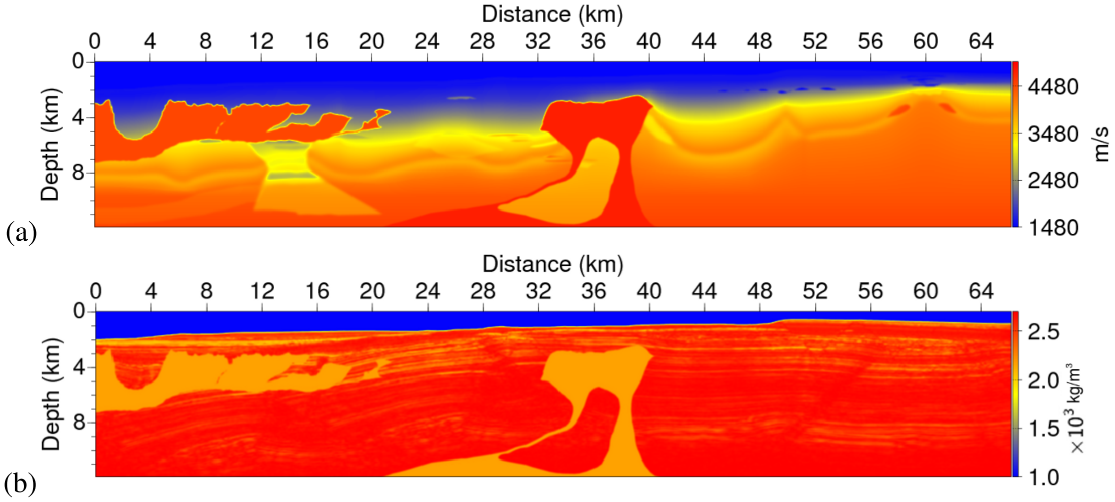


Figure 3.1: BP2004 model: 67 km × 12 km P-wave velocity model (a) and density model (b). This model is characterized by its large size and its strong heterogeneities.

The model was built on a 6.25 m × 6.25 m grid. The top part is a water layer with a constant P-wave velocity 1486 m/s. The left part presents a complex rugose salt body with a constant velocity of 4510 m/s. A challenge comes from the sub-salt low velocity layers that are meant to represent over-pressured zones. The center part of the model is representative for the Gulf of Mexico and West Africa with a constant P-wave velocity 4510 m/s in the salt body.

3.1.2 Numerical results

The 2D frequency-domain acoustic wave equation is discretized using the fourth-order staggered-grid finite difference method (Levander, 1988). Perfectly matched layers (Bérenger, 1994) are implemented at the edges of the domain to absorb the outgoing waves. $N_{PML} = 5$ grid points are used in the PMLs and the damping coefficient in the PML is set to $C_{PML} = 1500$. The explosive Dirac source is located at the middle top of the domain at $x_s = 33$ km and $z_s = 150$ m. The medium is considered without attenuation by setting the quality factor to $q_{att} = 1000$. Several experiments are performed to illustrate the performance of the CGMN method for the solution of the 2D frequency-domain acoustic wave equation on the BP2004 model. For each experiment, the starting point $\mathbf{x}^{(0)}$ is set to $\mathbf{0}_n$. The relaxation parameter used for the Kaczmarz projections is set to 1.2 which is the optimal value selected from several experiments.

The BP2004 model, presented in Figure 3.1 is filtered and then sampled for several frequencies $f = \{1,\ 5,\ 10,\ 20,\ 40\}$ Hz. I use 5 grid points per wavelength to ensure the accuracy of the fourth-order finite-difference scheme as promoted by Levander (1988). For the low frequencies, a smoothing operator is applied on the model to avoid aliasing effects. Table 3.1 summarizes the geometric parameters of the models used for each frequency. The solutions for the different frequencies at displayed in Figure 3.4 at the end of this section.

f	h	N_z	N_x	N_u
1	300	41	224	11551
5	60	201	1117	232617
10	30	401	2234	904128
20	15	801	4468	3564132
40	7.5	1601	8934	14152068

Table 3.1: Geometric parameters of the discretization of the BP2004 model $(h,\ N_x,\ N_z)$ and size of the linear systems N_u. Such frequency simulations correspond to the propagation of few tens of wavelengths (8 in the vertical direction and 44 in the horizontal direction for $f = 1$ Hz) and up to several hundreds of wavelengths (320 in the vertical direction and 1780 in the horizontal direction for $f = 40$ Hz).

In FWI, performing frequency-domain acoustic wave simulations are usually not required at high frequencies such as $f = 40$ Hz. In this study, I am mainly interested in the behavior of the iterative solver CGMN for such high frequency problems and more importantly whether the preconditioning still applies for such indefinite linear systems.

The preconditioner P is computed upon a strongly damped medium with $q_{att} = 1$ and using the 9-point sparsity pattern. The results obtained with this preconditioner are shown in Table 3.2.

The number of iterations performed by the CGMN method for the solution of each linear system with respect to the frequency are given together with the computation time. The same data are provided for the preconditioned CGMN method (PCGMN). Note that the computation time of the preconditioner is not included in the run times. I consider that the preconditioner can be computed once and therefore be used for the solution of the same linear system but with multiple right-hand sides for instance. The results in Table 3.2 show that the number of iterations is divided by a factor ranging from 5.1 to 6.9 between the CGMN method (without preconditioning) and the PCGMN (with preconditioning). More

		N_{iter}		Time (s)	
f	N_u	CGMN	PCGMN	CGMN	PCGMN
1	11551	845	164	2.3	1.4
5	232617	3313	569	206.9	109.7
10	904128	5959	951	1443.5	728.5
20	3564132	9525	1424	9195	4355.4
40	14152068	13469	1942	51480.9	23402.6

Table 3.2: Number of iterations and computation time performed by the CGMN method for the solution of the 2D frequency-domain acoustic wave equation on the BP2004 model. PCGMN denotes the preconditioned iterative method CGMN. The size of the linear systems ranges from ten thousands to ten millions of unknowns. Note that the computation time denotes the CGMN and the PCGMN run times and does not include the preconditioner computation time. The latter is performed apart, independently from the PCGMN runs.

interestingly, the efficiency of the preconditioner increases with respect to the frequency. Such reduction in the number of iterations even allows to note a gain in the computation time by a factor up to 2.2. Note the difference in the reduction in the number of iterations and the computation time. Indeed, this gain in the number of iterations is compensated by the increase of the cost of each iteration. The matrix AP has 45 nonzero coefficients per row whereas the matrix A has only 13 nonzero coefficients. The improvement of the convergence profile of the residuals for $f = 5$ Hz is shown in Figure 3.2.

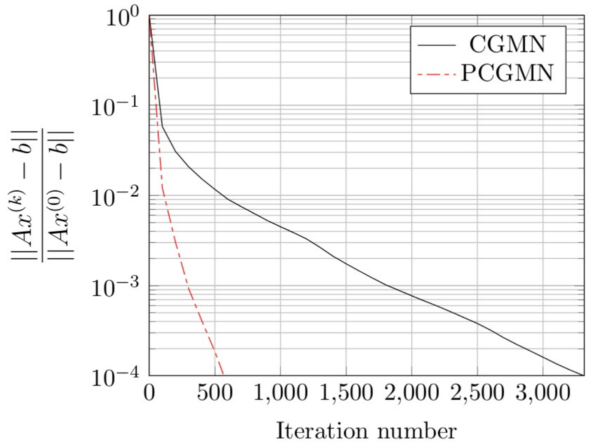


Figure 3.2: Convergence histories of the relative residuals of the 2D frequency-domain acoustic wave simulations using the CGMN method (black solid curve) and the preconditioned CGMN method (red dashed curve) for $f = 5$ Hz on the BP2004 model.

One can see a drastic convergence towards the stopping criterion when considering the preconditioned CGMN method (PCGMN). Unfortunately, this is mitigated by the fill-in in the matrix AP.

A complexity analysis of the CGMN method is presented in Figure 3.3. The number of iterations N_{iter} performed by the preconditioned and non-preconditioned CGMN method is plotted as a function of the average number of discretization points per direction N

$$N = \sqrt{N_x \times N_z},$$

on a log-log scale. The black solid line represents the linear increase of the computational complexity

$$N_{iter} = \mathcal{O}(N).$$

The solid red curve indicates the dependence between N_{iter} and N for the CGMN method. A standard linear regression, indicated by the solid blue line, is performed on the dashed red curve. A slope equal to 0.71 for the CGMN method without preconditioning is obtained. The computational complexity is sub-linear with respect to N, which is consistent with previous studies performed in the previous Chapter 2 Section 2.1.6 and in (Li et al., 2015). Based on this analysis, the complexity of the CGMN method is approximately evaluated as

$$N_{iter} = 20.9\ N^{0.79}.$$

Using the preconditioner, the complexity is improved to

$$N_{iter} = 5.9\ N^{0.71}.$$

Therefore, following the analysis presented in Chapter 1 Section 1.6, the total computational complexity of the CGMN method for the 2D simulations with one source is

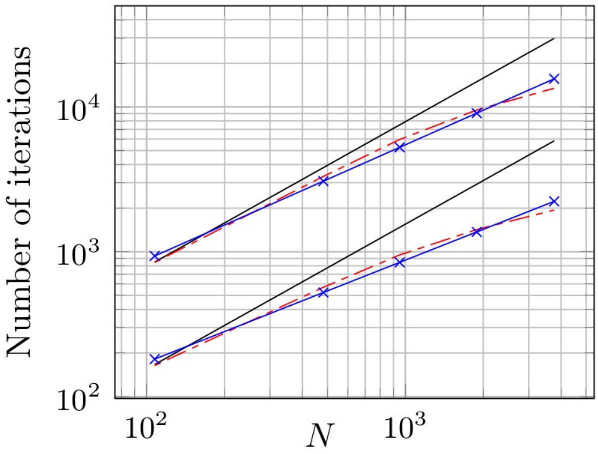


Figure 3.3: Complexity analysis of the CGMN method (red dashed curve) with and (red dashed curve) without the preconditioner for the solution of the 2D frequency-domain acoustic wave equation on the BP2004 model. The black solid line and the blue “x”-marked line represent respectively the linear increase and trend line.

$$\mathcal{O}(N_{iter} \times N^2) = \mathcal{O}(N^{2.71}), \tag{3.1}$$

which is a noticeable improvement compared to $\mathcal{O}(N^3)$ performed by standard time-domain modeling techniques. However, such reduction is not as important as one may wish. One possible reason is the difficulty that is introduced by the heterogeneous density.

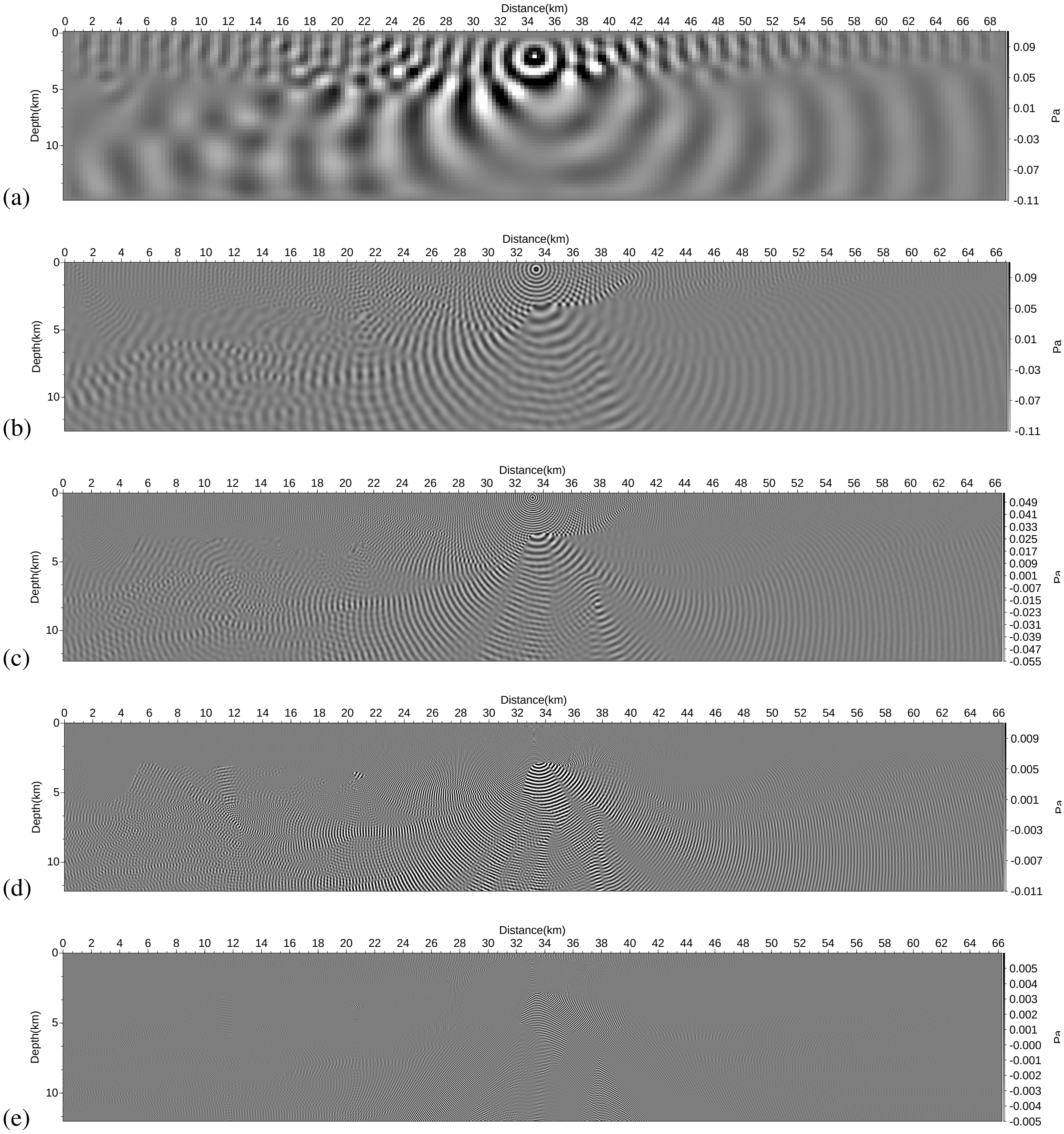


Figure 3.4: 2D frequency-domain pressure wavefield (real part) solutions on the BP2004 model: 1 Hz (a), 5 Hz (b), 10 Hz (c), 20 Hz (d) and 40 Hz (e). One can guess the the reflectivity in the wavefield at the surface, at the edges of the center part of the salt body and below the left part salt body. This geometry creates a waveguide that can be seen in the Figures (b) and (c) between the two salt bodies.

3.2 Numerical applications in the 2D elastic approximation

In this section, the 2D frequency-domain elastic wave equations are solved on the well-known heterogeneous Marmousi 2 density and velocity model. The parallel solver CARP-CG is used to solve the resulting linear systems. A complexity analysis and the scaling properties of the CARP-CG method are investigated. The sparsity pattern for the preconditioner is introduced. The performances of the CARP-CG method on the preconditioned system are compared with the results obtained from the non-preconditioned system.

3.2.1 Presentation of the Marmousi 2 model

Following the description of the Marmousi 2 model performed in Chapter 2 Section 2.2.4, we simply note that we consider the shear-wave velocity model necessary for elastic wave simulations.

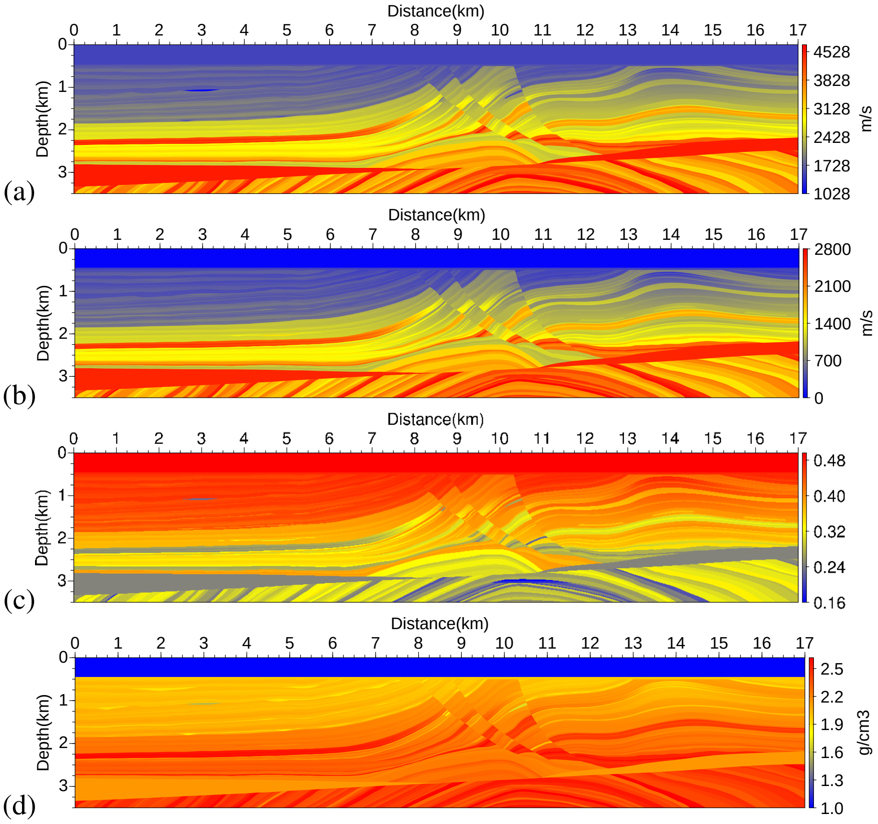


Figure 3.5: Marmousi 2 model: P-wave velocity (a), S-wave velocity (b), Poisson's ratio (c) and density fields (d). Oil reservoirs are located $x = 3000$ m $z = 1000$ m, $x = 2500$ m $z = 1700$ m, $x = 10000$ m $z = 1200$ m, $x = 11000$ m $z = 2000$ m and $x = 10500$ m $z = 3000$ m. They can be seen in the Poisson's ratio panel with the low values (blue).

The Marmousi 2 model has very low V_S values, i.e. high Poisson's ratios (see Figure 3.5(c)). This requires a fine discretization with respect to the lowest shear wave velocity which gives a large number of grid points and thus large linear systems to solve. This feature makes the model particularly interesting to study. For this reason, the parallel iterative solver CARP-CG is used to solve the linear systems.

3.2.2 Numerical results

The 2D frequency-domain elastic wave equations in velocity formulation is discretized using the fourth-order staggered-grid finite-difference method with optimal coefficients (see Chapter 1 Section 1.5.1). The discretization of the Marmousi 2 model is performed using 5.4 grid points per minimum shear wavelength. An explosive Dirac source is used at the position $x_s = 8500$ m and $z_s = 100$ m. The initial point $\mathbf{x}^{(0)}$ is set to $\mathbf{0}_n$. The relaxation parameter used for the Kaczmarz projections is set to 1 and the stopping criterion for the iterations is based on the relative residual

$$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|}{\|\mathbf{b}\|} < 10^{-4}.$$

The sparsity pattern $9/0$ is used for the computation of the preconditioner. For all numerical experiments, the computation time of the preconditioner is not included in the computation time presented for the preconditioned CARP-CG method. This aspect will be discussed for the 3D application in Chapter 4.

In the following, I investigate the influence of the different physical parameters on the convergence of the CARP-CG method. These investigations aim to characterize the influence of the Poisson's ratio, the PMLs and the free-surface boundary condition. In the first part, the optimal value for the quality factor for the computation of the preconditioner is investigated. The value of q_{att} providing the minimum number of iterations for the preconditioned CARP-CG method is therefore retained for the following studies. In the second part, numerical experiments are performed on the Marmousi 2 model with constant Poisson's ratio. The shear velocity model is derived from the P-wave velocity model using the Poisson's relation (3.2). The values for the Poisson's ratio $\{0.25,\ 0.33,\ 0.45\}$ are investigated with the PML configuration. For the model with the constant Poisson's ratio $\xi = 0.25$, numerical experiments are carried out to investigate the influence of the free-surface. The results are then compared with those of the PML case. In the third part, the original Marmousi 2 model is considered with its space-dependent Poisson's ratio. The numerical experiments are performed with both PMLs and free-surface boundary conditions. In the last part, the scaling properties of the CARP-CG method (with and without preconditioning) are investigated.

3.2.2.1 Optimal value of damping for the best preconditioner efficiency

The good performances of the preconditioner come from the level of damping introduced in the medium for the computation of the preconditioner so that a good balance between accuracy of the preconditioner and diagonal dominance is obtained. In this part, I investigate the sensitivity of the convergence of the preconditioned CARP-CG method in number of iterations with respect to the quality factor q_{att} used to compute the preconditioner.

Consider the Marmousi 2 model built with a constant Poisson's ratio which is set to $\xi = 0.25$. The shear velocity model V_S is derived from V_P using the Poisson's relation

$$\frac{V_S}{V_P} = \sqrt{\frac{0.5 - \xi}{1 - \xi}}. \tag{3.2}$$

The frequency-domain elastic wave simulations are performed using PMLs on all boundaries for the frequencies $\{1.25,\ 2.5,\ 5,\ 10,\ 20\}$ Hz. Details on the geometric parameters of the discretization can be found in Table 3.3 in the next section. Figure 3.6 shows the number of iterations performed by

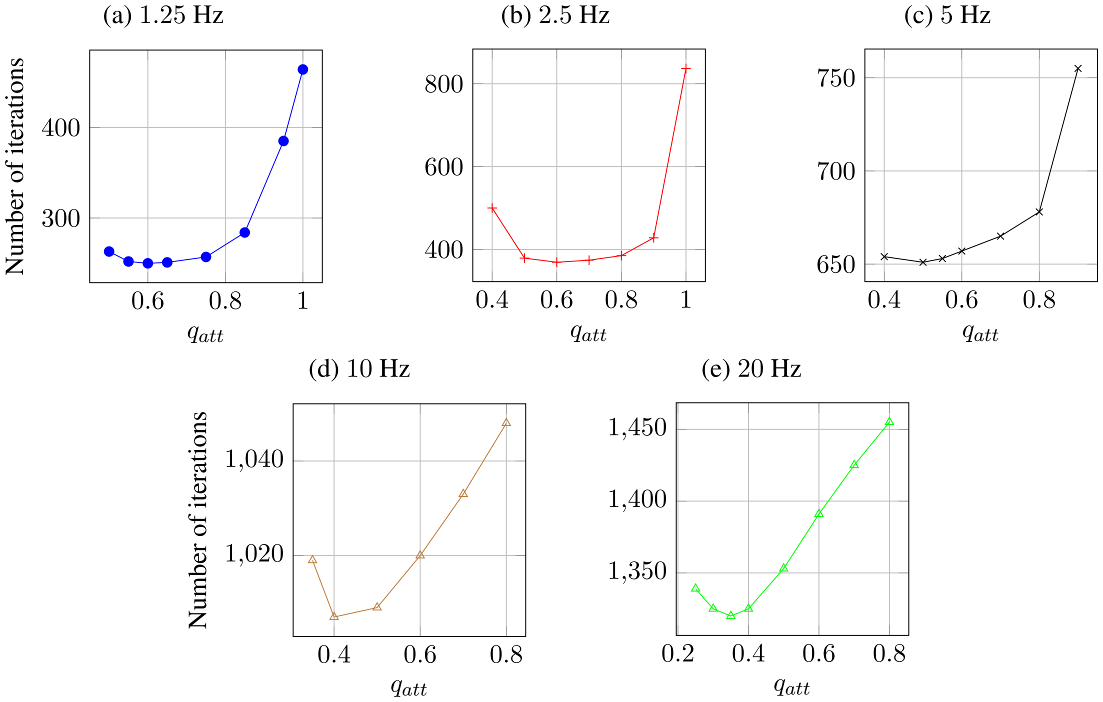


Figure 3.6: Sensitivity analysis of the convergence of the preconditioned CARP-CG method (PCARP-CG) with respect to the damping used to build the preconditioner. The simulations are performed on the Marmousi 2 model where the shear wave velocity model is derived from the P-wave velocity model using the Poisson's relation (3.2) with a constant Poisson's ratio $\xi = 0.25$.

the preconditioned CARP-CG method using a preconditioner computed upon a damped medium using q_{att}. The results clearly show that the optimal value for the damping is frequency-dependent. These optimal values range from 0.35 for the highest frequency ($f = 20$ Hz) to 0.6 for the lowest frequency ($f = 1.25$ Hz). Nevertheless, within this range the value $q_{att} = 0.6$ seems to be optimal for all frequencies. The number of iterations varies slightly (couple of tens) for the high frequencies and this is completely insignificant compared to the tens of thousands of iterations performed by the CARP-CG method without preconditioning.

Therefore, in the following the optimal value of damping $q_{att} = 0.6$ is used to compute the preconditioner for all simulated frequencies.

3.2.2.2 Marmousi 2 case study with constant Poisson's ratio

Poisson's ratio The Poisson's ratio is an important factor for describing an elastic medium. It is related to the P- and S-wave velocities through the relation (3.2). If $\xi = 0.5$, i.e. $V_S = 0$, it reduces the elastic wave modeling to the acoustic approximation. A Poisson's ratio close to 0.5 corresponds to soft materials. In this case, the ratio V_S/V_P is small and the discretization resulting from such configuration gives large problems to solve as the discretization is performed according to the smallest wavelength which corresponds to the shear wave velocity (see Figure 3.7). On the contrary, a lower Poisson's ratio, around the value 0.25 for instance, corresponds to harder media. The discretization is thus coarser which yields linear systems which are easier to solve.

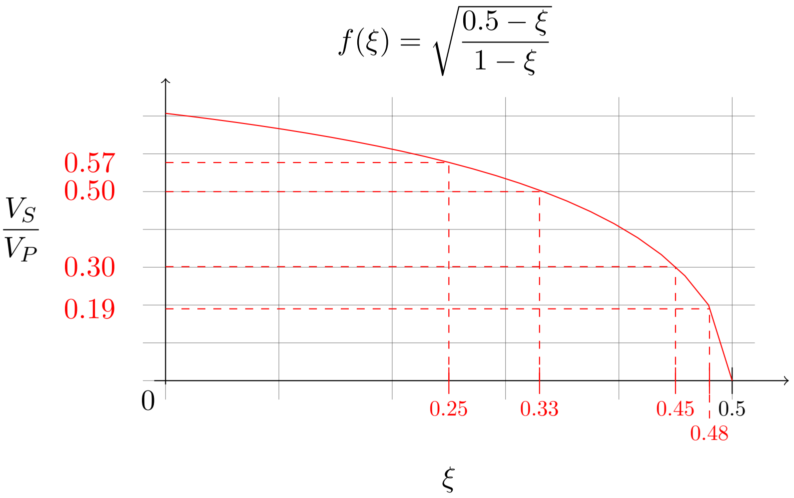


Figure 3.7: Ratio V_S/V_P profile.

To investigate the influence of the Poisson's ratio, the frequency-domain elastic wave simulations are performed in three cases with a constant increasing Poisson's ratio. In the first case, a constant Poisson's ratio equal to 0.25 is used. The ratio is increased to 0.33 in the second case and finally to 0.45 in the last case. For each case, the shear wave velocity model is derived from the P-wave velocity model using the Poisson relation (3.2). Therefore, the minimum shear wave velocities are $V_{S,min} = \{593.52,\ 517.82,\ 309.95\}\ m/s$. Numerical experiments using the space-dependent Poisson's ratio from the original Marmousi 2 model are performed in the next part.

The discretization is performed with respect to the minimum shear wavelength for all three cases. The preconditioner is computed based on a strongly damped medium using the quality factor $q_{att} = 0.6$. The sparsity pattern $9/0$ is used (see Figure 2.26a in Chapter 2 Section 2.4.3). The details of the discretization of the media are presented in Tables 3.3, 3.4 and 3.5. The number of iterations and the computation time (in seconds) are presented for the experiments using the CARP-CG method and the preconditioned CARP-CG method (PCARP-CG). Representative wavefields of the horizontal particle velocity v_x and the vertical particle velocity v_z at each frequency are shown in the Figures 3.9 and 3.10 at the end of this part.

These results show that for the constant Poisson's ratio $\xi = 0.25$ simulations, the preconditioned method PCARP-CG is able to solve the linear systems using a small number of iterations. In Table 3.3, one can see that the number of iterations is divided by a factor ranging from 4.71 up to 6.75. Such reductions in the number of iterations allow to reduce the computation time by a factor going from 1.64 up to 2.52.

Similarly, from the results shown in Table 3.4, one can see that in the case where the constant Poisson's ratio $\xi = 0.33$ is used, the preconditioner allows to reduce the number of iterations by factor 4.75 and up to 6.8. A speed up in computation time by a factor 1.94 and up to 2.62 is obtained.

Finally, for the last case $\xi = 0.45$, the results from Table 3.5 show a reduction in the number of iterations by a factor going from 3.91 up to 5.45 allowing as well to reduce the computation time by a factor 1.55 and up to 2.3.

These results show that for constant Poisson's ratio simulations, the preconditioned method PCARP-CG is able to achieve substantial reductions in the number of iterations. The first conclusion which can

$\xi = 0.25$					CARP-CG		PCARP-CG	
f	h (m)	N_z	N_x	N_u	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	88	41	194	37908	1532	20.5	325	12.5
2.5	44	81	388	103576	2487	101.0	429	46.8
5	22	160	774	325600	4211	556.2	698	243.6
10	11	319	1548	1140184	9226	338.8	1366	127.4
20	5.5	628	3094	4187024	12568	505.8	2080	200.1

Table 3.3: Geometric parameters, total number of unknowns N_u, computing time and number of iterations N_{iter} for the solution of the 2D frequency-domain elastic wave equations using CARP-CG and PCARP-CG with PMLs at the boundaries on the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.25$. The preconditioner is computed upon the Marmousi 2 model with a strong damping using $q_{att} = 0.6$ and according to the $9/0$ sparsity pattern.

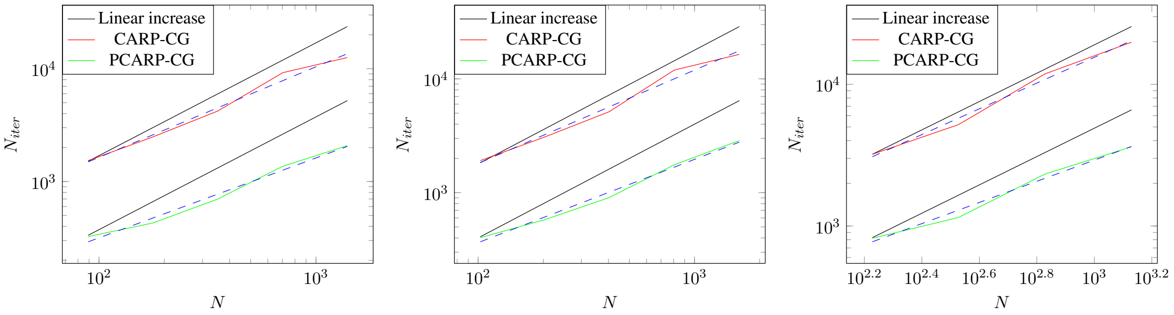
$\xi = 0.33$					CARP-CG		PCARP-CG	
f	h (m)	N_z	N_x	N_u	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	76.71	47	223	45762	1913	36.9	402	19.0
2.5	38.36	92	444	127776	3074	162.9	577	77.6
5	19.19	183	887	413442	5143	878.6	905	406.2
10	9.59	366	1774	1472968	11947	162.4	1758	61.8
20	4.79	731	3547	5531154	16419	841.2	2870	355.1

Table 3.4: Geometric parameters, total number of unknowns N_u, computing time and number of iterations N_{iter} for the solution of the 2D frequency-domain elastic wave equations using CARP-CG and PCARP-CG with PMLs at the boundaries on the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.33$. The preconditioner is computed upon the Marmousi 2 model with a strong damping using $q_{att} = 0.6$ and according to the $9/0$ sparsity pattern.

$\xi = 0.45$					CARP-CG		PCARP-CG	
f	h (m)	N_z	N_x	N_u	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	45.92	77	371	6174	3207	128.7	819	82.9
2.5	22.96	153	741	301466	5191	625.7	1148	374.2
5	11.48	306	1482	1053224	11839	398.5	2324	199.5
10	5.74	611	2963	3909906	19794	728.4	3619	317.1

Table 3.5: Geometric parameters, total number of unknowns N_u, computing time and number of iterations N_{iter} for the solution of the 2D frequency-domain elastic wave equations using CARP-CG and PCARP-CG with PMLs at the boundaries on the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.45$. The preconditioner is computed upon the Marmousi 2 model with a strong damping using $q_{att} = 0.6$ and according to the $9/0$ sparsity pattern.

be drawn from these results is, despite the heterogeneities in the velocity models and density model, the combination of the preconditioner with the CARP-CG method allows to retrieve a significant reduction in the number of iterations permitting as well a reduction in the computation time.



(a) $\xi = 0.25$ (b) $\xi = 0.33$ (c) $\xi = 0.45$

Figure 3.8: Complexity analysis of the CARP-CG method with and without preconditioning for the solution of the 2D frequency-domain elastic wave equations with PMLs on the Marmousi 2 model with homogeneous Poisson's ratio: $\xi = 0.25$ (a), $\xi = 0.33$ (b) and $\xi = 0.45$ (c). The number of iterations N_{iter} is plotted as a function of the average model dimension N on a log-log scale. The straight dashed line indicates the results obtained from a line regression.

Figure 3.8 shows that the complexity in number of iterations of the CARP-CG method (red solid curves) remains sub-linear with respect to the average number of grid points per dimension N.

	N_{iter}	
ξ	CARP-CG	PCARP-CG
0.25	$49 \times N^{0.76}$	$15 \times N^{0.67}$
0.33	$51 \times N^{0.78}$	$14 \times N^{0.71}$
0.45	$35 \times N^{0.87}$	$20 \times N^{0.71}$

Table 3.6: Complexities in number of iterations of the CARP-CG method with and without preconditioning for the 2D frequency-domain elastic wave simulations on the Marmousi 2 model with homogeneous Poisson's ratio with PMLs on the boundaries.

Performing a standard linear regression (dashed blue line), one can obtain the complexities which are summarized in Table 3.6. Using the preconditioner, the complexity of the CARP-CG method is improved. The slopes as well as the constants multiplying these complexities are reduced which is the expected result.

The second conclusion which can be drawn is, however, as the problem becomes increasingly more complex to solve (this is denoted by the increasing Poisson's ratio from 0.25 to 0.45), the complexity in number of iterations of the CARP-CG method increases (see CARP-CG column in Table 3.6) and the performances of the preconditioner are a little degenerated due to the finer discretization which is used for high Poisson's (see PCARP-CG column in Table 3.6).

In the following part, the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.25$ is considered and the effect of the free-surface is investigated.

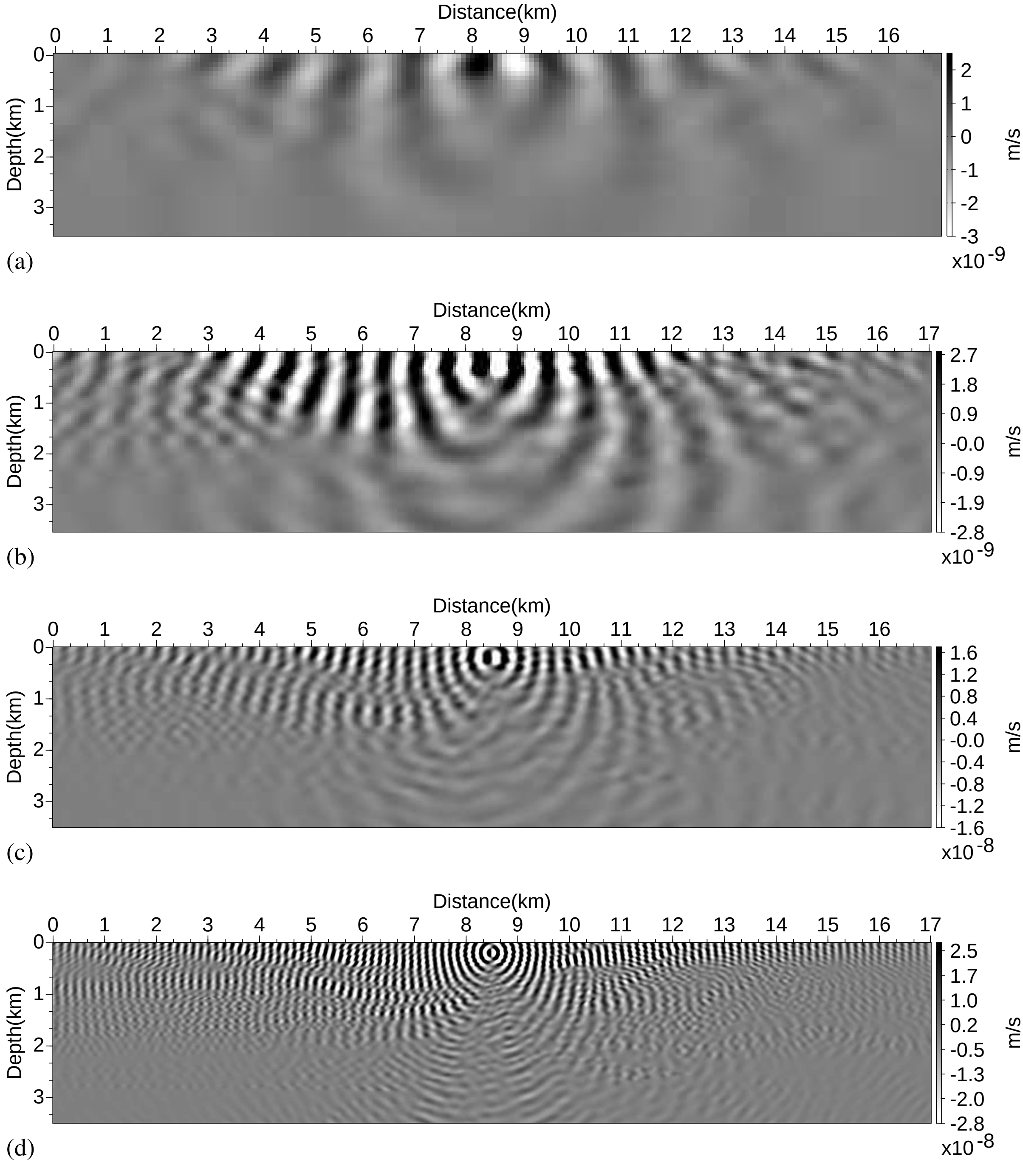


Figure 3.9: 2D frequency-domain horizontal particle velocity v_x wavefields in the constant Poisson's ratio $\xi = 0.25$ Marmousi 2 model with PMLs at 1.25 (a), 2.5 (b), 5 (c) and 10 Hz (d). For high frequencies, note the reflections induced by the change in density and velocities at the water bottom.

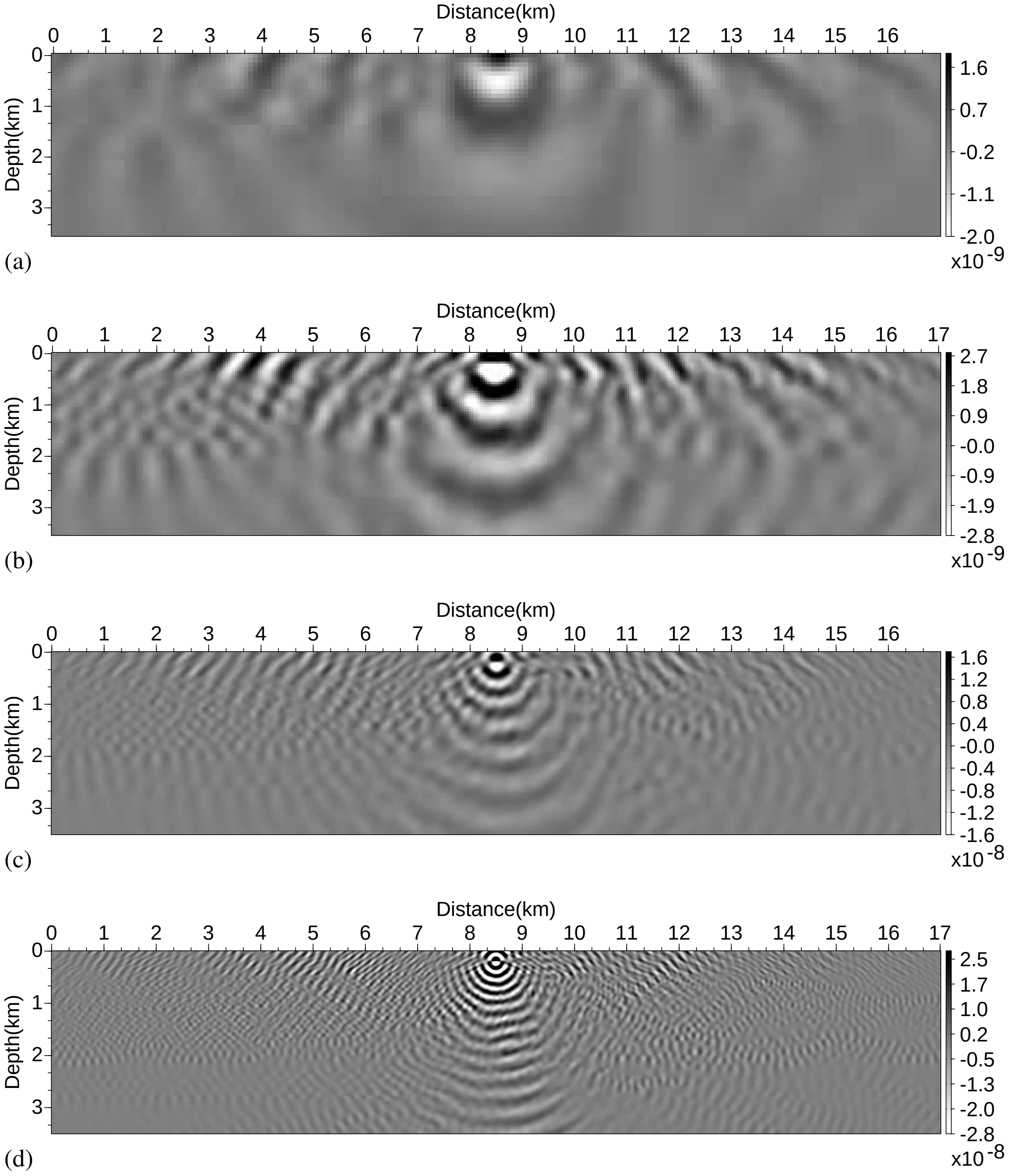


Figure 3.10: 2D frequency-domain vertical particle velocity v_z wavefields in the constant Poisson's ratio $\xi = 0.25$ Marmousi 2 model with PMLs at 1.25 (a), 2.5 (b), 5 (c) and 10 Hz (d). For high frequencies, note the reflections induced by the change in density and velocities at the water bottom.

Free-surface Realistic simulations of the propagation of seismic waves require a free-surface boundary condition (FSBC) at the air/subsurface interface. In this paragraph, the influence of the FSBC on the convergence of the CARP-CG method using the preconditioner is investigated. The previous model settings are kept, i.e. the investigations are restricted to the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.25$. The FSBC is used at the top boundary and PMLs are attached to the other three boundaries of the model. The geometric parameters, the computation time and the number of iterations performed by the CARP-CG method and the preconditioned CARP-CG method are summarized in Table 3.7. Representative wavefields of the horizontal particle velocity v_x and the vertical particle velocity v_z at the given frequencies are presented at the end of this part in Figures 3.12 and 3.13.

					CARP-CG		PCARP-CG	
f	h (m)	N_z	N_x	N_u	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	88	41	194	28548	5034	22.2	766	8.9
2.5	44	81	388	86456	8766	130.5	1153	46.5
5	22	160	774	293040	14764	812.2	1647	232.4
10	11	319	1548	1076664	23764	1324.2	2653	371.2

Table 3.7: Geometric parameters, total number of unknowns N_u, number of iterations N_{iter} and computing time for the solution of the 2D frequency-domain elastic wave equations using CARP-CG and PCARP-CG with the FSBC on the Marmousi 2 model with homogeneous Poisson's ratio $\xi = 0.25$. The preconditioner is computed upon the Marmousi 2 model with a strong damping using $q_{att} = 0.6$ and according to the $9/0$ sparsity pattern.

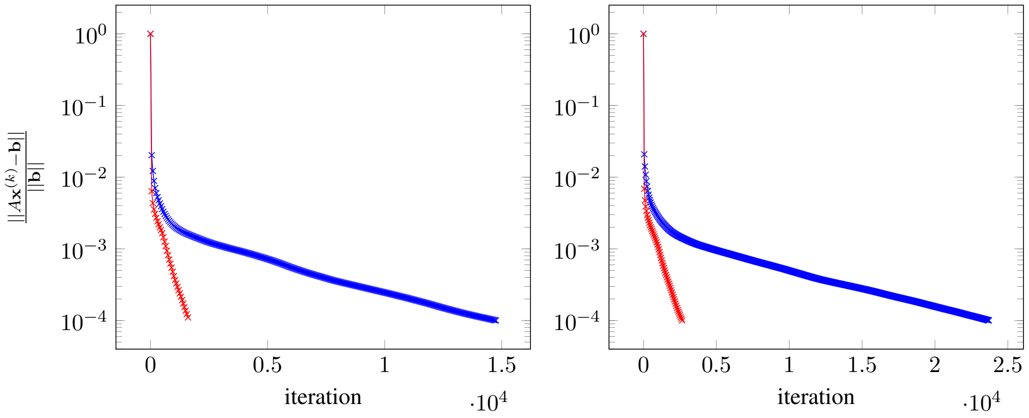
Table 3.7 shows the performances of the CARP-CG method with and without preconditioning when the free-surface boundary condition is used. The FSBC generates the propagation of surface waves which interact with the P and S-waves. Therefore, it strongly affects the convergence of the CARP-CG method. The number of iterations is multiplied by more than a factor 2 compared to the simulations performed with PMLs on all boundaries despite the smaller size linear system (see previous results in Table 3.3). When the preconditioner is applied, the convergence of the CARP-CG method is significantly improved. The number of iterations is reduced by a factor going from 6.57 and up to 8.96 inducing a computation time speed-up by a factor 2.5 and up to 3.5.

In Figure 3.11, the convergence histories of the CARP-CG method clearly show the improvement of the convergence thanks to the preconditioner. The relative residual tends very quickly towards zero, while, without preconditioning, the CARP-CG method converges slowly.

The complexity analysis of the CARP-CG method (with and without preconditioning) is presented in Figure 3.11d. The following result is obtained

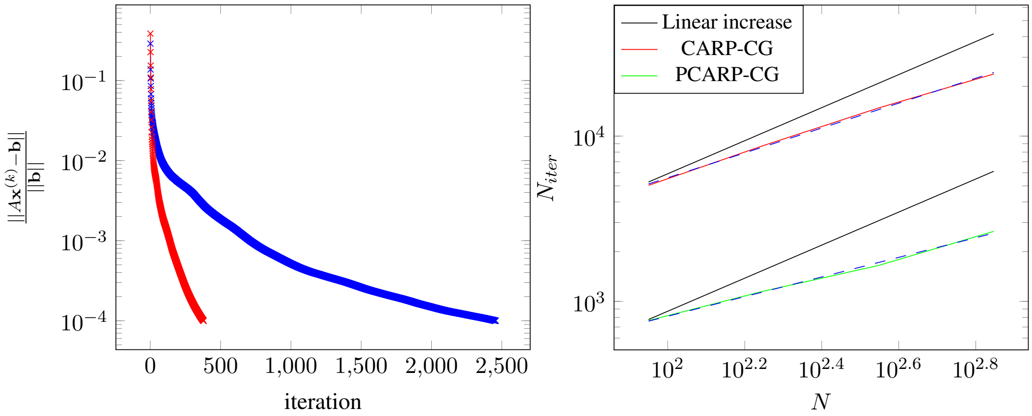
$$N_{iter} = \begin{cases} 172 \times N^{0.75}, & \text{for CARP-CG,} \\ 51 \times N^{0.6}, & \text{for PCARP-CG.} \end{cases} \tag{3.3}$$

In comparison with the simulation performed with PMLs, for which the complexities of the CARP-CG method are recalled in Table 3.8, one may say that the free-surface boundary condition is responsible for degenerating the constant multiplying the complexity of the CARP-CG method. The exponent remains roughly the same. However, thanks to the preconditioner, both constant and exponent are reduced.



(a) 5 Hz

(b) 10 Hz



(c) 2.5 Hz

(d) Complexity analysis

Figure 3.11: Convergence histories of the CARP-CG method with (red curve) and without preconditioning (blue curve) for the solution of the 2D frequency-domain elastic wave simulations with the FSBC on the Marmousi 2 model with the constant Poisson's ratio $\xi = 0.25$ (a), (b) and (c). Complexity analysis of the CARP-CG method with and without preconditioning. The number of iterations N_{iter} is plotted as a function of the average model dimension N on a log-log scale. The straight dashed line indicates the results obtained from a line regression.

	N_{iter}	
$\xi = 0.25$	CARP-CG	PCARP-CG
PML	$49 \times N^{0.76}$	$15 \times N^{0.67}$
FSBC	$172 \times N^{0.75}$	$51 \times N^{0.60}$

Table 3.8: Comparison of the complexities of the CARP-CG method with and without preconditioning for solution of the 2D frequency-domain elastic wave equation on the Marmousi 2 model with homogeneous Poisson's ratio with PMLs and the FSBC.

In the following part, the original Marmousi 2 model with its space-dependent Poisson's ratio is considered. The numerical experiments are performed using both PMLs and the FSBC.

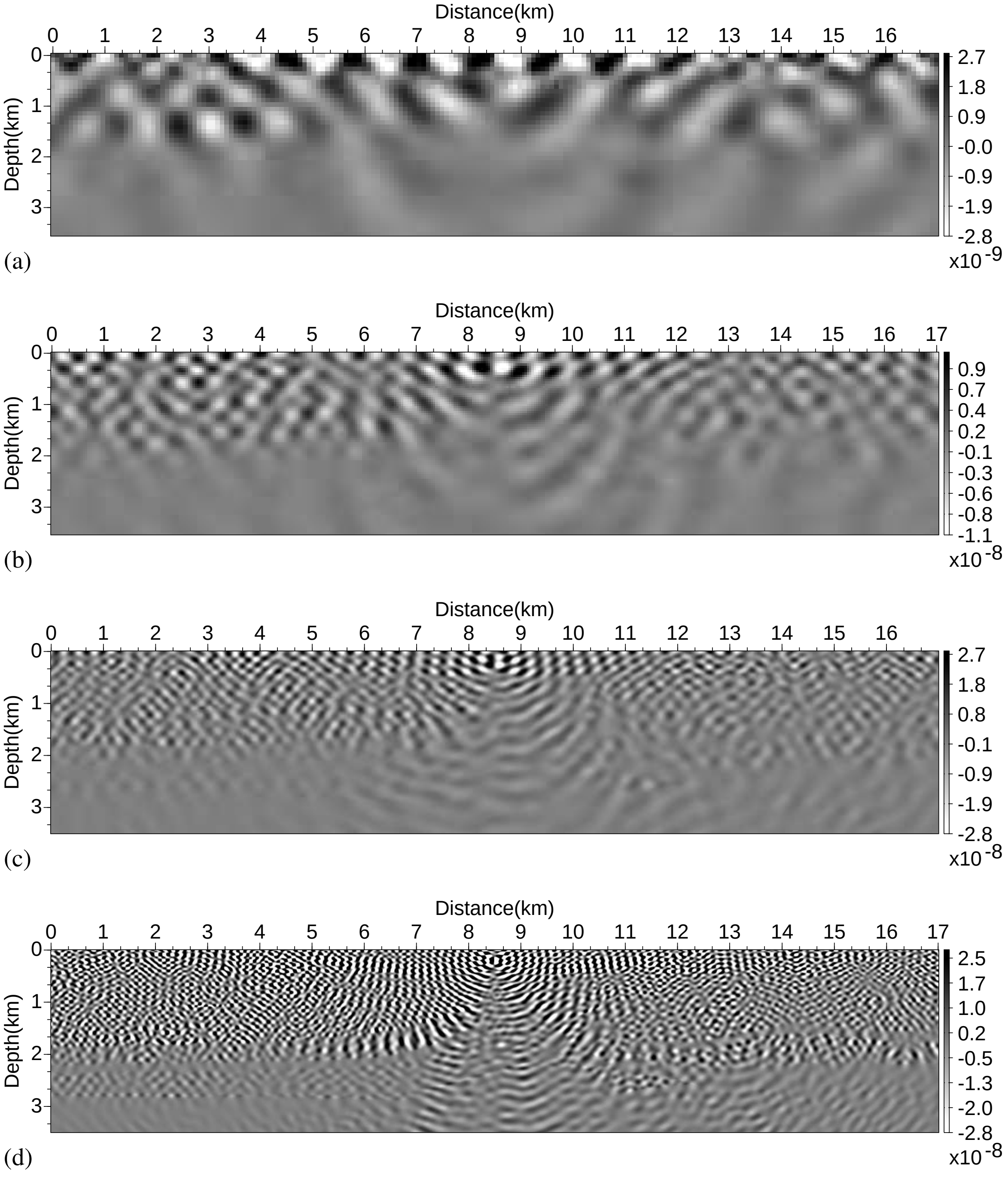


Figure 3.12: 2D frequency-domain horizontal particle velocity v_x wavefields in the constant Poisson's ratio $\xi = 0.25$ Marmousi 2 model with the FSBC at 1.25 (a), 2.5 (b), 5 (c) and 10 Hz (d). The FSBC introduces multiple reflections at the top boundary.

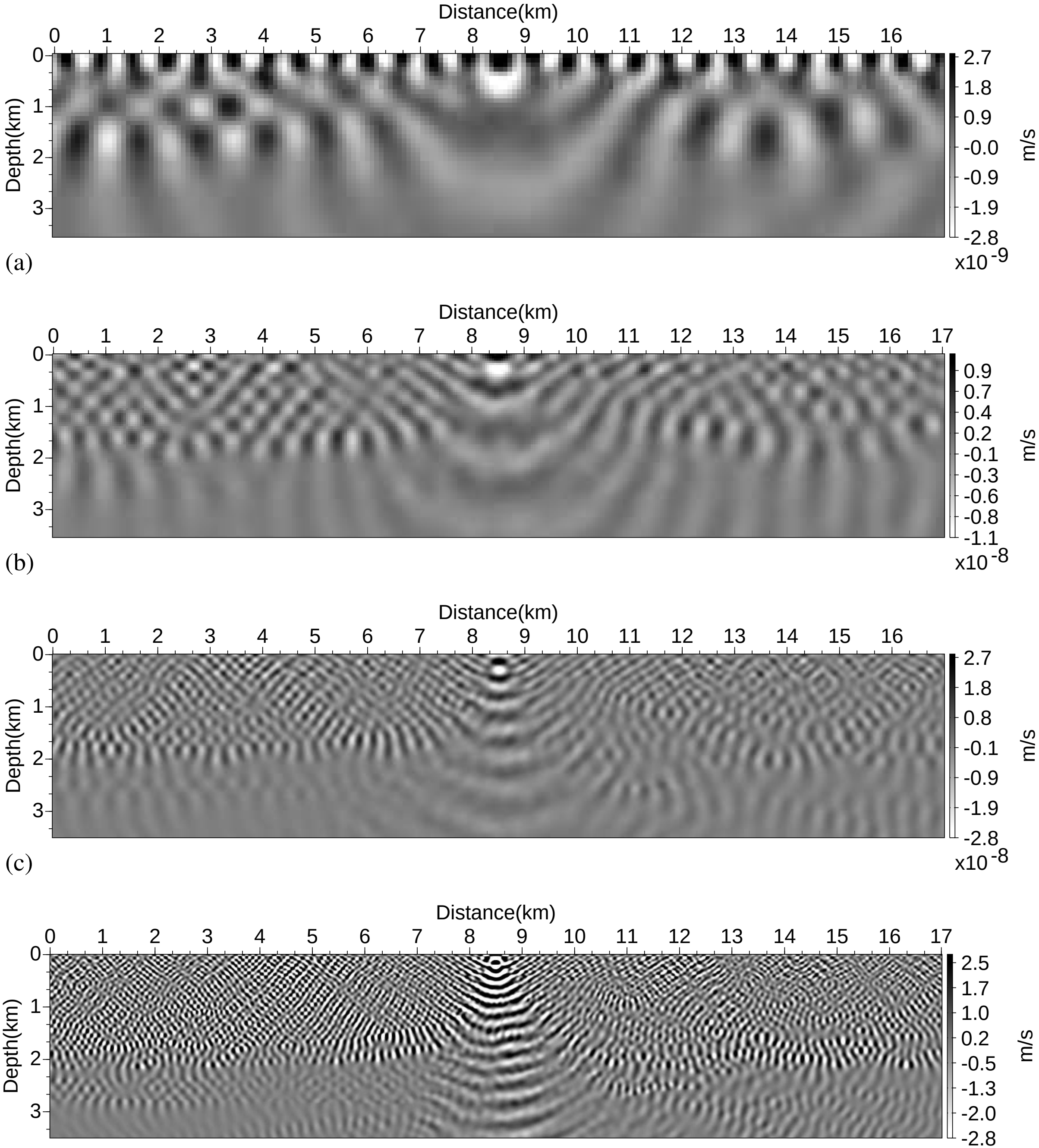


Figure 3.13: 2D frequency-domain vertical particle velocity v_z wavefields in the constant Poisson's ratio $\xi = 0.25$ Marmousi 2 model with the FSBC at 1.25 (a), 2.5 (b), 5 (c) and 10 Hz (d). The FSBC introduces multiple reflections at the top boundary.

3.2.2.3 Marmousi 2 case study with the space-dependent Poisson's ratio

In this part the original Marmousi 2 model is considered with its space-dependent Poisson's ratio. This minimum wavelength is obtained for $V_{S,min} = 271.53\ m/s$ corresponding to the Poisson's ratio $\xi = 0.46$. Two configurations are investigated: the first one considers the frequency-domain elastic wave simulations with the PMLs on all boundaries. The second configuration uses the free-surface (FSBC) at the top boundary and PMLs on the remaining boundaries. Table 3.9 summarizes the discretization parameters of these two problems for several frequencies.

f (Hz)	h (m)	$N_z \times N_x$	N_u(FSBC)	N_u(PML)
1.25	40	89×426	$101,588$	$120,228$
2.5	20	176×851	$349,272$	$384,912$
5	10	351×1701	$1,291,822$	$1,361,462$
10	5	701×3401	$4,961,922$	$5,099,562$
15	3.35	1046×5075	$10,905,180$	$11,109,780$

Table 3.9: Geometry parameters, total number of unknowns N_u at different frequencies with PMLs and the FSBC boundary conditions for the Marmousi 2 model with the space-dependent Poisson's ratio.

Several experiments have shown that the preconditioner built on the strongly damped medium with a space-dependent Poisson's ratio does not give a substantial improvement in the convergence of CARP-CG method.

PML						
f	CARP-CG	Time (s)	PCARP-CG ξ_p: space-dependent	Time (s)	PCARP-CG $\xi_p = 0.15$	Time (s)
1.25	**3647**	74.7	**2846**	154.2	**1071**	56.1
2.5	**7349**	76.8	**3031**	81.2	**1541**	44.7
5	**12571**	155.4	**4254**	116.3	**2487**	68.8
FSBC						
1.25	**9997**	163.9	**4970**	212.9	**2900**	124.2
2.5	**22054**	400.3	**10950**	516.4	**5338**	250.6
5	**43975**	514.2	**16359**	421.2	**10316**	265.6

Table 3.10: Number of iterations and computation time for the solution of 2D frequency-domain elastic equations on a portion of the space-dependent Marmousi 2 model with PMLs and FSBC using the CARP-CG method. The preconditioner is computed on the Marmousi 2 model with the space-dependent Poisson's ratio and a constant Poisson's ratio $\xi_p = 0.15$ both with strong damping ($q_{att} = 0.6$) according to the $9/0$ sparsity pattern.

In Table 3.10, several experiments are performed on a portion of the Marmousi 2 model where the water layer is removed. The simulations are carried out with PMLs and the FSBC boundary conditions. The results show that when the preconditioner is computed on a medium with homogeneous low Poisson's ratio $\xi_p = 0.15$ with strong damping, the performances of the preconditioner are far better than the configuration where the preconditioner is computed on the original medium (with space-dependent Poisson's ratio) with a strong damping. Several experiments have shown that the Poisson's ratio $\xi_p = 0.15$ for the preconditioner computation is optimal. This is partially due to the fact that the minimum Poisson's ratio of the Marmousi 2 model is approximately equal to $\xi_{min} = 0.16$. Moreover, the constant Poisson's ratio $\xi_p = 0.15$ provides a shear velocity model with higher velocity values. Therefore, the amount of damping introduced in the preconditioner is accordingly higher. This makes the preconditioner more diagonal dominant. This value of the Poisson's ratio $\xi_p = 0.15$ will be used to compute the preconditioner in the following experiments.

PML				CARP-CG		PCARP-CG	
f	h (m)	N_u	N_{proc}	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	40	$120,228$	2	**5918**	125.7	**2969**	167.1
2.5	20	$384,912$	4	**9174**	341.5	**3474**	331.9
5	10	$1,361,462$	32	**16267**	376.9	**4650**	266.5
10	5	$5,099,562$	64	**27284**	1295.9	**5274**	598.1

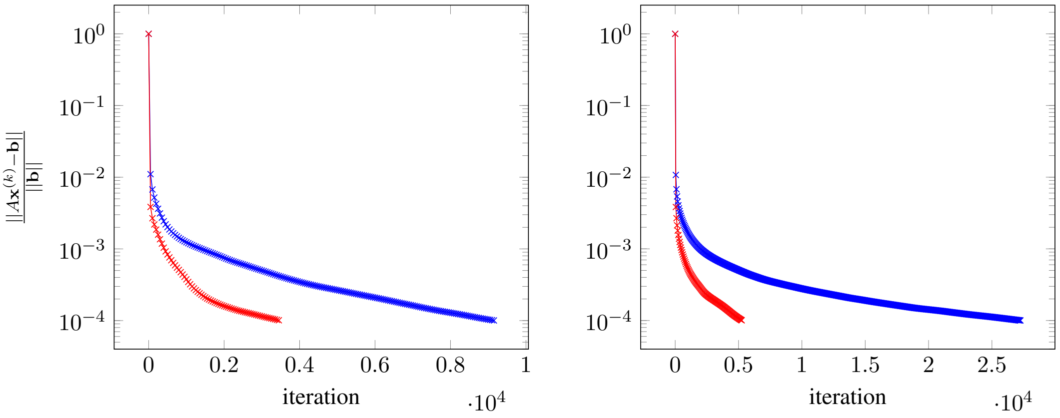
FSBC				CARP-CG		PCARP-CG	
f	h (m)	N_u	N_{proc}	N_{iter}	Time (s)	N_{iter}	Time (s)
1.25	40	$101,588$	2	**10452**	189.0	**3186**	151.9
2.5	20	$349,272$	4	**21278**	706.3	**6417**	549.5
5	10	$1,297,822$	32	**35471**	833.7	**7217**	389.1
10	5	$4,961,922$	64	**43144**	2012.8	**9052**	993.9
15	3.35	$10,905,180$	128	**44779**	2779.3	**9529**	1337.1

Table 3.11: Geometric parameters, total number of unknowns N_u, number of iterations N_{iter} and computation time of the frequency-domain elastic wave simulations using the CARP-CG method with PMLs (top) and FSBC (bottom) on the Marmousi 2 model with the space-dependent Poisson's ratio. N_{proc} denotes the number of cores used for the parallel solver CARP-CG. The preconditioner is computed on the Marmousi 2 model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$.

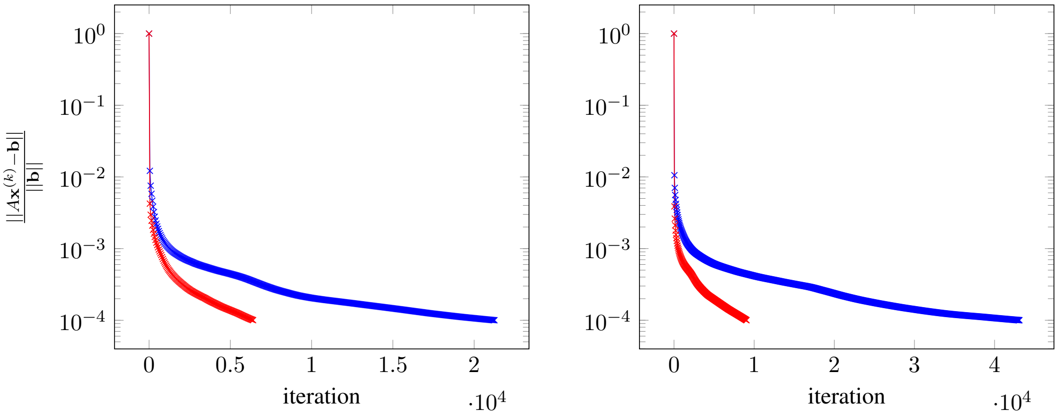
In Table 3.11, the number of iterations and the computation time of the CARP-CG method for the solution of the frequency-domain elastic wave equations on the Marmousi 2 model with the space-depend Poisson's ratio with the PML and FSBC configurations are presented. The preconditioner is computed on the Marmousi 2 model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$.

In Figure 3.14, the convergence histories of the CARP-CG method clearly show the improvement of the convergence thanks to the preconditioner. Overall, the relative residual tends quickly towards zero, while, without preconditioning, the CARP-CG method converges slowly. However, in comparison with the constant Poisson's ratio case, one can see the difficulty introduced by the variable Poisson's ratio. The convergence histories of the preconditioned CARP-CG stagnates near the stopping criterion.

In Figure 3.15, the complexity analysis of the CARP-CG method for the two experiments with PML (see Figure 3.15a) and the FSBC (see Figure 3.15b) is presented. The number of iterations N_{iter}



(a) PML: 2.5 Hz (b) PML: 10 Hz



(c) FSBC: 2.5 Hz (d) FSBC: 10 Hz

Figure 3.14: Convergence histories of the CARP-CG method for the solution of the 2D frequency-domain elastic wave simulations using the CARP-CG method with PMLs (a), (b) and the FSBC (c), (d) on the Marmousi 2 model with the space-dependent Poisson's ratio.

is plotted as a function of the geometric mean size in one dimension

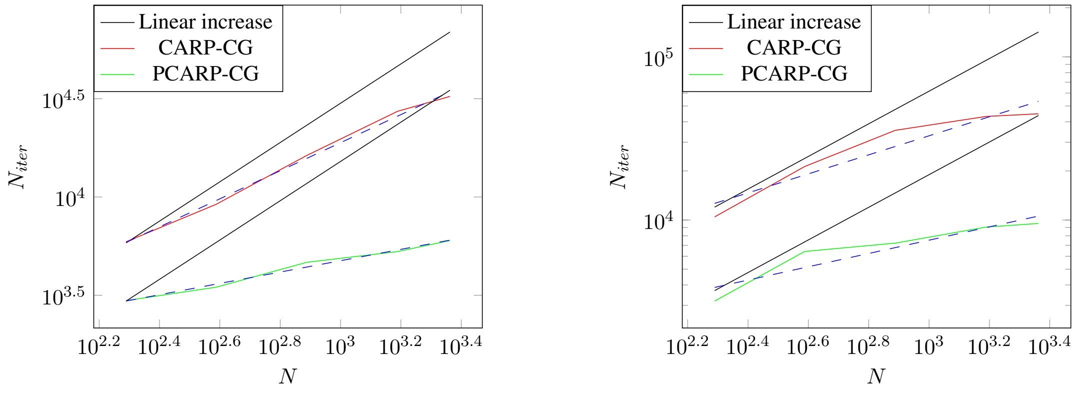
$$N = \sqrt{N_x \times N_z},$$

on a log-log scale. The solid black line represents the linear increase that is

$$N_{iter} = \mathcal{O}(N),$$

and the dashed lines indicate the standard linear regression. The complexities of the CARP-CG method for these two experiments are presented in Table 3.12. The comparison of the performances of CARP-CG on the simulations using PMLs and the FSBC shows that the slopes of the complexity are roughly the same ($\mathcal{O}(N^{0.74})$ and $\mathcal{O}(N^{0.77})$). However, some irregularities arises when the FSBC is used which indicate the influence of the surface waves and the reflections at the air/subsurface interface. In this case, the convergence of the CARP-CG method is worsened and this is noted by the constant multiplying the complexities. This constant increases from 121 in the PML configuration to 216 in the FSBC case.

The use of the preconditioner improves the complexity of the CARP-CG method in the sense that, for experiments using PMLs, the complexity of the CARP-CG method reduces from 121 $N^{0.74}$ to



(a) PML (b) FSBC

Figure 3.15: Complexity analysis of the CARP-CG method with and without preconditioning for the 2D frequency-domain elastic wave simulations on the Marmousi 2 model with the space-dependent Poisson's ratio: PML (a) and FSBC (b). The number of iterations N_{iter} is plotted as a function of the average model dimension N on a log-log scale. The straight dashed line indicates the results obtained from line regression. The preconditioner is computed on the Marmousi 2 model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$.

	N_{iter}	
	CARP-CG	PCARP-CG
PML	$121 \times N^{0.74}$	$690 \times N^{0.28}$
FSBC	$216 \times N^{0.77}$	$472 \times N^{0.41}$

Table 3.12: Complexities of the CARP-CG method with and without preconditioning for the 2D frequency-domain elastic wave simulations on the space-dependent Poisson's ratio Marmousi 2 model.

690 $N^{0.28}$. The decrease in the complexity is compensated with an increase in the constant multiplying the complexity, but the preconditioned CARP-CG method allows to retrieve substantial gains in the number of iterations and the computation time. The same conclusion is drawn from the results with the FSBC. Using the preconditioner, the computational complexity in number of iterations of the CARP-CG method is reduced from 216 $N^{0.77}$ to 472 $N^{0.41}$. These results denote a substantial improvement compared to $\mathcal{O}(N)$ performed by the combination of the time-domain modeling and the Fourier Transform approach and the previous complexities of the CARP-CG method derived from constant Poisson's ratio experiments with both PMLs and the FSBC.

Remark 3.1. *The computation are performed in double precision code. For large size linear systems and particularly when the number of iterations performed by the CARP-CG method is too large, the CARP-CG method diverges when a single-precision implementation is used. This is the case in the 2D approximation at 10 Hz and 20 Hz and when the free-surface boundary condition is used on the elastic Marmousi 2 case study. The large number of iterations (more than 30 000 at 10 Hz and more than 50 000 iterations at 20 Hz) increases the round-off errors. In addition, performing the sweeps on large blocks worsens the problem. The double-precision implementation is used to remedy to this issue. This is discussed as well by Li et al. (2015).*

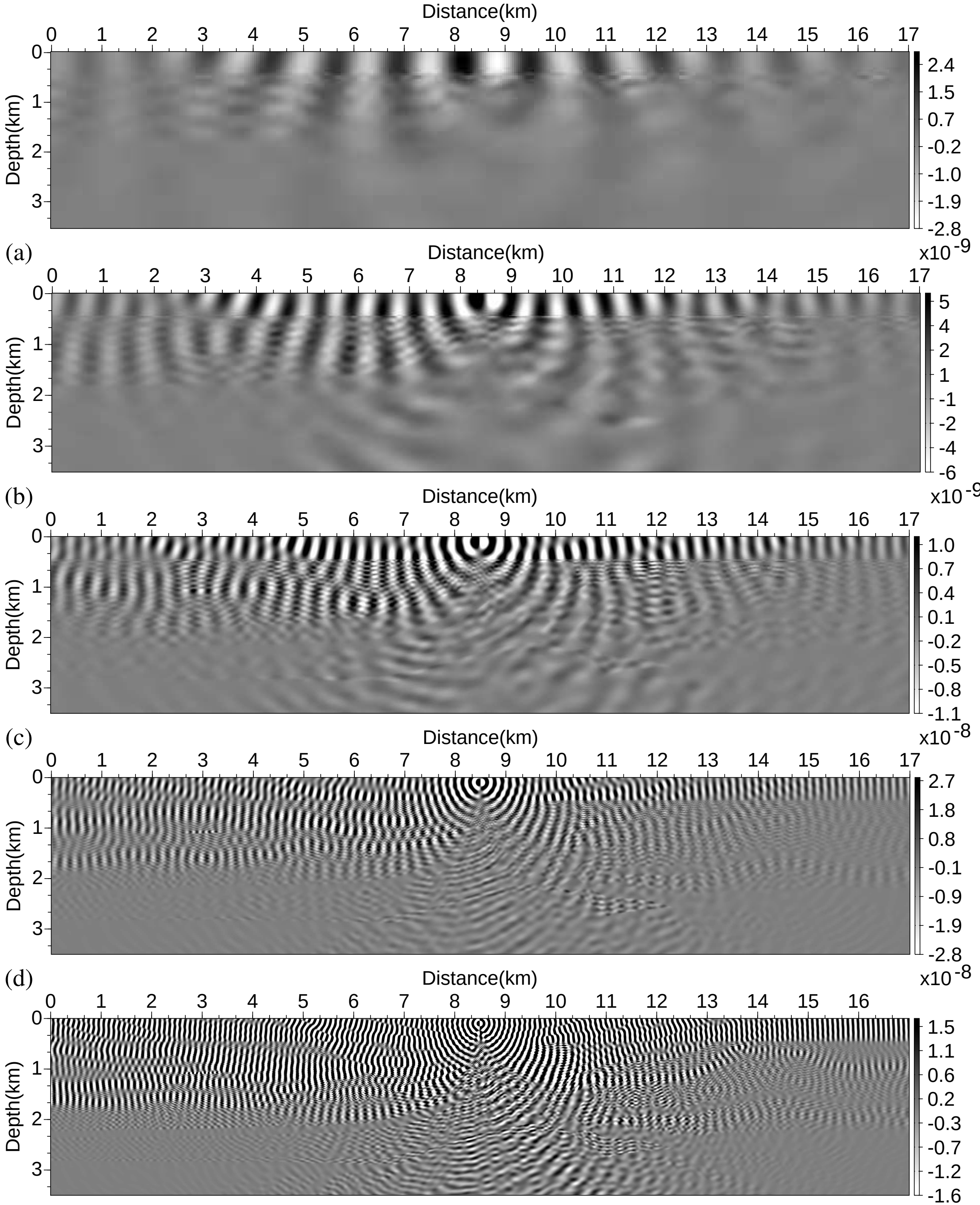


Figure 3.16: 2D frequency-domain horizontal particle velocity v_x wavefields in the the space-dependent Poisson's ratio Marmousi 2 model with PMLs at 1.25 (a), 2.5 (b), 5 (c), 10 Hz (d) and 15 Hz (e).

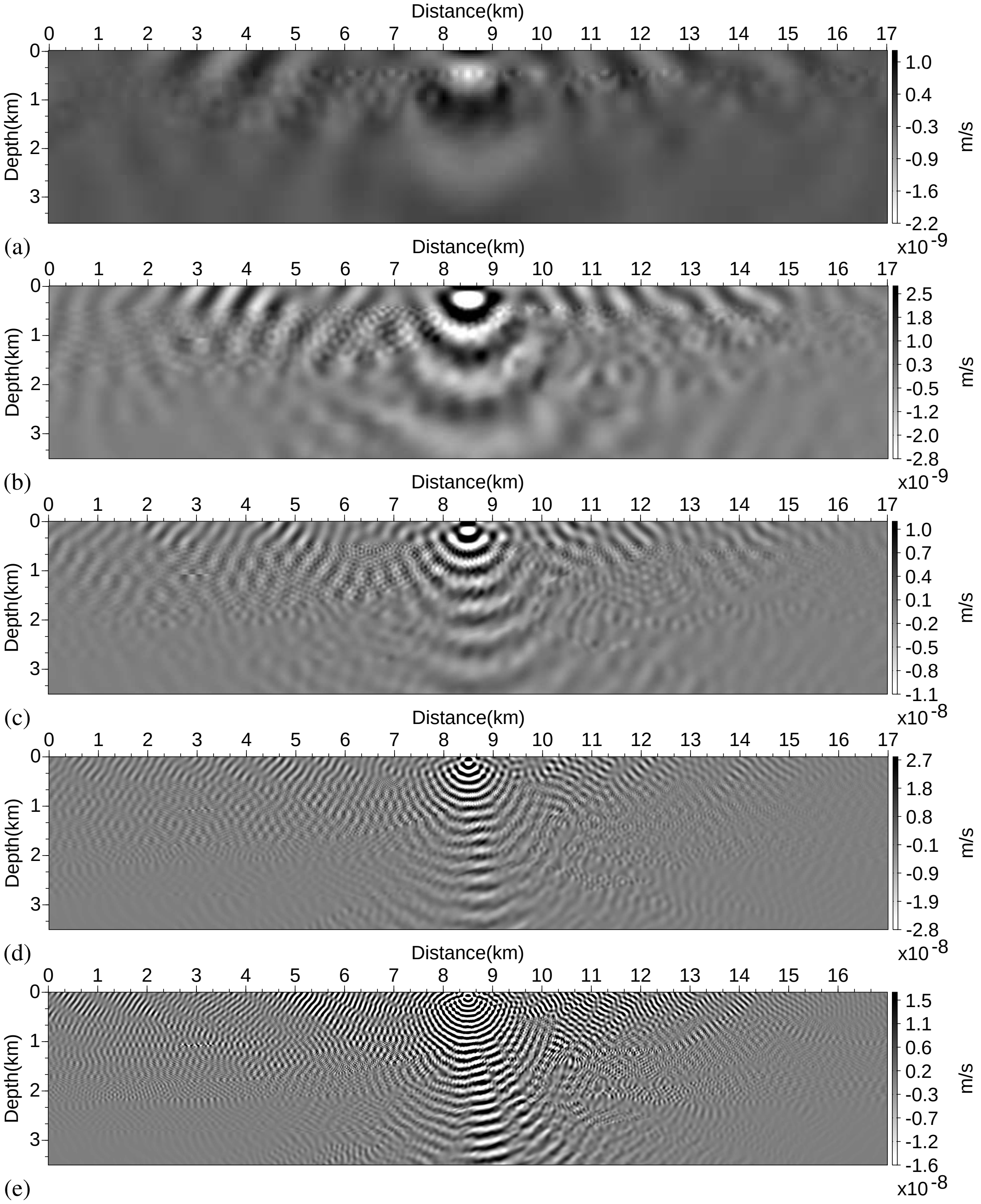


Figure 3.17: 2D frequency-domain vertical particle velocity v_z wavefields in the the space-dependent Poisson's ratio Marmousi 2 model with PMLs at 1.25 (a), 2.5 (b), 5 (c), 10 Hz (d) and 15 Hz (e).

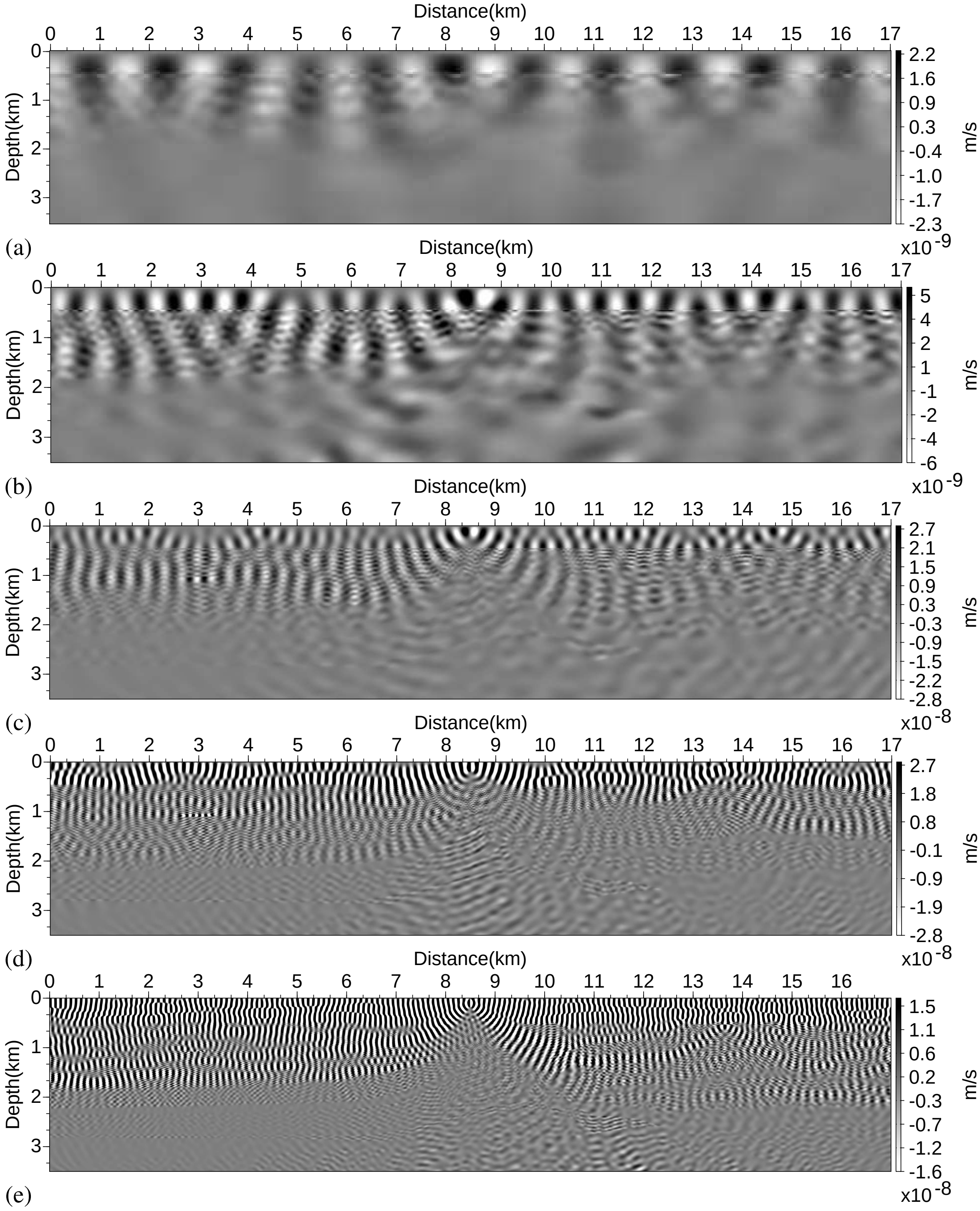


Figure 3.18: 2D frequency-domain vertical particle velocity v_x wavefields in the the space-dependent Poisson's ratio Marmousi 2 model with the FSBC at 1.25 (a), 2.5 (b), 5 (c) , 10 (d) and 15 Hz (e).

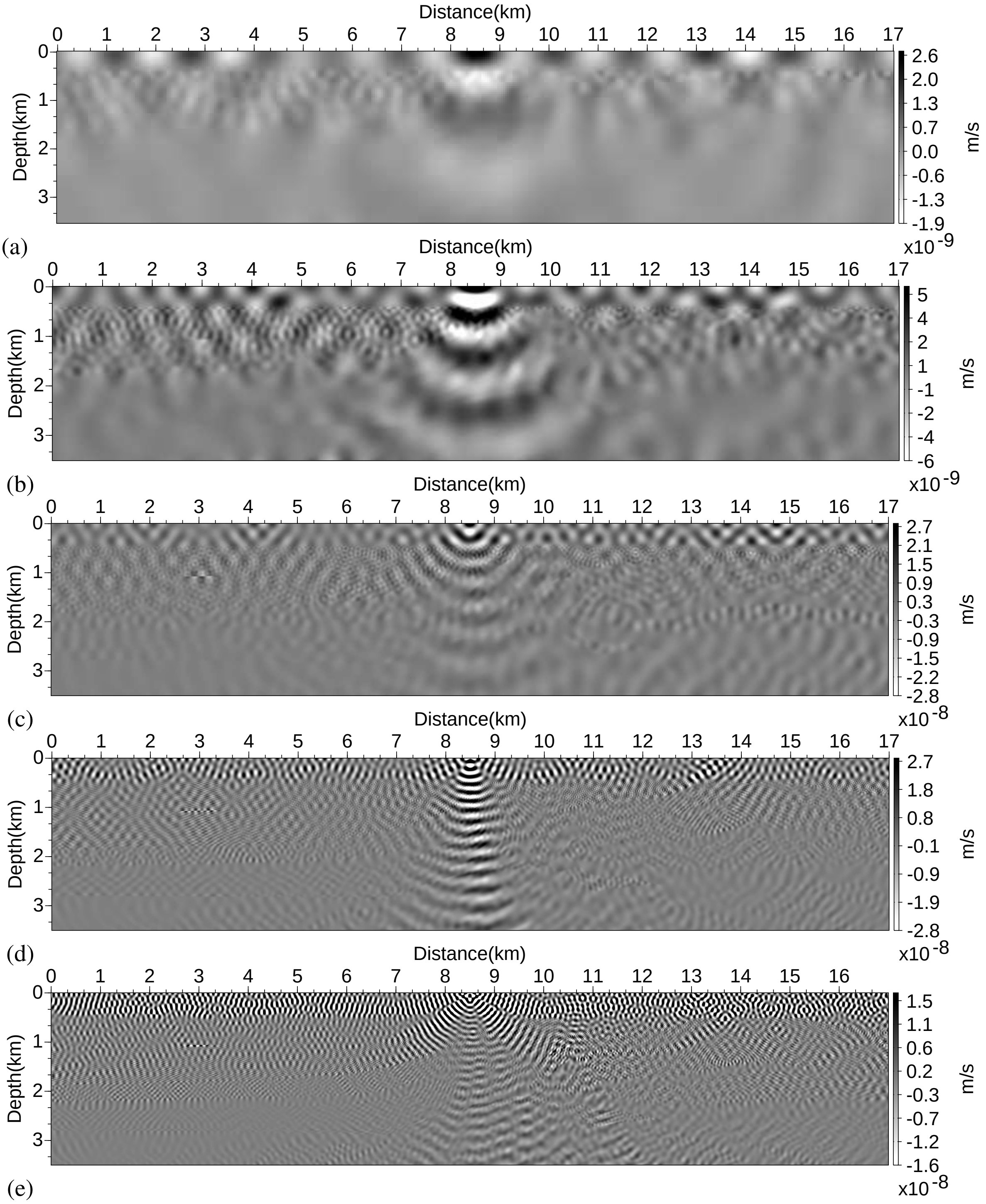
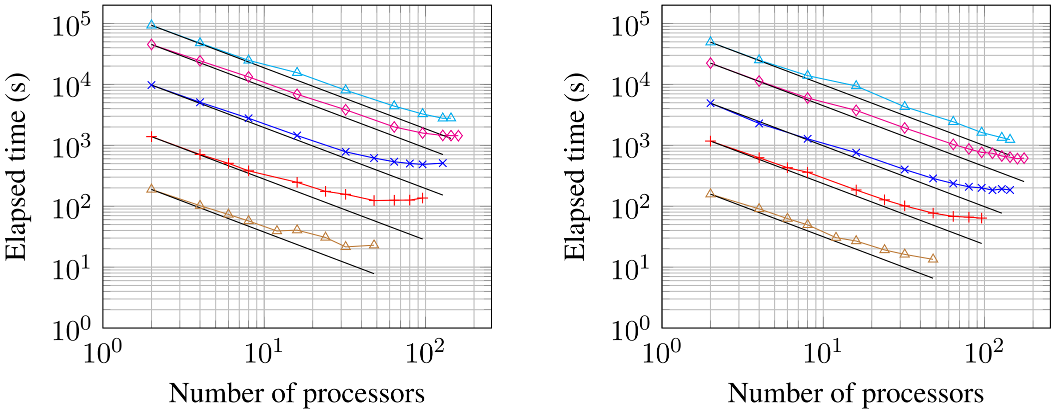


Figure 3.19: 2D frequency-domain vertical particle velocity v_z wavefields in the the space-dependent Poisson's ratio Marmousi 2 model with the FSBC at 1.25 (a), 2.5 (b), 5 (c) , 10 (d) and 15 Hz (e).

3.2.2.4 Scalability tests

In this part, the scaling properties of the CARP-CG method are investigated for each frequency.



(a) CARP-CG

(b) PCARP-CG

Figure 3.20: Scaling properties of the CARP-CG method (a) and the preconditioned CARP-CG method (b) using different numbers of cores for the frequency-domain elastic wave simulations with the FSBC on the Marmousi 2 model with the space-dependent Poisson's ratio. The frequencies are 1.25 —△—, 2.5 —+—, 5 —×—, 10 —◇— and 15 Hz —△— respectively. The black lines represent the ideal scalability for parallel computing. The preconditioner is computed on the Marmousi 2 model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$.

In Figure 3.20, the experiments are formed on the Marmousi 2 model with the FSBC using the space-dependent Poisson's ratio. The frequency-domain elastic wave simulations are performed for the frequencies 1.25, 2.5, 5, 10 and 15 Hz and using an increasing number of processors. The computation times are plotted as a function of the number of processors. The black solid lines represent the ideal scalability for parallel computing and the colored marked curves are the results for the CARP-CG method without preconditioning (Figure 3.20a) and with preconditioning (Figure 3.20b). A satisfactory scaling properties of CARP-CG is observed up to 16 cores for low frequencies and 64 cores for higher frequencies.

Let $T(n,1)$ be the run-time of the CARP-CG method using 1 core (namely, it is the CGMN method) and let $T(n,p)$ be the run-time of the parallel algorithm CARP-CG executed using p cores. Let n denote the size of the linear system. The speedup is then defined as

$$S(p) = \frac{T(n,1)}{T(n,p)},$$

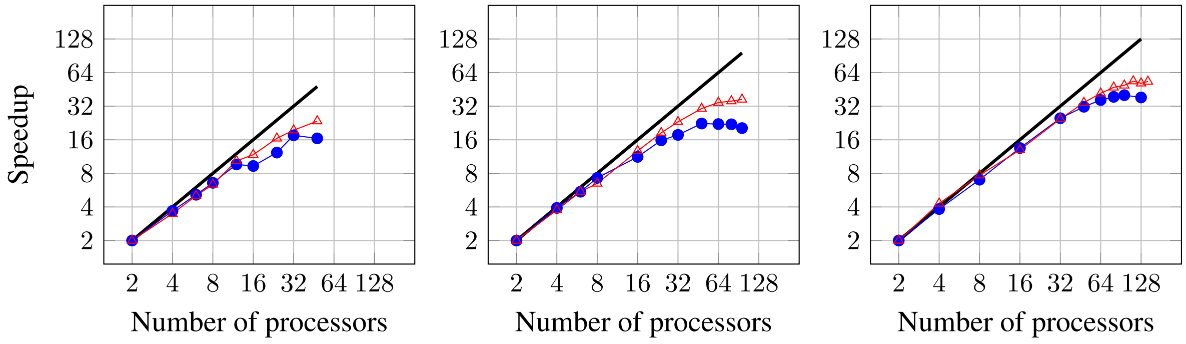
i.e., the ratio of the sequential execution time to the parallel execution time. Ideally, one would hope to obtain

$$S(p) = p,$$

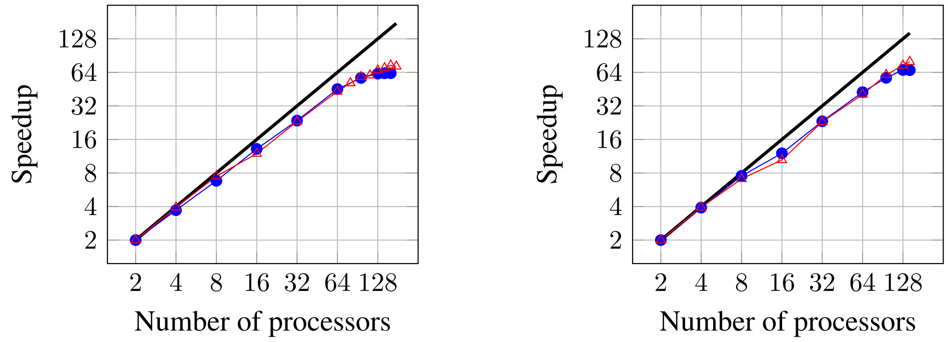
which is called perfect or ideal speedup. The parallel efficiency $E(p)$ is defined as well as

$$E(p) = \frac{S(p)}{p}.$$

Ideally, the efficiency is equal to 1 for any number of cores p. I shall use the speedup and the efficiency to analyze the scaling properties of the CARP-CG and the PCARP-CG algorithms using the run-time complexity. I shall consider as well the sequential run-time as the one performed by CARP-CG using 2 cores, for convenience, since the sequential CGMN leads to extremely high run-times.



(a) 1.25 Hz (b) 2.5 Hz (c) 5 Hz

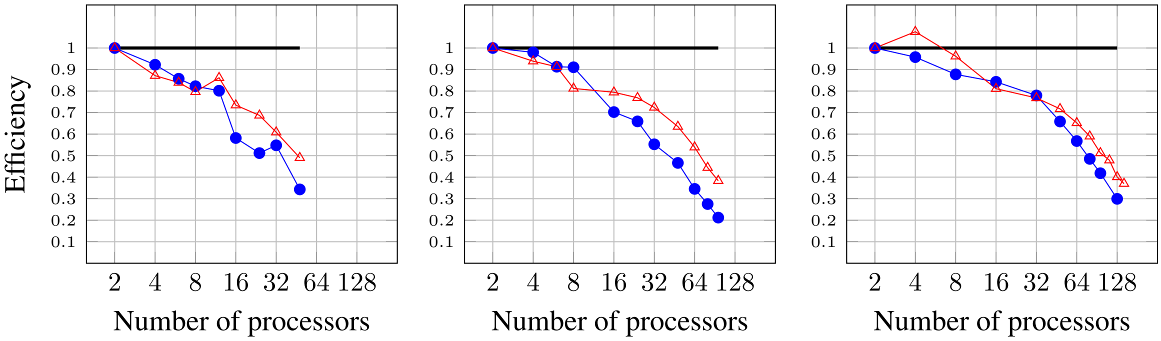


(d) 10 Hz (e) 15 Hz

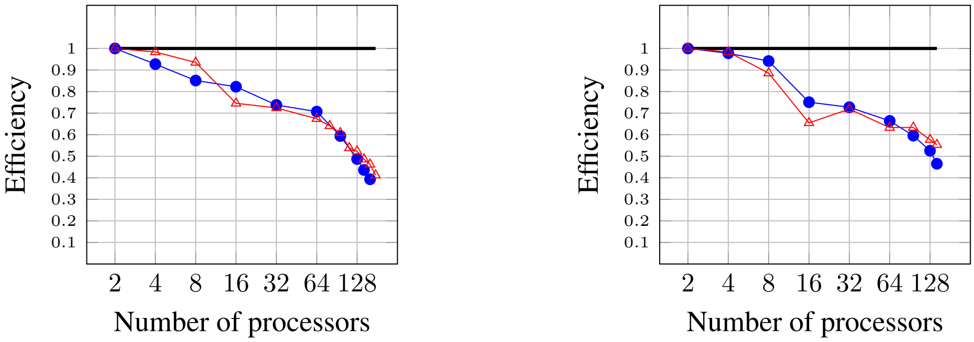
Figure 3.21: Speedup analysis for the 2D frequency-domain elastic wave simulations with the FSBC and the space-dependent Poisson's ratio Marmousi 2 model at 1.25 Hz (a), 2.5 Hz (b), 5 Hz (c), 10 Hz (d) and 15 Hz (e). The blue curves —●— are used to plot the results of the CARP-CG method without preconditioning and the red curves —△— are used for the CARP-CG method with preconditioning. The black solid lines represent the ideal speedup.

In Figures 3.21 and 3.22, the speedup and the efficiency plots are presented for the CARP-CG method when applied to the Marmousi 2 model with space-dependent Poisson's ratio and the FSBC. On each plot, both curves for the CARP-CG method and the preconditioned CARP-CG method are compared. The speedup and the efficiency are improved for both methods (CARP-CG and PCARP-CG) as the frequency increases. Note the irregularity in the efficiency when the number of cores goes from 8 to 16. First, the nodes on which the experiments are performed are composed of 2 CPUs. Each of them has 8 cores. Communications are performed between CPUS. Therefore when running the CARP-CG and PCARP-CG program on less than 16 cores, the node uses one CPU completely and the remaining cores (<8) are used on the second CPU, therefore the total amount of communication per CPU is not perfectly balanced. Thus, efficiency plots show a slight drop when going from 8 to 16 cores. When using more than 16 cores, the efficiency curves are more regular. Overall, the CARP-CG method presents satisfying speedup and efficiency properties.

The same comments can be made of the PCARP-CG method. Even more, the latter presents better speedup and efficiency properties compared to the CARP-CG method. The preconditioned matrix is denser than the unpreconditioned one. Therefore, a good speed-up is observed as the computation time is much larger than the communication time in comparison with the non-preconditioned experiments.



(a) 1.25 Hz (b) 2.5 Hz (c) 5 Hz



(d) 10 Hz (e) 15 Hz

Figure 3.22: Efficiency analysis for the 2D frequency-domain elastic wave simulations with the FSBC and the space-dependent Poisson's ratio Marmousi 2 model at 1.25 Hz (a), 2.5 Hz (b), 5 Hz (c), 10 Hz (d) and 15 Hz (e). The blue curves —●— are used to plot the results of the CARP-CG method without preconditioning and the red curves —△— are used for the CARP-CG method with preconditioning. The black solid lines represent the ideal efficiency.

3.3 Conclusion

In this chapter, I have presented two realistic 2D applications for the frequency-domain wave simulations using the iterative methods CGMN and CARP-CG. A sensitivity analysis is performed on different parameters, namely the quality factor which controls the damping in the medium, the sparsity pattern of the preconditioner, the Poisson's ratio on which the shear velocity model is derived for the computation of the preconditioner and the scaling properties of the CARP-CG method.

In the first part, the frequency-domain wave simulations are performed in the acoustic approximation. The numerical experiments are performed on the BP2004 model. The CGMN method is used to the solve the resulting linear systems at various frequencies (up to 40 Hz). The number of iterations performed by the CGMN method has shown a sub-linear computational complexity with respect to the average number of grid points per dimension N which is an expected result from previous studies (Li et al., 2015). The use of the preconditioner with the CGMN method allowed to improve the convergence of the CGMN method in terms of number of iterations. A study on the sensitivity of the quality factor q_{att} allowed to determine the optimal value of damping the preconditioner computation. In this acoustic case applied to the BP2004 model, the value $q_{att} = 1$ is noted. A reduction in the number of iterations by a factor up to 6.9 is obtained. As a result, the complexity of the preconditioned CGMN method is decreased and furthermore, the constant multiplying this complexity is reduced as well. Such reduction induced a gain in the computation time by a factor 2.2.

In the second part, the frequency-domain wave simulations are carried out in the elastic approximation. The simulations are performed on the well-known Marmousi 2 model. The CARP-CG method is used as a solver. Using the sparsity pattern $9/0$, the optimal value of damping $q_{att} = 0.6$ gives optimal reduction in the number of iterations for this 2D elastic application. Numerical experiments are performed to investigate the influence of the Poisson's ratio and the boundary condition (PML or FSBC) on the convergence of the CARP-CG method. The original Marmousi 2 P-wave velocity model is used and the S-wave velocity model is derived from the P-wave velocity model using the Poisson relation. The numerical experiments were performed using PMLs on all boundaries with several models derived with constant Poisson's ratios. This study has shown that using the preconditioner, the PCARP-CG is able to retrieve a significant reduction in the number of iterations and the computational time. The complexity of the CARP-CG method is improved with preconditioning whatever the Poisson's ratio value. However, for high Poisson's ratios, the convergence is slightly degenerated. In the third study, the influence of boundary conditions on the convergence of CARP-CG are investigated. The behavior of the CARP-CG method with and without preconditioning is studied on the Marmousi 2 model derived with a constant Poisson's ratio. Simulations were performed using PMLs and the FSBC. The results have shown substantial reductions in both number of iterations and computation time for both configurations (PML and FSBC). The configuration with the FSBC has shown better results with a reduction by a factor almost equal to 9 in the number of iterations and a factor 3.5 speedup in computation time. In the fourth investigation, the original Marmousi 2 model with its space-dependent Poisson's ratio is studied. In this case, a better preconditioner is obtained by using a V_S model derived from a constant low Poisson's ratio for the computation of the preconditioner. Using this strategy, the convergence of the CARP-CG method is improved in both PML and FSBC configurations in terms of number of iterations. The complexities of the CARP-CG method are significantly reduced. Finally, the scaling properties of the CARP-CG method are investigated. As expected, the combination of the preconditioner with the CARP-CG method improves the scaling properties of this method. A better speedup and parallel efficiency are obtained when using the preconditioner. The preconditioned system is denser, therefore the ratio of the computation time and the communication time is smaller in comparison with the results obtained from the original system.

In both acoustic and elastic approximations, the preconditioning strategy which is developed so far seems to be satisfying. Significant reductions in the number of iterations and the complexities of the CARP-CG method are obtained. The quality factor q_{att} and the Poisson's ratio ξ_p used for the computation of the preconditioner are case and frequency dependent. But the latter can be easily determined by investigating the Poisson's ratio of the model which is used. The computation of the preconditioner based on the strongly damped medium derived from the minimum Poisson's ratio seems to be the governing rule. The optimal value for the quality factor q_{att} ranges around 1 and is frequency-dependent. However, the gain which is obtained by tuning the quality factor around the value 1 does not provide tremendous gain in the number of iterations.

In the next chapter, I shall extend this investigation to the 3D elastic approximation. Serious concerns rise as regard the memory request. The latter could represent the main limitations for such developments.

Chapter 4

Application to a 3D realistic case

Contents

In this chapter, the CARP-CG method is investigated for the solution of the 3D frequency-domain elastic wave equations. The pioneer work of (Li et al., 2015) in this framework leads to attempt to use the preconditioner introduced in the previous chapters to improve the convergence of the CARP-CG method when solving this problem.

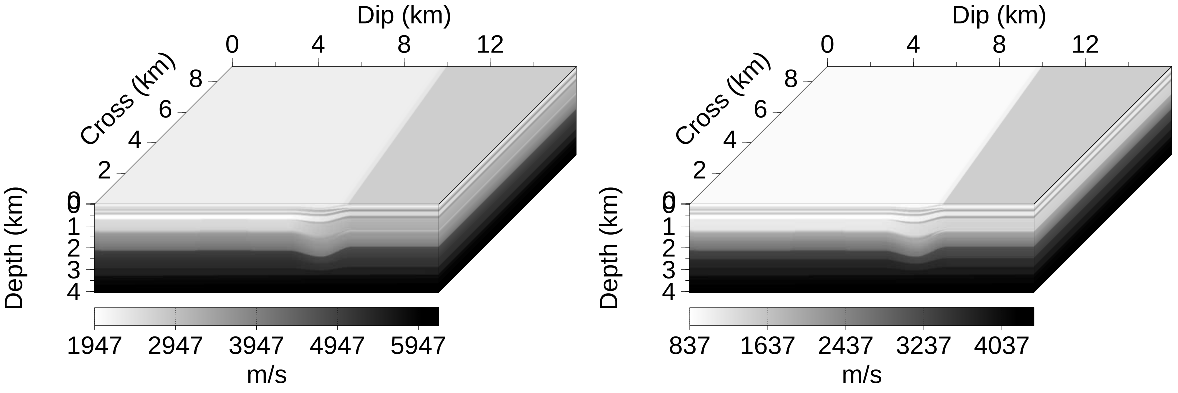
In the first part, the 3D heterogeneous elastic model used by Li et al. (2015) (courtesy of Shell) is presented. The size of this model is 48 × 20 × 4 km which covers the surface of 960 km^2. Besides its size, this model is quite challenging due to the fast-slow-fast layers resembling a waveguide geometries. For a more reasonable application, the size of this model is reduced to 16 × 9 × 4 km. Therefore it covers the same surface as the Valhall model. The numerical experiments are performed on this reduced model.

In the second part, the different sparsity patterns are investigated. For the 3D frequency-domain elastic wave equations, three components for the particle velocity are computed on a staggered grid. Therefore, one may consider several approaches to design the sparsity pattern for the computation of the preconditioner. The numerical tests are performed on a small 3D homogeneous elastic model to determine efficiency (reduction in the number of iterations and memory cost induced by the fill-in of the matrix-matrix product AP) of these sparsity patterns.

In the third part, numerical experiments are performed using the preconditioner on the 3D heterogeneous elastic model with the iterative solver CARP-CG. The 3D elastic wave simulations are performed at the frequencies $f = \{1.25,\ 2.5,\ 5,\ 7.5\}$ Hz. Following the results obtained in Chapters 2 and 3, the preconditioner is computed based on a strongly damped medium and using a shear wave velocity model which is derived from the minimum Poisson's ratio using Poisson's relation. The complexity analysis of the preconditioned iterative solver is carried out. On conclusion, the improvements of the convergence of the CARP-CG method and the main limitations of this preconditioning strategy are discussed.

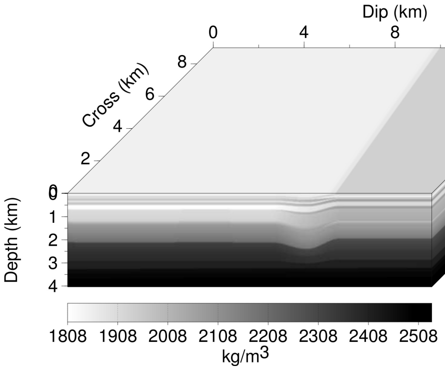
4.1 3D heterogeneous elastic model

The 3D heterogeneous elastic model provided by Shell is used for the investigations on the CARP-CG method and its preconditioner. The size of this model is 16 × 9 × 4 km. The P- and S- wave velocities and the density models are presented in Figure 4.1.



(a) V_P

(b) V_S



(c) Density

Figure 4.1: 3D heterogeneous elastic model of 16 × 9 × 4 km (courtesy of Shell): P-wave velocity (a), S-wave velocity (b) and density (c).

This model presents an interesting feature which is its layered structure. More precisely, a low P-wave velocity layer is surrounded by two fast P-wave velocity layers with $2800/1960/2600$ m/s (fast/low/fast) at the depth $z = 550$ m. Such a structure creates a waveguide geometry for which, if a source is located in this area, the linear system resulting from the discretization of such problem is particularly difficult to solve.

	V_P(m/s)	V_S(m/s)	Poisson's ratio ξ	ρ(kg/m^3)
Min	1947.0	837.2	0.0196	1808.0
Max	6000.0	4200.0	0.4047	2500.0

Table 4.1: Geological properties of the 3D heterogeneous elastic model.

The geological properties of the model are presented in Table 4.1. The geological layered structure of the model with high P-wave velocity denotes mostly shales (Poisson's ratio approximately equal to 0.4 representative of hard material). Low P-wave velocity zones with low Poisson's ratio, i.e. around the value 0.16, are representative of water wet sand or salt zones.

4.2 3D sparsity pattern for the computation of the preconditioner

The preconditioning strategy relies on the computation of sparse approximations of the particle velocity components v_x, v_y and v_z based on a strongly damped medium. Following the approach used in the 2D elastic approximation, I investigate the sparsity patterns which can be adopted in the 3D approximation.

In this section, the numerical experiments are performed on a 3D homogeneous visco-elastic problem. Consider the 3D visco-elastic problem of size 1900 $\times$ 1430 $\times$ 950 m. The P-wave velocity is homogeneous and is defined by $V_P = 1500\ m/s$. The shear wave velocity is derived from the P-wave model using the Poisson's relation (3.2) with the constant Poisson's ratio $\xi = 0.1$ yielding $V_S = 1000\ m/s$. The density is considered constant as well and is equal to $\rho = 1000$ kg/m^3. The problem is discretized using the fourth-order staggered-grid finite-difference method with the optimal coefficients as it is described in Chapter 1 Section 1.5.3. The frequency $f = 5$ Hz is considered and 4.2 grid points per minimum shear wavelength are used. The discrete domain of 41 $\times$ 31 $\times$ 21 grid points is obtained. The domain is surrounded by PMLs using $N_{PML} = 10$ grid points per layer. The damping coefficient in the PMLs is set to $C_{PML} = 500$. A horizontal force source in the x direction is used at the position $x_s = 950$ m, $y_s = 715$ m and $z_s = 475$ m corresponding to the center of the domain. The linear system is solved using the CARP-CG method. The relaxation parameter is set to $\lambda = 1.0$ and the stopping criterion is defined as usual by

$$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2}{\|\mathbf{b}\|_2} < 10^{-4}.$$

In the 3D approximation, three particle velocity wavefields are computed v_x, v_y and v_z. The solution of the homogeneous problem in the medium without attenuation is presented in Figure 4.2. Slices of the particle velocity wavefields in the planes xz, yz and xy at the source position are presented. When the force source is applied in the x direction, the amplitude of the particle velocity component v_x is higher compared to the amplitude of v_y and v_z. The same effect is observed when the force source is applied on the y direction and the z direction for the particle velocities v_y and v_z.

When the simulations are performed in a strongly damped medium (the quality factor $q_{att} = 0.6$ is used), the particle velocity wavefields are strongly attenuated. The amplitude of the wavefield decays rapidly to zero after few discretization points away from the source position. Representative wavefields in the strongly damped medium in the planes xz, yz and xy at the source position are presented in Figure 4.3. The amplitude of the wavefield is mainly dominated by the particle velocity component v_x.

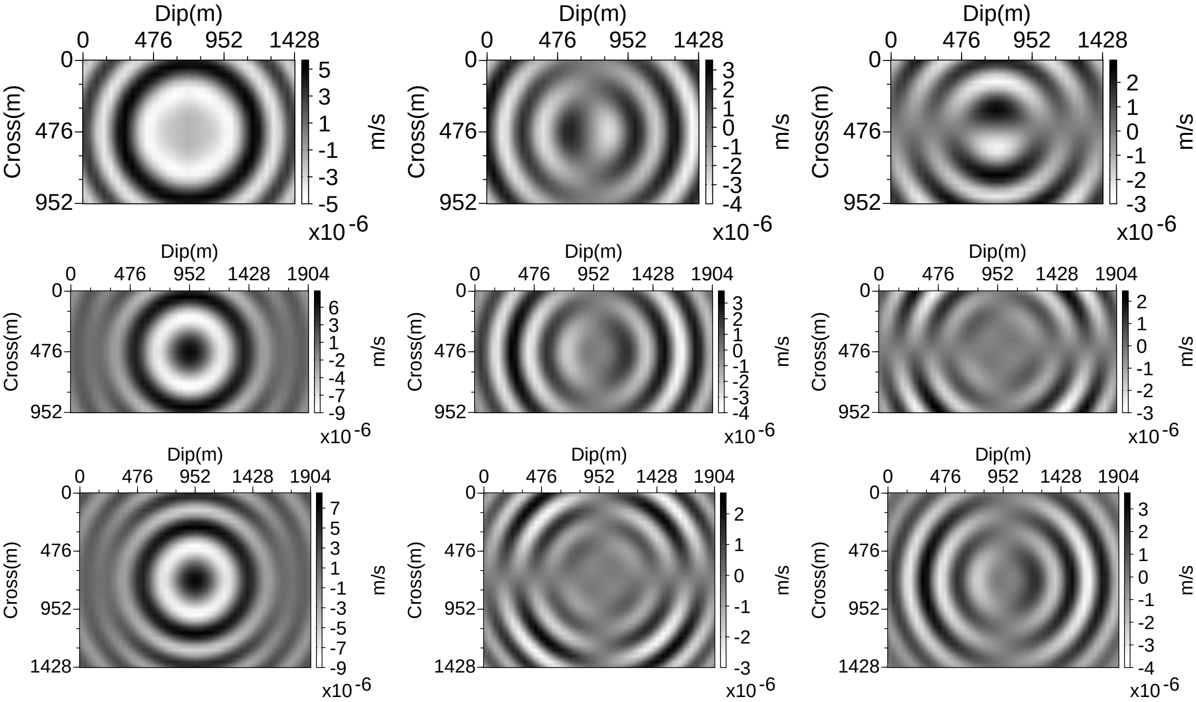


Figure 4.2: 3D solution of the frequency-domain elastic wave problem in a homogeneous medium without attenuation at $f = 5$ Hz and using a horizontal force source in the x-direction. The figures in the first column represent the particle velocity component v_x. The figures in the second column represent the particle velocity component v_y. The figures in the third column represent the vertical particle velocity component v_z. Figures in row 1 are the wavefields in the plane yz at the source position. Figures in row 2 are the wavefields in the plane xz at the source position. Figures in row 3 are the wavefield in the plane xy at the source position.

In the following, the gain in number of iterations is denoted by η_{iter} with

$$\eta_{iter} = \frac{N_{iter}(A)}{N_{iter}(AP)},$$

where $N_{iter}(A)$ denotes the number of iterations performed by the CARP-CG method to solve the non-preconditioned linear system and $N_{iter}(AP)$ denotes the number of iterations performed by the CARP-CG method to solve the preconditioned linear system. The increase in memory cost is introduced as well through the ratio η_{nnz}

$$\eta_{nnz} = \frac{\text{nnz}(AP)}{\text{nnz}(A)}$$

where nnz(A) and nnz(AP) represent respectively of the number of nonzero coefficients of the matrix A and AP. The ratio η_{nnz} denotes the fill-in effect induced by the matrix-matrix product AP in comparison with A.

4.2.1 27-point sparsity patterns

The 27-point sparsity pattern presented in Figure 4.4 takes into account the source point (colored in red) and the adjacent discretization points (colored in blue). When the force source is applied in the

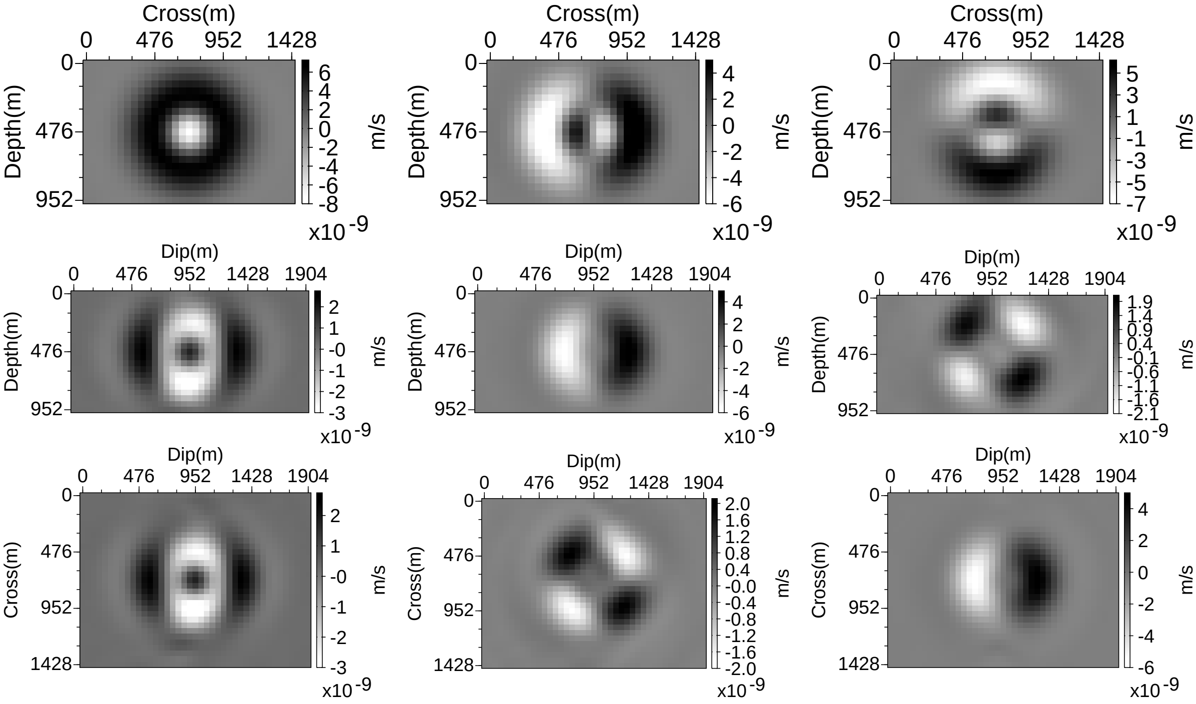


Figure 4.3: 3D solutions of the frequency-domain visco-elastic wave problem in a homogeneous medium with strong damping at $f = 5$ Hz and using a horizontal force source in the x direction. The figures in the first column represent the particle velocity component v_x. The figures in the second column represent the particle velocity component v_y. The figures in the third column represent the vertical particle velocity component v_z. Figures in row 1 are the wavefields in the plane yz at the source position. Figures in row 2 are the wavefields in the plane xz at the source position. Figures in row 3 are the wavefield in the plane xy at the source position.

x direction, the preconditioner is computed by selecting the particle velocity component v_x defined by the discretization points presented in the Figure 4.4. The particle velocities v_y and v_z are set to 0. The method is repeated for the particle velocity component v_y when the force source is applied in the y direction and the particle velocity component v_z when the force source is applied in the z direction. The reduction in the number of iterations and the memory request induced by such sparsity pattern are given by

$$\eta_{iter} = 4.8, \quad \eta_{nnz} = 7.6.$$

4.2.2 Alternative sparsity pattern

9-point sparsity pattern The 9-point sparsity pattern presented in Figure 4.5 takes into account the source point (colored in red) and the adjacent discretization points (colored in blue) located either in the plane xy as in the left figures or yz as in the right figures. The 3D particle velocity components are thus approximated in a 2D fashion. Similarly to the 27-point sparsity pattern, when the force source is applied in the x direction, the particle velocities v_y and v_z are set to 0.

3-point sparsity pattern The 9-point sparsity pattern presented in Figure 4.6 takes into account the source point (colored in red) at the center of the cube and the adjacent discretization points (colored

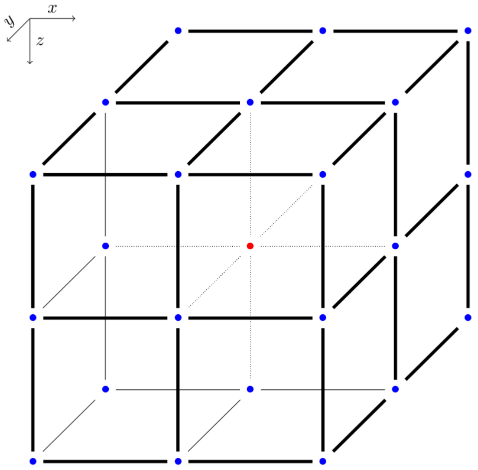


Figure 4.4: 27-point sparsity pattern. The source point is marked by the red bullet at the center of the cube. The discretization points involved in the stencil are colored in blue. It is the grid points which are adjacent to the source point. The particle velocities v_y and v_z are set to 0.

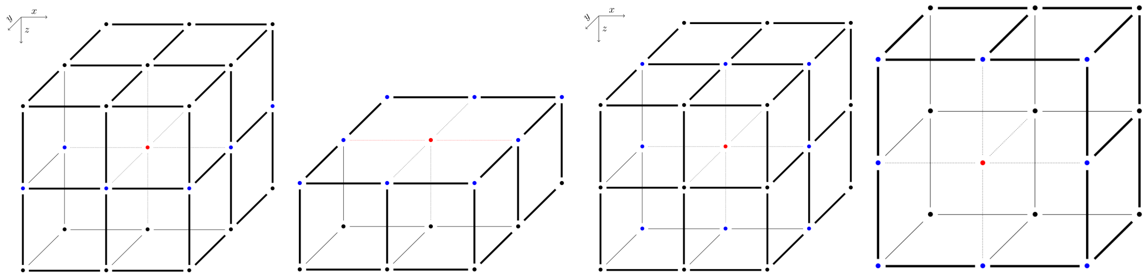


Figure 4.5: 9-point sparsity patterns. The source point is marked by the red bullet at the center of the cube. The discretization points involved in the stencil are colored in blue. It is the grid points which are adjacent to the source and are located either in the plane xy (see the two right figures) or in the plane xz (see the two left figures). The particle velocities v_y and v_z are set to 0

in blue) located on the axis where the force source is applied (Ox in this case). when the force source is applied in the x direction, the particle velocities v_y and v_z are set to 0. The 3D particle velocity components are thus drastically approximated in a 1D fashion.

The results obtained from the application of the CARP-CG method with the 27-point, 9-point and 3-point sparsity patterns are summarized in Table 4.2. The 9-point sparsity pattern in both configurations xy-plane and xz-plane gives exactly the same results when applied to the homogeneous problem.

	27-point	9-point	3-point
η_{iter}	4.8	2.8	1.9
η_{nnz}	7.6	3.9	1.8

Table 4.2: Reduction in the number of iterations and extra memory cost induced by the 27-point, 9-point and 3-point sparsity patterns with the CARP-CG method.

Note that the above sparsity patterns (4.4),(4.5) and (4.6) are tested on a small size homogeneous

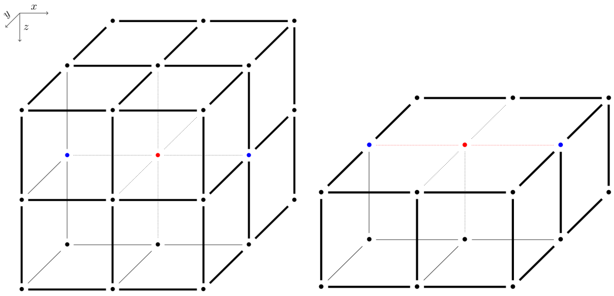


Figure 4.6: 9-point sparsity pattern. The source point is marked by the red bullet at the center of the cube. The discretization points involved in the stencil are colored in blue. It is the grid points which are adjacent to the source and are located on the axis where the force source is applied (Ox in this case). The particle velocities v_y and v_z are set to 0.

problem. The behavior changes when large scale problems with heterogeneities are experimented. A better reduction in the number of iterations is expected however, the memory increase remains the same.

For realistic applications which shall be introduced in the next section, the 27-point sparsity pattern is selected. This choice is motivated by the substantial reduction in the number of iterations performed by the CARP-CG method using a preconditioner computed on this stencil. Such reduction in the number of iterations underlines a good conditioning of the matrix AP. Therefore, thanks to the sweeps performed by CARP-CG which introduce a second level of preconditioning, the number of iterations is expected to grow slowly as the size of the linear system increases. Furthermore, the extra memory request induced by the use of this sparsity pattern remains affordable compared to what would be requested by direct solvers.

4.3 Numerical results

4.3.1 Setup

In this section, the CARP-CG method is used to solve the 3D frequency-domain visco-elastic wave equations using the heterogeneous model of size 16 × 9 × 4 km presented earlier in Figure 4.1. The wave simulations are performed at the frequencies $1.25,\ 2.5,\ 5$ and 7.5 Hz. The fourth-order staggered-grid finite-difference scheme with optimal coefficients is used (see Chapter 1 Section 1.5.3). To ensure the accuracy of the finite-difference scheme, the discretization is performed using 4.2 grid points per minimum shear wavelength which is equal to $V_{S,min} = 837$ m/s. The number of grid points in the PMLs is equal to $N_{PML} = 10$. The damping coefficient in the PMLs is set to $C_{PML} = 200$. The explosive source is located in the layer with the low P-wave velocity surrounded by the two layers with higher P-wave velocities. The source position is $x_s = 8$ km $y_s = 4.5$ km and $z_s = 0.55$ km. The relaxation parameter is set to $\lambda = 1.0$ and the stopping criterion is defined as usual by $\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2/\|\mathbf{b}\|_2 < 10^{-4}$. The geometry parameters, the total number of unknowns N_u and the estimated memory requirements are summarized in Table 4.3. Representative particle velocity wavefields are presented at the end of this section in Figures 4.9, 4.10 and 4.11. For medium and high frequencies in the low P-wave velocity layer, the waveguide geometry can be seen through the several reflections in this zone. In the shallow part of the medium, the small P-wave velocity generates the propagation of waves with short wavelengths. This is particularly visible on the left side of the domain.

f (Hz)	Δ (m)	$n_x \times n_y \times n_z$	N_u
1.25	160	$101 \times 57 \times 26$	1 285 746
2.5	80	$201 \times 113 \times 51$	6 260 709
5	40	$401 \times 226 \times 101$	37 594 458
7.5	26.6	$601 \times 338 \times 152$	114 716 088

Table 4.3: Geometry parameters (number of grid points per direction n_x, n_y, n_z and discretization space step Δ in meters) and total number of unknowns N_u which takes into account the $N_{PML} = 10$ grid points in the PML layers for the 3D frequency-domain visco-elastic wave simulations in the heterogeneous model (4.1).

I give a particular interest to the single/double-precision implementations aspect of the CARP-CG method which has been discussed as well by Li et al. (2015). For large size linear systems and particularly when the number of iterations performed by the CARP-CG method is too large, the CARP-CG method diverges when a single-precision implementation is used. Such behavior is observed in the 2D approximation at 10 Hz and 20 Hz and when the free-surface boundary condition is used on the elastic Marmousi 2 case study (see Chapter 3 Section 3.2.2.3). The large number of iterations (more than 30 000 at 10 Hz and more than 50 000 iterations at 20 Hz) increases the round-off errors. In addition, performing the sweeps on large blocks worsens the problem. The double-precision implementation is used to remedy to this issue.

			Simple-precision			**Double-precision**		
f (Hz)	N_u	N_proc	Mem (GB)	N_iter	Time (s)	Mem (GB)	N_iter	Time (s)
1.25	1 285 746	8	0.7	1 578	133.8	1.2	1 578	204.8
2.5	6 260 709	16	3.6	3 075	770.4	6.0	3 061	1 217.1
5	37 594 458	32	21.7	5 451	4 575.8	36.2	5 404	7 088.3
7.5	114 716 088	54	67.1	$-$	$-$	110.8	6 640	16 466.9

Table 4.4: 3D frequency-domain elastic wave simulations with the CARP-CG solver using the 3D heterogeneous elastic model (4.1). The single-precision and the double-precision implementations are tested. For the 7.5 Hz case, a divergence occurs in the single-precision implementation of the CARP-CG method. The double-precision allows to save a couple of iterations thanks to its accuracy. The computation time for the double-precision code is multiplied by a factor 1.5 compared to the single-precision code. The memory request is increased by a factor 1.6. "$-$" denotes that a divergence occurs.

Table 4.4 shows the results of the single- and double-precisions tests of the CARP-CG method for the solution of the 3D frequency-domain elastic wave equations using the heterogeneous model. The double-precision implementation of the CARP-CG method allows to save a couple of iterations although this is not the main objective of using such implementation. The computation time is increased by factor smaller than 2 (approximately 1.5) and the memory requirements for the storage of the nonzero coefficients of the matrix A and the right-hand side $\mathbf{b}$ are increased by a factor 1.6 approximately. During the numerical experiment on the $f = 7.5$ Hz case, a divergence occurs. It is due to a lack of accuracy because of the size of linear system and the number of iterations performed. The divergence occurs when the L_2-norm of the relative residual becomes small, i.e. near $1.5\ 10^{-4}$. Performing the double sweeps on smaller blocks (this is achieved by increasing the number of cores) does not help mitigating this problem. The use of the double-precision implementation is therefore mandatory for this large problem. The double-precision implementation is used for the numerical test on the 7.5 Hz case. Otherwise, the single-precision code is used for memory savings.

4.3.2 Computation of the preconditioner and the matrix AP: computational costs and run-times

For this 3D application, I pay a close attention to the computational resources required for the computation of the preconditioner and the matrix AP although their computations are embarrassingly parallel.

The computation of the preconditioner requires for each grid point of the discrete domain the solution of three small 3D frequency-domain strongly damped visco-elastic problems. The size of this local domain is 13 grid points per dimension (3 grid points per dimension according to the 27-point sparsity pattern and 5 grid points per PML). The resulting size of these small linear systems is 6 591. The latter are solved using roughly 100 iterations of the CGMN method. Assuming the most costly operation of one iteration of CGMN is the Kaczmarz double sweep operation which requires $4sn$ operations (s is the number of nonzero coefficients per row of the small impedance matrix, $s = 51$ due to the fourth-order staggered-grid stencil and n is the size of this matrix), the computational cost of one iteration of CGMN for the solution of the small linear system is roughly equal to $1.3\ 10^6$. Thus the computational cost of the solution of one small linear system using CGMN is approximately equal to $1.3\ 10^8$. Finally, the total computational cost of the preconditioner is

$$N_P^{op} = 3.9\ 10^8\ N^3,$$

with N is the average number of grid points per dimension.

The computation of the matrix-matrix product AP relies on the computation of sparse scalar products. Each row of the matrix AP contains at most 405 nonzero coefficients and each coefficient is, in average, the result of the sparse scalar product of one row of the matrix A and one column of the matrix P. Assuming the computational cost of the sparse scalar product is $2s$ with s the number of nonzero coefficient per row of the matrix A ($s = 51$), the total computational cost of the matrix-matrix product AP is roughly estimated at

$$N_{AP}^{op} = 4.1\ 10^4\ N^3.$$

		P			**AP**		
f (Hz)	N	N_{proc}	Mem (GB)	Time	N_{proc}	Mem (GB)	Time (s)
1.25	75	512	0.38	0 h 27 m 12 s	32	5.5	6.4
2.5	128	1024	1.96	1 h 08 m 00 s	64	27.4	15.9
5	232	2048	11.58	3 h 05 m 37 s	128	167.5	48.2
7.5	337	2048	35.85	8 h 11 m 11 s	256	514.2	72.8

Table 4.5: Average number of grid points per dimension (PML included), number of cores N_{proc}, total memory used and run time for the computation of the preconditioner and the matrix-matrix product AP.

Remark 4.1. *Exceptionally, the preconditioner is computed on the cluster "Ada" of the Institute for Development and Resources in Intensive Scientific Computing (IDRIS* `http://www.idris.fr`*).*

This cluster is composed of 332 compute nodes, a quadri-socket node of 4 Intel Sandy Bridge E5-4650 8-core processors at 2.7 GHz, with 32 cores per node with 128 GB of memory (4 GB/core). Nodes are linked through an InfiniBand FDR10 Mellanox network (2 links per node). In usage, the resources are accessible rapidly for up to 2048 execution cores. The matrix-matrix product AP and the resolutions using the CARP-CG methods are performed the CIMENT cluster.

The computation times in Table 4.5 show that the matrix AP is computed efficiently using a small amount of time. However, the computation of the preconditioner requires much more computational effort. Improvements can be done to speedup the computation of the preconditioner at two levels: implementation and reduction of the computational costs. The current implementation uses a MPI parallelization where each MPI process is in charge of the computation of the solution of the $3N^3/N_{proc}$ small linear systems. As for each grid point, the solution of these three linear systems can be computed independently as well, a second level of parallelism using multi-thread can be introduced. A speedup by a factor 3 can be obtained. Alternatively, as these local solutions are very similar after an adequate rotation in space, the computation of the preconditioner can be reduced to the computation of the solution of only one small linear system. The two remaining solutions can then be obtain by performing a space rotation of the computed wavefield.

4.3.3 Numerical simulations with CARP-CG and PCARP-CG

In Table 4.6, the results from the CARP-CG and the PCARP-CG numerical experiments on the 3D heterogeneous elastic model (4.1) are presented. For each frequency, the size of the linear system N_{u} is recalled and the memory used for the storage of the linear system (A and $\mathbf{b}$ for the CARP-CG method and AP and $\mathbf{b}$ for the preconditioned CARP-CG method) is given in the column "Mem" in GigaBytes (GB). The computations are performed using the single-precision implementation of the code. The preconditioner is computed on the strongly damped medium with the quality factor $q_{att} = 0.6$. Several experiments have shown that when the shear-wave velocity model is derived from the P-wave velocity model using the Poisson's relation (3.2) with the constant Poisson's ratio $\xi_p = 0.15$, optimal gain in the number of iterations is obtained. This behavior is observed in the 2D case study as well. Numerical tests performed in double-precision implementation are labeled by "*" in Table 4.6 (this is relative only to the CARP-CG results without preconditioning for the 7.5 Hz case).

It is interesting to note that thanks to the preconditioner, the single-precision CARP-CG code is able converge for the 7.5 Hz case. This result allows significant memory savings as the memory requirements for the storage of the matrix AP and the right-hand side $\mathbf{b}$ increases from 514.2 GB in single-precision to 867.9 GB in double-precision. The number of iterations are divided respectively by a factor 3.1, 5, 6 and 5.5 for the 1.25, 2.5, 5 and 7.5 Hz experiments. The number of iterations performed by the preconditioned CARP-CG method increases modestly. However, the run times show that despite the preconditioning strategy, the reduction in the number of iterations does not provide a substantial gain in the computation time. The number of iterations performed by PCARP-CG is still high in comparison with the computational burden induced by the fill-in of the matrix AP. Figure 4.7 shows the fast convergence of the relative residual of the preconditioned CARP-CG method whereas the relative residual of the CARP-CG (without preconditioning) has a slow convergence particularly starting from 10^{-3}.

The complexity analysis of the CARP-CG method and the preconditioned CARP-CG method are presented in Figure 4.8. The number of iterations N_{iter} is plotted as a function of the geometric mean

			CARP-CG			PCARP-CG		
f (Hz)	N_{u}	N_{proc}	Mem (GB)	N_{iter}	Time (s)	Mem (GB)	N_{iter}	Time (s)
1.25	1 285 746	8	0.7	1 578	133.8	5.5	510	284.7
2.5	6 260 709	16	3.6	3 075	770.4	27.4	620	1 005.7
5	37 594 458	32	21.7	5 451	4 575.8	167.5	910	4 068.2
7.5	114 716 088	54	110.8*	6 640*	16 466.9*	514.2	1 218	11 873.9

Table 4.6: Geometry parameters, total number of unknowns N_{u}, number of cores N_{proc}, total memory used, number of iterations N_{iter} and computing time for the 3D elastic wave simulations in a heterogeneous model. The preconditioner is computed on original model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$. Results which are labeled with "*" denote numerical experiments performed using the double-precision implementation of code due to a divergence of the single-precision version because of a lack of accuracy.

size of the discrete domain in one dimension

$$N = (n_x \times n_y \times n_z)^{1/3},$$

on a log-log scale. The solid black line indicates the linear increase that is

$$N_{iter} = \mathcal{O}(N),$$

and the dashed line represents the standard linear regression performed on the results from the CARP-CG method.

The computational costs of both methods are presented in Table 4.7.

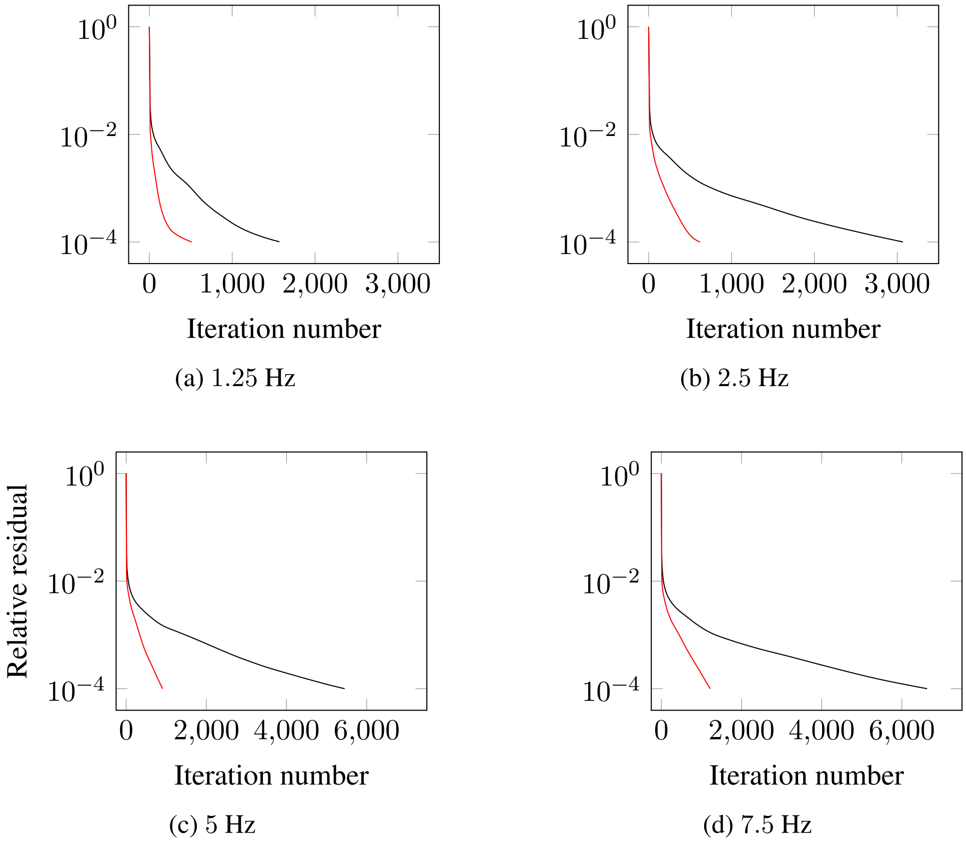
	N_{iter}	
	CARP-CG	PCARP-CG
Computational cost	$43 \times N^{0.90}$	$95 \times N^{0.42}$

Table 4.7: Computational cost of the CARP-CG method with and without preconditioning for the solution of the 3D frequency-domain elastic wave equations on the 3D heterogeneous elastic model.

The complexity analysis shows clearly the advantage of the preconditioning method. The number of iterations grows slowly with respect to the geometric mean size of the discretization N for the CARP-CG method. This is mainly due to the ill-conditioning of the problem and more precisely the source position which creates a difficult problem to solve by iterative solvers. Thanks to the preconditioning, the complexity of the CARP-CG method is nicely improved. The exponent β of the complexity αN^{β} is divided by factor slightly bigger than 2 yielding the complexity $N_{iter} = \mathcal{O}(N^{0.42})$. However, the overhead α has increased.

4.3.4 Numerical tests with an alternative source position

In the previous experiments, the position of the source in the low P-wave velocity layer clearly introduced a difficulty for the solution of this problem using both CARP-CG and PCARP-CG solvers. In this



(a) 1.25 Hz

(b) 2.5 Hz

(c) 5 Hz

(d) 7.5 Hz

Figure 4.7: Convergence histories of the 3D frequency-domain elastic wave simulations using the CARP-CG method and the preconditioned CARP-CG method (PCARP-CG) for the 3D heterogeneous elastic model. The convergence curves relative to the CARP-CG method are plotted in black. The convergence curves relative to the preconditioned CARP-CG method (PCARP-CG) are plotted in red. The preconditioner is computed with the quality factor $q_{att} = 0.6$. Its computation is performed on the medium where the shear wave velocity model is derived from the P-wave velocity model using the Poisson's relation and a constant Poisson's ratio $\xi_p = 0.15$.

paragraph, I investigate the previous numerical tests with a source position outside of the low P-wave velocity layer. Therefore, I choose to put the source far from the heterogeneities of the shallow part of the medium. The explosive source is located at depth in a more homogeneous zone at $x_s = 4$ km, $y_s = 4.5$ km and $z_s = 3$ km.

The number of iterations and the computation time are presented in Table 4.8. In this configuration, the CARP-CG method performs less iterations to solve the problem. More interestingly, the performances of the PCARP-CG method are also better than expected. The preconditioned CARP-CG method performed impressively a small number of iterations. The reduction in the number of iterations by factors 8.9, 9.4, 9.8 and 11 is obtained which is more than twice the reduction obtained with the numerical tests using the source in the low P-wave velocity layer. The time speedup ranges from 1.3 to 2.2.

The computational costs of both methods are presented in Table 4.9. The gain in computational cost is clearly noticeable in the reduction of the constant multiplying the complexity. Time speedups is thus obtained at early frequencies.

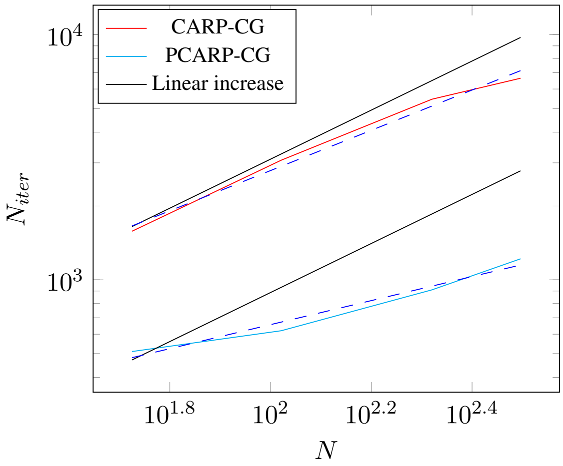


Figure 4.8: Complexity analysis of the CARP-CG method with and without preconditioning for the 3D frequency-domain elastic wave simulations on the 3D heterogeneous elastic model. The number of iterations N_{iter} is plotted as a function of the geometric mean size of the discrete domain in one dimension N on a log-log scale. The straight dashed line indicates the results obtained by a line regression. The preconditioner is computed with the quality factor $q_{att} = 0.6$. Its computation is performed on the medium where the shear wave velocity model is derived from the P-wave velocity model using the Poisson's relation and a constant Poisson's ratio $\xi_p = 0.15$.

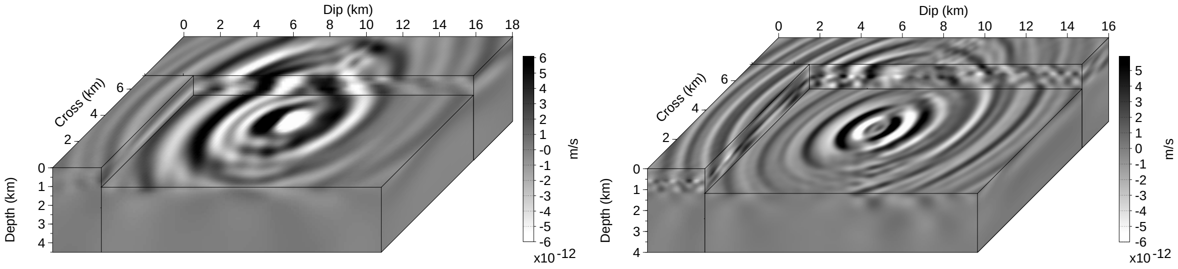
			CARP-CG			**PCARP-CG**		
f (Hz)	N_{u}	N_{proc}	Mem (GB)	N_{iter}	Time (s)	Mem (GB)	N_{iter}	Time (s)
1.25	1 285 746	8	0.7	1 574	131.7	5.5	176	98.6
2.5	6 260 709	16	3.6	2 267	567.2	27.4	241	388.4
5	37 594 458	32	21.7	3 463	2 910.8	167.5	355	1 601.2
7.5	114 716 088	54	110.8*	4 954*	13 197.1*	514.2	451	4 065.3

Table 4.8: Geometry parameters, total number of unknowns N_{u}, number of cores N_{proc}, total estimated memory used, number of iterations N_{iter} and computing time for the 3D elastic wave simulations in a heterogeneous model with an explosive source located in depth outside the low P-wave velocity layer at $x_s = 4$ km, $y_s = 4.5$ km and $z_s = 3$ km. The preconditioner is computed on original model where the shear wave velocity model is derived using the constant Poisson's ratio $\xi_p = 0.15$ and where a strong damping is introduced through the quality factor $q_{att} = 0.6$. Results which are labeled with "*" correspond to numerical experiments performed using the double-precision implementation of code due to a divergence of the single-precision version because of a lack of accuracy.

Such configuration, i.e. with the source located in a homogeneous zone, is quite comparable to simulations using models with a water column with a source in this zone, similar to the BP2004 and the Marmousi 2 case studies presented in Chapter 3. One can expect that for such model, i.e. with a water column, similar gains could be obtained.

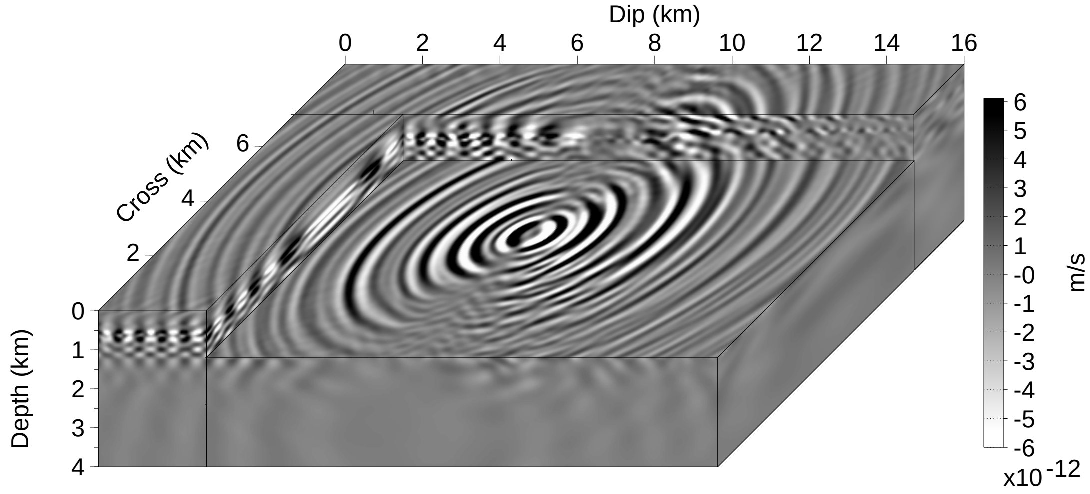
	N_{iter}	
	CARP-CG	PCARP-CG
Complexity	$122 \times N^{0.64}$	$21 \times N^{0.53}$

Table 4.9: Computational cost of the CARP-CG method with and without preconditioning for the solution of the 3D frequency-domain elastic wave equations on the 3D heterogeneous elastic model with an explosive source located outside the low P-wave velocity layer at $x_s = 4$ km, $y_s = 4.5$ km and $z_s = 3$ km.

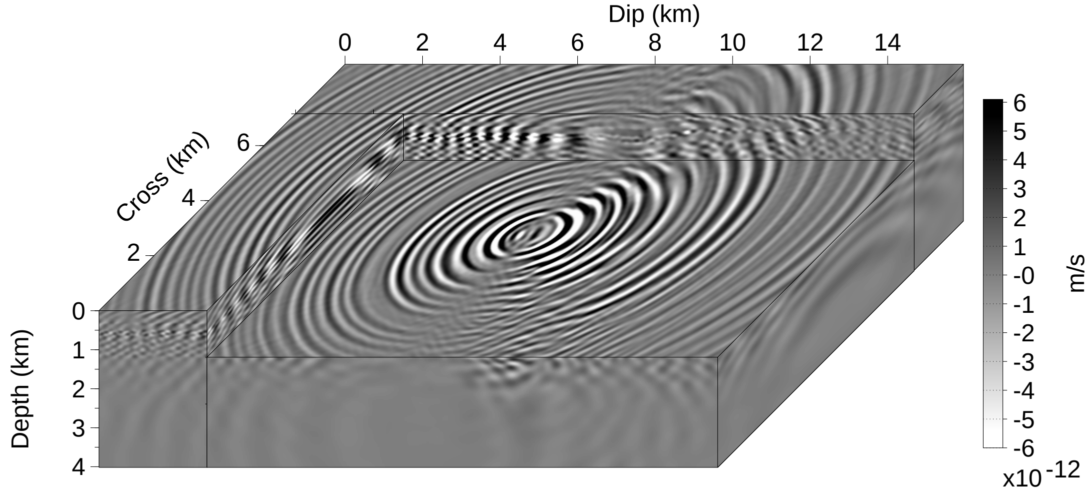


(a) 1.25 Hz

(b) 2.5 Hz

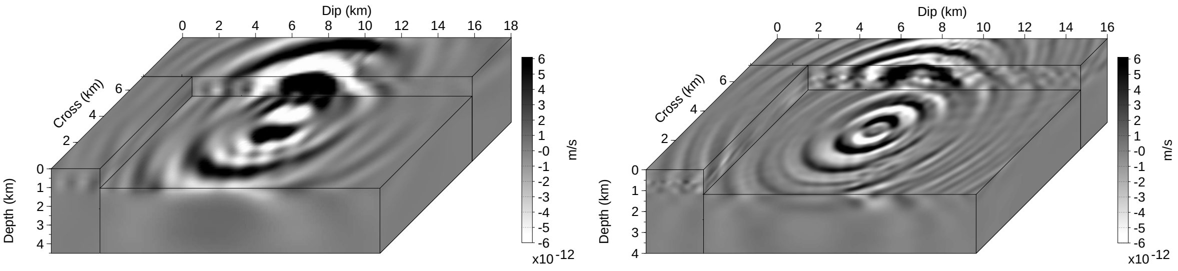


(c) 5 Hz

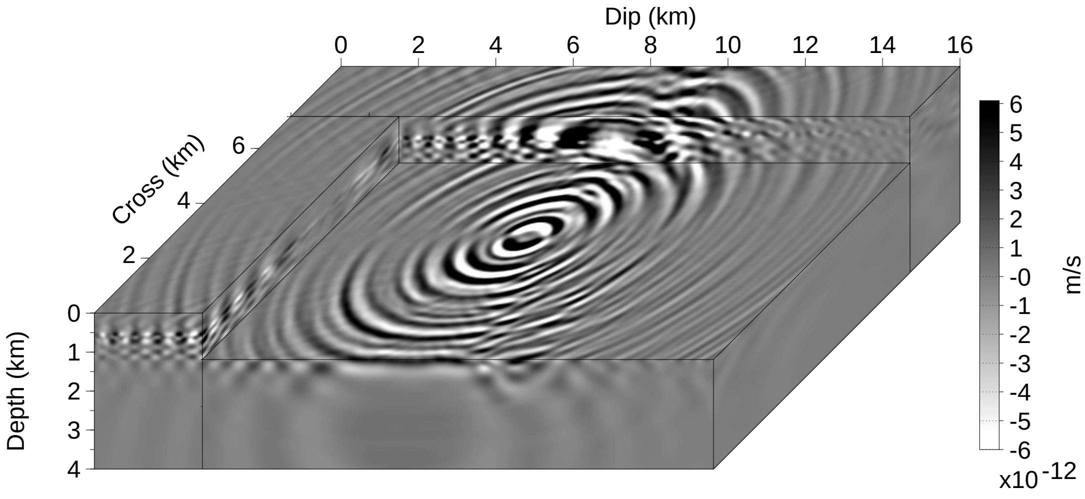


(d) 7.5 Hz

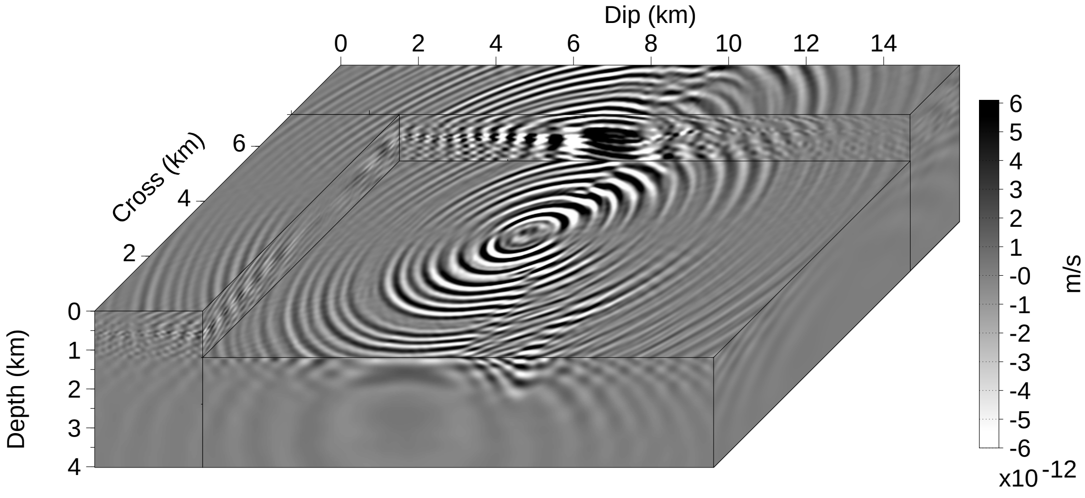
Figure 4.9: 3D frequency-domain particle velocity component v_x at 0.625 (a), 1.25 (b), 2.5 (c), 5 Hz (d) and 7.5 (e). In the low P-wave velocity layer, the waveguide geometry can be seen through the several reflections in this zone for the medium and high frequencies. In the shallow part, the small P-wave velocity generates the propagation of waves with short wavelengths. This is particularly the case on the left side of the domain.



(a) 1.25 Hz (b) 2.5 Hz

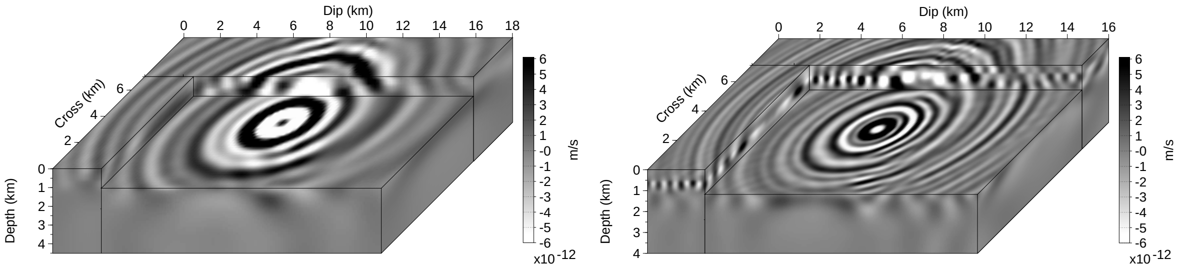


(c) 5 Hz



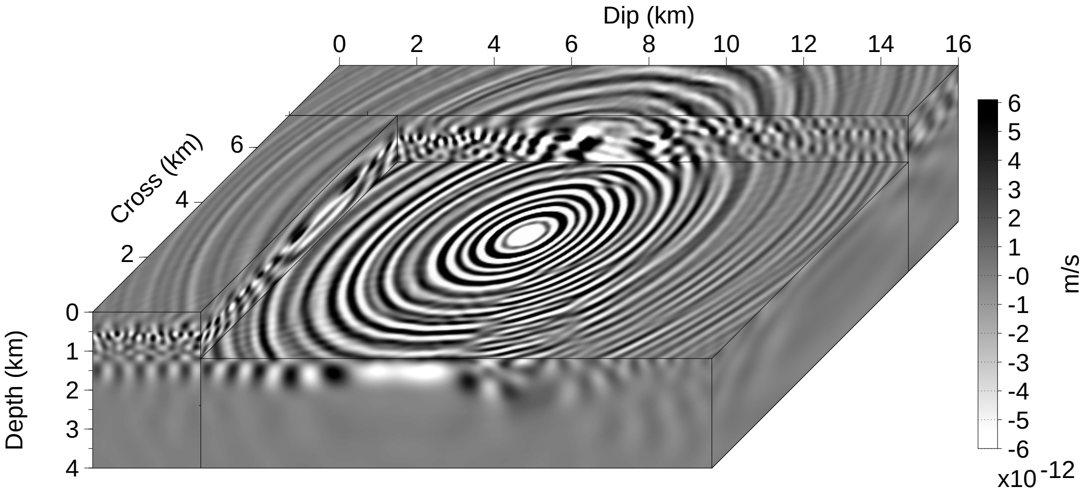
(d) 7.5 Hz

Figure 4.10: 3D frequency-domain particle velocity component v_y at 0.625 (a), 1.25 (b), 2.5 (c), 5 Hz (d) and 7.5 (e). In the low P-wave velocity layer, the waveguide geometry can be seen through the several reflections in this zone for the medium and high frequencies. In the shallow part, the small P-wave velocity generates the propagation of waves with short wavelengths. This is particularly the case on the left side of the domain.

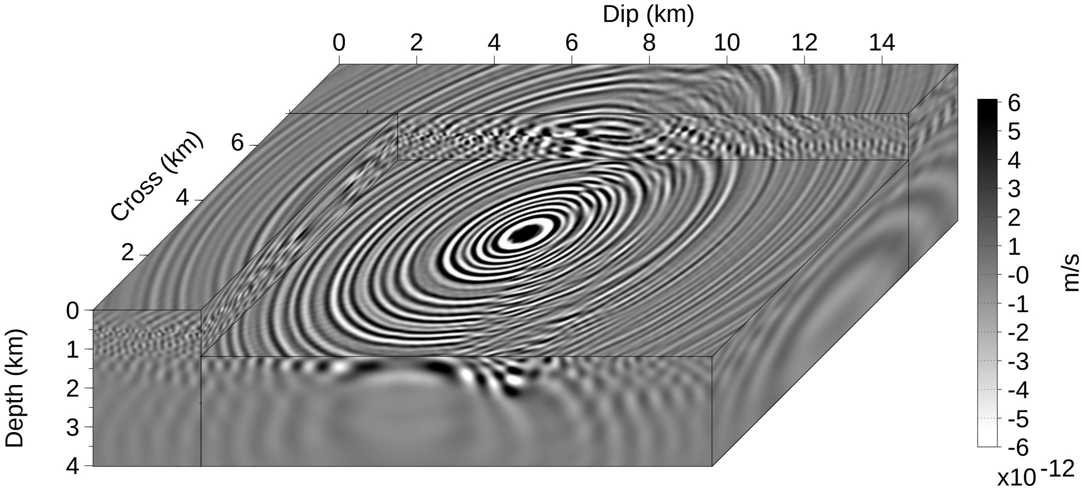


(a) 1.25 Hz

(b) 2.5 Hz



(c) 5 Hz



(d) 7.5 Hz

Figure 4.11: 3D frequency-domain particle velocity component v_z at 0.625 (a), 1.25 (b), 2.5 (c), 5 Hz (d) and 7.5 (e). In the low P-wave velocity layer, the waveguide geometry can be seen through the several reflections in this zone for the medium and high frequencies. In the shallow part, the small P-wave velocity generates the propagation of waves with short wavelengths. This is particularly the case on the left side of the domain.

4.4 Computational complexities comparison with the time-domain 3D elastic wave modeling and the sparse parallel direct solver approaches

Consider the 3D time-domain elastodynamic wave equations in velocity-stress formulation. The time-domain approach consists in the discretization of the system of equations using an explicit second-order scheme in time and the fourth-order scheme in space. Discrete Fourier Transform (DFT) is used to extract the needed frequencies. Assuming that an addition and a multiplication count for one operation, the evaluation of one time step requires $46\ N^3$ operations. Therefore, the total number of operations is roughly estimated as $46\ N_t\ N^3$ operations where N_t is the number of time steps. The latter is constrained by the CFL condition which is given for the fourth-order scheme by

$$\Delta t = \alpha_1 \frac{h}{V_{max}}, \quad \alpha_1 = 0.495,$$

and the recording time $T = \Delta t\ N_t$ which depends on the size of the problem $L = hN$ and the smallest wave speed V_{min}

$$T = \alpha_2 \frac{L}{V_{min}}, \quad 1 \leq \alpha_2 \leq 2.$$

This gives

$$N_t = \frac{\alpha_2}{\alpha_1} \frac{V_{max}}{V_{min}} N.$$

By considering the 3D heterogeneous elastic model (4.1), the ratio V_{max}/V_{min} is equal to 7.16 (see Table 4.1). Assuming $\alpha_2 = 2$, as $n_x = 2N$ approximately, the number of operations for the time-domain approach is estimated at

$$N^{op}_{TD} = 1332\ N^4.$$

The DFT is performed at each time step. The computational cost of this operation can be roughly estimated at

$$N^{op}_{DFT} = 1738\ N^4.$$

Note that the numerical cost of the evaluation of the trigonometric functions is roughly estimated at 1. One can see that the computational cost of the DFT is larger than the time-domain simulation. A comparative analysis on the time complexities is presented in the 3D acoustic approximation in (Brossier et al., 2014b). Therefore, the total number of operations for the time-domain modeling combined with the DFT approach is estimated at

$$N^{op}_{TD+FDT} = 3070\ N^4.$$

Using the preconditioned CARP-CG method as the frequency-domain solver, the numerical results and the complexity analysis given in Table 4.7, the total number of operations performed by the preconditioned iterative solver for the 3D simulation is given by

$$N^{op}_{FD+I} = 95\ N^{3.42}.$$

The results from the simulations with the source in the low- P-wave velocity layer are used.

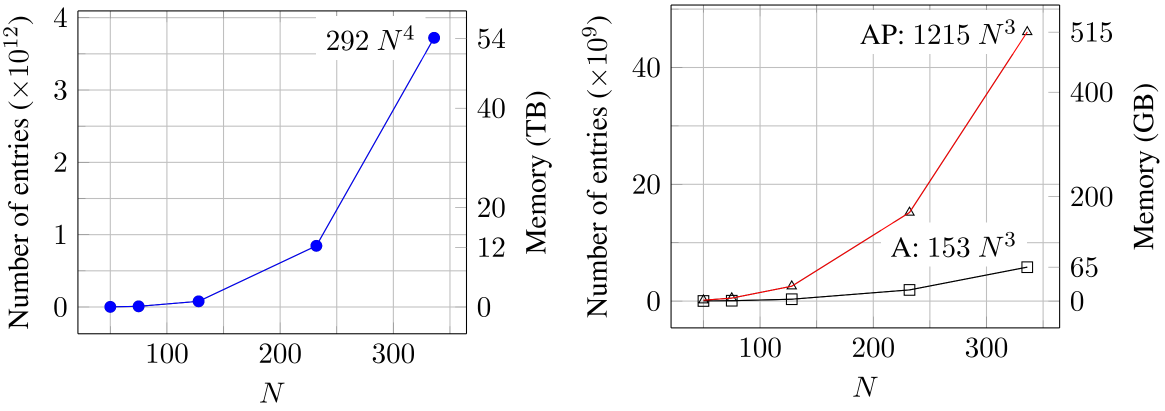
A sparse direct solver can be used as well to solve the linear system. Table 4.10 summaries the computational and memory complexities of these three modeling approaches in the frequency-domain.

The main limitations of direct solvers is the memory issue. Tackling large scale problems still remains out of reach given the current high performance computing clusters. Despite the latest developments carried out by Gosselin-Cliche and Giroux (2014) using compact stencils to reduce the

	Computational complexity	Memory complexity
TD + DFT	$3070\ N^4 = \mathcal{O}(N^4)$	$\mathcal{O}(N^3)$
FD + Iterative	$95\ N^{3.42} = \mathcal{O}(N^{3.42})$	$\mathcal{O}(N^3)$
FD + Direct	$\mathcal{O}(N^4)$	$\mathcal{O}(N^4)$

Table 4.10: Theoretical computational and memory complexities of the three modeling approaches in the frequency-domain (FD) using one source for a 3D elastic problem of size N^3 inspired from the 3D elastic case study. “TD + DFT" denotes the time-domain (TD) approach combined with the Discrete Fourier Transform (DFT). “FD + Iterative" denotes the resolution of the frequency-domain (FD) linear system using an iterative method, i.e. PCARP-CG. ‘FD + Direct" denotes the resolution of the frequency-domain linear system using the sparse parallel direct solver (MUMPS-team, 2006).

bandwidth of the matrix A or Wang et al. (2012) using compression schemes to reduce the memory requirements induced by the fill-in, the memory requirements of direct solvers become tremendous. For a simulation performed on a 601×338×152 discrete elastic model (see Figure 4.12 (a)), a memory requirement of 54 TB could be expected for the MUMPS solver.



(a) Number of entries and memory request for L and U

(b) Number of entries and memory request for A and AP

Figure 4.12: Estimated number of nonzero coefficients in the LU factors and memory for the storage of the LU factors issued by the MUMPS (MUMPS-team, 2006) sparse parallel direct solver (a). The estimation is reproduced from (Gosselin-Cliche and Giroux, 2014) using the second-order staggered-grid finite-difference method with optimal coefficients and a single-precision implementation. Estimated number of nonzero coefficients in the matrix A (squares) and AP (triangles) and memory request for their storage (b). Note the difference of magnitude for the storage of LU on the one hand and A and AP on the other hand.

On the other hand, time-domain and iterative solvers fully benefit from the sparsity of the discretization method. Iterative methods require the storage of the matrix A which remains quite modest compared to the direct solvers memory complexity. Figure 4.12 (b) shows the memory requirements for the storage of the matrix A and AP. In comparison with direct solvers, the memory complexity of iterative solvers is very small.

Thanks to the low computational complexity of the preconditioned CARP-CG method, the latter can be seriously considered for 3D frequency domain elastic wave simulations for high frequencies up to 15 Hz considering the current frequency range used in FWI applications.

4.5 Conclusion

In this chapter, I have presented a realistic 3D application for the simulation of frequency-domain elastic wave propagation using the iterative solver CARP-CG. The preconditioning strategy in 3D is investigated using a homogeneous model. The sparsity pattern with a compact support and conserving the highest amplitude in the particle velocity components is used. This sparsity pattern is introduced as a 27-point stencil.

The numerical tests are performed on the 3D heterogeneous elastic model (courtesy of Shell). The model is of size $16 \times 9 \times 4$ km and presents heterogeneities in the P-wave velocity model, the S-wave velocity model and the density model. The Poisson's ratio panel is heterogeneous as well and varies from 0.01 to 0.40. This model is particularly chosen for the difficulties it rises for iterative solvers. The model is formed by a superposition of layers of different P-wave velocities. One particular layer of low P-wave velocity is surrounded between two layers of higher P-wave velocity creating a waveguide geometry. When the source is located in this particular layer, iterative solvers have difficulties converging and this is confirmed by the numerical results.

The 3D heterogeneous model is discretized at the frequencies 0.625, 1.25, 2.5, 5 and 7.5 Hz. The size of the linear systems resulting from such discretization ranges from $3.4\ 10^5$ to $1.1\ 10^8$. First, the CARP-CG is used to the solve the linear systems using a single-precision implementation. Due to the lack of accuracy and the accumulation of round off errors, a divergence occurs on the 7.5 Hz problem. Therefore, a double-precision implementation is used to remedy this issue. The complexity analysis of the CARP-CG method gives a total number of operations for the 3D frequency-domain wave simulations

$$N^{op}_{FD+I} = 43\ N^{3.9}, \quad (CARP - CG)$$

where N is the geometric mean size of the discrete model ($N = (n_x \times n_y \times n_z)^{1/3}$ with $n_x,\ n_y$ and n_z the number of grid points along the corresponding space direction). The numerical results show that the complexity of the CARP-CG is nearly linear. Second, the CARP-CG method is used with the preconditioner. The latter is computed based on the strongly damped medium by using $q_{att} = 0.6$. Several experiments have shown that an optimal preconditioner, i.e. giving the best reduction in the number of iterations, is obtained by considering the medium parameter $V_P, \tilde{V_S}, \rho$ where $\tilde{V_S}$ is derived from V_P using a constant Poisson's ratio $\xi_p = 0.15$. Using this preconditioner, the complexity of the preconditioned CARP-CG method is improved to

$$N^{op}_{FD+I} = 95\ N^{3.42}, \quad (PCARP - CG)$$

which is very satisfying. Thanks to the preconditioner, the divergence problem which occurs at 7.5 Hz with the singly-precision implementation is mitigated.

Other numerical tests are performed using an explosive source outside the low P-wave velocity zone. The source is located at depth with the medium is almost homogeneous. Such configuration gives very encouraging results. Using the preconditioner, the number of iterations is divided by a factor up to 11 allowing a time speedup by a factor 2.2.

The comparison with the direct solver and time-domain combined with the DFT approaches shows the crucial advantages of the iterative method PCARP-CG. On the one hand, direct solvers suffer from the huge memory requirements for the storage of the factors L and U which makes them impracticable for simulations on a $601 \times 338 \times 152$ discretization grid. More than 50 TB would be requested for such a problem size with a direct solver. Iterative solvers only need the storage of the matrix AP which roughly amounts to 514 GB of storage. On the other hand, the time-domain approach combined with Discrete Fourier Transform requires relatively high computational cost

$$N_{TD+DFT}^{op} = 2201\ N^4.$$

In addition, such approach suffers from the lack of multiple source modeling techniques. The time-domain simulation relative to each source needs is performed independently which denotes a linear increase of the computational cost with respect to the number of sources. CARP-CG may benefit from multiple right-hand side strategies such as the block approach (O'Leary, 1980) or the projection method (Chan and Wan, 1997). Theoretical results show that for a number of right-hand sides N_{rhs}, the block conjugate gradient method may converge in roughly N_{rhs} times fewer iterations than a naive application of CG over the N_{rhs} linear systems independently. Therefore, the frequency-domain approach using the preconditioned CARP-CG method seems to be appealing.

The computation of the preconditioner and the matrix AP is indeed not costless. The complexity of both operations is in the order of $\mathcal{O}(N^3)$. Numerical results show that the product AP is computed efficiently in a small amount of time but the preconditioner requires much more computational efforts. However, one can mitigate this computational cost as soon as multiple right-sides are considered. For a given linear system, the preconditioner and the matrix AP are computed once and used for the solution of several right-hand sides.

Finally, one may consider the memory cost induced by the fill-in due to the matrix-matrix product AP as a possible limitation of this method. Indeed, using the fourth-order scheme for the discretization of the frequency-domain elastic wave equations and the 27-point sparsity pattern for the preconditioner yield a matrix A with at most 51 nonzero coefficients per row and a matrix AP with at most 405 nonzero coefficients per row. For the 7.5 Hz problem, the storage of the matrix AP and the right-hand side requires 514.2 GB. Performing numerical experiments at 10 Hz would require 1.1 TB using the single-precision implementation and 1.9 TB in double-precision which may seem close to the limit of current computing clusters today. This memory issue can be mitigated by using more compact finite-difference schemes. In this direction, the compact second-order scheme of Gosselin-Cliche and Giroux (2014) is suggested. Such compact scheme should reduce the fill-in in the matrix AP. This compact second-order stencil together with a preconditioner computed with the 27-point sparsity pattern give a matrix A and a matrix AP with, respectively, at most 51 and 261 nonzero coefficients per row. A reduction of the memory requirements by a factor 1.5 approximately is obtained which helps reducing the memory request for the storage of the matrix AP from 1.9 TB to 1.3 TB. This is due to the smaller bandwidth of the matrix A.

Conclusions and perspectives

Conclusions for this work

The work performed during this thesis was carried out in the objective to develop a preconditioning strategy for the frequency-domain wave propagation problem adapted to the iterative solver CARP-CG. In the framework of the seismic imaging techniques such as the Full Waveform Inversion or the Reverse Time Migration, the solution of the forward problem, i.e. frequency-domain wavefields, for large scale problems and numerous sources is required at each iteration of the optimization procedure to build high resolution quantitative images of the subsurface parameters. Therefore, a robust and efficient method for the solution of the forward problem is the cornerstone for these imaging methods.

In chapter 1, the equations governing the wave propagation in the subsurface are introduced. These equations are presented in the frequency-domain which are derived from the time-domain formulation. Both acoustic and elastic approximation are considered. At the exploration and regional scales, the subsurface can be assimilated to a semi-infinite domain. Absorbing boundary conditions are used to mimic infinite domain wave propagation in the horizontal direction and the vertical direction in depth. The Perfectly Matched Layers of Bérenger (1994) are preferred for their efficiency and the simplicity of their implementation. The free-surface boundary condition (Mittet, 2002) is introduced at the top boundary of the domain as it allows to perform realistic wave propagation modeling. The discretization of the acoustic and elastic equations is performed using the staggered-grid finite-difference method. In the frequency-domain, this discretization leads to the linear system

$$A\mathbf{x} = \mathbf{b}, \tag{4.1}$$

of size n which can be solved using a direct or an iterative method. A review of the different strategies for the computation of frequency-domain wavefields is presented. A state of the art regarding iterative solvers and preconditioning techniques in particular for the wave equation is carried out.

In chapter 2, an overview of the iterative solvers CGMN (Björck and Elfving, 1979) and its parallel version called CARP-CG (Gordon and Gordon, 2010b) is given. These particular solvers attract special interest due to their robustness for solving the 2D and 3D frequency-domain wave equations in the acoustic and elastic approximations in complex media and for large scale problems (Li et al., 2015). These iterative methods solve the linear system (4.1) by the mean of the Kaczmarz row projections (Kaczmarz, 1937) performed in double sweeps (projections from the first row to the last row then from the last row to the first row) which transforms the matrix A into an hermitian positive semi-definite matrix $I - Q(A)$. The conjugate gradient (CG) is used to accelerate the convergence. It amounts to apply the CG method the linear system

$$(I - Q(A))\mathbf{x} = R(A)\mathbf{b}.$$

Two preconditioning strategies are introduced to improve the convergence of these iterative methods. The first strategy relies on the computation of a preconditioner P of the matrix A and the resolution of the system of size n

$$P(I - Q(A))P^H\mathbf{y} = PR(A)\mathbf{b}, \quad \mathbf{x} = P^H\mathbf{y}.$$

The second strategy is based on the application the CGMN or CARP-CG method to the preconditioned system

$$AP\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P\mathbf{y}. \tag{4.2}$$

For both approaches, the preconditioner P needs to be computed on a diagonal dominant matrix . In the context of seismic wave modeling, this is achieved by computing the preconditioner based on a wave operator G in a strongly damped medium using complex-valued P-wave and S-wave velocities for instance. The second preconditioned strategy gave the best results and is therefore retained. However, it requires P to be sparse and banded and the explicit computation of the matrix AP as well. The new approach for the computation of preconditioner which is introduced in this thesis is to identify the columns of the matrix G^{-1} as the Green's function associated with the frequency-domain wave propagation operator G in a strongly damped medium. Due to the strong attenuation, the wavefield is closely located around the source point. Therefore, the each column of the preconditioner P can be sparsely approximated by restraining the Green's function to few discretization points around the source. The computation of the preconditioner is performed column-wise by solving n frequency-domain wave problems in small strongly damped domains defined by a sparse approximation. Perfectly Matched Layers (Bérenger, 1994) are used to truncate the computational domain to the local approximation. As each column of P can be computed independently and using $\mathcal{O}(1)$ operations, the computation of the preconditioner is embarrassingly parallel and can be performed in low complexity $\mathcal{O}(n)$. Thanks to the sparse approximation of the preconditioner and its small bandwidth, the matrix AP remains sparse and banded and its computation can be performed in linear complexity as well $\mathcal{O}(n)$. The spectral properties of the preconditioned system AP are evaluated in the acoustic and elastic approximation. Despite the sparse approximation, the conditioning of the matrix AP is improved and its spectrum is closely gathered around 1.

In chapter 3, 2D realistic case studies are presented to evaluate the performances of the preconditioner for large scale and heterogeneous problems. In the acoustic approximation, the frequency-domain wave simulations are performed on the BP2004 model (Billette and Brandsberg-Dahl, 2004) which provides heterogeneous P-wave velocity and density models of size 67 km × 12 km. The size of the linear systems to be solved ranges from $1.4\ 10^4$ to $1.4\ 10^7$. The complexity of the CGMN method is evaluated as

$$N_{iter} = 20.9\ N^{0.79}, \quad \text{(CGMN)}$$

where N_{iter} denotes the number of iterations performed by the iterative solver and N is the geometric average number of discretization points per direction (n_x, n_z)

$$N = \sqrt{n_x \times n_z}.$$

Using the preconditioner, the complexity of the CGMN method is improved to

$$N_{iter} = 5.9\ N^{0.71}, \quad \text{(PCGMN).}$$

A reduction in the number of iterations by factor 6.9 is obtained, yielding a computation time acceleration by a factor 2.2 in average over all the simulated frequencies. The total computational complexity of the CGMN method for the 2D frequency-domain acoustic modeling with one source is then

$$N^{op}_{FD+I} = 5.9\ N^{2.71}.$$

Compared to the time-domain approach with the DFT which the computational complexity is in $\mathcal{O}(N^3)$, this result denotes an improvement. In the 2D elastic approximation, the frequency-domain wave equations are solved on the Marmousi 2 model (Martin et al., 2006) using the parallel solver CARP-CG. The numerical experiments are performed with PMLs at the surface and a free-surface as well. The size of the linear system ranges from 10^5 to 10^7. Very satisfying results are obtained for both configurations. The number of iterations is divided by a factor approximately equal to 9 allowing a speedup in the computation time by a factor 3.5 when the simulations are performed with the free-surface. The numerical complexity for the simulation with one source is given by

$$N_{FD+I}^{op} = \begin{cases} 690\ N^{2.28} & \text{PMLs,} \\ 472\ N^{2.41} & \text{Free-Surface,} \end{cases} \quad (\mathrm{PCARP-CG}).$$

Such complexities denote an number of iterations roughly proportional to $\sqrt{N}$. Due to the sparsity of the linear system, the memory complexity in the 2D approximation is

$$N_{FD+I}^{mem} = \mathcal{O}(N^2).$$

The time-domain approach allows the computations of the wavefields with a number of operations equal to

$$N_{TD}^{op} = 6200\ N^3.$$

A Discrete Fourier Transform (DFT) is included at each time step to extract the needed frequency yielding a computational cost for the DFT roughly equal to

$$N_{DFT}^{op} = 6200\ N^3$$

In these estimations, the numerical cost of the evaluation of the trigonometric functions is roughly estimated at 1. In the 3D acoustic approximation, Brossier et al. (2014b) show that the computation time of the DFT is almost equal to the time-domain simulation for the computation of the wavefields.

Alternatively, a sparse direct solver can be used as well to solve the linear system (4.1). Table 4.11 summaries the computational and memory complexities of the three modeling approaches in the frequency-domain. Despite the small computational and memory complexity of the iterative approach in 2D, the frequency-domain approach using a sparse parallel direct solver (MUMPS-team, 2006) is usually preferred due to the numerous right-hand sides to solve. Once the LU factorization is performed, the linear system is efficiently solved by forward and backward substitutions. This method

	Computational complexity	Memory complexity
Time-domain + DFT	$12400\ N^3 = \mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
Freq-domain + Iterative*	$472\ N^{2.28} = \mathcal{O}(N^{2.28})$	$\mathcal{O}(N^2)$
Freq-domain + Direct	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2 \log(N))$

Table 4.11: Theoretical computational and memory complexities of the three modeling approaches in the frequency-domain using one source for a 2D elastic problem of size N^2 inspired from the Marmousi 2 case study. "*" refers to the results using the free-surface at the top boundary of the domain.

is efficient for multiple right-hand sides linear system. Using the time-domain solver, each source is solved independently, making this approach not efficient at all as the computational cost scales linearly with the number of sources. Iterative solvers can be combined with multiple right-hand sides strategies such as the block conjugate gradient (O'Leary, 1980; Gutknecht, 2006) and projection methods (Chan and Wan, 1997). Theoretical results show that for a number of right-hand sides N_{rhs}, the block conjugate gradient may converge in roughly N_{rhs} times fewer iterations than a naive application of CG over the N_{rhs} linear systems independently.

The scaling properties of the iterative solver CARP-CG are investigated as well. Gordon and Gordon (2010b) and Li et al. (2015) have already shown the good scaling properties of this method. The scaling remains close to the ideal scalability up to 128 cores and this scaling improves for high frequencies , i.e. for large problem size. Applying the CARP-CG to the preconditioned system (4.2) gives better scaling properties. Using the same number of cores, the CARP-CG applied to (4.2) presents a better speedup and efficiency compared to applying CARP-CG to (4.1). Such behavior is mainly due the communication cost which becomes significantly smaller than the computation cost on the preconditioned system (4.2) as the matrix AP is denser than A. Therefore, the application of the CARP-CG method to the preconditioned system enables to use a larger number of cores, allowing thus faster computations.

In chapter 4, the 3D frequency-domain elastic approximation is investigated on a realistic case. The system of equations is discretized using the fourth-order staggered-grid finite-difference method with optimal coefficients (Li et al., 2016). A 3D heterogeneous elastic model (courtesy of Shell) is used to performed the simulations. The size of the model is 16 km$\times$9 km$\times$4 km. The model is formed of the superposition of several layers of sediments of increasing P-wave velocities. One difficulty in this model is the particular layer of low P-wave velocity which is trapped between two layer of higher P-wave velocity creating thus a waveguide resembling geometry. When the source is located in this particular layer, iterative solvers have difficulties to converge. The discretization is performed for the frequencies 1.25, 2.5, 5 and 7.5 Hz yielding linear systems of size from $3.4\,10^5$ to $1.1\,10^8$ corresponding to a 601$\times$338$\times$152 discretization grid. The complexity analysis of the CARP-CG method yields a total number of iterations for the 3D frequency-domain elastic wave simulation equal to

$$N^{op}_{FD+I} = 43\, N^{3.9}, \quad (\mathrm{CARP-CG}).$$

In 3D, N denotes the geometric mean size of the discrete model given by

$$N = (n_x \times n_y \times n_z)^{1/3},$$

where n_x, n_y and n_z are the number of grid points along the space directions. Using the preconditioner, this computational complexity is reduced to

$$N^{op}_{FD+I} = 95\, N^{3.42}, \quad (\mathrm{PCARP-CG}).$$

By moving the source to a more homogeneous zone which denotes an easier problem to solve, larger reduction in the number of iterations are obtained. Time speedups by a factor up to 2.2 is noted. The preconditioner is built on a operator $\tilde{A}$ using strong damping. The operator $\tilde{A}$ is computed in the media defined by $(V_P, \tilde{V}_S, \rho)$ where $\tilde{V}_S$ is derived from V_P using a low constant Poisson's ratio $\xi_p = 0.15$. This method enables the computation of a preconditioner which provides the best reduction in the number of iterations. These results are very satisfying in comparison with other frequency-domain wave modeling strategies and very encouraging for higher frequencies simulations. The comparison between the different 3D frequency-domain elastic wave modeling methods, i.e. the time-domain modeling with the DFT and the direct solver for the solution (4.1), shows the mains limitations of these approaches. Direct solvers suffer from the tremendous memory requirement issue. Tackling large scale problems still remains out of reach given the current high performance computing clusters despite the latest developments carried out by Gosselin-Cliche and Giroux (2014) using compact stencils to reduce the bandwidth of the matrix A or Wang et al. (2012) using compression schemes to reduce the memory requirements induced by the fill-in. For instance, the sparse parallel direct solver of MUMPS-team (2006) would require more than 54 TB of memory for the 3D elastic Shell model using the $601 \times 338 \times 152$ discretization grid whereas PCARP-CG requires only 514 GB. On the other hand, the time-domain approach with the DFT requires significant computational resources due to the cost of the DFT.

Although the memory request may not be the bottleneck for performing 3D frequency-domain elastic wave simulations using the CARP-CG method, the computation time plays however a key role for considering this iterative method in seismic imaging techniques such as RTM or FWI. Note that the solution of the 3D problem on the Shell model at 7.5 Hz takes approximately 3 hours and 20 minutes using 54 cores. Thus, considering this computation time for thousands of sources shows how time consuming this approach can be (4 months and 18 days for solving this linear system with 1000 sources). Interestingly, the resolution of these linear systems using the PCARP-CG method takes 2 months using the same number of cores (the computation of the preconditioner P and the matrix AP are included in the run time). Significant computation time savings are obtained using the preconditioner. From a theoretical point of view, the comparison of the computational complexities shows that this iterative solver clearly outperforms the time-domain solver. But a fair comparison would require to perform numerical tests using the time-domain solver on a same problem to be able to conclude. However, considering the numerous source simulations inquired by these seismic imaging techniques, this iterative solver can be considered as a serious candidate for the forward simulations.

Perspectives

Several future investigations can be conducted based on this thesis work on both methodological/implementation and application aspects. I only list some of the key perspectives below.

Method/Implementation

This work emphasizes the efficiency and the robustness of the CARP-CG method for the solution of the 3D frequency-domain elastic equations. The preconditioning strategy has enabled significant reduction in the number of iterations allowing to improve the computational complexity of this iterative solver. Although the memory issue is not the main limitation of this approach, memory savings can be achieved by considering compact discretization stencil such as the second-order stencil with optimized coefficient (Gosselin-Cliche and Giroux, 2014). However, the computation time denotes the main bottleneck which prevents considering this iterative solver of one of the most important application which is the seismic imaging area. In this perspective, further investigations on the CARP-CG method have to be carried out.

- The Kaczmarz sweeps through the rows of the matrix A are performed in the natural order, i.e. projections on the hyperplanes associated with the first row and proceed to the last row. It is suggested by Strohmerand and Vershynin (2008) and references therein that the convergence of the Kaczmarz method can be greatly improved by sweeping through the rows of the matrix A in a random order. This randomized Kaczmarz method is proposed with an exponential rate of convergence. Therefore, performing these randomized projections in double sweeps following the same order at each iteration of CARP-CG (the objective is to preserve the hermitian property of the matrix $I - Q(A)$) may enable a faster convergence compared to the standard approach.

- A close attention to the computation time performed by the CARP-CG method on the preconditioned 3D elastic problem (see Table 4.12) allows to see that the point-to-point communication time increases with respect to the number of cores which is expected because of the Component Averaging operation, but the computation time of the forward and backward sweeps remains high due to a denser matrix AP. These observations lead to investigate a second level of parallelism using multi-threads. The point-to-point communication time shall be kept to the minimum thanks to a lower number of cores. The forward and backward sweeps may be accelerated using a multi-thread implementation as the matrix AP is denser. Take for instance the 30-core example in Table 4.12, by using 2 threads per core, the computation time of the forward and backward

Operation	30 cores	40 cores	54 cores
Forward Sweep	5.0	3.6	2.7
Backward Sweep	6.0	4.3	3.2
Double Sweep	12.5	9.5	8.0
Point-to-Point Communication	1.5	1.6	2.1

Table 4.12: Computation time in seconds performed by one forward, backward and double sweep and the point to point communication time performed by the CARP-CG method on the preconditioned 3D frequency-domain elastic problem on the Shell model.

sweeps shall be divided by 2 (ideally) and the communication time remains the same. Therefore, the overall computation time resulting from such implementation may be improved compared to running the CARP-CG method on 60 cores.

- In the perspective of a purely threaded implementation of the CARP-CG method, GPUs can be considered. The latter can be used either as a computing unit replacing the CPU. This is the work of Elble et al. (2010) and Knibbe et al. (2011). Elble et al. (2010) present a comparison between GPU and CPU implementations of several iterative methods including CARP-CG on 9 test problems. The results show that the GPU implementation outperforms the CPU implementation with a substantial reduction of the computation time. However, this framework supposes that the matrix fits in the memory offered by the GPU which is very small (around 4 GB per GPU). Knibbe et al. (2011) compare the CPU and the GPU implementation of the iterative solver BI-CGSTAB preconditioned with the shifted Laplace preconditioner on the 2D and 3D Helmholtz equation. Similar results with the previous referenced work are obtained. Alternatively, GPUs can be used as accelerators which means that the problem is solved on the CPU while the some computational intensive parts of the algorithm are performed by the GPUs. This approach requires that the data are transferred to and from the GPU for each task. However, the main problem is the intensive data transfer which is limited by the small bandwidth of the computing architectures Knibbe et al. (2014). Therefore, the possible GPU implementation of the CARP-CG method may not be the best fit for the 3D elastic equations from my point of view. The memory bottleneck is even more emphasized.

- One interesting direction of research is the devoted work of Petrenko (2014); Petrenko et al. (2014) to optimize the implementation of the Kaczmarz sweeps. The use of reconfigurable hardware accelerator FPGA (field programmable gate arrays) is investigated. This approach is similar to GPU accelerators but offers larger memory resources. Applications on the 3D Helmholtz problem using the CGMN method are performed and an improvement by a factor 2 in the computation time is observed when running the CGMN method on the FPGA accelerator.

- Finally, multiple-right hand side accelerations (Gutknecht, 2006) are one of the main advantages of using CARP-CG in comparison to the time-domain approach. The "single seed" method (Chan and Wan, 1997) which is a projection-based method (Smith et al., 1989), is one possible solution although no theoretical proof have been made for the improvement of the convergence rate. The single seed method is tested on the 2D acoustic problem (the 5 Hz acoustic problem on the BP2004 model is considered, see Chapter 3 Section 3.1.1) with 1000 right-hand sides (Hamitou et al., 2015) (see Appendix B). A reduction in the number of iterations and the computation time by a factor 1.5 is obtained. Further investigations can be driven using the block conjugate gradient method (O'Leary, 1980). With N_{RHS} right-hand sides, the block CG method may converge, theoretically, with N_{RHS} times fewer number of iterations compared to a sequential approach where each linear system is solved independently.

Improvements for the computation of the preconditioner can be made as well. Substantial accelerations can be achieved with a second level of parallelism for the computation of the preconditioner. For the 3D frequency-domain elastic problem, three small linear systems are solved for each discretization point. Using the same local matrix, three solutions are computed for three force excitations: (1) along the x direction, (2) along the y direction and (3) along the z direction. As these solutions can be computed independently, a multi-thread implementation seems very appropriate to achieve a speedup by a factor 3. Further investigations take advantage of the similarities of the three wavefields computed for each excitation. For a given local medium, one can use the first computed wavefield (along the x direction for instance) and by a simple copy/rotation, avoid the computation of the two extra solutions along the y and z directions. Such strategy enables computational resource savings.

Applications

This work enables the preconditioning of the iterative solver CARP-CG for the solution of the 3D frequency-domain elastic wave equations. So far, the 3D heterogeneous model provided by Shell is tested. This case study can be further investigated with different configurations (free-surface). Several numerical applications can be carried out as well on other 3D realistic models. In the following, some possible applications are presented.

- 3D elastic wave modeling with a Free-Surface: realistic seismic wave modeling accounts for the Free-Surface. The positive results obtained in the 2D elastic approximation on the Marmousi 2 case study with the free-surface boundary condition (see Chapter 3 Section 3.2.2.3) should lead to investigate the free-surface effects in the 3D elastic approximation. Larger reduction in the number of iterations and thus in the computational complexity is expected.

- 3D SEG/EAGE Overthrust model: the 20 km × 20 km × 4.65 km 3D acoustic onshore model presents complex sedimentary succession similar to the model which is investigated here (see Figure 4.13). One particular difference is the presence of faults and more complex heterogeneities at the surface of the model. For elastic wave modeling, the V_S model can be build from the V_P model using a constant Poisson's ratio.

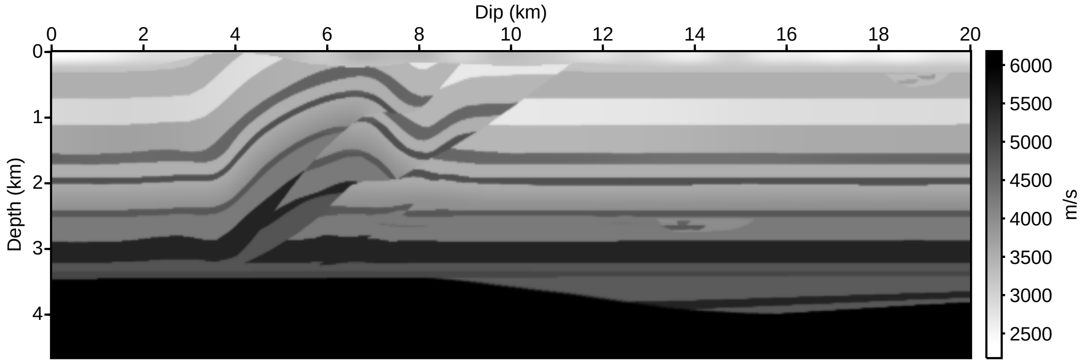


Figure 4.13: Slice of the Overthrust SEG/EAGE P-wave velocity model.

- 3D Valhall model: the 18 km × 11 km × 5 km elastic model provides less heterogeneities compared to Shell's model (see Figure 4.14). This model is interesting for its 70 m water layer and the smaller range of variation for the P-wave velocity with $V_P^{max} = 3200$ m/s and $V_P^{min} = 1480$ m/s which denotes less heterogeneities but high contrasts. A larger panel for the shear wave velocity model $V_S^{max} = 1594$ m/s and $V_S^{min} = 379$ m/s denotes large Poisson's ratio. This model is investigated in the acoustic approximation for the FWI application performed by (Operto et al., 2015).

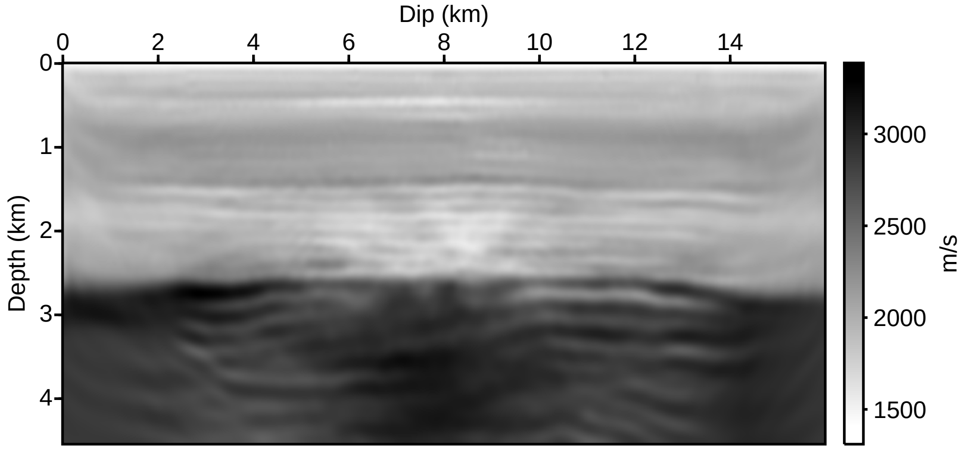


Figure 4.14: Slice of the Valhall P-wave velocity model.

- 3D Valhall model with anisotropy: Perhaps one interesting application is to look at the influence of anisotropy in the medium on the convergence of the CARP-CG with and without preconditioning. The fourth-order stencil of Li et al. (2016) provides the possibility to perform the elastic wave modeling in the frequency-domain for orthorhombic media. Numerical experiments can be carried out straightforwardly.

Bibliography

Aki, K. and Richards, P. (1980). *Quantitative Seismology: Theory and Methods*. W. H. Freeman & Co, San Francisco.

Aki, K. and Richards, P. G. (2002). *Quantitative seismology, theory and methods, second edition*. University Science Books, Sausalito,California.

Amestoy, P., Brossier, R., Buttari, A., L'Excellent, J.-Y., Mary, T., Métivier, L., Miniussi, A., Operto, S., Ribodetti, A., Virieux, J., and Weisbecker, C. (2015a). Efficient 3D frequency-domain full-waveform inversion of ocean-bottom cable data with sparse block low-rank direct solver: a real data case study from the North Sea. In *Expanded Abstracts, 85^{th} Annual SEG Meeting (New Orleans)*, pages 1303–1308. SEG.

Amestoy, P., Duff, I. S., and L'Excellent, J. Y. (2000a). Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4):501–520.

Amestoy, P. R., Ashcraft, C., Boiteau, O., Buttari, A., L'Excellent, J.-Y., and Weisbecker, C. (2015b). Improving multifrontal methods by means of block low-rank representations. *SIAM Journal on Scientific Computing*, 37(3):1451–1474.

Amestoy, P. R., Duff, I. S., Koster, J., and L'Excellent, J. Y. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23(1):15–41.

Amestoy, P. R., Duff, I. S., and L'Excellent, J. Y. (2000b). Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184:501–520.

Aruliah, D. A. and Ascher, U. M. (2003). Multigrid preconditioning for krylov methods for time-harmonic maxwell's equations in 3D. *SIAM Journal on Scientific Computing*, 24:702–718.

Bayliss, A., Goldstein, C. I., and Turkel, E. (1983). An iterative method for the Helmholtz equation. *Journal of Computational Physics*, 49(3):443–457.

Benzi, M. and Tuma, M. (1998). A Sparse Approximate Inverse Preconditioner for Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing*, 19(3):968–994.

Bérenger, J.-P. (1994). A perfectly matched layer for absorption of electromagnetic waves. *Journal of Computational Physics*, 114:185–200.

Billette, F. J. and Brandsberg-Dahl, S. (2004). The 2004 BP velocity benchmark. In *Extended Abstracts, 67th Annual EAGE Conference & Exhibition, Madrid, Spain*, page B035.

Björck, Å. and Elfving, T. (1979). Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT Numerical Mathematics*, 19(2):145–163.

Boman, E. G., Catalyurek, U. V., Chevalier, C., and Devine, K. D. (2012). The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering, and coloring. *Scientific Programming*, 20(2):129–150.

Brekhovskikh, L. M. and Godin, O. A. (1990). *Acoustic of layered media I : plane and quasi-plane waves.* Springer.

Brekhovskikh, L. M. and Godin, O. A. (1992). *Acoustic of layered media II : point sources and bounded beams.* Springer.

Brossier, R., Etienne, V., Operto, S., and Virieux, J. (2010). Frequency-domain numerical modelling of visco-acoustic waves based on finite-difference and finite-element discontinuous galerkin methods. In Dissanayake, D. W., editor, *Acoustic Waves*, pages 125–158. SCIYO.

Brossier, R., Pajot, B., Combe, L., Operto, S., Métivier, L., and Virieux, J. (2014a). Time and frequency-domain FWI implementations based on time solver: analysis of computational complexities. In *Expanded Abstracts, 76th Annual EAGE Meeting (Amsterdam).*

Brossier, R., Pajot, B., Combe, L., Operto, S., Métivier, L., and Virieux, J. (2014b). Time and Frequency-domain FWI implementations based on time solver: analysis of computational complexities. In *Expanded Abstracts*. EAGE.

Brossier, R., Virieux, J., and Operto, S. (2008). Parsimonious finite-volume frequency-domain method for 2-D P-SV-wave modelling. *Geophysical Journal International*, 175(2):541–559.

Bube, K. P. and Langan, R. T. (1997). Hybrid l_1/l_2 minimization with applications to tomography. *Geophysics*, 62(4):1183–1195.

Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M. (1985). A nonreflecting boundary condition for discrete acoustic and elastic wave equations. *Geophysics*, 50(4):2117–2131.

Chan, T. F. and Ng, M. K. (1999). Galerkin projection methods for solving multiple linear systems. *SIAM Journal on Scientific Computing*, 21(3):836–850.

Chan, T. F. and Wan, W. L. (1997). Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM J. Sci. Comput.*

Chapman, C. (2004). *Fundamentals of seismic waves propagation.* Cambridge University Press, Cambridge, England.

Clapp, R. (2008). Reverse time migration: Saving the boundaries. Technical Report SEP-136, Stanford Exploration Project.

Clayton, R. and Engquist, B. (1977). Absorbing boundary conditions for acoustic and elastic wave equations. *Bulletin of the Seismological Society of America*, 67:1529–1540.

Courant, R., Friedrichs, K., and Lewy, H. (1928). Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74.

Demko, S., Moss, W. F., and Smith, P. W. (1984). Decay rates for inverses of band matrices. *Mathematics of Computation*, 43(168):491–499.

Dessa, J. X., Operto, S., Kodaira, S., Nakanishi, A., Pascal, G., Virieux, J., and Kaneda, Y. (2004). Multiscale seismic imaging of the eastern nankai trough by full waveform inversion. *Geophysical Research Letters*, 31(L18606):doi:10.1029/2004GL020453.

Dormy, E. and Tarantola, A. (1995). Numerical simulation of elastic wave propagation using a finite volume method. *Journal of Geophysical Research*, 100:2123–2133.

Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). *Direct methods for sparse matrices*. Clarendon Press, Oxford, U. K.

Duff, I. S. and Reid, J. K. (1983). The multifrontal solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, 9:302–325.

Elble, J. M., Sahinidis, N. V., and Vouzis, P. (2010). GPU computing with kaczmarz's and other iterative algorithms for linear systems. *Parallel Computing*, 36(5-6):215–231.

Engquist, B., Poulson, J., and Ying, L. (2011). Sweeping preconditioner for the 3D Helmholtz equation. *SEG Technical Program Expanded Abstracts*, pages 3158–3163.

Engquist, B. and Ying, L. (2011). Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. *Communications on pure and applied mathematics*, 64(5):697–735.

Erlangga, Y. A. (2005). *A robust and efficient iterative method for the numerical solution of the Helmholtz equation*. PhD thesis, Delft University of Technology.

Erlangga, Y. A. and Nabben, R. (2008a). Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetrix matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(2):684–699.

Erlangga, Y. A. and Nabben, R. (2008b). Multilevel projection-based nested krylov iteration for boundary value problems. *SIAM Journal on Scientific Computing*, 30(3):1572–1595.

Erlangga, Y. A. and Nabben, R. (2008c). On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian. *Electronic Transactions on Numerical Analysis*, 31:403–424.

Erlangga, Y. A., Oosterleeand, C., and Vuik, C. (2006a). A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM Journal on Scientific Computing*, 27:1471–1492.

Erlangga, Y. A., Vuik, C., and Oosterlee, C. W. (2006b). Comparison of multigrid and Incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Applied Numerical Mathematics*, 56(5):648–666.

Erlangga, Y. A., Vuik, C., and Osterlee, C. W. (2003). On a class of preconditioners for solving the helmholz equation. Reports of the Departement of Applied Mathematical Analysis report 03-01, Delft University of Technology.

Erlangga, Y. A., Vuik, C., and Osterlee, C. W. (2004). On a class of preconditioners for solving the helmholtz equation. *Applied Numerical Mathematics*, 50:409–425.

Ernst, O. G. and Gander, M. J. (2012). Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods. In Graham, I. G., Hou, T. Y., Lakkis, O., and Scheichl, R., editors, *Numerical Analysis of Multiscale Problems*, volume 83 of *Lecture Notes in Computational Science and Engineering*, pages 325–363. Springer Berlin Heidelberg.

Fletcher, R. (1976). *Conjugate gradient methods for indefinite systems*, pages 73–89. Springer Berlin Heidelberg.

Futterman, W. (1962). Dispersive body waves. *Journal Geophysical Research*, 67:5279–5291.

Gander, M. J., Graham, I. G., and Spence, E. A. (2015). Applying gmres to the helmholtz equation with shifted laplacian preconditioning: what is the largest shift for which wavenumber-independent convergence is guaranteed? *Numerische Mathematik*, 131(3):567–614.

George, A. and Liu, J. W. (1981). *Computer solution of large sparse positive definite systems*. Prentice-Hall, Inc.

Golub, G. H. (1996). *Matrix Computation, third edition*. Johns Hopkins Studies in Mathematical Sciences.

Gordon, D. and Gordon, R. (2005). Component-averaged row projections: a robust block-parallel scheme for sparse linear systems. *SIAM Journal on Scientific Computing*, 27(3):1092–1117.

Gordon, D. and Gordon, R. (2010a). CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs. *Parallel Computing*, 36(9):495–515.

Gordon, D. and Gordon, R. (2010b). CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs. *Parallel Computing*, 36:495–515.

Gordon, D. and Gordon, R. (2010c). Row scaling as a preconditioner for some nonsymmetric linear systems with discontinuous coefficients. *Journal of Computational and Applied Mathematics*, 234(12):3480–3495.

Gordon, D. and Gordon, R. (2013). Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers. *Journal of Computational and Applied Mathematics*, 237(1):182–196.

Gosselin-Cliche, B. and Giroux, B. (2014). 3D frequency-domain finite-difference viscoelastic-wave modeling using weighted average 27-point operators with optimal coefficients. *Geophysics*, 79(3):T169–T188.

Griewank, A. (1992). Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and software*, 1(1):35–54.

Gutknecht, M. H. (2006). Block Krylov space methods for linear systems with multiple right-hand sides: an introduction.

Haidar, A. (2008). *On the parallel scalability of hybrid linear solvers for large 3D problems*. PhD thesis, Institut National Polytechnique de Toulouse - CERFACS TH/PA/08/57.

Hamitou, O., Métivier, L., Labbé, S., Brossier, R., and Virieux, J. (2015). Preconditioning and multiple-right hand sides strategies for the solution of the frequency-domain wave propagation problem using the CGMN method. In *SEG Technical Program Expanded Abstracts 2015*, pages 3612–3616.

Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradient for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436.

Kaczmarz, S. (1937). Angenäherte Auflösung von Systemen linearer Gleichungen (English translation by Jason Stockmann). *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, 35:355–357.

Karypis, G. and Kumar, V. (1999). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359 – 392.

Knibbe, H., Oosterlee, C., and Vuik, C. (2011). GPU implementation of a Helmholtz krylov solver preconditioned by a shifted Laplace multigrid method. *Journal of Computational and Applied Mathematics*, 236(3):281–293.

Knibbe, H., Oosterlee, C., and Vuik, C. (2014). Closing the performance gap between an iterative frequency-domain solver and an explicit time-domain scheme for 3D migration on parallel architectures. *Geophysics*, 79(2):547–561.

Kolsky, H. (1956). The propagation of stress pulses in viscoelastic solids. *Philosophical Magazine*, 1:693–710.

Laird, A. and Giles, M. (2002). Preconditioned iterative solution of the 2D Helmholtz equation. *St. Hugh's College Oxford University*.

Levander, A. R. (1988). Fourth-order finite-difference P-SV seismograms. *Geophysics*, 53(11):1425–1436.

Li, Y., Han, B., Métivier, L., and Brossier, R. (2016). Optimal fourth-order staggered-grid finite-difference scheme for 3D frequency-domain viscoelastic wave modeling. *Journal of Computational Physics*, in press.

Li, Y., Métivier, L., Brossier, R., Han, B., and Virieux, J. (2015). 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver. *Geophysics*, 80(3):T101–T118.

Luo, Y. and Schuster, G. T. (1990). Parsimonious staggered grid finite-differencing of the wave equation. *Geophysical Research Letters*, 17(2):155–158.

Madariaga, R. (1976). Dynamics of an expanding circular fault. *Bulletin of the Seismological Society of America*, 66:639–666.

Made, M. M., Beauwens, R., and Warzée, G. (2000). Preconditioning of discrete helmholtz operators perturbed by a diagonal complex matrix. *Communications in numerical methods in engineering*, 16:801–817.

Marfurt, K. (1984). Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations. *Geophysics*, 49:533–549.

Martin, G. S., Wiley, R., and Marfurt, K. J. (2006). Marmousi2: An elastic upgrade for Marmousi. *The Leading Edge*, 25(2):156–166.

Métivier, L., Brossier, R., Mérigot, Q., Oudet, E., and Virieux, J. (2016). Measuring the misfit between seismograms using an optimal transport distance: Application to full waveform inversion. *Geophysical Journal International*, 205:345–377.

Mittet, R. (2002). Free-surface boundary conditions for elastic staggered-grid modeling schemes. *Geophysics*, 67(5):1616–1623.

Moczo, P. (1989). Finite-difference technique for SH waves in 2D media using irregular grids: application to the seismic response problem. *Geophysical Journal International*, 99:321–329.

Moczo, P., Ampuero, J. P., Kristek, J., Galis, M., Day, S. M., and Igel, H. (2005). The European Network SPICE Code Validation. In *EOS Trans. AGU, abstract S13A-0180*. American Geophysical Union, San Francisco, USA.

Moczo, P., Kristek, J., Galis, M., Pazak, P., and Balazovjech, M. (2007). The finite-difference and finite-element modeling of seismic wave propagation and earthquake motion. *Acta Physica Slovaca*, 52(2):177–406.

Mulder, W. A. (2006). A multigrid solver for 3d electromagnetic diffusion. *Geophysical Prospecting*, 54:633–649.

MUMPS-team (2006). *MUMPS - MUltifrontal Massively Parallel Solver users' guide - version 4.6.3 (February 2006)*. ENSEEIHT-ENS Lyon, http://www.enseeiht.fr/apo/MUMPS/ or http://graal.ens-lyon.fr/MUMPS.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition.

O'Leary, D. P. (1980). The block conjugate gradient algorithm and related methods. *Linear algebra and its applications*, 29:293–322.

Operto, S., Brossier, R., Combe, L., Métivier, L., Ribodetti, A., and Virieux, J. (2014). Computationally-efficient three-dimensional visco-acoustic finite-difference frequency-domain seismic modeling in vertical transversely isotropic media with sparse direct solver. *Geophysics*, 79(5):T257–T275.

Operto, S., Miniussi, A., Brossier, R., Combe, L., Métivier, L., Monteiller, V., Ribodetti, A., and Virieux, J. (2015). Efficient 3-D frequency-domain mono-parameter full-waveform inversion of ocean-bottom cable data: application to Valhall in the visco-acoustic vertical transverse isotropic approximation. *Geophysical Journal International*, 202(2):1362–1391.

Operto, S., Virieux, J., Dessa, J. X., and Pascal, G. (2006). Crustal imaging from multifold ocean bottom seismometers data by frequency-domain full-waveform tomography: application to the eastern Nankai trough. *Journal of Geophysical Research*, 111(B09306):doi:10.1029/2005JB003835.

Osei-Kuffor, D. and Saad, Y. (2013). Preconditioning helmholtz linear systems. *Applied Numerical Mathematics*, 60:420–431.

Osei-Kuffuor, D. and Saad, Y. (2010). Preconditioning Helmholtz linear systems. *Applied Numerical Mathematics*, 60(4):420–431. Special Issue: {NUMAN} 2008.

Pajot, B., Li, Y., Berthoumieux, V., Weisbecker, C., Brossier, R., Métivier, L., Thierry, P., Operto, S., and Virieux, J. (2014). A review of recent forward problem developments used for frequency-domain fwi. In *Expanded Abstracts, 76*th *Annual EAGE Meeting (Amsterdam)*.

Pan, G., Liang, L., and Habashy, T. (2015). *Development of Three-dimensional Frequency-Domain Elastic Full-Waveform Inversion*, chapter 234, pages 1209–1214. Society of Exploration Geophysicists.

Petrenko, A. (2014). *Accelerating an iterative Helmholtz solver using reconfigurable hardware*. PhD thesis, University of British Columbia.

Petrenko, A., van Leeuwen, T., Oriato, D., Tilbury, S., and Herrmann, F. J. (2014). Accelerating an iterative Helmholtz solver with FPGAs. In *EAGE Annual Conference Proceedings*.

Plessix, R. E. (2006a). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503.

Plessix, R. E. (2006b). A robust three-dimensional iterative solver for the time-harmonic wave equation. In *Extended Abstracts*. Eur. Ass. Expl.Geophys.

Plessix, R. E. (2007). A Helmholtz iterative solver for 3D seismic-imaging problems. *Geophysics*, 72(5):SM185–SM194.

Plessix, R. E. (2009). Three-dimensional frequency-domain full-waveform inversion with an iterative solver. *Geophysics*, 74(6):WCC53–WCC61.

Plessix, R.-E., Baeten, G., de Maag, J. W., and ten Kroode, F. (2012). Full waveform inversion and distance separated simultaneous sweeping: a study with a land seismic data set. *Geophysical Prospecting*, 60:733 – 747.

Plessix, R. E. and Perkins, C. (2009). 3D full-waveform inversion with a frequency-domain iterative solver. In *Expanded Abstracts*. European Association of Geoscientists & Engineers.

Poulson, J., Engquist, B., Li, S., and Ying, L. (2013). A parallel sweeping preconditioner for heterogeneous 3d helmholtz equations. *SIAM Journal on Scientific Computing*, 35(3):C194–C212.

Pratt, R. G., Shin, C., and Hicks, G. J. (1998). Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. *Geophysical Journal International*, 133:341–362.

Pratt, R. G. and Worthington, M. H. (1990). Inverse theory applied to multi-source cross-hole tomography. Part I: acoustic wave-equation method. *Geophysical Prospecting*, 38:287–310.

Reddy, J. N. (2002). *Energy principles and variational methods in applied mechanics*. John Wiley & Sons.

Riyanti, C. D., Erlangga, Y. A., Plessix, R. E., Mulder, W. A., Vuik, C., and Oosterlee, C. (2006). A new iterative solver for the time-harmonic wave equation. *Geophysics*, 71(E):57–63.

Rizzuti, G. and Mulder, W. (2016). Multigrid-based shifted-laplacian preconditioning for the time-harmonic elastic wave equation. *Journal of Computational Physics*, 317:47–65.

Saad, Y. (1986). GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869.

Saad, Y. (1994). Ilut: A dual threshold incomplete lu factorization. *Numerical Linear Algebra with Applications*, 1(4):387–402.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM, Philadelphia.

Shin, C., Jang, S., and Min, D. J. (2001). Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophysical Prospecting*, 49:592–606.

Sirgue, L., Etgen, J. T., and Albertin, U. (2008). 3D Frequency Domain Waveform Inversion using Time Domain Finite Difference Methods. In *Proceedings 70th EAGE, Conference and Exhibition, Roma, Italy*, page F022.

Sirgue, L. and Pratt, R. G. (2004). Efficient waveform inversion and imaging : a strategy for selecting temporal frequencies. *Geophysics*, 69(1):231–248.

Sleijpen, G. L. G. and Fokkema, D. R. (1993). BiCGstab(L) for linear equations involving unsymmetric matrices with complex spectrum. *Electron. Trans. Numer. Anal.*, 1(Sept.).

Smith, B. F., Bjrstad, P. E., and Gropp, W. (1996). *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press.

Smith, C. F., Peterson, A. F., and Mittra, R. (1989). A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields. *Antennas and Propagation, IEEE Transactions on*, 37(11):1490–1493.

Sonneveld, P. (1989). Cgs, a fast lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10(1):36–52.

Sourbier, F., Haiddar, A., Giraud, L., Ben-Hadj-Ali, H., Operto, S., and Virieux, J. (2011). Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct/iterative solver. *Geophysical Prospecting*, 59(5):834–856.

Strohmerand, T. and Vershynin, R. (2008). A Randomized Kaczmarz Algorithm with Exponential Convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278.

Symes, W. W. (2007). Reverse time migration with optimal checkpointing. *Geophysics*, 72(5):SM213–SM221.

Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.

Tarantola, A. (1987). *Inverse problem theory: methods for data fitting and model parameter estimation*. Elsevier, New York.

Van der Vorst, H. A. (1992). BI-CGSTAB: a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13:631–644.

van Leeuwen, T. and Herrmann, F. J. (2014). 3D frequency-domain seismic inversion with controlled sloppiness. *SIAM Journal on Scientific Computing*, 36(5):S192–S217.

Versteeg, R. J. and Grau, G., editors (1991). *The Marmousi experience*. Proceedings of the 1990 EAEG workshop on Practical Aspects of Seismic Data Inversion, Eur. Ass. Expl. Geophys.

Virieux, J. (1984). SH wave propagation in heterogeneous media: Velocity-stress finite difference method. *Geophysics*, 49:1259–1266.

Virieux, J. (1986). P-SV wave propagation in heterogeneous media: Velocity-stress finite difference method. *Geophysics*, 51:889–901.

Virieux, J. and Operto, S. (2009). An overview of full waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.

Virieux, J., Operto, S., Ben Hadj Ali, H., Brossier, R., Etienne, V., Sourbier, F., Giraud, L., and Haidar, A. (2009). Seismic wave modeling for seismic imaging. *The Leading Edge*, 28(5):538–544.

Voigt, W. (1889). Uber die beziehung zwischen den beiden elastizitatkonstanten isotroper körper. *Annalen der Physik*, 38:573–587.

Wang, S., de Hoop, M. V., Xia, J., and Li, X. S. (2012). Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-d anisotropic media. *Geophysical Journal International*, 191:346–366.

Warner, M., Nangoo, T., Shah, N., Umpleby, A., and Morgan, J. (2013). *Full-waveform inversion of cycle-skipped seismic data by frequency down-shifting*, chapter 176, pages 903–907. Society of Exploration Geophysics.

Weisbecker, C., Amestoy, P., Boiteau, O., Brossier, R., Buttari, A., L'Excellent, J.-Y., Operto, S., and Virieux, J. (2013). 3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver. In *Society of Exploration Geophysics Technical Program Expanded Abstracts*, pages 3411–3416.

Appendix A

Notations

I introduce in this annexe the relevant concepts in linear algebra which are useful in the chapters of this thesis. I give an overview of the basic vector and matrix theory and introduce some basic notations which are used later. The reader may refer to Golub (1996) for relevant proofs.

Vectors and matrices

We denote by $\mathbb{K}$ the real $\mathbb{R}$ or complex $\mathbb{C}$ corps. For all $z \in \mathbb{C}$, $|z|$ denotes the modulus of the complex number z and $\bar{z}$ denotes the complex conjugate number of z. We denote by i the purely imaginary complex number such that $i^2 = -1$.

For n and m in $\mathbb{N} \setminus \{0\}$, we have

- $\mathbb{K}^n$ is the $\mathbb{K}$-vector space of $(z_1, z_2, \cdots, z_n)$ with $z_j \in \mathbb{K}$ for $j = 1, \cdots, n$,
- $\mathcal{M}_{n,m}(\mathbb{K})$ is the corp of matrices with n rows and m columns and whose coefficients are in $\mathbb{K}$. For simplicity, the corp of square matrices is denoted by $\mathcal{M}_n(\mathbb{K}) = \mathcal{M}_{n,n}(\mathbb{K})$.

$\mathbb{K}^n$ is identified to $M_{n,1}(\mathbb{K})$. Therefore, a vector writes as a single column matrix. For $A \in \mathcal{M}_{n,m}(\mathbb{K})$, the notation $A = (a_{ij})_{1 \leq i \leqslant n,\ 1 \leqslant j \leq m}$ is used when the coefficients of the matrix A need to be mentioned. The notation $\mathbf{a}_{\bullet j}$ refers to the j-th row vector of A

$$\mathbf{a}_{\bullet j} = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{pmatrix}.$$

The same way, the vector $\mathbf{a}_{i\bullet}$ denotes the i-th column vector of A

$$\mathbf{a}_{i\bullet} = (a_{i1}, a_{i2}, \cdots, a_{im}).$$

We denote by $0_n \in \mathcal{M}_n(\mathbb{K})$ the identity element for the addition in the matrix corp $\mathcal{M}_n(\mathbb{K})$. For any matrix $A \in \mathcal{M}_n(\mathbb{K})$

$$A + 0_n = 0_n + A = A.$$

Similarly, $I \in \mathcal{M}_n(\mathbb{K})$ the identity element for the multiplication in the matrix corp $\mathcal{M}_n(\mathbb{K})$. For any matrix $A \in \mathcal{M}_n(\mathbb{K})$

$$AI = IA = A.$$

For all $A \in \mathcal{M}_n(\mathbb{C})$, A^T is the transposed matrix of A and $A^H = \bar{A}^T$ the complex conjugate transposed matrix of A, that is, $a_{ij}^H = \bar{a}_{ji}$.

Definition 1 (Upper (respectively lower) triangular matrices)**.** *The matrix $A \in \mathcal{M}_n(\mathbb{K})$ is called upper (respectively lower) triangular if*

$$a_{ij} = 0, \textit{ pour } i > j, \textit{ (respectivement } i < j).$$

Definition 2 (Banded matrices)**.** *The matrix $A \in \mathcal{M}_n(\mathbb{K})$ is called banded if*

$$a_{ij} \neq 0, \textit{ only if } i - m_l \leqslant j \leqslant i + m_u,$$

where m_l and m_u are two positive integers. The number $m_l + m_u$ is then called the bandwidth of A.

Definition 3 ((Strictly) diagonally dominant matrices)**.** *The matrix $A \in \mathcal{M}_n(\mathbb{K})$ is diagonally dominant if*

$$|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n$$

strictly dominant if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n$$

Definition 4 (Trace of a matrix)**.** *The trace $\mathtt{tr}(A)$ of a matrix $A \in \mathcal{M}_n(\mathbb{K})$ is the scalar*

$$\mathtt{tr}(A) = \sum_{i=1}^{n} a_{ij}.$$

Definition 5 (Eigenvalues)**.** *A scalar $\lambda \in \mathbb{C}$ is called an eigenvalue of a matrix $A \in \mathcal{M}_n(\mathbb{C})$ if a non zeros vector $u \in \mathbb{C}^n$ exists such that $Au = \lambda u$. The vector u is called an eigenvector of A associated to λ. The set of all the eigenvalues of A is then called the spectrum of A and is denoted by $\sigma(A)$. The spectral radius of A is defined by*

$$\rho(A) = \max\{|\lambda|;\ \lambda \in \sigma(A)\}.$$

Vector inner products and norms

An inner product on a space vector $\mathbb{K}^n$ is any bi-linear, hermitian, positive definite mapping from $\mathbb{K}^n \times \mathbb{K}^n$ into $\mathbb{K}$.

Definition 6 (Euclidean inner product)**.** *The Euclidean inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{C}^n$ is defined by*

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i \bar{y}_i, \tag{A.1}$$

which writes in matrix form as

$$(\mathbf{x},\mathbf{y}) = \mathbf{y}^H\mathbf{x}.$$

The Euclidean inner product in $\mathbb{C}^n$ *reduced to*

$$(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} x_i y_i, \tag{A.2}$$

which writes in matrix form as

$$(\mathbf{x},\mathbf{y}) = \mathbf{y}^T\mathbf{x}.$$

Definition 7 (Symmetric, hermitian matrices)**.** *In the sens of the scalar product* $(\cdot,\cdot)$*, the matrix* $A \in \mathcal{M}_n(\mathbb{R})$ *is called symmetric if* $A^T = A$

$$(\mathbf{x}, A\mathbf{y}) = (A\mathbf{x},\mathbf{y}) \Longleftrightarrow \mathbf{y}^T A^T \mathbf{x} = \mathbf{y}^T A\mathbf{x}.$$

Similarly, for $A \in \mathcal{M}_n(\mathbb{C})$*, it is called hermitian or self-adjoint if* $A^H = A$*.*

Definition 8 (Orthogonality)**.** *Two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *of* $\mathbb{K}^n$ *are said orthogonal if*

$$(\mathbf{x},\mathbf{y}) = 0.$$

Definition 9 (Norms de $\mathbb{K}^n$)**.** *The norms* $\|\cdot\|_p$ *associated to the Euclidean vector inner product are defined by*

$$\|\mathbf{x}\|_p = (\mathbf{x},\mathbf{x})^{\frac{1}{p}} = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, \quad p \in \{1,2,\infty\}. \tag{A.3}$$

Therefore, we have

$$\begin{aligned} \|\mathbf{x}\|_1 &= \sum_{i=1}^{n} |x_i|, \\ \|\mathbf{x}\|_2 &= \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}, \\ \|\mathbf{x}\|_\infty &= \max_{i=1,\ldots,n} |x_i|. \end{aligned}$$

Definition 10 (Matrix norms)**.** *For* $A \in \mathcal{M}_n(\mathbb{K})$*, the p-norm of A is defined by*

$$\|A\|_p = \max_{\mathbf{x}\in\mathbb{K}^n,\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}. \tag{A.4}$$

The usual matrix norms are

$$\begin{aligned} \|A\|_1 &= \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{ij}|, \\ \|A\|_2 &= \rho\left(A^H A\right)^{\frac{1}{2}}, \\ \|A\|_\infty &= \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{ij}|, \\ \|A\|_F &= \left(\mathtt{tr}(A^H A)\right)^{\frac{1}{2}}. \end{aligned}$$

$\|A\|_F$ is called the Frobenius norm.

Definition 11 (Unitary matrices)**.** *The matrix $A \in \mathcal{M}_n(\mathbb{K})$ is called unitary if $A^H A = AA^H = I$. Then $A^{-1} = A^H$*

Condition number

Definition 12 (Condition number)**.** *Given the matrix norm $\|\cdot\|$ (see definition 10), the condition number of a matrix $A \in \mathcal{G}l_n(\mathbb{K})$ with respect to the norm $\|\cdot\|$ is defined by*

$$\texttt{cond}(A) = \|A\|\|A^{-1}\|.$$

For any matrix $A \in \mathcal{G}l_n(\mathbb{K})$, let $\mu_1 \leqslant \cdots \leqslant \mu_n$ be the singular values of A, then

$$\texttt{cond}(A) = \frac{\mu_n}{\mu_1}.$$

If the matrix A is hermitian, then

$$\texttt{cond}(A) = \frac{|\lambda_n|}{|\lambda_1|},$$

where $|\lambda_1|$ and $|\lambda_n$ are respectively the smallest and largest eigenvalue with respect to the module $|\cdot|$ of A.

Appendix B

Preconditioning and multiple-right hand sides strategies for the solution of the frequency-domain wave propagation problem using the CGMN method

This appendix contains a supplementary publication, a conference expanded abstract. This work was presented at the SEG conference in 2015. The preconditioning strategy for the CGMN method adapted to the 2D frequency-domain acoustic wave equation is presented on the BP2004 model. This part of the conference paper is presented in the first part of chapter 3. In addition, a multiple right-hand side acceleration called the single seed method (Chan and Wan, 1997) is investigated. An interesting extension to this work would be to perform a comparison with the block Conjugate Gradient method (O'Leary, 1980).

Preconditioning and multiple-right hand sides strategies for the solution of the frequency-domain wave propagation problem using the CGMN method

Okba Hamitou*,†,‡*, Ludovic Métivier*†,‡*, Stéphane Labbé*‡*, Romain Brossier*† *and Jean Virieux*†
**Speaker;*
†*ISTerre, University Grenoble Alpes;*
‡*LJK, University Grenoble Alpes*

SUMMARY

Frequency-domain waveform modeling in the acoustic and elastic approximations requires the solution of large ill-conditioned linear systems. In the context of frequency-domain full waveform inversion, the solutions of these systems are required for a large number of sources (i.e. right-hand sides). Because of their tremendous memory requirements, direct solvers are not yet adapted to the solution of 3D elastodynamics equations. We are thus interested in the use of efficient iterative solvers adapted to the solution of these systems. The CGMN method has shown robust convergence properties for 2D and 3D elastic problems in highly heterogeneous media, compared to standard Krylov methods, but still requires a large number of iterations to reach sufficient accuracy. In this study, the design of an efficient preconditioning strategy adapted to this method is investigated. This preconditioner is computed as a sparse approximate inverse of a heavily damped wave propagation operator. In addition, the single seed method is used to increase the efficiency of the solver for multiple right-hand sides. The efficiency of these two combined strategies is evaluated on the 2D BP2004 model in the visco-acoustic approximation, up to 40 Hz. An overall time speed-up equal to 3 and a reduction of the number of iterations by a factor 10 are observed.

INTRODUCTION

Solving 3D frequency-domain acoustic and elastodynamics equation is a highly challenging problem. After discretization, these problems amount to the solution of a large, sparse, ill-conditioned and non-positive linear systems. These difficulties are magnified by the high contrasts and discontinuities in the physical properties of the subsurface. In the context of frequency-domain Full Waveform Inversion (FWI), the solution of the wave propagation problem is needed for a large number of sources (up to several tens of thousands for realistic applications). In the acoustic approximation, direct methods can be used. The factorization of the matrix is performed once and the solutions of the linear systems relative to each right-hand side can be obtained efficiently by forward and backward substitutions. Current FWI studies demonstrate the ability of using direct solvers in the 3D acoustic approximation for reasonable target size (Operto et al., 2014, 2015). However, the high memory requirement of these methods, due to the fill-in associated with the matrix factorization, makes them still inappropriate for solving 3D elastic problems.

In the perspective of performing 3D frequency-domain elastodynamics FWI, two strategies could thus be adopted. The first consists in solving the elastodynamics in the time-domain and using a Discrete Fourier Transform (DFT) on-the-fly to transform the solution into the frequency-domain. This strategy is appealing because of its straightforward implementation, however, three limitations can be identified. First, no Multiple Right Hand Sides (MRHS) acceleration can be implemented. Second, the time-integration is controlled by a CFL condition depending on the highest P-wave-velocity and the smallest discretization step. For elastodynamics simulations in media with slow S-wave velocity, this could lead to severe restrictions on the time step. Finally, the integration time should be long enough for the DFT to be sufficiently accurate.

These limitations lead us to rather investigate the possibility of solving the elastodynamics equations using iterative solvers. These algorithms fully benefit from the sparsity of the linear system as they only require the computation of scalar products and sparse matrix vector products, and therefore present significantly lower memory requirement. In this study, we are interested in the CGMN iterative solver, introduced by Björck and Elfving (1979). This method is based on the row projection strategy proposed by Kaczmarz (1937) to transform the initial non-symmetric system into a symmetric positive one which can be solved using a Conjugate Gradient (CG) method. This strategy can be efficiently parallelized using row-block decomposition and averaging techniques (Gordon and Gordon, 2010). It has been successfully applied to the frequency-domain wave modeling in the 3D acoustic approximation (van Leeuwen et al., 2012) as well as in the 2D and 3D elastic approximations (Li et al., 2014). In the latter case, CGMN reveals to be extremely robust: the convergence is obtained in media with fast variations, presenting high Poisson's ratio, and with a free surface boundary condition responsible for the generation of surface waves. In the same configuration, standard Krylov solvers such as GMRES, BiCGSTAB, or Conjugate Gradient on Normal Equations (CGNR) (Saad, 2003) fail to converge.

Despite these relatively good properties, the CGMN method still requires a large number of iterations to reach sufficient accuracy. We are thus interested in the design of a suitable preconditioner for the CGMN method, allowing to increase the performance of this algorithm. In addition, in the perspective of FWI applications, a MRHS strategy based on the seed method proposed by Chan and Wan (1997) is applied to investigate the potential acceleration which can be expected.

A numerical example on the BP2004 case study is provided in the 2D visco-acoustic approximation. Using a 4th order finite-difference scheme the combination of the two strategies (preconditioning and MRHS) allow for an overall speed-up equal to 3 and a reduction of the total number of iterations by a factor 10 for a case study involving 1000 sources.

THEORY

Overview of the CGMN method

We consider the linear system derived from the discretization of the frequency-domain wave equation

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where A is a complex and invertible matrix of size n and $\mathbf{b}$ is a complex vector. The rows of A are denoted by $\mathbf{a}_{i\bullet}$. Assuming these rows have been previously normalized, the Kaczmarz's method cyclically projects the iterate $\mathbf{x}^k$ following

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda \left(b_i - \mathbf{a}_{i\bullet}^T \mathbf{x}^k \right) \mathbf{a}_{i\bullet}, \quad i = k+1\,[n], \tag{2}$$

where $\lambda \in]0;2[$ is a relaxation parameter. Performing the projections (2) from the first to the last row of A is referred to as a forward sweep. Performing these projections in the reverse order is referred to as a backward sweep. A succession of a forward sweep and a backward sweep is referred to as a double sweep operation. Within this framework, the well known Symmetric Successive Over Relaxation (SSOR) iterative method (Saad, 2003) is equivalent to a sequence of double sweep operations. The solution of (1) through the SSOR method is computed through the iteration $\mathbf{x}^{k+1} = Q\mathbf{x}^k + R\mathbf{b}$, where Q and R are deduced from A and represent the double sweep operation. An acceleration of this fixed-point iteration can be obtained by considering the system

$$(I - Q)\mathbf{x} = R\mathbf{b}, \tag{3}$$

where the identity operator is denoted by I. The matrix $I - Q$ is symmetric and positive semi-definite (Björck and Elfving, 1979). Hence, the CG method can be applied to solve (3). The resulting method is called CGMN.

Using a finite-difference method for the discretization of the wave equation yields a sparse matrix A with s non-zero diagonals, s depending on the order of the discretization scheme. A double sweep requires $2ns$ operations ($2n$ projections on sparse rows), with s being in $O(1)$. Using the double sweep operation, the complexity of the matrix vector product $(I - Q)x$ is thus in $O(n)$, despite the matrix $I - Q$ is dense.

Preconditioning strategy for the CGMN method

The preconditioning method presented in this study relies on applying the CGMN method to the right preconditioned system

$$AP\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = P\mathbf{y}, \tag{4}$$

where the matrix P is an approximate inverse of A. Following earlier results on the use of complex-shifted Laplacian preconditioners (Erlangga and Nabben, 2008), a good preconditioner for A can be deduced from a heavily damped wave propagation operator B. In this study, we build a preconditioner P as a sparse approximate inverse of B. The k-th column of B^{-1}, denoted by $\mathbf{b}_{\bullet k}^{-1}$, is the solution of the system

$$B\mathbf{b}_{\bullet k}^{-1} = \mathbf{e}_k, \tag{5}$$

where $\mathbf{e}_k$ is a vector of size n with a single non-zero component equal to 1 at index k. Assuming here a 2D approximation, let (i_k, j_k) be the couple of indexes representing the spatial position associated with the index k, the column $\mathbf{b}_{\bullet k}^{-1}$ can be identified with the pressure wavefield generated by a Dirac source located at the position (i_k, j_k) in a strongly attenuating medium. Each column of P can thus be sparsely approximated by restraining $\mathbf{b}_{\bullet k}^{-1}$ to few discretization points around (i_k, j_k) (see Figure 1). An efficient way to compute the preconditioner P is to solve the n local frequency-domain wave propagation problems in a strongly damped small medium (9 or 25 grid points according to the sparsity pattern which has been chosen). Perfectly Matched Layers (PML) (Bérenger, 1994) surround this local domain to absorb the outgoing waves. Using this method, the computation of the preconditioner presents low memory requirements as it only requires the solution of small frequency-domain wave propagation problems. In addition it is highly scalable as each column of the preconditioner can be computed independently.

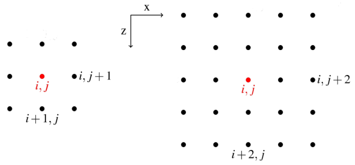


Figure 1: Sparsity pattern for computing the preconditioner using 9-points (left) and 25-points (right) per column.

This approach ensures P is composed of only few non-zero diagonals (9 or 25 in Figure 1). The number of non-zero diagonals of AP, denoted by $\tilde{s}$, increases with respect to the number s of non-zero diagonals of A. A gain in number of arithmetic operations is obtained if the increase from s to $\tilde{s}$ is compensated by the decrease of the number of iterations. This is quantified by the ratio $\eta = (Ns)/(\tilde{N}\tilde{s})$, where N and $\tilde{N}$ denote respectively the number of iterations performed by CGMN to solve the system (1) and (4).

Seed method for linear systems with MRHS

Consider the original linear system (1) with N_{rhs} MRHS

$$AX = B, \tag{6}$$

where $X = \left[x^{(1)}, \ldots, x^{(N_{rhs})}\right]$ are the solutions of the linear system related to the right-hand sides $B = \left[b^{(1)}, \ldots, b^{(N_{rhs})}\right]$. The straightforward treatment for an iterative approach to solve the linear systems (6) is to solve each linear system independently. One way to accelerate this process is to compute initial guesses for the unsolved systems while solving the linear systems. This is efficiently done using the single seed method introduced by Smith et al. (1989) and Chan and Wan (1997).

The single seed method takes advantage of the Krylov subspace generated along the solution of one linear system in (6), referred to as the seed system. The residuals of the others systems are projected orthogonally on this Krylov subspace to compute an approximate initial guess. Once the seed system is solved, this process is repeated by selecting the next "non seed" systems as the seed system. This strategy is presented in Algorithm 1. For seismic experiments, the MRHS are associated with distinct source positions. As a consequence, the MRHS are orthogonal one to each other. In practice the seed method performs poorly in such configurations. This is understandable as in this case the residuals associated with one system are nearly orthogonal to the Krylov space generated through the solution of the seed system. Therefore the ini-

Algorithm 1 Single seed method algorithm

```
Compute Conjugate Gradient initial residuals
for $k = 1, \ldots, N_{rhs}$ do
    Set the $k$-th system as the seed system
    for $i = 1, \ldots,$ until the seed system is solved do
        Perform Conjugate Gradient iteration:
        Compute $p_i^{(k)}$, $r_i^{(k)}$ and $x_i^{(k)}$
        for $j = k+1, \ldots, N_{rhs}$ do
            Compute the projected solutions and residuals
            $\eta \leftarrow (p_i^{(k)})^T r_i^{(j)} / (p_i^{(k)})^T (I-Q) p_i^{(k)}$
            $x_{i+1}^{(j)} \leftarrow x_i^{(j)} + \eta p_i^{(k)}$
            $r_{i+1}^{(j)} \leftarrow r_i^{(j)} - \eta (I-Q) p_i^{(k)}$
        end for
    end for
end for
for $k = 1, \ldots, N_{rhs}$ do
    $r_0^{(k)} \leftarrow Rb^{(k)} - (I_n - Q) x_0^{(k)}$
end for
```

tial guess computed through the seed method is close to zero. However, one can circumvent this problem by transforming the system (6) into

$$AY = MB, \quad X = M^{-1}Y, \tag{7}$$

where $M \in \mathbb{R}^{N_{rhs} \times N_{rhs}}$ is an invertible matrix which we will refer to as the blending matrix. For the numerical experiments, we will choose the matrix M of the form

$$M = \begin{pmatrix} 1-\gamma & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \cdots & 1 & 1-\gamma \end{pmatrix}, \tag{8}$$

with $\gamma \in]0,1[$. The choice of γ should be done with caution. Indeed, as γ tends towards 1, the seed strategy tends to be particularly efficient, as the right hand sides MB tend to be similar. However, the condition number of M also increases, and the multiplication of Y by M^{-1} degrades the accuracy of the solution. The accuracy required for the solution of Y should thus be done accordingly, which decreases the efficiency of the seed strategy. In the case study presented here, a choice of $\gamma = 0.2$ seems to represent an acceptable trade-off between these two requirements.

TEST ON 2D MODEL

We consider the frequency-domain visco-acoustic wave equation

$$-\frac{\omega^2}{\rho(x) v_P(x)^2} p(x) - \operatorname{div}\left(\frac{1}{\rho(x)} \nabla p(x)\right) = s(x, \omega), \tag{9}$$

where $\omega = 2\pi f$ is the angular frequency, $v_P(x)$ is the P-wave velocity, $\rho(x)$ is the density, $s(x,\omega)$ is the source term and $p(x,\omega)$ is the pressure wavefield.

The attenuation is accounted for using the Kolsky model, resulting in a complex-valued P-wave velocity. The velocity is therefore defined by $v_P(x) = v_{P,0}(x)\,(1 - 0.5i/q_{att})$, where q_{att} is the quality factor. PML are attached to the four sides of the domain to mimic an infinite medium and absorb outgoing energy.

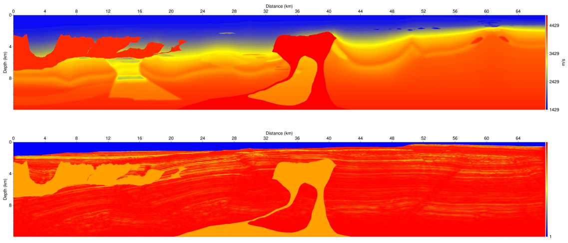


Figure 2: 2004 BP model: velocity (top) and density (bottom).

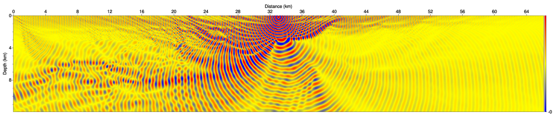


Figure 3: Pressure wavefield at frequency $f = 5$ Hz for a Dirac source located at $x = 33$ km and $z = 180$ m.

We perform the numerical experiments using the 67 km long and 12 km deep 2004 BP velocity and density model (see Figure 2). The quality factor q_{att} is taken equal to 1000, which corresponds to a non-attenuating medium. The discretization of the equation (9) is performed using the fourth-order staggered-grid finite-difference scheme of Levander (1988). At least 5 discretization points per wavelength are used for all frequencies. The pressure wavefield solution of (9) for $f = 5$ Hz is presented in Figure 3. The number of iterations performed by the iterative solver to converge is denoted by N_{iter}. The geometric mean size of the domain is denoted by $N = \sqrt{n}$. The initial estimation is taken as $x^0 = 0$ and the stopping criterion is fixed to 10^{-4} on the relative residual $||Ax^k - b|| / ||Ax^0 - b||$. The relaxation parameter λ is equal to 1.2.

Evaluating the efficiency of the preconditioner

The preconditioner P is computed upon a strongly damped medium with $q_{att} = 1$ and using the 9-points sparsity pattern (Fig. 1). The results obtained with this preconditioner are shown in Table 1. The computation time is divided by a factor up to 2.2. The gain in computational cost η reaches a factor 2. The preconditioner reduces the number of iterations up to a factor 6.9. However this gain is compensated by the increase of the

		N_{iter}		Time (s)	
f	n	CGMN	PCGMN	CGMN	PCGMN
1	11551	845	164	2.3	1.4
5	232617	3313	569	206.9	109.7
10	904128	5959	951	1443.5	728.5
20	3564132	9525	1424	9195	4355.4
40	14152068	13469	1942	51480.9	23402.6

Table 1: Number of iterations and time of computations performed by the CGMN method with and without the preconditioner.

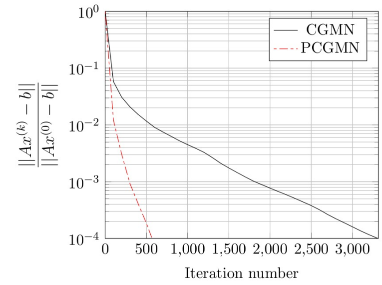


Figure 4: Convergence curves of the normalized residuals of the solution of the frequency-domain wave propagation problem using the CGMN method (black solid curve) and the preconditioned CGMN method (red dotted curve) for $f = 5$ Hz.

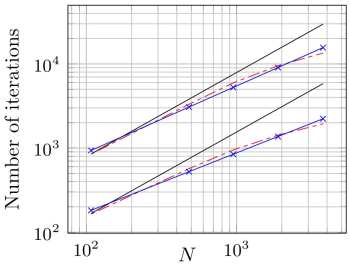


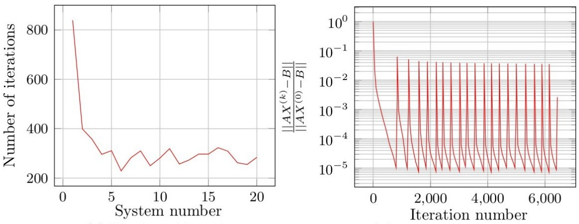
Figure 5: Complexity analysis of the CGMN method (lower red dotted curve) with and (upper red dotted curve) without the preconditioner. The black solid line and the blue "x"-marked line represent respectively the linear increase and trend line.

cost of each iteration. Indeed, the matrix AP has 45 non-zero values per row whereas the matrix A has only 13 non-zero entries. The improvement of the convergence profile of the residuals for $f = 5$ Hz is shown in Figure 4. A complexity analysis of the CGMN method is presented in Figure 5. The number of iterations N_{iter} performed by the preconditioned and non-preconditioned CGMN method is plotted as a function of N on a log-log scale. The computational complexity is sub-linear in N, which is consistent with previous studies (Li et al., 2014). Based on this analysis, the CGMN complexity is evaluated as $N_{iter} \approx 20.9\, N^{0.79}$. Using the preconditioner, the complexity is improved to $N_{iter} \approx 5.9\, N^{0.71}$.

A multiple right-hand side strategy

The gain in computational time and number of iterations of the single seed method is evaluated for $f = 5$ Hz yielding 209 grid points along the vertical direction, 1113 grid points along the horizontal direction and a space grid step $d = 60$ m. The number of sources N_{rhs} is set to 1000. The sources are located at $z_s = 180$ m and regularly spaced by a space grid step d along the horizontal direction. The blending matrix M is defined with $\gamma = 0.2$.

Applying the single seed method as it has been introduced in the previous section over all the N_{rhs} right-hand sides leads to poor results in terms of computational time. The large number of sources makes the computational cost related to the additional projection operations (lines 7 to 12 in algorithm 1) too expensive, and not compensated by the reduction of the overall number of iterations. Therefore, we apply the seed method only on smaller subsets of right-hand sides. We divide the 1000 RHS into 50 subsets of 20 RHS and apply our method sequentially on each subset. The improvement of the convergence profile when solving the linear systems with the subset of RHS is shown in Figure 6. Linear systems using the projected initial guesses converge in an increasingly smaller number of iterations. Compared to the CGMN method using the preconditioner, the computational time is then divided by a factor 1.5 whereas the number of iterations is divided by a factor 1.6. Compared to the CGMN method without preconditioner, the computational time is divided by a factor 2.9 whereas the number of iterations is divided by a factor 10.



(a) Number of iterations (b) Residual convergence

Figure 6: Number of iterations performed to solve each linear system with the subset of RHS (a), residual convergence of the linear systems with the subset of RHS (b).

CONCLUSION

A sparse approximate inverse preconditioner for frequency-domain wave equation and a MRHS strategy are presented in this study. The preconditioner is built as an approximate inverse of a strongly damped frequency-domain wave propagation operator. Its evaluation relies on the solution of local systems of small size which can be performed independently making its construction highly scalable. The preconditioner accelerates substantially the convergence of the CGMN solver. However this gain is compensated by the increase of the computational cost of each iteration and the final speed-up is around 2.2 for a system of 14 million unknowns. The single seed method is used to accelerate the solution of the system for 1000 right-hand-sides. When applied to small subsets of the right-hand sides, an additional speed up of 1.5 is observed.

Further work will be dedicated to the application of the preconditioner to 2D and 3D realistic elastic systems, and adapted to the parallel version of the CGMN algorithm, named as the CARP-CG method (Gordon and Gordon, 2010). Block conjugate gradient method (O'Leary, 1980) will be investigated together with the single seed method for improving the MRHS strategy. Ultimately, this method should serve as an efficient forward modeling engine for performing 3D frequency-domain FWI in the elastodynamics approximation.

ACKNOWLEDGMENTS

This study was funded by the SEISCOPE consortium (*http://seiscope2.osug.fr*), sponsored by BP, CGG, CHEVRON, EXXON-MOBIL, JGI, PETROBRAS, SAUDI ARAMCO, SCHLUMBERGER, SHELL, SINOPEC, STATOIL, TOTAL and WOODSIDE. This study was granted access to the HPC resources of CIMENT infrastructure (https://ciment.ujf-grenoble.fr) and CINES/IDRIS under the allocation 046091 made by GENCI.

REFERENCES

Bérenger, J.-P., 1994, A perfectly matched layer for absorption of electromagnetic waves: Journal of Computational Physics, **114**, 185–200.

Björck, Å., and T. Elfving, 1979, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations: BIT Numerical Mathematics, **19**, 145–163.

Chan, T. F., and W. L. Wan, 1997, Analysis of projection methods for solving linear systems with multiple right-hand sides: SIAM J. Sci. Comput.

Erlangga, Y. A., and R. Nabben, 2008, On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian: Electronic Transactions on Numerical Analysis, **31**, 403–424.

Gordon, D., and R. Gordon, 2010, CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs: Parallel Computing, **36**, 495–515.

Kaczmarz, S., 1937, Angenäherte Auflösung von Systemen linearer Gleichungen (English translation by Jason Stockmann): Bulletin International de l'Académie Polonaise des Sciences et des Lettres, **35**, 355–357.

Levander, A. R., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Li, Y., L. Métivier, R. Brossier, B. Han, and J. Virieux, 2014, 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver: Geophysics, Accepted.

O'Leary, D. P., 1980, The block conjugate gradient algorithm and related methods: Linear algebra and its applications, **29**, 293–322.

Operto, S., R. Brossier, L. Combe, L. Métivier, A. Ribodetti, and J. Virieux, 2014, Computationally-efficient three-dimensional visco-acoustic finite-difference frequency-domain seismic modeling in vertical transversely isotropic media with sparse direct solver: Geophysics, **79(5)**, T257–T275.

Operto, S., A. Miniussi, R. Brossier, L. Combe, L. Métivier, V. Monteiller, A. Ribodetti, and J. Virieux, 2015, Efficient three-dimensional frequency-domain full-waveform inversion of ocean-bottom cable data: application to Valhall in the visco-acoustic vertical transverse isotropic approximation: Geophysical Journal International, **submitted**.

Saad, Y., 2003, Iterative methods for sparse linear systems: SIAM.

Smith, C. F., A. F. Peterson, and R. Mittra, 1989, A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields: Antennas and Propagation, IEEE Transactions on, **37**, 1490–1493.

van Leeuwen, T., D. Gordon, R. Gordon, and F. Herrmann, 2012, Preconditioning the Helmholtz equations via row projections: 74^{th} EAGE Conference and Exhibition, Expanded Abstracts, EAGE, A002.