A perturbed two-level preconditioner for the solution of three-dimensional heterogeneous Helmholtz problems with applications to geophysics

Xavier Pinel

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Mathématiques, Informatiques et Télécommunication

Présentée et soutenue par :
Xavier Pinel

le : mardi 18 mai 2010

Titre :

A perturbed two-level preconditioner for the solution of three-dimensional heterogeneous Helmholtz problems with applications to geophysics

---

JURY
Hélène Barucq, Rapporteur , Henri Calandra, Membre du Jury
Iain Duff, Membre du Jury , Andreas Frommer, Rapporteur
Serge Gratton, Directeur de thèse , Cornelis Oosterlee, Rapporteur
Xavier Vasseur, co-encadrant

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
CERFACS

Directeur(s) de Thèse :

Serge Gratton

Rapporteurs :

Hélène Barucq, Andreas Frommer et Cornelis Oosterlee

Dissertation for the degree of doctor in Mathematics,
Computer Science and Telecommunications (ED MITT)

A perturbed two-level preconditioner for the solution of three-dimensional heterogeneous Helmholtz problems with applications to geophysics

Xavier Pinel (PhD student, CERFACS and INPT)

Hélène Barucq	Research director, INRIA and University of Pau	France	Referee
Henri Calandra	Senior advisor, TOTAL	France	Member of jury
Iain Duff	Professor, RAL and CERFACS	UK, France	Member of jury
Andreas Frommer	Professor, University of Wuppertal	Germany	Referee
Serge Gratton	Professor, ENSEEIHT and INPT/IRIT	France	PhD advisor
Cornelis Oosterlee	Professor, Delft University of Technology and CWI Amsterdam	The Netherlands	Referee
Xavier Vasseur	Senior researcher, CERFACS	France	PhD co-advisor

July 23, 2010

Remerciements

En premier lieu, je désirerais remercier le groupe énergétique TOTAL pour le financement de ma thèse au travers du CERFACS ainsi que les membres de mon Jury de thèse.

En particulier, je tiens à exprimer ma gratitude à mon directeur de thèse, le professeur Serge Gratton, et à mon co-encadrant, le docteur Xavier Vasseur, sans qui ce travail n'aurait pas été possible.

Il en va de même pour les rapporteurs de ma thèse: la directrice de recherche Hélène Barucq, le professeur Andreas Frommer et le professeur Kees Oosterlee.

Je sais gré à tous les membres de l'équipe ALGO du CERFACS et à son chef, le professeur Iain Duff, d'avoir été à mes côtés durant ces quatre dernières années: Anke, Antoine, Audrey, Azzam, Bora, Brigitte, Caroline, Mme Chatelain, Fabian, François, Jean, Kamer, Léon, Marc, Martin, Mélodie, Milagros, Mohamed, Nicole, Pablo, Pavel, Phillip, Rafael, Riadh, Selime, Tzvetomila, Xueping.

Je voudrais pareillement saluer l'équipe APO de L'ENSEEIHT et l'équipe MUMPS dont l'aide et les conseils m'ont été précieux.

Je souhaite également remercier les personnes dont la collaboration m'a permis de mener à bien ce projet: Henri Calandra et Pierre-Yves Aquilanti de TOTAL, Luc Giraud, Julien Langou, ainsi que les organismes de calcul intensif dont j'ai utilisé les super-calculateurs: le CINES, le CSC-IT Espoo, l'IDRIS et le Jülich Forschungszentrum.

Finalement, je tiens à témoigner ma reconnaissance à mes parents et amis: M. et Mme Pinel, Franzi, Philippe, Laetitia, Pépé, Mamie, Tatie Joe, Gérard, Constance, Bernie, Tatie Anne, Caroline, Marion, Elizabeth, Henri Pinel, Clément, les descendants de Jeannot, Jako, Glup, Choco, Biquet, Dani, Marc, Otto, Julie, Pierre, Célia, Manu, Poncho, Vincent, Kévin ...

Thesis Summary

The topic of this PhD thesis is the development of iterative methods for the solution of large sparse linear systems of equations with possibly multiple right-hand sides given at once. These methods will be used for a specific application in geophysics - seismic migration - related to the simulation of wave propagation in the subsurface of the Earth. Here the three-dimensional Helmholtz equation written in the frequency domain is considered. The finite difference discretization of the Helmholtz equation with the Perfect Matched Layer formulation produces, when high frequencies are considered, a complex linear system which is large, non-symmetric, non-Hermitian, indefinite and sparse. Thus we propose to study preconditioned flexible Krylov subspace methods, especially minimum residual norm methods, to solve this class of problems. As a preconditioner we consider multi-level techniques and especially focus on a two-level method. This two-level preconditioner has shown efficient for two-dimensional applications and the purpose of this thesis is to extend this to the challenging three-dimensional case. This leads us to propose and analyze a perturbed two-level preconditioner for a flexible Krylov subspace method, where Krylov methods are used both as smoother and as approximate coarse grid solver.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The target industrial application of this PhD thesis is related to the solution of wave propagation problems in seismics [24]. At a given frequency, a source is triggered at a certain position on the Earth's surface. As a consequence, a pressure wave propagates from the source. When a wave encounters discontinuities, it is scattered and propagated back to the surface. The pressure field is then recorded at several receiver locations located on the Earth's surface. This experimental process is repeated over a given range of frequencies and with multiple source locations. The main aim of the numerical simulation is thus to reproduce these wave propagation phenomena occurring in heterogeneous media. This leads to an interpretative map of the subsoil that helps to detect both the location and the thickness of the reflecting layers. The resulting frequency-domain problem is then solved using efficient solvers, able to take benefit of the structure of the system on modern parallel architectures. Afterward, an inverse Fast Fourier Transform is employed to obtain the time-domain solution from the set of frequency-domain solutions. This time-domain solution is of great importance in oil exploration for predicting correctly the structure of the subsurface. In this thesis, the wave propagation is modeled by the Helmholtz equation,

$$-\Delta u - k^2 u = s,$$

where u denotes the wave pressure, k the wavenumber and s a given source term. Absorbing boundary conditions are used to simulate an infinite domain and to limit spurious reflections. A key point for an efficient migration thus relies on a robust and fast solution method for the heterogeneous Helmholtz problem at high wavenumbers with multiple sources. For each considered frequency, the discretization of the Helmholtz operator by finite difference or finite element techniques leads to a linear system of equations of the following type

$$AX = B,$$

where $A \in \mathbb{C}^{n\times n}$ is a square matrix which is sparse, usually non-Hermitian, non-symmetric, large and indefinite at high wavenumbers, $X \in \mathbb{C}^{n\times p}$, $B \in \mathbb{C}^{n\times p}$ where p is the number of sources.

These large and indefinite linear systems can be handled very efficiently up to a certain point by sparse direct methods [29, 30] (e.g. sparse Gaussian elimination LU-factorization). In the indefinite non-symmetric case, pre-processing (permutation and scaling) can be performed before the factorization phase to minimize the fill-in and improve the accuracy of the factorization e.g. obtaining matrices with a zero-free diagonal [34]. In the two-dimensional case, these methods have proved efficient [65] since they enable both the solution of linear systems to machine precision and the reuse of the LU-factorization in multi-source situations. However, their memory requirement greatly increases with the size of the problem, compromising their use on a parallel distributed memory computer for large three-dimensional problems. In [89], the authors have used MUMPS [2, 3, 4] to solve three-dimensional Helmholtz problems formulated with a compact 27 point stencil discretization scheme. In fact they have reported that the memory complexity of the LU factorization is $O(35n^{4/3})$, the number of floating-point operations during the factorization phase is $O(n^2)$ and the computational complexity of the solution phase $O(n^{4/3})$. Despite this computational cost, the approach was used with success when solving a Helmholtz problem at 10 Hz on the SEG/EAGE Overthrust model [5] considering a $409 \times 109 \times 102$ grid and allocating 450 GB of memory.

To alleviate the memory constraint, iterative methods can be considered. One of the key points becomes then the design of an efficient preconditioner to obtain a fast convergence. Similarly the choice of the Krylov

method [101], especially in the multiple right-hand sides situation, has to be addressed. First, we describe some preconditioning techniques described in the literature for Helmholtz problems.

Incomplete factorizations (ILU) [8] are popular preconditioning techniques, that may however lead to unstable, highly ill-conditioned incomplete factors in the indefinite case. Some remedies have been proposed to manage these issues when considering Helmholtz problems. In [52] a specific factorization is designed that aims at performing an analytic incomplete factorization (AILU); this approach is yet difficult to extend to the heterogeneous case. We also note that incomplete LU factorization with threshold (ILUT [100]) is recommended in [68] for a finite element discretization of the Helmholtz operator (Galerkin Least Square (GLS)). Finally, an other approach consists in performing an incomplete factorization of a complex shifted Helmholtz operator as a preconditioner for the original Helmholtz problem [80, 90] (see Equation 1.1 and details hereafter). However, the convergence of ILU preconditioned Krylov methods is found to be generally slow at high wavenumbers and storing the ILU factors may not be always affordable. Furthermore it is recognized that ILU methods are difficult to parallelize [8, 63].

Another important class of preconditioners relies on domain decomposition techniques [94, 111, 114]. These methods solve the original problem by splitting the physical domain into smaller subdomains where the solution of the local problems is affordable with direct methods. For elliptic definite problems, their convergence rate becomes independent of the number of subdomains if a coarse space correction is included. Due to their indefinitness at high wavenumbers, Helmholtz type problems are challenging for domain decomposition preconditioners for two main reasons. First in order to be effective, a rather fine coarse space has to be considered. Consequently this leads to large coarse problems. Secondly local Dirichlet or Neumann problems may be close to singular. We refer the reader to Section (11.5.2) in [114] for further comments and references.

When nonoverlapping domain decompositions methods are considered, it is advocated to use Sommerfeld-like conditions on the subdomain boundaries to obtain well-posed local problems [10, 26, 51]. An efficient domain decomposition preconditioner for indefinite Helmholtz problem is FETI-H [46] where an auxiliary coarse problem based on plane waves is considered. This approach has been improved in [44, 45] introducing a dual primal variant of FETI-H (FETI-DPH) and allows to solve Helmholtz scattering problems at middle-range frequencies on a large number of cores [44]. To the best of our knowledge the most recent theoretical result related to domain decomposition preconditioners for homogeneous Helmholtz problems with Dirichlet boundary conditions (discretized with standard finite element techniques) is due to Li and Tu [76]. A bound for the condition number of the preconditioned operator $A\ M^{-1}$ has been proven for the case of exact local solvers:

$$\kappa(A\ M^{-1}) \leq C(1+k^2)(1+k^2H^2)\left(1+log\left(\frac{H}{h}\right)\right)^2.$$

where C is a positive constant independent of the diameter element h and the maximal diameter of the subdomains H. Consequently $\kappa(A\ M^{-1})$ is found to grow like k^4. Obviously this is a major drawback when considering high wavenumbers. Recently an algebraic formulation has been proposed for Helmholtz problems in [62, 120]. It consists in an algebraic additive Schwarz preconditioner and enables to solve problems for frequencies up to 12 Hz in a reasonable time on real-life velocity model (SEG/EAGE Saltdom) on 2000 BlueGene/P processors [1]. However a drawback of this method is its high memory cost.

Multigrid methods [15, 20, 61, 115] can also be used as a preconditioner for Helmholtz problems. Nevertheless they also encounter difficulties to cope with such indefinite problems. Regarding Helmholtz problems, classical multigrid ingredients such as standard smoothing and coarse grid correction are found ineffective [7, 19, 37, 42, 70]. First, smoothers cannot smooth error components on the intermediate grids. Second, the wavenumber k in the discrete Helmholtz operator makes its approximations poor on coarse meshes, the effect of the coarse grid correction being then deteriorated. In [31, 37, 42, 70, 78], strategies have been proposed to adapt the multigrid technique to the solution of Helmholtz problems.

A first strategy consists of the use of few grids in the hierarchy of the multigrid preconditioner [31, 37, 70] such that the grid approximation is effective on the considered grids. If more than two grids are considered, non-standard smoothers (Krylov based such as GMRES [102]) on the coarser levels should be used to alleviate the weakness of standard smoothers on intermediate grids [37]. However, in three

[1] http://www.idris.fr/docs/docu/projets-Babel/SEISCOPE/CR-projet-SEISCOPE.html

dimensions, a reduced number of grids in the multigrid hierarchy could lead to a coarse problem whose factorization is prohibitive in terms of computational resources.

A second approach is to solve Helmholtz problems with a *wave-ray* multigrid algorithm [77]. These methods are based on two representations of the error on the coarse grids of the hierarchy. These representations enable then both the smoother and coarse grid corrections to be efficient. This method performs well in the homogeneous case [74, 78, 118] but, in the heterogeneous case, ray functions must be computed. It implies to solve large eigenvalue problems [122, 123] that may be expensive in terms of computational resources.

Lately a third multigrid preconditioner - considered as a significant breakthrough - has been proposed in [42, 43], it is not directly applied to the discrete Helmholtz operator but to a complex shifted one defined as:

$$-\Delta u - (1 - i\beta)k^2 u \tag{1.1}$$

where β denotes the shift parameter. This shift parameter makes standard multigrid efficient on the preconditioning problem [42]. This solution method has proved efficient for relatively high wavenumbers considering both homogeneous and heterogeneous problems [42, 95, 96]. However the complexity of the method remains high (see [95] in two dimensions and [96] in three dimensions respectively). More recently, an algebraic multi-level preconditioner based on this shifted approach has been proposed in [14]. An incomplete LDL^T factorization is performed on each level of the multi-level hierarchy taking advantages of modern direct methods for sparse symmetric indefinite matrices [35, 103]. This method has shown efficient to improve the convergence of Krylov methods for both two-dimensional and three-dimensional heterogeneous problems but its complexity is still relatively high. Yet this class of multi-level preconditioners raises the question of the determination of the shift parameter β. Indeed it is depending on the multilevel components [14, 42] and of course on the discretization of the Helmholtz operator [116]. Therefore, the choice of the shift parameter is not obvious and often relies on extensive numerical experiments and/or on a Fourier analysis [15].

The choice of a shift parameter can be avoided if a two-grid preconditioner is applied to the original Helmholtz discrete operator [31] where a sparse direct method is employed for the coarse solution phase of the two-grid algorithm. As said before, the computational cost of a LU-factorization in three dimensions, even on the coarse grid, is too severe. Consequently, an iterative method seems to be the natural choice for solving the coarse grid problem. Thus, in this thesis, we consider a perturbed two-grid preconditioner applied to the original Helmholtz operator where the coarse problem is solved only approximately. The efficiency of such a preconditioner relies on both its monitorable computational memory requirements and its good preconditioning properties when using a really large convergence threshold on the coarse grid. This last point will be analyzed in the Fourier analysis framework and illustrated both by numerical experiments and a spectrum analysis.

Moreover we advocate the use of a preconditioned Krylov method on the coarse level of the two-level method. This leads us to the choice of the flexible GMRES (FGMRES [99]) as an outer Krylov method. Indeed the two-level preconditioner varies from one iteration to the next. In this work, we have extended GMRES with deflated restarting [85] to the flexible case (FGMRES-DR [53]). This method has shown efficient for two-dimensional Helmholtz problems with Dirichlet boundary conditions but relatively less efficient for absorbing boundary conditions. Another challenging issue in the geophysics application is an efficient treatment of multiple sources (up to few thousands). The design of efficient block Krylov methods to process several sources at once is then of crucial interest. In this thesis, starting from existing references related to block GMRES methods [59, 72, 73, 79, 97, 112, 121], we have developed efficient variants of Block Flexible GMRES (BFGMRES) implementing the deflation of the block residual at the restart: Block Flexible GMRES with SVD based Deflation (BFGMRESD) and Block Flexible GMRES with SVD based Truncation (BFGMREST). Both methods perform a SVD of the block residual ($R = U\Sigma W^H$ [54]) at each restart. The BFGMRESD method uses as an initial block vector at each restart the singular vectors corresponding to the largest singular values as defined by a threshold whereas BFGMREST keeps as an initial block residual a fixed number of singular vectors corresponding to the largest singular values.

Finally, all these methods have been evaluated in a parallel distributed memory environment. Extensive numerical experiments have shown the robustness and efficiency of the perturbed two-level preconditioner both on homogeneous and heterogeneous problems on thousands of cores. Moreover, the interest for

BFGMRESD and BFGMREST clearly appears when multiple right-hand sides have been considered.

The outline of the thesis is thus as follows:

- In Chapter 2, Krylov methods for both single and multiple right-hand sides situation are presented. First, a brief description of GMRES and Flexible GMRES (FGMRES) is given, introducing a spectrum analysis tool in the FGMRES context. Then the flexible GMRES method with spectral deflation at the restart is introduced. Finally, block flexible Krylov methods are presented. We describe some strategies to take advantage of the multiple right-hand sides context: deflation of the residual (computation of the numerical rank of the block residual at each restart) and truncation of the residual (use of a part of the block residual to compute the block solution corresponding to the whole block residual).
- In Chapter 3, we focus on multi-level methods used as a preconditioner for three-dimensional Helmholtz problems. First, basic elements on three-dimensional geometric multigrid are introduced. Then, a Fourier analysis is described for three-dimensional Helmholtz problems. A smoothing analysis is performed for both original and shifted Helmholtz operators. It is followed by an analysis of a two-level cycle where a preconditioned Krylov method is used on the coarse level. This analysis shows that the convergence factor of a two-grid method is nearly the same whether the coarse solution is exact or whether the coarse problem is solved within a rather large convergence threshold. This behavior is numerically confirmed using a perturbed two-level method as a preconditioner. Finally a spectrum analysis is included to show the evolution of the spectrum of the preconditioned Helmholtz operator according to several coarse tolerances.
- In Chapter 4, numerical experiments on parallel distributed memory computers are presented. First three-dimensional homogeneous Helmholtz problems are considered. Using the perturbed two-level method described in Chapter 3 as a preconditioner for FGMRES, a strong scalability property is obtained (growing numbers of cores for a fixed problem size) with experiments up to $65,536$ cores. Concerning the weak scalability (the number of cores is growing linearly with the size of the problem), the number of iterations of the method is found to grow linearly with the frequency parameter up to 2048^3. Then, heterogeneous problems are considered. Two public domain velocity fields are considered, the SEG/EAGE Salt dome and the SEG/EAGE Overthrust. The two-level preconditioner is found efficient for heterogeneous problems even if it does not scale as well as in the homogeneous case for a large number of cores (more than 2048). Finally, we present numerical results in the multiple right-hand side context for heterogeneous problems. We show that, using block methods presented in Chapter 2 in combination with the two-level preconditioner can greatly improve the overall number of iterations required when solving the multiple right-hand side problems.

Chapter 2

Krylov subspace methods

2.1 Introduction

In this section we focus on a class of iterative methods called Krylov subspace methods for solving linear systems of the following type:

$$Ax = b,\ A \in \mathbb{C}^{n\times n},\ b,\ x \in \mathbb{C}^n$$

where A is complex, non-symmetric, non-Hermitian, sparse and non-singular. Of course, the most robust way to solve linear systems is to use direct methods. For this class of problem, they consist in performing an LU-factorization of the matrix and forward backward substitutions to obtain the solution of the linear system. Once the LU factorization is obtained, they enable to solve easily several linear systems involving the same matrix (multiple right-hand sides situation). However, a direct method may need important computational resources. Indeed, the LU factors must be stored and they are less sparse than the matrix A in general [29, 30]. Furthermore, it has a computational cost of $O(n^{(4/3)})$ for a Laplacian like operator. Iterative methods can remedy these drawbacks, their memory requirement is generally low and can be controlled; matrix-vectors and dot products are their dominant operations in flops. Their principle is to look for the solution in a Krylov subspace. Krylov subspaces, denoted by $K_m(A, r_0)$, are vector subspaces of $\mathbb{C}^n$ spanned by monomials of A applied to the initial residual vector $r_0 = b - Ax_0$ where x_0 is the initial solution guess:

$$K_m(A, r_0) = \text{span}\left\{r_0, Ar_0, A^2r_0, ..., A^{m-1}r_0\right\}.$$

The parameter m is then an upper bound for the dimension of the space $K_m(A, r_0)$ since it is generated by m vectors.

The most popular Krylov methods for the non-Hermitian case are BiCGSTAB [117], GMRES [102] and QMR [50]. We are focusing on the GMRES (General Minimum RESidual) family of methods. In the first half of this chapter, GMRES methods for a single right-hand side will be presented. First, classical GMRES-type methods are depicted: restarted GMRES [102] with and without preconditioning, FGMRES (Flexible GMRES) [99]. Then, methods implementing spectral deflation at the restart (deflated restarting): GMRES with deflated restarting [85] (GMRES-DR), and its flexible variant: FGMRES-DR [53]. The second half of this chapter will be devoted to block Krylov methods, i.e. Krylov methods for multiple right-hand side problems. First a state of the art bibliographical description will be proposed. Then, block Flexible GMRES (BFGMRES) will be introduced followed by two methods that implement residual deflation (BFGMRESD) and residual truncation (BFGMREST) respectively.

2.1.1 Notations

We denote by $\|.\|$ the Euclidean norm, $I_k \in \mathbb{C}^{k\times k}$ the identity matrix of dimension k and $0_{i\times j} \in \mathbb{C}^{i\times j}$ the zero rectangular matrix with i rows and j columns. The operator ^{T} denotes the transpose operation, whereas ^{H} represents the Hermitian transpose operation. Given a vector $d \in \mathbb{C}^k$ with components d_i, $D = \text{diag}(d_1, \cdots, d_k)$ is the diagonal matrix $D \in \mathbb{C}^{k\times k}$ such that $D_{ii} = d_i$. Given a matrix Q we denote by q_j its $j-th$ column. The vector $e_m \in \mathbb{C}^m$ denotes the m-th canonical vector of $\mathbb{C}^m$. Finally, we denote by $\lambda(A)$ the spectrum of the matrix A. Regarding the algorithmic part, we adopt Matlab-like notations in the presentation. For instance $Q(i, j)$ denotes the entry of matrix Q and $Q(1 : m, 1 : j)$ refers to the submatrix made of the m first rows and first j columns of Q.

2.2 General Minimum RESidual (GMRES)

This method consists in finding the solution in the space $x_0 + K_m(A, r_0)$ minimizing the two-norm of the residual $b - Ax$, where x is the solution. It can be formulated as follows:

$$\text{Find } x_m \text{ such that it minimizes } \min_{x \in x_0 + K_m(A,r_0)} \|b - Ax\|.$$

$x_m \in x_0 + K_m(A, r_0)$ can be written as any vector x_m, $x_m = x_0 + V_m y_m$ where y_m is a vector of dimension m and V_m is a unitary $n \times m$-matrix whose columns span $K_m(A, r_0)$. The matrix V_m results from the orthogonalization of a basis of $K_m(A, r_0)$. This orthogonalization is usually made with an Arnoldi process using Modified Gram-Schmidt (MGS), so that V_m satisfies the Arnoldi relation:

$$AV_m = V_{m+1}\bar{H}_m, \tag{2.1}$$

where $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ is a Hessenberg matrix containing the orthogonalization coefficients.

Writing the residual $b - Ax$ with these matrices leads to, denoting by $\beta = \|r_0\|$:

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_m y_m) = r_0 - AV_m y_m & (2.2)\\ &= \beta v_1 - V_{m+1}\bar{H}_m y_m = V_{m+1}(\beta e_1 - \bar{H}_m y_m). & (2.3) \end{aligned}$$

As V_{m+1} is unitary, the residual norm is thus:

$$J(y_m) = \|b - Ax_m\| = \|b - A(x_0 + V_m y_m)\| = \|\beta e_1 - \bar{H}_m y_m\|.$$

Minimizing $\|b - Ax\|$ for $x \in \mathbb{C}^n$ is thus equivalent to minimize $\|\beta e_1 - \bar{H}_m y\|$ for $y \in \mathbb{C}^m$. Therefore, the GMRES algorithm can be divided into two parts:

1. Orthogonalization (Arnoldi process) of the basis for K_m which yields $\bar{H}_m$ and V_m.
2. Minimization of $\|\beta e_1 - \bar{H}_m y\|$. Its minimizer y_m is then used to compute x_m by $x_m = x_0 + V_m y_m$.

In practice, the computational and memory costs of GMRES are increasing with m. Indeed, the computational cost is $O(m^2 n)$ because of the Arnoldi process and the memory cost is $O(mn)$, which may be prohibitive for large m (the dimension n of the problem is fixed). To remedy these problems, restarting GMRES after a few iterations can be a satisfactory solution: GMRES is restarted again after m iterations with x_m replacing the initial guess x_0. The weakness of this method is that convergence is not as easily characterized as e. g. in the case of full GMRES where a Krylov space of dimension n (matrix dimension) contains the solution $A^{-1}b$ [126]. Algorithm 1 describes the classical restarted GMRES algorithm.

Remark 1. *In Algorithm 1 line 7, the convergence is verified on the Arnoldi residual norm ($\|c - \bar{H}_j y_j\|$) normalized by the norm of the right-hand side. However, $\|c - \bar{H}_j y_j\|$ can be different from the norm of the true residual $\|b - Ax_j\|$. Furthermore, the convergence should rather be checked on the backward error $\dfrac{\|b - Ax_j\|}{\|b\| + \|A\|\|x_j\|}$, to scale the matrix and right-hand side entries and so to insure the convergence up to a threshold tol with machine precision ψ. Nevertheless, in [28], the authors show that the backward stability of GMRES using MGS is verified at each step if the matrix is real and if its smallest singular value is much larger than $n^2\psi\|A\|_F$. These last results have led us to consider a convergence criterion based on the Arnoldi's residual.*

2.2.1 Restarted GMRES with right preconditioning

The convergence of restarted GMRES is not guaranteed in general (unless $Re(x^T Ax) > 0,\ \forall x \neq 0$) but can hopefully be improved with preconditioning. Preconditioning consists in improving the numerical properties of the matrix. Some desirable properties satisfied by the preconditioning matrix, denoted by M, are given in [101]: it has to approximate the original matrix, it has to be non-singular and solving the linear system $Mx = b$ should not be too expensive. In restarted GMRES with right-preconditioning, the preconditioning phase appears both in the matrix vector product needed by the Arnoldi process and in the computation of the solution. In order to obtain the right-preconditioned variant of restarted GMRES, line 2 of Algorithm 2 is replaced by $w = AM^{-1}v_j$; and line 8 and 11 of Algorithm 1 by $x_j = x_0 + M^{-1}V_j y_j$;

Algorithm 1 Restarted GMRES (GMRES(m))

1: Choose $m > 0$, $itermax > 0$, $tol > 0$, $x_0 \in \mathbb{C}^n$. Let $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T$ where $c \in \mathbb{C}^{m+1}$, $v_1 = r_0/\beta$.
2: **for** $iter = 1, itermax$ **do**
3: Set $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T$ and $v_1 = r_0/\beta$.
4: **for** $j = 1, m$ **do**
5: *Completion of V_{j+1} and $\bar{H}_j$:* apply Algorithm 2 from line 2 to 8 to obtain $V_{j+1} \in \mathbb{C}^{n\times(j+1)}$ and the upper Hessenberg matrix $\bar{H}_j \in \mathbb{C}^{(j+1)\times j}$ such that:

$$AV_j = V_{j+1}\bar{H}_j \quad \text{with} \quad V_{j+1}^H V_{j+1} = I_{m+1}.$$

6: Compute $y_j = \text{argmin}_{y\in\mathbb{C}^j} \|\beta e_1 - \bar{H}_j y\|$;
7: **if** $\|c - \bar{H}_j y_j\|/\|b\| \leq tol$ **then**
8: $x_j = x_0 + V_j y_j$; stop;
9: **end if**
10: **end for**
11: Compute $x_m = x_0 + V_m y_m$;
12: Set $x_0 = x_m$, $r_0 = b - Ax_0$;
13: Return to line 2.
14: **end for**

Algorithm 2 Arnoldi process with Modified Gram-Schmidt (MGS): computation of V_{m+1} and $\bar{H}_m$

1: **for** $j = 1, m$ **do**
2: $w = Av_j$
3: **for** $i = 1, j$ **do**
4: $h_{i,j} = w^H v_i$
5: $w = w - h_{i,j} v_i$
6: **end for**
7: $h_{i+1,j} = \|w\|$, $v_{j+1} = w/h_{i+1,j}$
8: Define $V_{j+1} = [v_1, \cdots, v_{j+1}]$, $\bar{H}_j = \{h_{i,l}\}_{1\leq i\leq j+1, 1\leq l\leq j}$
9: **end for**

and $x_m = x_0 + M^{-1}V_m y_m$. In fact, right preconditioned restarted GMRES is equivalent to solve the linear system $(AM^{-1})t = b$ with a restarted GMRES and to compute the solution x of the original system $Ax = b$ via $x = M^{-1}t$.

When preconditioning is considered, at each iteration, $M^{-1}v_j$ is computed, which is equivalent to compute the solution z_j of the linear system $Mz_j = v_j$. Preconditioners can be divided into two classes: explicit and implicit preconditioners. For explicit preconditioners, the preconditioning matrix M is built. Diagonal preconditioning, $M = diag(A)$, incomplete LU factorization (ILU) [8], $M = L_{inc}U_{inc}$, and domain decomposition techniques with exact local solvers [94, 111, 114] are such preconditioners. Implicit preconditioners are solution methods aiming at solving approximately $Az_j = v_j$. M is never explicitly formed but has to be non-variable to be used in GMRES with right-preconditioning. Iterative methods like relaxation methods and standard multigrid [115] are such preconditioners.

One may want to use GMRES itself to precondition restarted GMRES; this is not possible with right-preconditioned GMRES. Indeed, the GMRES solution x_m is not depending linearly on the right-hand side b (unlike standard multigrid for instance) except if x_m satisfies $Ax_m = b$. As a consequence, the solution cannot be computed as in line 11 of Algorithm 1. However, various methods have been developed to use variable operators as a preconditioner: this is the class of flexible Krylov methods. These methods allow to use a different preconditioner at each preconditioning step. The next section describes one of these flexible methods: the flexible variant of GMRES (FGMRES).

2.3 Flexible GMRES

FGMRES is a minimum residual norm subspace method based on the GMRES approach that allows variable preconditioning [99]. We denote by M_j the non singular matrix that represents the preconditioner at step j of the method. Algorithm 3 depicts the FGMRES(m) method. Starting from an initial guess x_0, it is based on the flexible Arnoldi relation with $Z_m \in \mathbb{C}^{n\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ and the upper Hessenberg matrix $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ defined below:

Definition 1. *The matrices computed with the FGMRES algorithm [99] satisfy the so-called Flexible Arnoldi relation:*

$$AZ_j = V_{j+1}\bar{H}_j$$

where $Z_j \in \mathbb{C}^{n\times j}$, $V_{j+1} \in \mathbb{C}^{n\times(j+1)}$ such that $V_{j+1}^H V_{j+1} = I_{j+1}$ and $\bar{H}_j \in \mathbb{C}^{(j+1)\times j}$. FGMRES computes an approximation of the solution in a j-dimensional affine space $x_0 + Z_j y_j$ where $y_j \in \mathbb{C}^j$.

An approximate solution $x_m \in \mathbb{C}^n$ is then found by minimizing the residual norm $\|b - A(x_0 + Z_m y)\|$ over the space $x_0 + \text{range}(Z_m)$, the corresponding residual being $r_m = b - Ax_m \in \mathbb{C}^n$ with $r_m \in \text{range}(V_{m+1})$. A similar relation as in GMRES (relation 2.2) is obtained. However, it has to be noted that, on one hand, FGMRES(m) has a greater memory cost than GMRES(m): the preconditioning solutions must be stored in Z_m ; i.e., m additional vectors of length n have to be stored. On the other hand, convergence results related to GMRES cannot be extended to FGMRES since the subspace range(Z_m) is a subspace which is not generated by a single fixed matrix. Nevertheless, a breakdown analysis of FGMRES can be found in [99]. These last considerations lead us to develop a practical tool to obtain a better understanding of the convergence of FGMRES based on a spectrum analysis; this is the topic of the next section.

Algorithm 3 Flexible GMRES (FGMRES(m))

1: Choose $m > 0$, $itermax > 0$, $tol > 0$, $x_0 \in \mathbb{C}^n$. Let $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T$ where $c \in \mathbb{C}^{m+1}$, $v_1 = r_0/\beta$.
2: **for** $iter = 1, itermax$ **do**
3: Set $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T$ and $v_1 = r_0/\beta$.
4: **for** $j = 1, m$ **do**
5: *Completion of V_{j+1}, Z_j and $\bar{H}_j$:* Apply Algorithm 4 from line 2 to 8 with preconditioning to obtain $V_{j+1} \in \mathbb{C}^{n\times(j+1)}$, $Z_j \in \mathbb{C}^{n\times j}$ and the upper Hessenberg matrix $\bar{H}_j \in \mathbb{C}^{(j+1)\times j}$ such that:

$$AZ_j = V_{j+1}\bar{H}_j \quad \text{with} \quad V_{j+1}^H V_{j+1} = I_{m+1}.$$

6: Compute $y_j = \text{argmin}_{y\in\mathbb{C}^j} \|\beta e_1 - \bar{H}_j y\|$;
7: **if** $\|c - \bar{H}_j y_j\|/\|b\| \leq tol$ **then**
8: $x_j = x_0 + Z_j y_j$; stop;
9: **end if**
10: **end for**
11: Compute $x_m = x_0 + Z_m y_m$;
12: Set $x_0 = x_m$, $r_0 = b - Ax_0$;
13: Return to line 2.
14: **end for**

2.3.1 Spectrum analysis in the Flexible GMRES method

It is known that unpreconditioned GMRES(m) converges for any m when the eigenvalues of the matrix A are lying in a convex set, called the field of values, located in a half plane of the complex plane [101, Section 6.11.4]. This property can be partly shown by computing approximations of the extremal eigenvalues of A [38] thanks to Ritz ($\lambda(H_m)$) or harmonic Ritz values ($\lambda(H_m + h_{m+1,m}^2 H_m^{-H} e_m^T e_m)$) [55], [9] where $H_m = \bar{H}_m(1 : m, 1 : m)$ and $\lambda(H_m)$ denotes the spectrum of H_m.

However, in the flexible case, since the Arnoldi relation is $AZ_m = V_{m+1}\bar{H}_m$ (see Definition 1), the Ritz or harmonic Ritz values, are then not corresponding to approximate eigenvalues of A [53].

Algorithm 4 Flexible Arnoldi process: computation of V_{m+1}, Z_m and $\bar{H}_m$

for $j = 1, m$ **do**
  $z_j = M_j^{-1} v_j$
  $w = A z_j$
  for $i = 1, j$ **do**
    $h_{i,j} = w^H v_i$
    $w = w - h_{i,j} v_i$
  end for
  $h_{i+1,j} = \|w\|$, $v_{j+1} = w / h_{i+1,j}$
  Define $Z_j = [z_1, \cdots, z_j]$, $V_{j+1} = [v_1, \cdots, v_{j+1}]$, $\bar{H}_j = \{h_{i,l}\}_{1 \le i \le j+1, 1 \le l \le j}$
end for

Proposition 1. *At the end of the restart in FGMRES, the Ritz or harmonic Ritz values approximate eigenvalues of a certain matrix $\underline{A} \in \mathbb{C}^{n \times n}$ which can be expressed as:*

$$\underline{A} = A Z_m V_m^H + X \underline{V}^H,$$

where X is a $n \times (n-m)$ matrix and $\underline{V}$ is a $n \times (n-m)$ matrix whose columns span the orthogonal complement of $Span\{V_m\}$. Note: $\underline{A}$ changes at each restart.

Proof. Indeed, $\underline{A}$ is satisfying the GMRES Arnoldi relation

$$\begin{aligned} \underline{A} V_m &= \left(A Z_m V_m^H + X \underline{V}^H \right) V_m = A Z_m, \\ \underline{A} V_m &= V_{m+1} \bar{H}_m, \end{aligned}$$

and so, the GMRES method applied to $\underline{A}$ produces the same iterates as FGMRES applied to A. □

Furthermore, we note that FGMRES does not require $\underline{V}$ nor X to fro the computation of the solution. We can choose them appropriately for our convergence analysis.

Proposition 2. *We propose to choose $\underline{V} = X = \left[v_{m+1}, \underline{V_\natural} \right]$ in $\underline{A}$, where v_{m+1} is the $(m+1)^{th}$ column of V_{m+1} and $Span\left\{ \underline{V_\natural} \right\} \perp Span\{V_{m+1}\}$. The spectrum of $\underline{A}$, where multiple eigenvalues are not repeated, is the spectrum of $\left[\bar{H}_m, e_{m+1} \right]$.*

Proof. We have

$$\begin{aligned} \underline{A} &= \left[A Z_m, \underline{V} \right] \begin{bmatrix} V_m^H \\ \underline{V}^H \end{bmatrix} = \left[V_{m+1} \bar{H}_m, \underline{V} \right] \begin{bmatrix} V_m^H \\ \underline{V}^H \end{bmatrix} = \left[V_{m+1} \left[\bar{H}_m, e_{m+1} \right], \underline{V_\natural} \right] \begin{bmatrix} V_{m+1}^H \\ \underline{V_\natural}^H \end{bmatrix}, \\ &= \left[V_{m+1}, \underline{V_\natural} \right] \begin{bmatrix} \left[\bar{H}_m, e_{m+1} \right] & 0_{(m+1) \times (n-m-1)} \\ 0_{(n-m-1) \times (m+1)} & I_{n-m-1} \end{bmatrix} \begin{bmatrix} V_{m+1}^H \\ \underline{V_\natural}^H \end{bmatrix}. \end{aligned}$$

Thus, since $H_{m+1} = \left[\bar{H}_m, e_{m+1} \right]$, we obtain:

$$\underline{A} = \left[V_m, \underline{V} \right] \begin{bmatrix} H_{m+1} & 0_{(m+1) \times (n-m-1)} \\ 0_{(n-m-1) \times (m+1)} & I_{n-m-1} \end{bmatrix} \begin{bmatrix} V_m^H \\ \underline{V}^H \end{bmatrix},$$

Therefore, $\underline{A}$ is similar to the matrix $\begin{bmatrix} H_{m+1} & 0_{(m+1) \times (n-m-1)} \\ 0_{(n-m-1) \times (m+1)} & I_{n-m-1} \end{bmatrix}$, because $\left[V_m, \underline{V} \right]$ is orthonormal. The spectrum of $\underline{A}$ is then equal to the spectrum of H_{m+1}:

$$\lambda(\underline{A}) = \lambda(H_{m+1}).$$

□

Therefore, a spectrum analysis is possible when considering the matrices $\underline{A}$ at each restart. It requires to compute the eigenvalues of H_{m+1}. We propose to compute the eigenvalues of H_{m+1} at the end of each restart and display them on the same plot. The distribution of these eigenvalues will enable us to show some information related to the performance of a given flexible preconditioner.

We now experiment the relevance of this spectrum analysis. When considering preconditioners of different quality, FGMRES - for the same restart parameter - may need more or less iterations to converge, depending on the quality of preconditioner [108]. We plan to show the correlation between the spectrum distributions and the histories of convergence. The easiest way to generate variable preconditioners of different quality is to use full GMRES with different prescribed numbers of iterations as an inner solver. We denote by m_{inner} this number of iterations. and full GMRES with a Krylov subspace of size m_{inner}: $GMRES(m_{inner})$. Therefore, we will use $FGMRES(m)$ preconditioned by $GMRES(m_{inner})$, denoted by $FGMRES(m)/GMRES(m_{inner})$. We will compute the eigenspectra of H_{m+1} at each restart of FGMRES for all m_{inner} values. We denote by $H_{m+1}^{(i)}$ the Hessenberg matrix corresponding to the ith restart and by $\lambda(H_{m+1}^{(i)}(m_{inner}))$ its eigenspectrum corresponding to the inner restart parameter m_{inner}. Finally, we denote by $\Lambda(H_{m+1}(m_{inner}))$ the union of all $\lambda(H_{m+1}^{(i)}(m_{inner}))$ for $i \geq 1$:

$$\Lambda(H_{m+1}(m_{inner})) = \cup_i \, \lambda(H_{m+1}^{(i)}(m_{inner})).$$

This represents the spectrum to be analyzed in our study. We will consider one academic test case and one real life test case from the University of Florida matrix collection [32]. For both test cases, the iterative method is stopped when the normalized residual is below 10^{-6}:

$$\frac{\|b - A\, x_j\|}{\|b\|} \leq 10^{-6}.$$

Example 1: a two-dimensional convection diffusion problem

We consider a two-dimensional convection diffusion problem with Dirichlet boundary conditions in the unit square $[0,1]^2 = \Omega \cup \partial\Omega$ with $\Omega = (0,1)^2$ such as:

$$\begin{cases} -\varepsilon \Delta u + c u_x + d u_y = g & \text{in} \quad \Omega, \\ u = 1 & \text{on} \quad \partial\Omega. \end{cases} \tag{2.4}$$

This problem is discretized with a second-order finite difference scheme for a vertex-centered location of unknowns. The Péclet condition ([115] equation (7.1.9)) is satisfied: $\frac{h}{\varepsilon} \max(|c|, |d|) = 2$ where $h = \frac{1}{N-1}$ is the mesh size and N the number of points per direction. For the spectrum study, we consider a $257^2 = 66049$ grid ($h = \frac{1}{256}$) for $c = d = 512$ and $\varepsilon = 1$. The matrix has 196099 non zero entries, it is real, sparse and non-symmetric. The right-hand side is $b = A\, e$ where e is a vector of ones. We consider as an outer solver FGMRES(5) and five values for m_{inner} : $1, 2, 3, 4, 5$. Histories of convergence are plotted in Figure 2.1. Each symbol on the convergence curves corresponds to one application of the variable preconditioner. We can first notice that the value of m_{inner} has a direct impact on the preconditioner quality: a large value of m_{inner} implies a smaller number of iterations for FGMRES(5). Then, looking at the spectrum in Figure 2.2, we remark that the better the quality of the preconditioner, the larger the minimum value of $\Lambda(H_{m+1}(m_{inner}))$ on the real axis. Therefore, there is a correlation for this model problem between the quality of the inner preconditioner and the distribution of $\Lambda(H_{m+1}(m_{inner}))$.

Example 2: a three-dimensional Navier-Stokes problem

We now consider a matrix from the FIDAP group in the University of Florida collection, the $ex11$ matrix [1]. This matrix is real, sparse and non-symmetric. Its dimension is $16,614$ and it has $1,096,948$ non zero entries. It models a three-dimensional fully coupled Navier-Stokes problem. As advised in [108], we use a diagonal preconditioner for the inner GMRES. The right-hand side is $b = A\,e$ where e is a vector of ones. We perform the same tests as for the convection-diffusion problem: we consider as an outer solver FGMRES(5) and five values for m_{inner} : $1, 2, 3, 4, 5$. Histories of convergence are plotted in Figure 2.3. Increasing m_{inner} tends to decrease significantly the number of iterations. However a different behavior can be observed for $m_{inner} = 2$. Indeed, although $FGMRES(5)/GMRES(2)$ is converging faster than $FGMRES(5)/GMRES(1)$, before its normalized residual is below 5.10^{-6}, a large plateau appears close to convergence. Such behaviors have already been remarked in [39], yet their analysis has been done for small restart parameters and GMRES without preconditioning. Our spectrum analysis gives information about this behavior in a more general framework. Indeed, looking at Figure 2.4, for $m_{inner} = 2$, the minimum value of $\Lambda(H_{m+1}(2))$ on the real axis is negative (-3.55×10^{-4}), whereas the one related to $\Lambda(H_{m+1}(1))$ is positive (1.59×10^{-4}). It seems then that GMRES applied to a matrix with a spectrum distribution such as $\Lambda(H_{m+1}(2))$ would converge slower than when it is applied to a matrix with a spectrum distribution such as $\Lambda(H_{m+1}(1))$. For the other values of m_{inner}, we remark that the better is the quality of the preconditioner, the larger is the minimum value on the real axis of $\Lambda(H_{m+1}(m_{inner}))$ and the fewer are the eigenvalues close to zero.

Thus, this spectrum analysis can give some indication why a preconditioner could be efficient or not looking at the H_{m+1} along the restart. Indeed, if the minimal real part value of $\Lambda(H_{m+1})$ is positive, or if its maximal real part is negative, and far from the origin, the preconditioner may improve the convergence. Notwithstanding, if the spectrum of H_{m+1} has values with a negative real part and values with a positive real part, convergence may be slow even if preconditioning is performed.

This study points out once again the practical importance of the spectrum distribution for Krylov methods even if a flexible preconditioner is used. However, as for the non flexible case, this result has to be balanced with the theoretical result by Greenbaum, Ptak and Strakos [56]: any convergence curve can be generated by GMRES applied to a matrix having any desired eigenvalues. Nevertheless, since the right-hand side is fixed, it remains an useful tool to understand in more details the convergence of FGMRES.

Besides, this approximate spectral information can be easily computed thanks to the Hessenberg matrix (Ritz, harmonic Ritz vectors). It could even be used to improve the convergence properties of FGMRES. This has already been realized for GMRES with GMRES-DR [84] which preserves spectral information from one restart to the next. Thus, we propose to extend such a technique to the flexible case. Therefore, after depicting the GMRES-DR in the next section, we will present the flexible variant of GMRES-DR method: FGMRES-DR [53].

[1] http://www.cise.ufl.edu/research/sparse/matrices/FIDAP/ex11.html

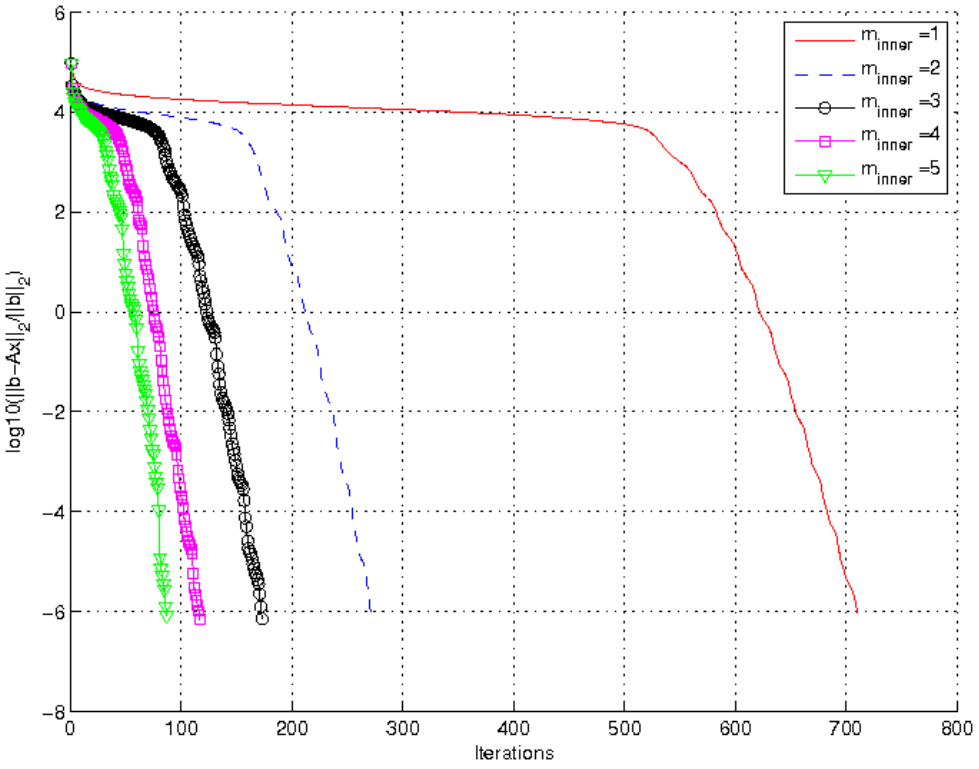

Figure 2.1: Histories of convergence for the convection-diffusion problem of *FGMRES*(5) preconditioned by full *GMRES*(m_{inner}) for different values of m_{inner}.

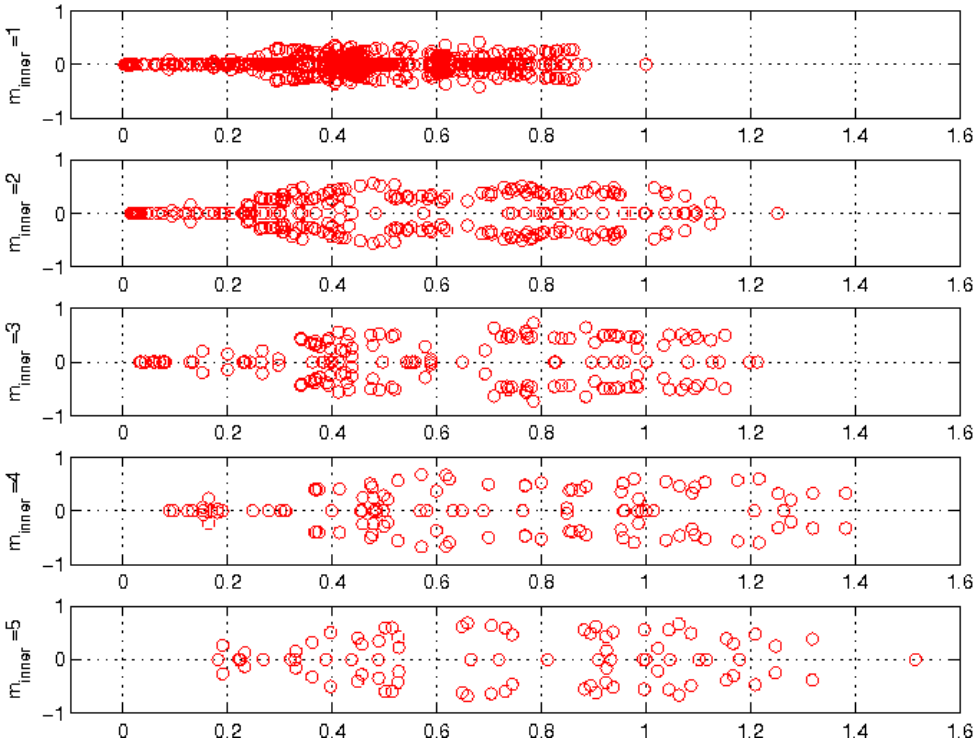

Figure 2.2: Plot of $\Lambda(H_{m+1}(m_{inner}))$ with the convection-diffusion problem, for *FGMRES*(5) preconditioned by a full *GMRES*(m_{inner}) for different values of m_{inner}.

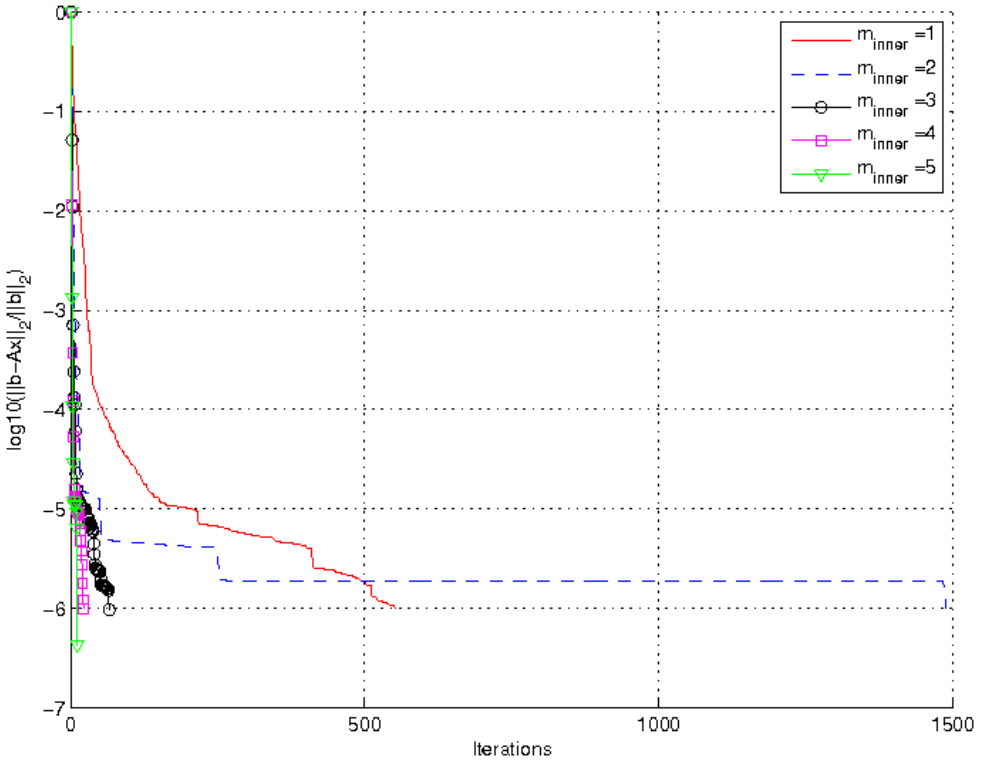

Figure 2.3: Histories of convergence for the FIDAP-ex11 matrix of $FGMRES(5)$ preconditioned by a diagonal preconditioned full $GMRES(m_{inner})$ for different values of m_{inner}.

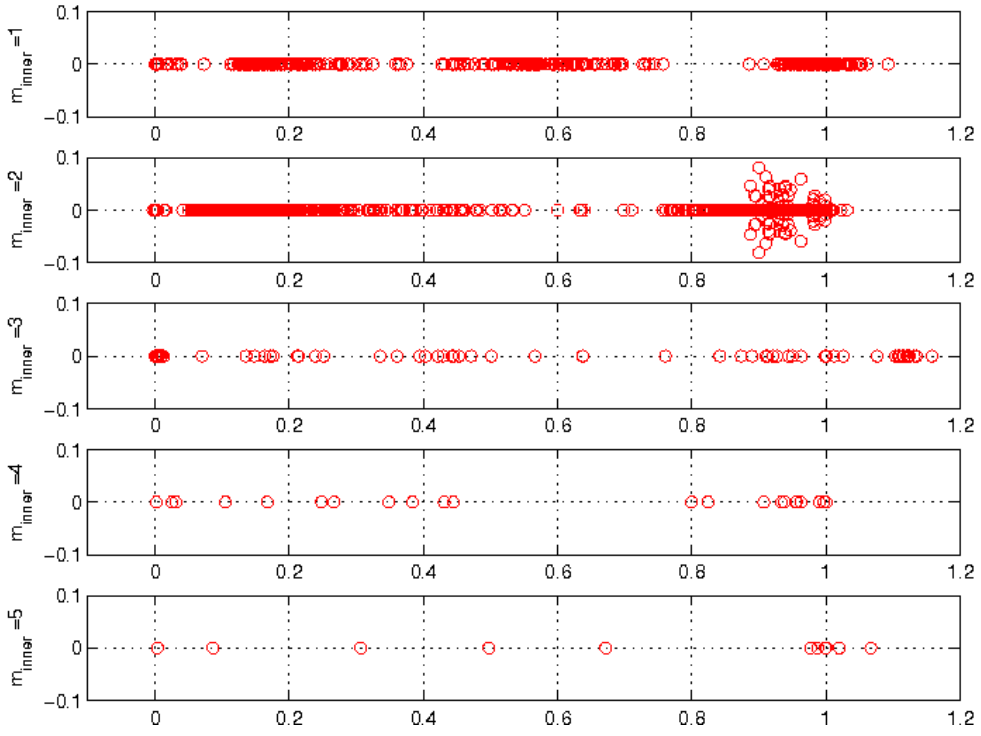

Figure 2.4: Plot of $\Lambda(H_{m+1}(m_{inner}))$ for the FIDAP-ex11 matrix, with $FGMRES(5)$ preconditioned by a diagonal preconditioned full $GMRES(m_{inner})$ for different values of m_{inner}.

2.4 GMRES with deflated restarting

Krylov subspace methods with standard restarting implement a scheme where the maximal dimension of the approximation subspace is fixed (m here). After m steps, the method is then restarted, in order to control both the memory requirements and the computational cost of the orthogonalization scheme of the method. In the case of GMRES(m) it means in practice that the orthonormal basis is thrown away after m steps. Since some information is discarded at the restart, the convergence is expected to be slower compared to full GMRES.

Nevertheless more sophisticated procedures have been proposed to enhance convergence properties of restarted Krylov subspace methods. Basically these methods fall in the category of augmented or deflated methods and we refer the reader to [109, Sections 8 and 9] for a review and detailed references. In this section we focus on GMRES with deflated restarting, and more particularly to one of those methods, referred to as GMRES-DR [84]. This method aims at using spectral information at a restart mainly to improve the convergence of restarted GMRES. A subspace of dimension k (with $k < m$) spanned by harmonic Ritz vectors (and not only the approximate solution with minimum residual norm) is retained in this restarting scheme. Property 1 describes how this subspace of dimension k is obtained in GMRES with deflated restarting, when a fixed right preconditioning matrix noted M is considered.

Before introducing the principle of GMRES-DR, we recall the definition of a harmonic Ritz pair [91, 110] since this notion plays an important role when considering deflated restarting.

Definition 2. *Harmonic Ritz pair. Consider a subspace $\mathcal{U}$ of $\mathbb{C}^n$. Given a matrix $B \in \mathbb{C}^{n\times n}$, $\lambda \in \mathbb{C}$ and $y \in \mathcal{U}$, (λ, y) is a harmonic Ritz pair of B with respect to $\mathcal{U}$ if and only if*

$$By - \lambda y \perp B\mathcal{U}$$

or equivalently, for the canonical scalar product,

$$\forall w \in \text{range}(B\mathcal{U}) \quad w^H (By - \lambda y) = 0.$$

We call y a harmonic Ritz vector associated with the harmonic Ritz value λ.

Property 1. *GMRES with deflated restarting relies on the computation of k harmonic Ritz vectors $Y_k = V_m G_k$ of $AM^{-1}V_m V_m^H$ with respect to* range(V_m) *with $Y_k \in \mathbb{C}^{n\times k}$ and $G_k \in \mathbb{C}^{m\times k}$.*

Proof. Let us denote $Y_k = [y_1, \ldots, y_k]$ and $G_k = [g_1, \ldots, g_k]$. Since $y_j = V_m g_j$ is a harmonic Ritz vector of $AM^{-1}V_m V_m^H$ with respect to range(V_m), the following relation holds (see Definition 2)

$$(AM^{-1}V_m V_m^H V_m)^H (AM^{-1}V_m V_m^H y_j - \lambda_j y_j) = 0 \tag{2.5}$$

which is equivalent to

$$(AM^{-1}V_m)^H (AM^{-1}V_m g_j - \lambda_j V_m g_j) = 0. \tag{2.6}$$

Thanks to the Arnoldi relation $AM^{-1}V_m = V_{m+1}\bar{H}_m$ we deduce

$$\bar{H}_m^H \bar{H}_m g_j - \lambda_j \bar{H}_m^H \begin{pmatrix} g_j \\ 0 \end{pmatrix} = 0. \tag{2.7}$$

Since $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ has the following form

$$\bar{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m} e_m^T \end{bmatrix}$$

where $H_m \in \mathbb{C}^{m\times m}$ is supposed to be non-singular, the eigenvalue problem becomes then

$$(H_m + |h_{m+1,m}|^2 H_m^{-H} e_m e_m^T) g_j - \lambda_j g_j = 0 \tag{2.8}$$

which corresponds to the formulation originally proposed by Morgan [84].

□

Next, the QR factorization of the following $(m+1)\times(k+1)$ matrix

$$\left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix} \quad V_{m+1}^H\, r_0\right] = \left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix} \quad c - \bar{H}_m y^*\right] \quad \text{with} \quad r_0 = V_{m+1}(c - \bar{H}_m y^*)$$

is performed where $c \in \mathbb{C}^{m+1}$ and $y^* \in \mathbb{C}^m$. This allows to compute new matrices $V_{k+1}^{new} \in \mathbb{C}^{n\times(k+1)}$ and $\bar{H}_k^{new} \in \mathbb{C}^{(k+1)\times k}$ such that

$$\begin{aligned} AM^{-1}V_k^{new} &= V_{k+1}^{new}\,\bar{H}_k^{new}, \\ V_{k+1}^{newH}\,V_{k+1}^{new} &= I_{k+1}, \\ \text{range}([Y_k, r_0]) &= \text{range}(V_{k+1}^{new}) \end{aligned}$$

where $\bar{H}_k^{new}$ is a $(k+1)\times k$ rectangular matrix. GMRES-DR then carries out $m-k$ Arnoldi steps with fixed preconditioning and starting vector v_{k+1}^{new} to eventually build V_{m+1} and $\bar{H}_m$. At the end of the GMRES cycle with deflated restarting we have a final relation similar to the Arnoldi relation (2.1) with $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ and $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$

$$AM^{-1}V_m = V_{m+1}\bar{H}_m \quad \text{with} \quad V_{m+1}^H\,V_{m+1} = I_{m+1}$$

where $\bar{H}_m$ is *no longer* upper Hessenberg after the first cycle. An approximate solution $x_m \in \mathbb{C}^n$ is then found by minimizing the residual norm $\|b - A(x_0 + M^{-1}V_m y)\|$ over the space $x_0 + M^{-1}\text{range}(V_m)$, the corresponding residual being $r_m = b - Ax_m \in \mathbb{C}^n$ with $r_m \in \text{range}(V_{m+1})$. An optimality property is thus also obtained. We refer the reader to [84, 98] for further comments on the algorithm and computational details. This approach has proved efficient on many academic examples [84]. We note that GMRES with deflated restarting is equivalent to GMRES with eigenvectors [82] and to implicitly restarted GMRES [83]. Details of the method are given in Algorithms 5 and 6 respectively. GMRES-DR(m,k) does require only $m-k$ matrix vector products and preconditioning operations per cycle while GMRES(m) needs m. Finally we note that Krylov subspace methods with deflated restarting have been exclusively developed in the case of a fixed preconditioner. In Section 2.5 we extend the GMRES-DR method to the case of variable preconditioning.

Algorithm 5 Right-preconditioned GMRES with deflated restarting: GMRES-DR(m,k)

1: *Initialization:* Choose $m > 0$, $k > 0$, $tol > 0$, $x_0 \in \mathbb{C}^n$. Let $r_0 = b - Ax_0$; $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T \in \mathbb{C}^{m+1}$, $v_1 = r_0/\beta$.

2: *Computation of V_{m+1} and $\bar{H}_m$:* Apply m steps of the Arnoldi procedure (algorithm 2) with right preconditioning to obtain $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ and the upper Hessenberg matrix $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ such that:

$$AM^{-1}V_m = V_{m+1}\bar{H}_m \quad \text{with} \quad V_{m+1}^H V_{m+1} = I_{m+1}.$$

Loop

3: *Minimum norm solution:* Compute the minimum norm solution $x_m \in \mathbb{C}^n$ in the affine space $x_0 + M^{-1}\text{range}(V_m)$; that is, $x_m = x_0 + M^{-1}V_m y^*$ where $y^* = \underset{y\in\mathbb{C}^m}{\text{argmin}}\,\|c - \bar{H}_m y\|$. Set $x_0 = x_m$ and $r_0 = b - Ax_0$.

4: *Check the convergence criterion:* If $\|c - \bar{H}_m y^*\|/\|b\| \le tol$, exit

5: *Computation of V_{k+1}^{new} and $\bar{H}_k^{new}$:* see Algorithm 6. At the end of this step the following relations hold:

$$AM^{-1}V_k^{new} = V_{k+1}^{new}\bar{H}_k^{new} \text{ with } V_{k+1}^{newH}V_{k+1}^{new} = I_{k+1} \text{ and } r_0 \in \text{range}(V_{k+1}^{new}).$$

6: *Arnoldi procedure:* Set $V_{k+1} = V_{k+1}^{new}$, $\bar{H}_k = \bar{H}_k^{new}$ and apply $(m-k)$ steps of the Arnoldi procedure with right preconditioning and starting vector v_{k+1} to build $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ and $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ such that:

$$AM^{-1}V_m = V_{m+1}\bar{H}_m \quad \text{with} \quad V_{m+1}^H V_{m+1} = I_{m+1}.$$

7: *Setting:* Set $c = V_{m+1}^H r_0$.

End of loop

Algorithm 6 GMRES-DR(m,k): computation of V_{k+1}^{new} and $\bar{H}_k^{new}$

1: *Input:* A, V_{m+1} such that $AM^{-1}V_m = V_{m+1}\bar{H}_m$ and $c - \bar{H}_m y^*$ such that $r_0 = V_{m+1}(c - \bar{H}_m y^*)$.
2: *Settings:* Define $h_{m+1,m} = \bar{H}_m(m+1,m)$, $H_m \in \mathbb{C}^{m\times m}$ as $H_m = \bar{H}_m(1:m,1:m)$.
3: *Compute k harmonic Ritz vectors:* Compute k independent eigenvectors g_i of the matrix $H_m + |h_{m+1,m}|^2 H_m^{-H} e_m e_m^T$. Set $G_k = [g_1,\ldots,g_k] \in \mathbb{C}^{m\times k}$.
4: *Augmentation of G_k:* Define $G_{k+1} \in \mathbb{C}^{(m+1)\times(k+1)}$ as

$$G_{k+1} = \left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix}, c - \bar{H}_m y^*\right].$$

5: *Orthonormalization of the columns of G_{k+1}:* Perform a QR-factorization of G_{k+1} as $G_{k+1} = P_{k+1}\Gamma_{k+1}$. Define $P_k \in \mathbb{C}^{m\times k}$ as $P_k = P_{k+1}(1:m,1:k)$.
6: *Settings and final relation:* Set $V_{k+1}^{new} = V_{m+1}P_{k+1}$ and $\bar{H}_k^{new} = P_{k+1}^H \bar{H}_m P_k$. At the end of this step the following relations are satisfied:

$$AM^{-1}V_m P_k = V_{m+1}P_{k+1}P_{k+1}^H\bar{H}_m P_k \quad ;\text{i.e.,} \quad AM^{-1}V_k^{new} = V_{k+1}^{new}\,\bar{H}_k^{new}$$

where $\bar{H}_k^{new}$ is generally a dense matrix.

2.5 Flexible GMRES with deflated restarting

In this section we present the new subspace method that allows deflated restarting and variable preconditioning simultaneously. We suppose that a flexible Arnoldi relation holds ($AZ_m = V_{m+1}\bar{H}_m$) and analyze one cycle of this method.

2.5.1 Analysis of a cycle

We discuss now the two main points related to the extension of GMRES-DR in a flexible setting: what is the harmonic Ritz information recovered at restart and is it still possible as in GMRES-DR to restart at low computational cost the flexible Arnoldi relation? Both questions will be answered in this section.

Harmonic Ritz formulation

Property 2 presents the harmonic Ritz formulation used in the flexible variant of GMRES with deflated restarting. It is a straightforward adaptation of Property 1 now when flexible preconditioning is considered.

Property 2. *Flexible GMRES with deflated restarting relies on the computation of k harmonic Ritz vectors $Y_k = V_m G_k$ of $AZ_m V_m^H$ with respect to* range(V_m) *with $Y_k \in \mathbb{C}^{n\times k}$ and $G_k \in \mathbb{C}^{m\times k}$ respectively.*

Proof. Following Definition 2, each harmonic Ritz pair $(\lambda_k, V_m g_k)$ satisfies the following relation

$$\forall w \in \text{range}(AZ_m V_m^H V_m) \quad w^H\,(AZ_m V_m^H V_m g_k - \lambda_k\, V_m g_k) = 0,$$

or equivalently since $V_m^H V_m = I_m$

$$\forall w \in \text{range}(AZ_m) \quad w^H\,(AZ_m g_k - \lambda_k\, V_m g_k) = 0, \tag{2.9}$$

where λ_k denotes the harmonic Ritz value associated to $V_m g_k$. Exploiting the flexible Arnoldi relation $AZ_m = V_{m+1}\bar{H}_m$ leads to the following eigenvalue problem

$$\bar{H}_m^H \bar{H}_m g_j - \lambda_j \bar{H}_m^H \begin{pmatrix} g_i \\ 0 \end{pmatrix} = 0$$

or equivalently

$$(H_m + |h_{m+1,m}|^2 H_m^{-H} e_m e_m^T) g_j - \lambda_j g_j = 0 \tag{2.10}$$

which is the same as in GMRES with deflated restarting (see relation (2.8)). Due to relation (2.9) we also note that the harmonic residual vectors $AZ_m V_m^H V_m g_k - \lambda_k V_m g_k \in \text{range}(V_{m+1})$ are orthogonal to a subspace of dimension m spanned by the columns of AZ_m.

□

In Lemma 1 we detail a useful relation satisfied by the harmonic Ritz vectors.

Lemma 1. *In Flexible GMRES with deflated restarting, the harmonic Ritz vectors are given by $Y_k = V_m G_k$ with corresponding harmonic Ritz values λ_k. $G_k \in \mathbb{C}^{m\times k}$ satisfies the following relation:*

$$AZ_m G_k = V_{m+1}\left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix}, \rho_m\right]\begin{bmatrix} \text{diag}(\lambda_1, \cdots, \lambda_k) \\ \alpha_{1\times k}\end{bmatrix} \tag{2.11}$$

where $\rho_m \in \mathbb{C}^{m+1}$ is such that $r_0 = V_{m+1}\rho_m = V_{m+1}(c - \bar{H}_m y^*)$ and $\alpha_{1\times k} = [\alpha_1, \cdots, \alpha_k] \in \mathbb{C}^{1\times k}$.

Proof. The harmonic residual vectors $AZ_m V_m^H V_m g_i - \lambda_i V_m g_i$ and the residual vector r_0 all reside in a subspace of dimension $m+1$ (spanned by the columns of V_{m+1}) and are orthogonal to the same subspace of dimension m (spanned by the columns of AZ_m, a subspace of range(V_{m+1}), so they must be collinear. Consequently there exist k coefficients noted $\alpha_i \in \mathbb{C}$ with $1 \le i \le k$ such that

$$\forall i \in \{1, \cdots, k\}\ AZ_m g_i - \lambda_i V_m g_i = \alpha_i r_0 = \alpha_i V_{m+1}\rho_m. \tag{2.12}$$

Setting $\alpha_{1\times k} = [\alpha_1, \cdots, \alpha_k] \in \mathbb{C}^{1\times k}$, the collinearity expression (2.12) can be written in matrix form

$$AZ_m G_k = V_{m+1}\left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix}, \rho_m\right]\begin{bmatrix} \text{diag}(\lambda_1, \cdots, \lambda_k) \\ \alpha_{1\times k}\end{bmatrix}.$$

□

Flexible Arnoldi relation

Let us further denote by $G_k = P_k\Gamma_k$ the QR-factorization of G_k, where $P_k \in \mathbb{C}^{m\times k}$ has orthonormal columns and $\Gamma_k \in \mathbb{C}^{k\times k}$ is a non-singular upper triangular matrix. We denote $G_{k+1} \in \mathbb{C}^{(m+1)\times(k+1)}$ the following matrix that appears in Lemma 1:

$$G_{k+1} = \left[\begin{bmatrix} G_k \\ 0_{1\times k}\end{bmatrix}, \rho_m\right]. \tag{2.13}$$

Proposition 3 shows that a flexible Arnoldi relation can be recovered at low computational cost when restarting with some harmonic information; i.e., without involving any matrix-vector product with A as in [23].

Proposition 3. *At each restart of Flexible GMRES with deflated restarting, the flexible Arnoldi relation*

$$AZ_k^{new} = V_{k+1}^{new}\bar{H}_k^{new}$$

holds with

$$Z_k^{new} = Z_m P_k,$$
$$V_{k+1}^{new} = V_{m+1}P_{k+1},$$

and

$$\bar{H}_k^{new} = P_{k+1}^H \bar{H}_m P_k.$$

Proof. After orthogonalization of the vector ρ_m against the columns of $\begin{bmatrix} P_k \\ 0_{1\times k}\end{bmatrix}$ we obtain the unit norm vector $p_{k+1} \in \mathbb{C}^{m+1}$ that satisfies

$$p_{k+1} = \bar{p}_{k+1}/\|\bar{p}_{k+1}\| \quad \text{with} \quad \bar{p}_{k+1} = \rho_m - \begin{bmatrix} P_k \\ 0_{1\times k}\end{bmatrix}\begin{bmatrix} P_k \\ 0_{1\times k}\end{bmatrix}^H \rho_m.$$

We note $a = \|\bar{p}_{k+1}\|$ and $u_{k\times 1} \in \mathbb{C}^k$ the following quantity $u_{k\times 1} = \begin{bmatrix} P_k \\ 0_{1\times k} \end{bmatrix}^H \rho_m$ respectively. Thus

$$\rho_m = \left[\begin{bmatrix} P_k \\ 0_{1\times k} \end{bmatrix},\ p_{k+1}\right]\begin{bmatrix} u_{k\times 1} \\ a \end{bmatrix}.$$

Consequently the QR factorization of $G_{k+1} = P_{k+1}\Gamma_{k+1}$ can be written as

$$\left[\begin{bmatrix} G_k \\ 0_{1\times k} \end{bmatrix},\ \rho_m\right] = \left[\begin{bmatrix} P_k \\ 0_{1\times k} \end{bmatrix},\ p_{k+1}\right]\begin{bmatrix} \Gamma_k & u_{k\times 1} \\ 0_{1\times k} & a \end{bmatrix}.$$

From relation (2.11) of Lemma 1 we deduce

$$AZ_mP_k = V_{m+1}P_{k+1}\Gamma_{k+1}\begin{bmatrix} \mathrm{diag}(\lambda_1,\cdots,\lambda_k) \\ \alpha_{1\times k} \end{bmatrix}\Gamma_k^{-1}. \tag{2.14}$$

Using the flexible Arnoldi relation $AZ_m = V_{m+1}\bar{H}_m$ and $P_{k+1}^HP_{k+1} = I_{k+1}$ we obtain

$$P_{k+1}^H\bar{H}_mP_k = \Gamma_{k+1}\begin{bmatrix} \mathrm{diag}(\lambda_1,\cdots,\lambda_k) \\ \alpha_{1\times k} \end{bmatrix}\Gamma_k^{-1}.$$

If we denote $Z_k^{new} = Z_mP_k$, $V_{k+1}^{new} = V_{m+1}P_{k+1}$ and

$$\bar{H}_k^{new} \quad = \quad \Gamma_{k+1}\begin{bmatrix} \mathrm{diag}(\lambda_1,\cdots,\lambda_k) \\ \alpha_{1\times k} \end{bmatrix}\Gamma_k^{-1} = P_{k+1}^H\bar{H}_mP_k,$$

Equation (2.14) can be written in the following flexible Arnoldi relation

$$AZ_k^{new} = V_{k+1}^{new}\bar{H}_k^{new}.$$

□

Next, setting $Z_k = Z_k^{new}$, $V_{k+1} = V_{k+1}^{new}$ and $\bar{H}_k = \bar{H}_k^{new}$ respectively flexible GMRES with deflated restarting then carries out $(m-k)$ flexible Arnoldi steps with flexible preconditioning and starting vector v_{k+1} leading to

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

where $Z_m \in \mathbb{C}^{n\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ and $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$.

2.5.2 Algorithm and computational aspects

Details of flexible GMRES with deflated restarting are depicted in Algorithms 7 and 8 respectively. We will call this algorithm FGMRES-DR(m,k) and compare this method with both FGMRES(m) and GMRES-DR(m,k) from a computational and storage point of view.

Computational cost

We summarize now in Table 2.1 the main computational costs associated with each generic cycle of FGMRES(m), GMRES-DR(m,k) and FGMRES-DR(m,k). We have only included the costs proportional to the size of the original problem n which is supposed to be much larger than m and k. We denote op_A and op_M the floating point operation counts for the matrix-vector product and the preconditioner application respectively. The main computational differences are in the calculation of V_{k+1} and Z_k when comparing FGMRES and FGMRES-DR. In FGMRES-DR those vectors are computed using dense matrix-matrix operations efficiently implemented in BLAS-3 libraries, while in FGMRES they are obtained through a sequence of matrix-vector products, possibly sparse, depending on the nature of A and the preconditioners.

Algorithm 7 Flexible GMRES with deflated restarting: FGMRES-DR(m, k)

1: *Initialization:* Choose $m > 0$, $k > 0$, $tol > 0$, $x_0 \in \mathbb{C}^n$. Let $r_0 = b - Ax_0$; $\beta = \|r_0\|$, $c = [\beta, 0_{1\times m}]^T \in \mathbb{C}^{m+1}$, $v_1 = r_0/\beta$.

2: *Computation of V_{m+1}, Z_m and $\bar{H}_m$:* Apply m steps of the Arnoldi procedure with *flexible* preconditioning (Algorithm 4) to obtain $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$, $Z_m \in \mathbb{C}^{n\times m}$ and the upper Hessenberg matrix $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ such that:

$$AZ_m = V_{m+1}\bar{H}_m \quad \text{with} \quad V_{m+1}^H V_{m+1} = I_{m+1}.$$

Loop

3: *Minimum norm solution:* Compute the minimum norm solution $x_m \in \mathbb{C}^n$ in the affine space $x_0 + \text{range}(Z_m)$; that is, $x_m = x_0 + Z_m y^*$ where $y^* = \underset{y\in\mathbb{C}^m}{\text{argmin}} \|c - \bar{H}_m y\|$. Set $x_0 = x_m$ and $r_0 = b - Ax_0$.

4: *Check the convergence criterion:* If $\|c - \bar{H}_m y^*\|/\|b\| \leq tol$, exit

5: *Computation of V_{k+1}^{new}, Z_k^{new} and $\bar{H}_k^{new}$:* see Algorithm 8. At the end of this step the following relations hold:

$$AZ_k^{new} = V_{k+1}^{new}\bar{H}_k^{new} \text{ with } V_{k+1}^{newH}V_{k+1}^{new} = I_{k+1} \text{ and } r_0 \in \text{range}(V_{k+1}^{new}). \tag{2.15}$$

6: *Arnoldi procedure:* Set $V_{k+1} = V_{k+1}^{new}$, $Z_k = Z_k^{new}$, $\bar{H}_k = \bar{H}_k^{new}$ and apply $(m-k)$ steps of the Arnoldi procedure with *flexible* preconditioning and starting vector v_{k+1} to build $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$, $Z_m \in \mathbb{C}^{n\times m}$ and $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ such that:

$$AZ_m = V_{m+1}\bar{H}_m \quad \text{with} \quad V_{m+1}^H V_{m+1} = I_{m+1}.$$

7: *Setting:* Set $c = V_{m+1}^H r_0$.

End of loop

Algorithm 8 FGMRES-DR(m, k): computation of V_{k+1}^{new}, Z_k^{new} and $\bar{H}_k^{new}$

1: *Input:* A, Z_m, V_{m+1} such that $AZ_m = V_{m+1}\bar{H}_m$ and $c - \bar{H}_m y^*$ such that $r_0 = V_{m+1}(c - \bar{H}_m y^*)$.

2: *Settings:* Define $h_{m+1,m} = \bar{H}_m(m+1, m)$, $H_m \in \mathbb{C}^{m\times m}$ as $H_m = \bar{H}_m(1:m, 1:m)$.

3: *Compute k harmonic Ritz vectors.* Compute k independent eigenvectors g_i of the matrix $H_m + |h_{m+1,m}|^2 H_m^{-H} e_m e_m^T$. Set $G_k = [g_1, \ldots, g_k] \in \mathbb{C}^{m\times k}$.

4: *Augmentation of G_k:* Define $G_{k+1} \in \mathbb{C}^{(m+1)\times(k+1)}$ as

$$G_{k+1} = \begin{bmatrix} \begin{bmatrix} G_k \\ 0_{1\times k} \end{bmatrix}, c - \bar{H}_m y^* \end{bmatrix}. \tag{2.16}$$

5: *Orthonormalization of the columns of G_{k+1}:* Perform a QR-factorization of G_{k+1} as $G_{k+1} = P_{k+1}\Gamma_{k+1}$. Define $P_k \in \mathbb{C}^{m\times k}$ as $P_k = P_{k+1}(1:m, 1:k)$.

6: *Settings and final relation:* Set $V_{k+1}^{new} = V_{m+1}P_{k+1}$, $Z_k^{new} = Z_m P_k$ and $\bar{H}_k^{new} = P_{k+1}^H \bar{H}_m P_k$, so that the following relations are satisfied:

$$AZ_m P_k = V_{m+1}P_{k+1}P_{k+1}^H \bar{H}_m P_k \quad ; \text{i.e.,} \quad AZ_k^{new} = V_{k+1}^{new}\bar{H}_k^{new} \tag{2.17}$$

where $\bar{H}_k^{new}$ is generally a dense matrix.

For deflating variants, the reduction of this total cost is still possible. The right-hand side c of the least-squares problem is computed as $c = V_{m+1}^H r_0$ which involves $2n(m+1)$ operations as shown in Table 2.1. This cost can be first reduced by observing that the residual r_0 belongs to the subspace spanned by the columns of V_{k+1}, consequently only its first $(k+1)$ entries are non-zero. These quantities can be obtained by computing $V_{k+1}^H r_0$ and it only requires $2n(k+1)$ operations. This has been notably investigated in [98]. The calculation of c can be even more reduced as described in Proposition 4.

Computation of	FGMRES(m)	GMRES-DR(m,k)	FGMRES-DR(m,k)
$V_m(:,1:k+1)$	$kop_A + nk(2k+5)$	$2n(m+1)(k+1)$	$2n(m+1)(k+1)$
$Z_m(:,1:k)$	kop_M	-	$2nmk$
$V_m(:,k+2:m+1)$	$(m-k)op_A+$ $n(m-k)(2m+2k+5)$	$(m-k)(op_A+op_M)+$ $n(m-k)(2m+2k+5)$	$(m-k)op_A+$ $n(m-k)(2m+2k+5)$
$Z_m(:,k+1:m)$	$(m-k)op_M$	-	$(m-k)op_M$
c	2n	$2n(m+1)$	$2n(m+1)$

Table 2.1: Computational cost of a generic cycle of FGMRES(m), GMRES-DR(m,k) and FGMRES-DR(m,k).

Proposition 4. *The first* $(k+1)$ *components of the right-hand side* c *of the next least-squares problem are given by the last column of* Γ_{k+1}*, the triangular factor of the QR factorization of the matrix* G_{k+1} *defined in relation (2.13).*

Proof. In Proposition 3 we have shown that $\rho_m = P_{k+1}\begin{bmatrix} u_{k\times 1} \\ a \end{bmatrix}$. Consequently $r_0 = V_{m+1}\rho_m = V_{k+1}^{new}\begin{bmatrix} u_{k\times 1} \\ a \end{bmatrix}$. Thus the right-hand side of the new least-squares problem is given by

$$c = V_{m+1}^H r_0 = V_{m+1}^H V_{k+1}^{new}\begin{bmatrix} u_{k\times 1} \\ a \end{bmatrix} = \begin{bmatrix} u_{k\times 1} \\ a \\ 0_{(m-k)\times 1} \end{bmatrix}.$$

□

We note that Proposition 4 holds for both GMRES-DR(m,k) and FGMRES-DR(m,k).

Storage requirements

Regarding storage, we have only included the storage proportional to the size of the original problem n which is supposed to be much larger than m and k.

Standard With this convention FGMRES-DR(m,k) requires the storage of Z_m, V_{m+1} and at most $k+1$ additional vectors to store in turn V_{k+1}^{new} and Z_k^{new}. Thus FGMRES-DR(m,k) requires the storage of $(2m+k+2)$ vectors of length n.

Buffered If an extra memory block of *buffer size* can be allocated, a blocked matrix-matrix product can be implemented to perform $V_{k+1}^{new} = V_{m+1}P_{k+1}$ and $Z_k^{new} = Z_m P_k$, that computes these matrices block-row by block-row before overwriting the result in the data structure allocated for V_{m+1} (Z_m respectively). The definition of this block size can be governed by the BLAS-3 performance of the targeted computer.

Economic A reduction of storage is however still possible. It can indeed be remarked that Z_k^{new} and V_{k+1}^{new} can overwrite Z_k and V_{k+1}. This can be accomplished by performing the matrix multiplications $V_{k+1} \leftarrow V_{m+1}P_{k+1}$ and $Z_k \leftarrow Z_m P_k$ of Step 6 in Algorithm 8 *in place*, i.e., within the arrays V_{m+1} and Z_m. Here we have exploited the fact that multiplications involving triangular factors can be done in place. It is therefore advisable to perform a LU factorization with complete pivoting of P_{k+1} to obtain a very good approximation $\Pi P_{k+1}\Sigma = LU$, and then, to perform successively the operations $X \leftarrow XL$ and $X \leftarrow XU$ and the corresponding permutations e.g. for X being V. This approach leads to a storage of $(2m+1)$ vectors of length n only. It is clearly saving a lot of memory when k is close to m, but may introduce additional round-off errors that can hopefully be monitored by inspecting the quantity $\dfrac{\|\Pi P_k\Sigma - LU\|}{\|P_k\|}$.

Table 2.2 summarizes the requirements related to the storage for both GMRES-DR(m,k) and FGMRES-DR(m,k). We note that the economic variant of FGMRES-DR(m,k) needs the same amount of memory as FGMRES(m) and that flexible variants require m additional vectors with respect to non flexible variants.

Strategy	GMRES-DR(m,k)	FGMRES-DR(m,k)
Standard	$n(m+k+2)$	$n(2m+k+2)$
Buffered	$n(m+1)+buff\ size$	$n(2m+1)+buff\ size$
Economic	$n(m+1)$	$n(2m+1)$

Table 2.2: Storage required for GMRES-DR(m,k) and FGMRES-DR(m,k).

2.5.3 Numerical experiments

In this section we investigate the numerical behavior of the FGMRES-DR(m,k) algorithm on academic problems. We consider the case of both sparse matrices in either real or complex arithmetic. All the examples include a detailed comparison with FGMRES(m). This allows us to show the effects of incorporating the deflation strategy in the flexible preconditioning framework.

In the following experiments, the right-hand sides are computed as $b = A\mathbf{1}$ where $\mathbf{1}$ is the vector of appropriate dimension with all components equal to one. A zero initial iterate x_0 is considered as an initial guess and the following stopping criterion is used:

$$\frac{\|b - A\,x_j\|}{\|b\|} \leq 10^{-12} \tag{2.18}$$

where j represents the step when the iterations are stopped. The choice of such a small tolerance relies on the fact that methods have to be restarted to be compared. Indeed, spectral deflation occurs only when the methods are restarted. If a larger convergence threshold is chosen, convergence could occur before restarting and no relevant comparison could be done.

Harwell-Boeing and Matrix Market test problems

In order to illustrate the numerical behavior of FGMRES-DR(m,k), we first consider a few test matrices from the Harwell-Boeing [33] and Matrix Market [13] libraries so that any reader could reproduce these experiments. The sparse matrices named Sherman4, Saylor4 and Young1c have been chosen. Sherman4 and Saylor4 are real matrices, whereas Young1c is a complex-valued one. They represent challenging sparse matrices coming from realistic applications (reservoir modeling, acoustics) that are often used to analyze the behavior of numerical algorithms. For those experiments, the preconditioner consists in five steps of preconditioned full GMRES, where the preconditioner is based on an ILU(0) factorization. In the case of Sherman4 only, the inner solver corresponds to five steps of unpreconditioned full GMRES.

In Table 2.3, we depict the total number of matrix-vector products performed in the inner and outer parts of the solver (Mv) and the total number of dot products (dot) for several flexible methods. We also display the ratios of total memory and total floating point operations where the reference is the corresponding quantity of the full FGMRES method; i.e.,

$$r_{ops} = \frac{flops(Krylov\ solver)}{flops(full\ FGMRES)} \text{ and } r_{mem} = \frac{mem(Krylov\ solver)}{mem(full\ FGMRES)}, \tag{2.19}$$

where we assume that the memory allocated for full FGMRES is exactly what is needed to store Z_j and V_{j+1}, j being the step where convergence is achieved.

In order to illustrate the possible benefit of using the economic implementation presented in Section 2.5.2 we effectively consider different combinations of restart parameters and harmonic Ritz values for the flexible methods. Indeed the performance of FGMRES-DR(5,3) can be compared with FGMRES(5) if the economic variant is implemented or with FGMRES(7) if a standard implementation is considered (see Table 2.2). The total amount of floating point operations spent in matrix-vector products, dot products, preconditioning and basis orthogonalization has been computed for each solution method, excluding however the cost of the ILU(0) factorization that is identical for each proposed method. We have also indicated the results related to full FGMRES as a reference solution method; i.e., when memory is not constrained.

	Sherman4				Saylor4				Young1c			
	Mv	dot	r_{ops}	r_{mem}	Mv	dot	r_{ops}	r_{mem}	Mv	dot	r_{ops}	r_{mem}
FGMRES-DR(5,3)	373	1288	1.41	0.14	115	384	1.10	0.30	1633	5698	2.60	0.08
FGMRES(5)	1273	3813	3.56	0.14	409	1221	3.22	0.30	6145	18430	7.41	0.08
FGMRES(7)	877	2771	2.54	0.19	295	931	2.39	0.41	5095	16126	6.33	0.11
FGMRES-DR(10,5)	247	951	1.02	0.27	109	396	1.08	0.57	967	3831	1.71	0.15
FGMRES(10)	979	3331	2.97	0.27	175	590	1.46	0.57	3619	12351	4.69	0.15
FGMRES(13)	649	2358	2.06	0.35	145	517	1.25	0.73	3205	11742	4.33	0.19
full FGMRES	229	1311	1	1	109	441	1	1	421	3535	1	1

Table 2.3: Performance of FGMRES(m) and FGMRES-DR(m,k) to satisfy the convergence threshold (2.18); Mv is the total number of matrix vector products, dot the total number of dot products and r_{ops} and r_{mem} are the ratios of floating point operations and memory respectively where the reference method is full FGMRES (see Equation (2.19)).

It can be noticed that flexible methods with deflated restarting enables a faster convergence than those with standard restarting. It also results in a faster calculation since a significant amount of floating point operations is saved. Moreover we also note that the performances of FGMRES-DR(10,5) in terms of floating point operations are close to that of full flexible GMRES especially when considering the Sherman4 and Saylor4 matrices. Those results also highlight the benefit of using deflated restarting as it may lead to important memory savings. FGMRES-DR has also been found efficient on both two-dimensional wave propagation problems (Helmholtz equation with Dirichlet boundary conditions) and three-dimensional electromagnetic problems related to Maxwell's equations. We refer the reader to [53] for further details.

Thus, to improve standard Krylov methods, reusing spectral information can be of great interest to save time and memory when solving a linear system. These improvements could have a deeper impact in the multiple right-hand side case. Indeed, the spectral information would be shared by all linear systems, this will be the topic of a future work. In the next section, Krylov methods for multiple right-hand side situations are considered but without any deflated restarting. Generalization of FGMRES to the block case will be presented. Several strategies benefiting from the multiple right-hand side situation will be investigated. First, notations and principles of block methods are presented; we introduce later the related state-of-the art techniques.

2.6 Block Krylov methods

In this section, we consider block Krylov methods for the solution of linear systems with multiple right-hand sides given at once:

$$AX = B,$$

with $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$, $X \in \mathbb{C}^{n\times p}$ and where p is the number of right-hand sides. Their principle relies on the same idea as in standard Krylov methods except that the subspace to be considered is a block Krylov subspace:

$$\mathcal{K}_m(A, R_0) = \text{span}\left\{R_0, AR_0, A^2R_0, ..., A^{m-1}R_0\right\}.$$

where the initial residual vector $R_0 = B - AX_0$ with X_0 is the block initial iterate. The product $m \times p$ is then an upper bound of the dimension of the space $\mathcal{K}_m(A, R_0)$ since $m-1$ is the highest degree of the monomials and p is the number of columns of R_0. Most of the Krylov subspace methods for non-Hermitian case have a block counterpart (block GMRES (BGMRES) [121], block BiCGStab (BBiCGSTAB) [58] and block QMR [49]).

2.6.1 Principles of block Krylov methods

The idea of block Krylov method is to solve several linear systems simultaneously in order to save computational time. This idea is justified at least by two reasons. The first one is enabling matrix-vector products

involving several vectors. Indeed, applying the matrix to a block of vectors instead of each vector independently may reduce, depending on the sparsity of A, the number of accesses to the memory ([6], [72, Section 3.7.2.3]). Considering parallel computers, this may also reduce the number of messages sent by MPI and therefore the latency cost. The second reason lies in the fact that the solution of each linear system is sought in a larger Krylov subspace. In fact, a block Krylov subspace contains all Krylov subspaces generated by each initial residual $K_m(A, R_0(:, i))$ for i such that $1 \leq i \leq p$ and all possible linear combinations of the vectors contained in these subspaces. Indeed, block Krylov subspaces can be expressed as:

$$\mathcal{K}_m(A, R_0) = \left\{ \sum_{k=0}^{m-1} A^k R_0 \gamma_k \text{ with } \gamma_k \in \mathbb{C}^{p\times p},\ \forall\, k \mid 0 \leq k \leq m-1 \right\} \subset \mathbb{C}^{n\times p}$$

and as the Cartesian product ($\times$) of the sum of the p Krylov subspaces $\mathcal{B}_m(A, R_0) = +_{i=0}^{p} K_m(A, R_0(:, i))$ [60]:

$$\mathcal{K}_m(A, R_0) = \underbrace{\mathcal{B}_m(A, R_0) \times \cdots \times \mathcal{B}_m(A, R_0)}_{p \text{ times}}.$$

Thus each column $X_m(:, i)$ of the block solution X_m is searched in the space $\mathcal{B}_m(A, R_0)$ whereas the solution obtained with Krylov methods for a single right-hand side is searched in $K_m(A, R_0(:, i)) \subset \mathcal{B}_m(A, R_0)$. Therefore, block Krylov space methods have more information to obtain the solution of each linear system than the Krylov subspace for a single right-hand side.

However, the extra cost of block Krylov subspace due to orthogonalization can make these methods more expensive in terms of flops compared to a Krylov method solving one linear system after the other unless the gain in iteration count is large enough. This is clearly highlighted when considering the costs in operations of the block Arnoldi process ([67, 121]) and of the classical Arnoldi process for p vectors (see Table 2.4).

Operations	block Arnoldi cost	p times Arnoldi costs
mvp	$2nnz(A)mp$	$2nnz(A)mp$
Orthogonalization	$(4np^2 + np)(m(m+1)/2) + (m+1)(5np + 2np^2)$	$nm(2m+5)p$

Table 2.4: Cost of the block Arnoldi and the classical Arnoldi process according to the matrix dimension n, its number of non-zero elements $nnz(A)$, the Krylov subspace restart parameter m and the number of right-hand sides p.

The use of block operations in certain steps can accelerate the block methods but this speed-up essentially depends on the sparsity of A. The denser it is, the larger the speed-up will be, since memory accesses to the entries of the matrix are made more efficient by an appropriate use of the memory hierarchy that is implemented on most modern supercomputers.

Nevertheless, as $\mathcal{B}_m(A, R_0)$ may not be a direct sum, it seems natural to improve block Krylov methods by removing from the block Krylov subspaces useless information for the convergence. This technique is called later deflation. The first strategy to remove useless information from a block Krylov subspace is *initial deflation*. It consists in detecting linear dependency in the right-hand side block B or/and in the initial residual block R_0 ([59, Section 12] and [72, Section 3.7.2]). This requires to compute its/their numerical rank thanks to a rank-revealing QR-factorization [21] or a singular value decomposition [54] according to a certain deflation tolerance [64]. The linear dependency in the block residual can also be detected at each iteration of the block Krylov method. This has been first implemented for the symmetric case in block CG [86] and for non-symmetric problems in Lanczos and Arnoldi methods [1, 27]. It has then been extended to GMRES, FOM [97] and GCR [73]. A cheap variant in memory of block GCR with deflation is also proposed in [112], this method is building the block solution with only one column of its block residual (the one with maximal two-norm). It is only keeping the residual with maximal norm over the residual norms from one iteration to the next. Deflation can also be performed at each initial computation of the residual block when a restarted method is used ([79], [59, Section 14]). The advantage of such methods is that they save some rank revealing QR-factorizations or singular value decompositions and can in some cases be as efficient as methods based on deflation at each iteration.

In the next sections, we will focus on certain methods derived from block flexible GMRES. This choice is governed by the fact that algorithms using a constant preconditioner can easily be deduced from the

variants available for a variable preconditioner. This is also a natural extension of the methods investigated in the single right-hand side case. We will first propose the block flexible GMRES and then two versions of block FGMRES algorithms with deflation at the restart. We will finally show some numerical experiments.

2.6.2 Block FGMRES

In this section, we present the block flexible GMRES algorithm, a combination of block GMRES [121] and FGMRES [99]. Block GMRES (BGMRES) has been presented for the first time in Vital's thesis [121]. Since then, numerous variants have been proposed. We refer to [36, 66, 67, 75, 105, 106, 107] for different variants of block GMRES and to [57, 85] for block GMRES with deflated restarting. However, we will focus on algorithms which stay close to FGMRES (for a single right-hand side). Indeed, a block version of the modified Gram-Schmidt method (MGS) is used as a block Arnoldi process and convergence is detected from the Arnoldi's residual. The block MGS (modified Gram-Schmidt) orthogonalization scheme is described in Algorithm 9.

Algorithm 9 Flexible block Arnoldi process (MGS implementation): computation of $\mathcal{V}_{j+1}$, $\mathcal{Z}_j$ and $\bar{\mathcal{H}}_j$ for $j \leq m$

for $j = 1, \ldots, m$ **do**
  $Z_j = M_j^{-1} V_j$
  $W = AZ_j$
  for $i = 1, \ldots, j$ **do**
    $H_{i,j} = V_i^H W$
    $W = W - V_i H_{i,j}$
  end for
  Compute the QR decomposition $W = QR$, $V_{j+1} = Q$, $H_{j+1,j} = R$;
  Set $H_{i,j} = 0_{p\times p}$ for $i > j + 1$
  Define $\mathcal{Z}_j = [Z_1, \ldots, Z_j]$, $\mathcal{V}_j = [V_1, \ldots, V_j]$, $\bar{\mathcal{H}}_j = (H_{k,l})_{1\leq k\leq j+1, 1\leq l\leq j}$.
end for

As in the standard flexible Arnoldi process (Algorithm 4), the flexible block Arnoldi process produces matrices $\mathcal{Z}_j \in \mathbb{C}^{n\times jp}$, $\mathcal{V}_j \in \mathbb{C}^{n\times(j+1)p}$ and $\bar{\mathcal{H}}_j \in \mathbb{C}^{(j+1)p\times jp}$, which satisfy a flexible Arnoldi relation, for $j \leq m$,

$$A\mathcal{Z}_j = \mathcal{V}_{j+1}\bar{\mathcal{H}}_j. \tag{2.20}$$

Combining the expressions of W in Algorithm 9, we obtain for all j such that $1 \leq j \leq m$:

$$W = V_{j+1}H_{j+1,j} = AZ_j - \sum_{i=1}^{j} V_i H_{i,j}$$

which can be written:

$$AZ_j = \left[V_1, V_2, \ldots, V_{j+1}\right] \begin{bmatrix} H_{1,j} \\ H_{2,j} \\ \vdots \\ H_{j+1,j} \end{bmatrix}.$$

Finally, we generalize this expression for all j such that $1 \leq j \leq m$:

$$A\left[Z_1, \ldots, Z_j\right] = \left[V_1, V_2, \ldots, V_{j+1}\right] \begin{bmatrix} H_{1,1} & H_{1,2} & \ldots & H_{1,j} \\ H_{2,1} & H_{2,2} & \ldots & H_{2,j} \\ 0_{p\times p} & H_{3,2} & \ldots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0_{p\times p} & 0_{p\times p} & 0_{p\times p} & H_{j+1,j} \end{bmatrix}.$$

Using the definitions given in Algorithm 9 line 10 of $\mathcal{Z}_j$, $\mathcal{V}_{j+1}$ and $\bar{\mathcal{H}}_j$, we deduce the block flexible Arnoldi relation:

$$A\mathcal{Z}_j = \mathcal{V}_{j+1}\bar{\mathcal{H}}_j.$$

Remark 2. *It should be noticed that $\bar{\mathcal{H}}_j$ is no longer a Hessenberg matrix but a block Hessenberg matrix. Indeed, its block sub-diagonal consists of diagonal blocks of size $p \times p$.*

We depict now the block flexible GMRES algorithm that we derive from the algorithm involving a constant preconditioner [105, 121].

Algorithm 10 Block Flexible GMRES (BFGMRES(m))

1: Choose a convergence threshold *tol*, the size of the restart m and the maximum number of iterations *itermax*.
2: Choose initial guess X_0;
3: Compute the initial block residual $R_0 = B - AX_0$;
4: **for** $iter = 1, \ldots, itermax$ **do**
5: Compute the QR decomposition $R_0 = QT$, $V_1 = Q$, $\mathcal{B}_j = \begin{bmatrix} T \\ 0_{jp\times p} \end{bmatrix}$, $1 \le j \le m$.
6: **for** $j = 1, \ldots, m$ **do**
7: *Completion of $\mathcal{V}_{j+1}$, $\mathcal{Z}_j$ and $\bar{\mathcal{H}}_j$:* Apply Algorithm 9 from line 2 to 10 with flexible preconditioning ($Z_j = M_j^{-1}V_j$, $1 \le j \le m$) to obtain $\mathcal{V}_{j+1} \in \mathbb{C}^{n\times(j+1)p}$, $\mathcal{Z}_j \in \mathbb{C}^{n\times jp}$ and the matrix $\bar{\mathcal{H}}_j \in \mathbb{C}^{(j+1)p\times jp}$ such that:

$$A\mathcal{Z}_j = \mathcal{V}_{j+1}\bar{\mathcal{H}}_j \quad \text{with} \quad \mathcal{V}_{j+1}^H\mathcal{V}_{j+1} = I_{(j+1)p}.$$

8: Solve the minimization problem $Y_j = \text{argmin}_{Y\in\mathbb{C}^{jp\times p}} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y\|_F$;
9: **if** $\|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y_j(:,l)\|_2 / \|B(:,l)\|_2 \le tol$, $\forall\, l \mid 1 \le l \le p$ **then**
10: compute $X_j = X_0 + \mathcal{Z}_j Y_j$; stop
11: **end if**
12: **end for**
13: Compute $X_m = X_0 + \mathcal{Z}_m Y_m$ and $R_m = B - AX_m$;
14: Set $R_0 = R_m$ and $X_0 = X_m$;
15: **end for**

In the following propositions, we first derive the relation between the true residual $R_j = B - AX_j$ and the Arnoldi's residual $\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j$ that holds in the block case (Proposition 5). Then, we prove that BFGMRES minimizes the Euclidean norm of each residual (Proposition 6).

Proposition 5. *At the end of the restart or at the convergence in Algorithm 10, the computed solution X_j and the least-squares solution Y_j satisfy the following block relation for j such that $1 \le j \le m$:*

$$R_j = B - AX_j = \mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j).$$

Proof. We first recall that the initial residual can be written as (see Algorithm 10 line 5):

$$R_0 = \mathcal{V}_{j+1}\mathcal{B}_j.$$

We then deduce the proposed relation using this last equality and the block flexible Arnoldi relation (2.20):

$$\begin{aligned} B - A(X_0 + \mathcal{Z}_j Y_j) &= R_0 - \mathcal{V}_{j+1}\bar{\mathcal{H}}Y_j, \\ &= \mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j). \end{aligned}$$

□

Proposition 6. *Algorithm 10 minimizes the Euclidean norm of the residual of each linear system.*

Proof. This is a direct consequence of Proposition 5 and of some properties related to the Frobenius norm:

$$\begin{aligned}\|B - AX_j\|_F^2 &= \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j\|_F^2 = \min_{Y\in\mathbb{C}^{jp\times p}} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y\|_F^2 = \sum_{l=1}^{p} \min_{Y(:,l)\in\mathbb{C}^{jp}} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y(:,l)\|_2^2,\\ &= \sum_{l=1}^{p} \min_{Y(:,l)\in\mathbb{C}^{jp}} \|R_0(:,l) - A\mathcal{Z}_j Y(:,l)\|_2^2 = \sum_{l=1}^{p} \min_{Y(:,l)\in\mathbb{C}^{jp}} \|B(:,l) - A(X_0(:,l) + \mathcal{Z}_j Y(:,l))\|_2^2.\end{aligned}$$

This last equality proves the proposition. □

Corollary 1. *The convergence of BFGMRES is monotone in the Euclidean norm for the residual of each linear system.*

Proof. Since

$$\min_{y\in\mathbb{C}^{jp}} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j y\|_2 \leq \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j \begin{bmatrix} y_{j-1} \\ 0_{p\times p} \end{bmatrix}\|_2 = \|\mathcal{B}_{j-1}(:,l) - \bar{\mathcal{H}}_{j-1} y_{j-1}\|_2$$

where $y_{j-1} = \text{argmin}_{y\in\mathbb{C}^{(j-1)p}} \|\mathcal{B}_{j-1}(:,l) - \bar{\mathcal{H}}_{j-1} y\|_2$, we have, for all l such that $1 \leq l \leq p$,

$$\min_{y\in\mathbb{C}^{jp}} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j y\|_2 \leq \min_{y\in\mathbb{C}^{(j-1)p}} \|\mathcal{B}_{j-1}(:,l) - \bar{\mathcal{H}}_{j-1} y\|_2.$$

Since $\|B(:,l) - A(X_0(:,l) + \mathcal{Z}_j y)\|_2 = \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j y\|_2$, the corollary is proved. □

Corollary 2. *In Algorithm 10, detecting the convergence on the true residual is equivalent to detecting the convergence on the Arnoldi's residual in exact arithmetic:*

$$\frac{\|B(:,l) - AX_j(:,l)\|_2}{\|B(:,l)\|_2} \leq tol,\ \forall\, l \mid 1 \leq l \leq p \Leftrightarrow \frac{\|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y_j(:,l)\|_2}{\|B(:,l)\|_2} \leq tol,\ \forall\, l \mid 1 \leq l \leq p.$$

Proof. This is a direct consequence of Proposition 6. □

Remark 3. *The stopping criterion in Algorithm 10 (line 9) has been chosen considering Corollary 2. The Frobenius norm could be used to check convergence ($\frac{\|R_j\|_F}{\sqrt{p}} \leq \varepsilon$) instead of the Euclidean norm of each residual since*

$$\max_{1\leq l\leq p} \|R_j(:,l)\|_2^2 \leq \|R_j\|_F^2 \leq p \max_{1\leq l\leq p} \|R_j(:,l)\|_2^2.$$

Despite the fact that the Frobenius norm would be convenient for detecting the convergence at once, it can be too severe for detecting at the right time the convergence of each right-hand side. Indeed, if one right-hand side converges much earlier than the others, the Frobenius norm cannot detect it. Thus, even simple strategies, like removing converged solutions, cannot be considered using a Frobenius norm.

Remark 4. *In Algorithm 10, the true residual is computed at each restart whereas it could be computed thanks to Proposition 5: $R_m = \mathcal{V}_{m+1}(\mathcal{B}_m - \bar{\mathcal{H}}_m Y_m)$. Indeed, it is usually cheaper to compute explicitly $R_m = B - AX_m$ for a sparse matrix A ($2nnz(A)p + np$ operations) than evaluating $\mathcal{V}_{m+1}(\mathcal{B}_m - \bar{\mathcal{H}}_m Y_m)$ explicitly ($2n(m+1)p^2$ operations).*

There exists a lot of applications in the literature for which traditional block methods are very efficient but these methods are not consistently profitable; floating point operations and memory have to be considered carefully. In the next section, we derive block method found efficient on the numerical tests addressed in this thesis.

2.6.3 Block FGMRES with deflation

Then it exists an elegant but complex way to introduce *deflation* during each iteration of block GMRES due to Robbé and Sadkane [97]. It consists in detecting linear dependency in the block of residuals at each iteration. Of course, this requires additional operations at each iteration but can really improve convergence [69] at the same memory cost as BGMRES. However, since small restart parameters are considered in practice for memory issues, we propose a simpler algorithm implementing deflation at the restart. It consists in detecting linear dependency in the true block residual $B - AX$ at the beginning of each restart. The main ideas of this method are presented in ([59, Section 14], [79]), it is a generalization of initial deflation techniques [72]. Thanks to a small restart parameter m, linear dependencies in the block residual could then be detected nearly when they occur. This detection is performed with a rank revealing QR-factorization or a SVD of the block residual. Of course, since exact deflation never occurs in practice, a deflation tolerance has to be selected. This deflation tolerance introduces a numerical error which may badly influence the convergence [59], the question of its choice will be discussed later in this section.

The block Flexible GMRES with deflation (Algorithm 11) introduces deflation at the restart in BFGMRES. This method is a direct adaptation of block FGMRES (Algorithm 10) to the case of deflation. It uses deflation techniques really close to the one depicted in [59] for BGMRES and in [79] for BQMR. The deflation is performed thanks to the SVD of the upper triangular factor arising from the QR-factorization of the true block residual $B - AX$. It consists in selecting the p_d singular vectors corresponding to the p_d singular values larger than a deflation tolerance $\varepsilon_d \, tol$ where $\varepsilon_d \in (0, 1]$ and tol is the convergence threshold for the linear systems. The philosophy behind this process is to detect linear combinations of the columns of the block residual which have converged. The value of ε_d has to be carefully chosen to guarantee convergence to a tolerance tol. To make this choice easier, a "quality of convergence criterion" $\varepsilon_q \in [0, 1]$ is introduced. The parameter ε_q sets a convergence criterion for the small residual ρ_j. We will remark that if ε_d and ε_q are chosen such that $\varepsilon_d + \varepsilon_q \leq 1$, the convergence will be guaranteed on the true residual. In order to keep the same scaled stopping criterion as in BFGMRES ($\frac{\|B(:,l)-AX(:,l)\|_2}{\|B(:,l)\|_2} \leq tol, \ \forall l \leq p$) and to avoid the scaling on the deflation condition (Algorithm 11 lines 8 and 24), Algorithm 11 deals with scaled right-hand sides and scaled initial solutions (Algorithm 11 line 6). Moreover, in this section, we assume that the singular values during the SVD ($T = U_T \Sigma_T W_T^H$, $T \in \mathbb{C}^{p \times p}$) are sorted in a increasing order:

$$\Sigma_T(l+1, l+1) \leq \Sigma_T(l, l), \ \forall \ l \mid 1 \leq l < p.$$

Algorithm 11 Block Flexible GMRES with SVD based deflation (BFGMRESD(m))

1: In this algorithm, we consider that $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$.
2: Choose a convergence threshold tol, a deflation criterion ε_d, a convergence criterion ε_q, the size of the restart m and the maximum number of iterations $itermax$.
3: Choose initial guess $X_0 \in \mathbb{C}^{n\times p}$;
4: Compute: $D_B(l,l) = \|B(:,l)\|_2$ for all l such that $1 \le l \le p$.
5: Compute the initial block residual $R_0 = B - AX_0$;
6: Compute the QR decomposition $R_0\, D_B^{-1} = QT$;
7: Compute the SVD of T: $T = U_T \Sigma_T W_T^H$,
8: Select p_d singular values of T such that
$\Sigma_T(l,l) \ge \varepsilon_d tol$ for all l such that $1 \le l \le p_d$;
9: Compute V_1: $V_1 = QU_T(:, 1:p_d)$.
10: **for** $iter = 1, \ldots, itermax$ **do**
11: Let $\mathcal{B}_j = \begin{bmatrix} I_{p_d} \\ 0_{jp_d\times p_d} \end{bmatrix}$, $1 \le j \le m$.
12: **for** $j = 1, \ldots, m$ **do**
13: *Completion of $\mathcal{V}_{j+1}$, $\mathcal{Z}_j$ and $\bar{\mathcal{H}}_j$ (see Algorithm 9):* Apply Algorithm 9 from line 2 to 10 with flexible preconditioning ($Z_j = M_j^{-1} V_j$, $1 \le j \le m$) to obtain $\mathcal{V}_{j+1} \in \mathbb{C}^{n\times(j+1)p_d}$, $\mathcal{Z}_j \in \mathbb{C}^{n\times jp_d}$ and the matrix $\bar{\mathcal{H}}_j \in \mathbb{C}^{(j+1)p_d\times jp_d}$ such that:

$$A\mathcal{Z}_j = \mathcal{V}_{j+1}\bar{\mathcal{H}}_j \quad \text{with} \quad \mathcal{V}_{j+1}^H \mathcal{V}_{j+1} = I_{(j+1)p_d}.$$

14: Solve the minimization problem $Y_j = \operatorname{argmin}_{Y\in\mathbb{C}^{jp_d\times p_d}} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y\|_F$;
15: Compute $\rho_j = (\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_T(1:p_d, 1:p_d) W_T(1:p, 1:p_d)^H$
16: **if** $\|\rho_j(:,l)\|_2 \le tol\,\varepsilon_q$, $\forall\, l \mid 1 \le l \le p$ **then**
17: Compute $X_j = X_0 + \mathcal{Z}_j Y_j \Sigma_T(1:p_d, 1:p_d) W_T(1:p, 1:p_d)^H D_B$; stop;
18: **end if**
19: **end for**
20: $X_m = X_0 + \mathcal{Z}_m Y_m \Sigma_T(1:p_d, 1:p_d) W_T(1:p, 1:p_d)^H\, D_B$,
21: $R_m = B - AX_m$,
22: Compute the QR decomposition $R_m\, D_B^{-1} = QT$;
23: Compute the SVD of T: $T = U_T \Sigma_T W_T^H$,
24: Select p_d singular values of T such that
$\Sigma_T(l,l) \ge \varepsilon_d tol$ for all l such that $1 \le l \le p_d$;
25: Compute V_1: $V_1 = QU_T(:, 1:p_d)$.
26: Set $R_0 = R_m$ and $X_0 = X_m$.
27: **end for**

Proposition 7 gives a generalization of Proposition 5 to the deflation case (relation between the true residual and the Arnoldi's one). In order to simplify the notations, we set $U_+ = U_T(:, 1:p_d)$, $\Sigma_+ = \Sigma(1:p_d, 1:p_d)$, $W_+ = W_T(:, 1:p_d)$ and $U_- = U_T(:, p_d+1:p)$, $\Sigma_- = \Sigma_T(p_d+1:p, p_d+1:p)$, $W_- = W_T(:, p_d+1:p)$.

Proposition 7. *At the end of one restart or at convergence in Algorithm 11, the block true residual $R_j = B - AX_j$ and the small residual $\rho_j = (\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H$ satisfy the following property for j such that $1 \le j \le m$:*

$$\begin{cases} R_j = \mathcal{V}_{j+1}\rho_j D_B \text{ if } p_d = p, \\ R_j = \left[\mathcal{V}_{j+1}\rho_j + QU_-\Sigma_- W_-^H\right] D_B \text{ if } p_d < p. \end{cases}$$

Proof. The first equality is a direct consequence of the fact that Algorithm 11 without deflation is equivalent to block FGMRES (Algorithm 10). To obtain the second equality, we develop R_j:

$$\begin{aligned} R_j &= B - AX_j = B - A(X_0 + \mathcal{Z}_j Y_j \Sigma_+ W_+^H\, D_B), \\ &= R_0 - \mathcal{V}_{j+1}\bar{\mathcal{H}}_j Y_j \Sigma_+ W_+^H\, D_B = \left[QU_T\Sigma_T W_T^H - \mathcal{V}_{j+1}\bar{\mathcal{H}}_j Y_j \Sigma_+ W_+^H\right] D_B. \end{aligned}$$

Since $V_1 = QU_+$ and $\mathcal{V}_{j+1}\mathcal{B}_j = V_1$, we have

$$\begin{aligned} R_j &= \left[\mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H + QU_-\Sigma_- W_-^H\right] D_B, \\ &= \left[\mathcal{V}_{j+1}\rho_j + QU_-\Sigma_- W_-^H\right] D_B. \end{aligned}$$

□

Proposition 8. *In Algorithm 11 for any $\varepsilon_d \in (0, 1]$, the Frobenius norm of the block residual is decreasing from one iteration to the next. This holds even if ε_d is allowed to vary at each restart.*

Proof. Proposition 7 gives:

$$\begin{aligned} \|R_j D_B^{-1}\|_F^2 &= \|\mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H + QU_-\Sigma_- W_-^H\|_F^2, \\ &= tr((\mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H + QU_-\Sigma_- W_-^H)(\mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H + QU_-\Sigma_- W_-^H)^H), \\ &= \|(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+\|_F^2 + \|\Sigma_-\|_F^2 + tr(\mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+^H W_-\Sigma_- U_-^H Q^H) + \\ &\quad tr(QU_-\Sigma_- W_-^H W_+\Sigma_+(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)^H \mathcal{V}_{j+1}^H) \\ &= \|(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+\|_F^2 + \|\Sigma_-\|_F^2, \end{aligned}$$

since $W_T = [W_+, W_-]$ is unitary. Furthermore, the definition of the Frobenius norm yields:

$$\begin{aligned} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j\|_F^2 &= \min_{Y \in \mathbb{C}^{jp_d \times p_d}} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y\|_F^2 = \min_{Y \in \mathbb{C}^{jp_d \times p_d}} \sum_{l=1}^{p_d} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y(:,l)\|_2^2, \\ &= \sum_{l=1}^{p_d} \min_{Y(:,l) \in \mathbb{C}^{jp_d}} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y(:,l)\|_2^2 = \sum_{l=1}^{p_d} \min_{y \in \mathbb{C}^{jp_d}} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j y\|_2^2. \end{aligned}$$

Thus, Y_j also minimizes $\|(\mathcal{B}_j - \bar{\mathcal{H}}_j Y)\Sigma_+\|_F$ because

$$\|(\mathcal{B}_j - \bar{\mathcal{H}}_j Y)\Sigma_+\|_F^2 = \sum_{l=1}^{p_d} \|(\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y(:,l))\Sigma_+(l,l)\|_2^2 = \sum_{l=1}^{p_d} \|\mathcal{B}_j(:,l) - \bar{\mathcal{H}}_j Y(:,l)\|_2^2 \Sigma_+(l,l)^2.$$

Therefore, according to the proof of Corollary 1, we have

$$\begin{aligned} \|R_j D_B^{-1}\|_F^2 &= \|(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+\|_F^2 + \|\Sigma_-\|_F^2, \\ &\leq \|(\mathcal{B}_{j-1} - \bar{\mathcal{H}}_{j-1} Y_{j-1})\Sigma_+\|_F^2 + \|\Sigma_-\|_F^2. \end{aligned}$$

Coming back to the first equality of the proof, and writing it for $j-1$ instead of j, it follows that

$$\|R_{j-1} D_B^{-1}\|_F^2 = \|(\mathcal{B}_{j-1} - \bar{\mathcal{H}}_{j-1} Y_{j-1})\Sigma_+\|_F^2 + \|\Sigma_-\|_F^2.$$

The monotonicity of BFGMRESD in the Frobenius norm is then proved, since we have

$$\|R_j D_B^{-1}\|_F \leq \|R_{j-1} D_B^{-1}\|_F.$$

□

This proposition shows that the Frobenius norm of the block residual decreases along the iterations when deflation is used. Nevertheless, it does not ensure a monotone behavior for the Euclidean norm of each residual; the following remark explores this issue.

Remark 5. *In order to guarantee monotonicity on the Euclidean norm of each residual, we should include the quantity $QU_-\Sigma_- W_-^H$ in the least-squares problem solution (Algorithm 11 line 14). However, it implies to solve a least-squares problem in a space with a larger dimension than $\mathbb{C}^{jp_d \times p_d}$ for each residual. Indeed, since $V_1 = QU_+$, we have, for all l such that $1 \leq l \leq p$,*

$$\begin{aligned} R_j(:,l) &= \mathcal{V}_{j+1}(\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_+ W_+(l,:)^H + QU_-\Sigma_- W_-(l,:)^H, \\ &= [Q, V_2, ..., V_{j+1}]\left(\begin{bmatrix} U\Sigma W(l,:)^H \\ 0_{jp_d, p_d} \end{bmatrix} - \begin{bmatrix} U_+ & 0_{p_d, jp_d} \\ 0_{jp_d, p_d} & I_{jp_d, jp_d} \end{bmatrix} \bar{\mathcal{H}}_j Y_j \Sigma_+ W_+(l,:)^H\right). \end{aligned}$$

Thus, to minimize $\| \begin{bmatrix} U\Sigma W(l,:)^H \\ 0_{jp_d,p_d} \end{bmatrix} - \begin{bmatrix} U_+ & 0_{p_d,jp_d} \\ 0_{jp_d,p_d} & I_{jp_d,jp_d} \end{bmatrix} \bar{\mathcal{H}}_j Y \Sigma_+ W_+(l,:)^H \|_2$ *over* $Y \in \mathbb{C}^{jp_d \times p_d}$*, is not equivalent to minimize the true residual* $R_j(:,l)$ *Euclidean norm since* $[Q, V_2, ..., V_{j+1}]$ *is not orthogonal.* $[Q, V_2, ..., V_{j+1}]$ *has then to be taken into account in the minimization problem.*

The next corollaries (Corollaries 3 and 4) give upper bounds of the individual residual norms which guarantee convergence on the true residual when deflation has occurred and when the norm of $\rho_j(:,l)$ is less than *tol* for all $l \leq p$.

Corollary 3. *In Algorithm 11, at the end of the restart or at convergence, when deflation has occurred* ($p_d < p$)*,* R_j *and* ρ_j *satisfy the following property:*

$$\|R_j(:,l) - \mathcal{V}_{j+1}\rho_j(:,l)\, D_B(l,l)\|_2 \leq \Sigma_T(p_d+1, p_d+1)\, D_B(l,l),\ \forall\, l \mid 1 \leq l \leq p.$$

Proof. This is a direct consequence of Proposition 7 and the SVD properties, indeed, we have

$$\begin{aligned} \| R_j(:,l) - \mathcal{V}_{j+1}\rho_j(:,l)\, D_B(l,l)\|_2 &= \| QU_-\Sigma_- W_T(l, p_d+1:p)^H\, D_B(l,l)\|_2, \\ &= \|\Sigma_- W_T(l, p_d+1:p)^H\|_2\, D_B(l,l), \\ &\leq \Sigma_T(p_d+1, p_d+1)\, D_B(l,l). \end{aligned}$$

□

This corollary makes the relation between the norm of the true residual and the norm of the small residual ρ_j explicit. It shows that the norm of their difference is always lower than the largest deflated singular value $\Sigma_T(p_d+1, p_d+1)$ multiplied by the corresponding right-hand side norm $D_B(l,l)$. It means that if the deflation tolerance is well chosen, when deflation occurs the residual ρ_j will be close to be the orthogonal projection of R_j onto $\mathcal{V}_{j+1}$. The next corollary is a reformulation of Corollary 3. It will be used in Remark 6 to discuss possible choices for the deflation criterion.

Corollary 4. *When Algorithm 11 restarts and deflation has occurred* ($p_d < p$)*, the block true residual* R_j *verifies*

$$\frac{\|R_j(:,l)\|_2}{D_B(l,l)} \leq \|\rho_j(:,l)\|_2 + \Sigma_T(p_d+1, p_d+1),\ \forall\, l \mid 1 \leq l \leq p.$$

Furthermore, if Algorithm 11 has converged and deflation has occurred, R_j *verifies*

$$\frac{\|R_j(:,l)\|_2}{D_B(l,l)} \leq tol\, \varepsilon_q + \Sigma_T(p_d+1, p_d+1),\ \forall\, l \mid 1 \leq l \leq p.$$

Proof. Corollary 3 gives for all l such that $1 \leq l \leq p$:

$$\Sigma_T(p_d+1, p_d+1)\, D_B(l,l) \geq \|R_j(:,l)\|_2 - \|\rho_j(:,l)\|_2\, D_B(l,l).$$

It follows that

$$\frac{\|R_j(:,l)\|_2}{D_B(l,l)} \leq \|\rho_j(:,l)\|_2 + \Sigma_T(p_d+1, p_d+1).$$

This shows the first inequality of the corollary. The second inequality is straightforward: if $\|\rho_j(:,l)\|_2\, D_B(l,l)^{-1} \leq \varepsilon_q tol$, we have:

$$\frac{\|R_j(:,l)\|_2}{D_B(l,l)} \leq \varepsilon_q tol + \Sigma_T(p_d+1, p_d+1).$$

□

Remark 6. *A way to insure that convergence is well detected is then to choose a fixed quality convergence criterion* $\varepsilon_q \in (0,1)$ *such that* $\varepsilon_q + \varepsilon_d = 1$ *which means* $\varepsilon_q = 1 - \varepsilon_d$*. Indeed, if such a criterion is chosen, we have:*

$$\begin{aligned} tol\, \varepsilon_q + \Sigma_T(p_d+1, p_d+1) &\leq (\varepsilon_d + \varepsilon_q)\, tol, \\ &\leq tol. \end{aligned}$$

A variable quality convergence criterion ε_q can also be chosen at each restart. It aims at obtaining a higher convergence tolerance on the Arnoldi residual ρ_j and then to gain some iterations. Considering Corollary 4, if at each restart ε_q is taken such that:

$$\varepsilon_q = 1 - \frac{\Sigma_T(p_d + 1, p_d + 1)}{tol},$$

the convergence will be guaranteed and the convergence condition on ρ_j will be weaker.

The choice of the deflation tolerance (Remark 6) is yet an open question. It can be chosen such that, even if the true residual norm is guaranteed to be less than tol when $\|\rho_j(:, l)\|_2 \leq tol$, $\forall l \mid 1 \leq l \leq p$, it improves convergence and achieve it, even if $\varepsilon_d = \varepsilon_q = 1$. It can also be chosen differently at each restart considering in some way a convergence criterion for the current restart. However, BFGMRESD requires as much memory as BFGMRES (Table 2.5). From Proposition 8 we know that BFGMRESD will have a monotone convergence in the Frobenius norm for any deflation tolerance criterion. This memory requirement issue and this last observation lead us to propose a truncation strategy; instead of having a deflation tolerance, we keep the size of the block constant from one restart to the other. In Table 2.5, we remark that the memory requirement is significantly lower than for other block methods when the fixed block size is chosen such that $p_f < p$. Of course, this method will often need more iterations to converge than BFGMRES with SVD based deflation, but this has to be balanced with its memory requirements. The method is depicted in Algorithm 12.

Method	BFGMRES(m)	BFGMRESD(m)	BFGMREST(m,p_f)
Storage	$n(2m+1)p + 3np$	$n(2m+1)p + 3np$	$n(2m+1)p_f + 3np$

Table 2.5: Storage required for BFGMRES(m), BFGMRESD(m) and BFGMREST(m,p_f) considering a block size p and a problem dimension n.

As previously said, the convergence of such an algorithm will be monotone in the Frobenius norm and there is no rule about how the individual residual norms will vary along the iterations. We only know that it is led by the larger deflated singular value (Corollary 4). Indeed, Remark 6 states that convergence would occur if $\rho_j(:, l)/D_B(l, l)$ convergence threshold is chosen equal to $tol - \Sigma_T(p_b + 1, p_b + 1)$. Unfortunately, as singular vectors are chosen by truncation at the beginning of the restart in Algorithm 12, there is no guarantee that $\Sigma_T(p_b + 1, p_b + 1)$ is close to tol. The quantity $tol - \Sigma_T(p_b + 1, p_b + 1)$ has then many chances to be negative. Thereby, we consider as a convergence threshold on $\rho_j(:, l)$, the minimum between tol and $|tol - \Sigma_T(p_b + 1, p_b + 1)|$ (Algorithm 12 line 17) and to be sure the convergence is achieved, the convergence criterion is checked on the true residual (Algorithm 12 line 20).

Algorithm 12 Block Flexible GMRES with SVD based truncation (BFGMREST(m,p_f))

1: In this algorithm, we consider that $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$.
2: Choose a convergence threshold tol, a deflation criterion ε_d, a fixed block size $p_f < p$, a restart size m and the maximum number of iterations $itermax$.
3: Choose initial guess $X_0 \in \mathbb{C}^{n\times p}$;
4: Compute: $D_B(l,l) = \|B(:,l)\|_2$ for all l such that $1 \le l \le p$.
5: Compute the initial block residual $R_0 = B - AX_0$;
6: Compute the QR decomposition $R_0\, D_B^{-1} = QT$;
7: Compute the SVD of T: $T = U_T \Sigma_T W_T^H$,
8: Calculate the number, p_d, of singular values of T such that $\Sigma_T(l,l) \ge \varepsilon_d tol$ for all k such that $1 \le l \le p_d$;
9: Compute $p_b = \min(p_d, p_f)$;
10: Compute V_1: $V_1 = QU_T(:, 1 : p_b)$.
11: **for** $iter = 1, \ldots, itermax$ **do**
12: Let $\mathcal{B}_j = \begin{bmatrix} I_{p_b} \\ 0_{jp_b\times p_b} \end{bmatrix}$, $1 \le j \le m$.
13: **for** $j = 1, \ldots, m$ **do**
14: *Completion of $\mathcal{V}_{j+1}$, $\mathcal{Z}_j$ and $\bar{\mathcal{H}}_j$ (see Algorithm 9):* Apply Algorithm 9 from line 2 to 10 with flexible preconditioning ($Z_j = M_j^{-1} V_j$, $1 \le j \le m$) to obtain $\mathcal{V}_{j+1} \in \mathbb{C}^{n\times(j+1)p_b}$, $\mathcal{Z}_j \in \mathbb{C}^{n\times jp_b}$ and the matrix $\bar{\mathcal{H}}_j \in \mathbb{C}^{(j+1)p_b\times jp_b}$ such that:

$$A\mathcal{Z}_j = \mathcal{V}_{j+1}\bar{\mathcal{H}}_j \quad \text{with} \quad \mathcal{V}_{j+1}^H \mathcal{V}_{j+1} = I_{(j+1)p_b}.$$

15: Solve the least-squares problem $Y_j = \operatorname{argmin}_{Y\in\mathbb{C}^{jp_b\times p_b}} \|\mathcal{B}_j - \bar{\mathcal{H}}_j Y\|_F$;
16: Compute $\rho_j = (\mathcal{B}_j - \bar{\mathcal{H}}_j Y_j)\Sigma_T(1 : p_b, 1 : p_b) W_T(1 : p, 1 : p_b)^H$
17: **if** $\|\rho_j(:,l)\|_2 \le \min(tol, |tol - \Sigma_T(p_b+1, p_b+1)|)\ \forall l \le p$ **then**
18: Compute $X_j = X_0 + \mathcal{Z}_j Y_j \Sigma_T(1 : p_b, 1 : p_b) W_T(1 : p, 1 : p_b)^H D_B$;
19: Compute $R_j = B - AX_j$;
20: **if** $\|R_j(:,l)\|_2 / D_B(l,l) \le tol$, $\forall l \le p$ **then**
21: stop;
22: **else**
23: Return to 29;
24: **end if**
25: **end if**
26: **end for**
27: $X_m = X_0 + \mathcal{Z}_m Y_m \Sigma_T(1 : p_b, 1 : p_b) W_T(1 : p, 1 : p_b)^H D_B$,
28: $R_m = B - AX_m$,
29: Compute the QR decomposition $R_m\, D_B^{-1} = QT$;
30: Compute the SVD of T: $T = U_T \Sigma_T W_T^H$,
31: Select p_d singular values of T such that $\Sigma_T(l,l) \ge \varepsilon_d tol$ for all l such that $1 \le l \le p_d$;
32: Compute $p_b = \min(p_d, p_f)$;
33: Compute V_1: $V_1 = QU_T(:, 1 : p_b)$;
34: Set $R_0 = R_m$ and $X_0 = X_m$;
35: **end for**

However, for all the test cases we have considered, individual convergence on the Euclidean norm was always monotone for block methods with deflation or truncation. The next section is dedicated to numerical experiments in Matlab.

2.6.4 Numerical experiments

The aim of our experiments is to compare different flexible block methods with respect to both memory requirements and numerical efficiency, using on Matlab on academic test cases. The first method is the most natural way to solve a linear system with many right-hand sides using an iterative method. It consists

in using FGMRES (Algorithm 3) for each right-hand side, and solving the linear systems, one after the other. We call it *FGMRES sequence*. This strategy is the cheapest in memory to solve these problems but it does not benefit from the multiple right-hand side situation. The second method is a traditional block method (BFGMRES, Algorithm 10). The memory requirement of this method is quite high but we expect an improved convergence due to a larger search space. The third method is a block method using deflation (BFGMRESD, Algorithm 11) for a deflation tolerance equal to the convergence threshold ($\varepsilon_d = 1$). The stopping criterion on the small residual ρ_j is described at the end of Remark 6:

$$\varepsilon_q = 1 - \frac{\Sigma_T(p_d + 1, p_d + 1)}{tol}.$$

This method is still expensive in memory but should behave, at least, as well as BFGMRES. The fourth and the fifth methods are block methods using both truncation and deflation (BFGMREST, Algorithm 12) for two different truncated block sizes. The first size is equal to the number of right-hand sides divided by 2 rounded up (in Matlab ceil(p/2)), the second one is the number of right-hand sides divided by 3 rounded up (ceil(p/3)). This involves a cheaper cost in memory but the convergence may behave worse than BFGMRES. The choice of flexible methods is governed by the fact that one restart of GMRES(5), that cannot be represented by a matrix for any right-hand side, will be our preconditioning strategy. For block methods, the preconditioner will be applied to each block vectors one after the other. This allows to compare all the methods with the same preconditioner. The restart size m is taken equal to 5 for all the preconditioned methods. Several block sizes (number of right-hand sides processed at once), denoted by p, will be considered in order to determine the best block size for each test case. The values of p are taken equal to 5, 10, 20, 40, 80 and 160 respectively. Despite the fact that the last two values would not be very relevant in practice for both memory requirement reasons and orthogonalization costs, they have been chosen to show the effect of using a large block size. For all the experiments, the algorithms are stopped when the Euclidean norm of each residual normalized by the corresponding right-hand side norm is below 10^{-6}:

$$\frac{\|B(:,l) - AX(:,l)\|_2}{\|B(:,l)|_2} \leq 10^{-6}, \ \forall\, l \mid 1 \leq l \leq p.$$

The block methods are compared according to the number of iterations (equivalent to the number of applications of the preconditioner) and to the number of floating point operations (flops) to achieve convergence. This last comparison criterion ensures that the method with the smallest flops number required will be the fastest. However, these two measures do not take into account the possible computational speed-up of block methods especially the block matrix vector acceleration. Timing would show such a behavior but since Matlab timing is not reliable, we have decided not to provide this information. In Chapter 4 experiments with block flexible methods will be performed on a geophysical application in Fortran and timings will be reported, this will emphasize the real capabilities of block methods.

Poisson problem

The first test case is a two-dimensional Poisson problem in the unit square $[0,1]^2 = \Omega \cup \partial\Omega$, where $\Omega = (0,1)^2$, with Dirichlet boundary conditions:

$$\begin{cases} -\Delta u = g & \text{on} \quad \Omega, \\ \\ u = 0 & \text{on} \quad \partial\Omega. \end{cases} \tag{2.21}$$

It is discretized with a second-order finite difference scheme for a vertex-centered grid arrangement. The mesh grid size is taken equal to $1/128$. Thus, the size of matrix is $n = 127^2 = 16129$, it is sparse, symmetric with a five-banded structure.

Canonical right-hand side First we take as right-hand side the canonical basis vectors:

$$B(:, 1 : p) = [e_1, e_2, \ldots, e_p].$$

The right-hand side matrix is unitary and no initial deflation can then be performed. The initial iterate X_0 is set to zero.

Numerical results Results are reported in Table 2.6.

number of RHS	$p = 5$		$p = 10$		$p = 20$		$p = 40$		$p = 80$		$p = 160$	
	It	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}
FGMRES(5) sequence	99	1	216	1	459	1	978	1	2087	*1*	4074	*1*
BFGMRES(5)	90	1.43	190	1.97	340	2.60	600	3.86	1120	6.06	2240	11.90
BFGMRESD(5)	**40**	*0.56*	**73**	0.65	**134**	0.88	**248**	1.36	**465**	2.29	**875**	4.43
BFGMREST(5,ceil(p/2))	45	0.59	75	*0.52*	140	0.62	260	0.84	480	1.30	900	2.38
BFGMREST(5,ceil(p/3))	53	0.65	90	0.61	145	*0.57*	265	*0.73*	490	*1*	909	1.73

Table 2.6: Number of iterations (It) and operation ratio (r_{ops}) for the 127^2-Poisson problem for p canonical basis right-hand sides.

In Table 2.6, the parameter p is the number of columns of the right-hand side block. The quantity It is the number of preconditioner applications required to converge. The quantity r_{ops} is the ratio between the number of operations, including the preconditioning operations, needed by FGMRES(5) in sequence over the number of operations needed by each other methods. For instance, r_{ops} in the second row is:

$$r_{ops} = \frac{ops(BFGMRES(5))}{ops(FGMRES(5)\ sequence)},$$

where $ops(method)$ denotes the number of operations needed by the relevant method to converge.

FGMRES sequence We first observe that the number of iterations needed by FGMRES(5) sequence is increasing almost linearly with the block size p; it is multiplied from one column to the next by a factor of two. It means that FGMRES converges in more or less the same number of iterations for each right-hand side given in the sequence.

BFGMRES Looking at the second row of Table 2.6, it can be remarked that BFGMRES(5) requires a reduced number of iterations with respect to FGMRES(5) sequence : the larger is p, the smaller is the number of iterations needed to converge. For $p = 160$, BFGMRES(5) needs only a slightly more than the half of FGMRES(5) sequence iterations to converge, respectively 2240 and 4074. However, although BFGMRES(5) needs less iterations, its computational cost is especially high (up to twelve times more (when $p = 160$)). In fact, the extra orthogonalizations make BFGMRES slower than solving the linear systems in sequence. Furthermore, we note that the operations performed by the preconditioner are taken into account in the computational cost calculation. Since the cost of the preconditioner application is low, to improve significantly the number of iterations does not decrease significantly the total computational cost. However, if the block matrix vector product computations significantly speed up the block method, it could be interesting to use such a method on this problem.

BFGMRESD Notwithstanding, we also remark that BFGMRES(5) is greatly improved by deflation. Indeed, BFGMRESD(5) diminishes significantly the number of iterations. The ratio between the number of iterations of BFGMRESD(5) and FGMRES(5) sequence greatly increases with the block size p. This ratio starts at more than 2 for $p = 5$ and it ends at more than 4 for $p = 160$. However, despite such a behavior, the operation ratio r_{ops} does not vary similarly. The best situation for BFGMRESD(5) is met when $p = 5$ where ($r_{ops} = 0.56$) and r_{ops} is larger than one for $p = 40$, larger than two for $p = 80$ and larger than four for $p = 160$. This behavior is once again due to the extra-orthogonalization and the cheap cost of the preconditioning operations.

BFGMREST When truncation is used (rows 4 and 5), the numbers of iterations of BFGMREST are quite close to the ones of BFGMRESD(5). Of course, the number of iterations of BFGMREST is always larger than one of BFGMRESD, since it lacks information compared to BFGMRESD. Moreover, the same behavior is observed between the fourth and fifth rows, the larger the fixed block size parameter p_f is, the smaller the number of iterations is. Nevertheless, on this test case, truncated methods require less operations to converge than a traditional block deflated method and the smaller p_f, the smaller the r_{ops}. This is once again a direct consequence of the extra block orthogonalization cost: to decrease the size of the block has a direct impact on the operation cost. Indeed, the cost in operations of the block Arnoldi process involves the square of the block size (see Table 2.5). Since the number of iterations is kept close to BFGMRESD, BFGMREST has a cheaper cost in operation.

Comments on histories of convergence Histories of convergence for the first two values of p ($p = 5$, $p = 10$) are plotted in Figures 2.5 and 2.6 respectively. We do not show the histories of convergence for the other values of p because the plots for larger p would be too overloaded. Indeed, the histories are plotted for each right-hand side and for each method on the same figure.

How to read the plots Each method is associated to a color and a symbol; (magenta, $\circ$) for FGMRES(5) sequence, (black, $\square$) for BFGMRES(5), (blue, $\triangledown$) for BFGMRESD(5), (red, $+$) for BFGMREST(5, $ceil(p/2)$) and (green, $\cdot$) for BFGMREST(5, $ceil(p/3)$).

Concerning FGMRES(5) sequence, histories of convergence are drawn for each right-hand side. Once the history of convergence for a right-hand side has been plotted, the history of convergence for the next right-hand side is plotted from the abscissa where the previous history ends. For BFGMRES(5), histories of convergence are also plotted for each right-hand side but the normalized norm of each residual is plotted against block iterations. Thereby p squares ($\square$) appear in group at each iteration in the history of convergence of BFGMRES(5) . Histories of convergence of BFGMRESD(5) are plotted in almost the same way. The only difference happens when deflation occurs. Indeed, since the block size decreases ($p_d < p$) due to deflation, we note that at the end of BFGMRESD(5) convergence, p_d triangles ($\triangledown$) appear instead of p. In the truncation case, the small residual ρ_j do not give information on the true block residual R_j. The true block residual is only computed at the end at the restart. Thus, the normalized norms of each true residual are plotted against one block restart iteration. Therefore, residuals norms are plotted in groups of size mp_b where p_b is the minimum between the size of the truncated block p_f and the number of significant singular vectors for the convergence p_d.

The main purpose of these plotting conventions is to illustrate the ranking in Table 2.6. Of particular interest are the histories of convergence of deflated and truncated block methods. Indeed, we notice on both Figures 2.5 and 2.6 how the block size is varying for BFGMRESD(5). At the beginning, there is no deflation, BFGMRESD(5) behaves like BFGMRES(5) but after the first restart, singular vectors are removed and the block size p_d of BFGMRESD(5) is decreasing along the solution phase. At the end, in Figures 2.5 and 2.6, the block size p_d of BFGMRESD(5) is found to be one. The histories of convergence of truncated methods points out an interesting behavior. Some residual norms are not decreasing during the first restart whereas the others reach 10^{-4} for both values of p (Figures 2.5 and 2.6). Then, the lower residuals from the first restart decrease a little during the second restart ($p_f = ceil(p/2)$ in Figure 2.5) or not ($p_f = ceil(p/2)$ in Figure 2.5 and for both values of p_f in Figure 2.6) whereas, the higher residuals from the first restart decrease a lot. Finally, the residual decreases uniformly during the third restart. This particular behavior is due to the structure of the right-hand side. Indeed, the initial block residual is

$$R_0 = \begin{bmatrix} I_{p\times p} \\ 0_{n-p\times p} \end{bmatrix},$$

and so coincide with the first V_1. Then, the first restart only deals with the first p_f right-hand sides and does not manage the last $p - p_f$ ones. The second restart handles these last $p - p_f$ right-hand sides and does not affect the first p_f right-hand sides except for $p = 5$ and $p_f = ceil(p/2) = 3$. In fact, for $p_f = 3$, one column of the second restart V_1 contains a singular vector related to the three residuals of the first restart. The corresponding residual norms are then decreasing along the second restart. For all the values of p_f, the third restart is dealing with V_1 which columns are singular vectors related to all the right-hand side residuals. Therefore, all the residuals norms decrease along this restart and it continues similarly along the next restarts.

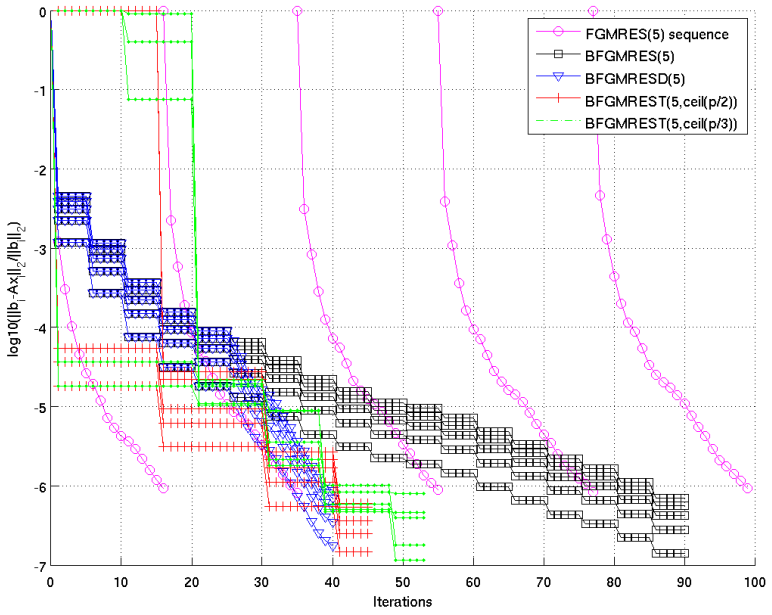

Figure 2.5: Histories of convergence of block methods when solving the Poisson problem with $p = 5$ canonical right-hand sides (Table 2.6)

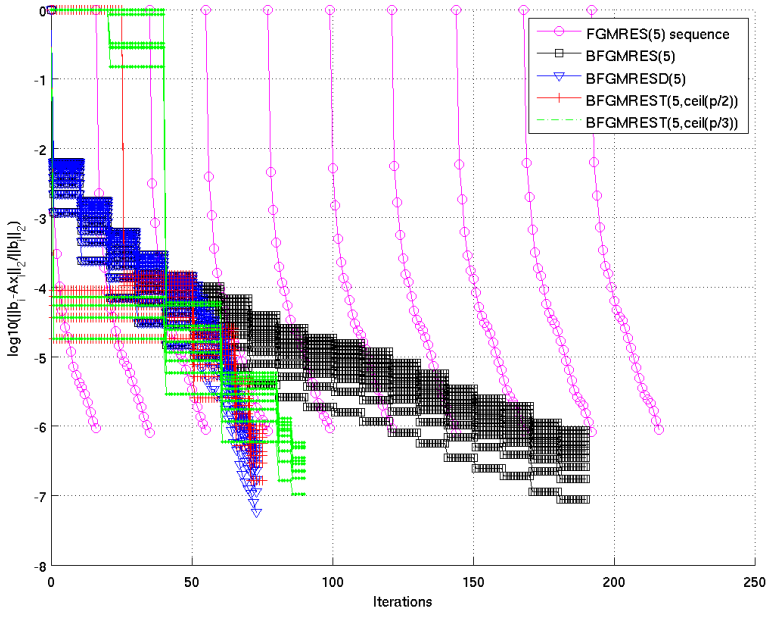

Figure 2.6: Histories of convergence of block methods when solving the Poisson problem with $p = 10$ canonical right-hand sides (Table 2.6)

Random right-hand sides Now, random vectors are chosen as right-hand sides for the Poisson problem. They are generated in Matlab using the *seed* random number generator (rand('seed',0)) and the command $B = rand(n, p_{max})$ where $p_{max} = 160$. The right-hand side block is then no more unitary nor orthogonal and has full rank. Once again, the initial iterate X_0 is set to zero.

Numerical results We report the results in Table 2.7.

number of RHS	$p = 5$		$p = 10$		$p = 20$		$p = 40$		$p = 80$		$p = 160$	
	It	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}
FGMRES(5) sequence	216	1	431	1	863	1	1730	1	3465	1	6928	*1*
BFGMRES(5)	225	1.67	310	1.63	400	1.68	400	1.46	560	1.71	960	2.90
BFGMRESD(5)	**113**	0.74	**160**	0.71	**235**	0.89	**315**	0.96	**428**	1.36	**801**	2.59
BFGMREST(5,ceil(p/2))	122	0.75	190	0.70	275	0.72	425	0.87	600	1.10	860	1.39
BFGMREST(5,ceil(p/3))	118	*0.67*	195	*0.67*	294	*0.64*	460	*0.74*	594	*0.77*	978	1.11

Table 2.7: Number of iterations (It) and operation ratio (r_{ops}) for the 127^2-Poisson problem for p random right-hand sides.

First we remark that this problem is more difficult than the previous one.

FGMRES sequence For each block size, FGMRES(5) sequence needs almost twice the number of iterations used in the test case with canonical basis vectors. However, the number of iterations still increases nearly linearly with the block size p.

BFGMRES BFGMRES(5) behaves in a weird way on this problem. First, BFGMRES(5) does not improve the convergence, it performs more iterations than FGMRES(5) sequence for $p = 5$. This result is the consequence of the block convergence detection: BFGMRES(5) stops when each solution has converged, even if some solutions have converged earlier than other. Afterward, the numbers of iterations is similar for $p = 20$ and $p = 40$ ($It = 400$). This phenomenon could be explained by the fact that the union of the Krylov subspaces generated at each restart for $p = 40$ contains the union of those generated for $p = 20$ after only 20 block iterations.

BFGMRESD For larger values of p, the numbers of iteration are in the range of those of deflated methods. BFGMRESD behaves similarly as for the canonical basis right-hand sides. Deflation always improves the number of iterations and reduces the number of operations, at least for the low values of p (10, 20, 40). Besides, the number of iterations of FGMRES(5) sequence for the last value of p ($p = 160$) is more than eight times the one of BFGMRESD.

BFGMREST The number of iterations of truncated methods is not as good as in the previous example. They converge in more iterations than BFGMRES for $p = 40$, 80 and BFGMREST(5, $ceil(p/3)$) converges in more iterations than BFGMRES for $p = 160$. Since BFGMRES works exceptionally well for this example, these results seem reasonable. Another unusual behavior can be observed for $p = 80$, BFGMREST(5, $ceil(p/3)$) converges in few less iterations than BFGMREST(5, $ceil(p/2)$). The only possible explanation of this behavior would be that the main information for the convergence is contained in the first $ceil(p/3)$ columns of the block right-hand sides. However, the numbers of operations of truncated methods are still lower than the deflated ones. Besides, BFGMREST(5, $ceil(p/3)$) improves nearly always the number of operations, it only fails for the larger block size $p = 160$.

Comment on the histories of convergence As in the previous example, histories of convergence are plotted for only two values of p ($p = 5$, $p = 10$), in Figures 2.7 and 2.8 respectively. Despite the scaling of the plot, we observe the same phenomena as for the previous set of right-hand sides. BFGMRESD starts like BFGMRES and then achieves convergence in half of the iterations of BFGMRES. It can also be noticed that the block size at the end of the BFGMRESD convergence is again found to be one for both values of p. BFGMREST is behaving in a slightly different manner than in the previous example. Contrary to the canonical right-hand sides case, all the residual norms have decreased after the first restart. This must be the consequence of the non-orthogonality of the block right-hand side. Indeed, the p_b first singular vectors seem to provide information about all the initial residuals. However, at the second restart, like for the orthonormal right-hand sides case, some residuals are decreasing more slowly than others, except for $p = 5$ and $p_b = ceil(p/2)$. This behavior could be the effect of the size of p_b, BFGMRES seems to lack information at the second restart to make converge all the residuals in an uniform way. However, after the two first restarts, the convergence rate is nearly the same for each residuals whatever the p_b parameter is.

Both examples illustrate the efficiency of block methods with deflation or truncation on this Poisson example. It has to be stressed that no computational speed-up is taken into account, like block matrix vector products acceleration, in this comparison. Nevertheless, both deflation and truncation strategies have shown efficient compared with a block flexible method and the sequence strategy.

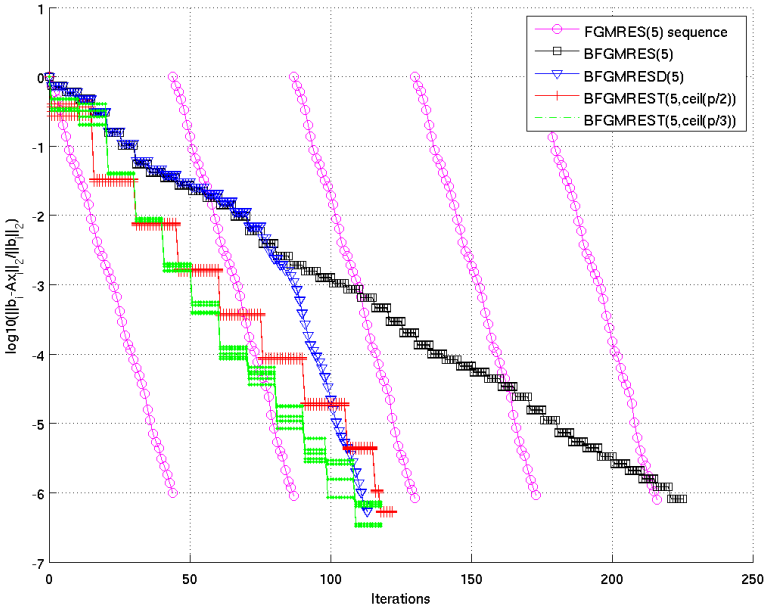

Figure 2.7: Histories of convergence of block methods when solving the Poisson problem with $p = 5$ random right-hand sides (Table 2.7).

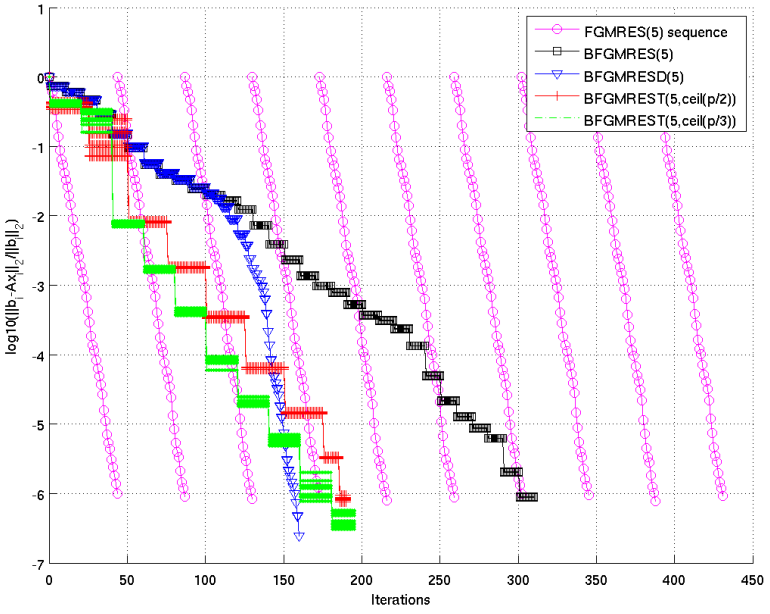

Figure 2.8: Histories of convergence of block methods when solving the Poisson problem with $p = 10$ random right-hand sides (Table 2.7).

Convection-diffusion problem

In this section, we focus on the convection diffusion problem (see Equation (2.4)). For these numerical experiments, the mesh grid size is taken equal to $1/128$. The parameters c and d are taken equal to 256 and

ε equal to 1; the Péclet condition is then satisfied. The problem size is then equal to $129^2 = 16641$ and the matrix is non-symmetric and five banded. Since Dirichlet boundary conditions are included in the linear system, the right-hand side B has to be generated such that $A^{-1}B$ satisfy them. Thus, in order to build the right-hand side, we first generate the solution X. The solution X is a random matrix which values on the boundaries of the domain are set to one to satisfy the boundary conditions. We still use the *seed* random number generator (rand('seed',0)) in Matlab. The solution is then multiplied by A to obtain the right-hand side: $B = AX$. The initial iterate X_0 is first set to zero in the interior whereas its values on the boundaries of the domain are set to one.

Numerical results Numerical results are displayed in Table 2.8.

number of RHS	$p = 5$		$p = 10$		$p = 20$		$p = 40$		$p = 80$		$p = 160$	
	It	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}	**It**	r_{ops}
FGMRES(5) sequence	172	1	346	1	687	*1*	1372	*1*	2760	*1*	5533	*1*
BFGMRES(5)	145	1.38	310	2.12	580	3.20	840	4.06	1280	5.69	1760	7.42
BFGMRESD(5)	**110**	1.02	**203**	1.29	**345**	1.79	**535**	2.37	**860**	3.69	**1510**	6.20
BFGMREST(5,ceil(p/2))	123	0.98	225	1.09	395	1.38	660	1.82	1055	2.53	1655	3.71
BFGMREST(5,ceil(p/3))	126	*0.90*	218	*0.95*	392	1.12	658	1.43	1080	1.89	1835	2.88

Table 2.8: Number of iterations (It) and operation ratio (r_{ops}) for the 129^2-convection-diffusion problem for p random right-hand sides.

FGMRES sequence In Table 2.8, we notice again that $FGMRES(5)$ needs for each right-hand side, almost always the same number of iterations to reach convergence. It can be also noticed that block methods still improve convergence. Nevertheless, it is not as efficient as in Table 2.7. Indeed, despite a lower number of iterations than FGMRES(5) sequence, the operation ratios are in most of the case greater than one. Exclusively truncated methods can improve the number of operations but only for the lower values of p (5 and 10). However, it can be noticed that BFGMREST$(5, ceil(p/3))$ needs more iterations than BFGMRES for the largest value of p. BFGMRESD(5) is yet the methods which always needs the lower number of iterations; the best iteration ratio, compared with FGMRES(5) sequence, is nearly 4 for $p = 160$.

Comments on the histories of convergence Histories of convergence for $p = 5$ and $p = 10$ are once again plotted in Figures 2.9 and 2.10 respectively. Most of the comments related to Figures 2.7 and 2.8 (Poisson problems for random right-hand sides) remain also valid on this experiment. Indeed, the block size p_d of BFGMRESD(5) at the end of the convergence is found to be only one. The truncated method still does not converge uniformly during the first two restarts for the same values of p and p_f as for the previous example ($p = 5$, $p_f = 2$ and $p = 10$, $p_f = 5, 4$). But then, it behaves nearly similarly for all right-hand sides.

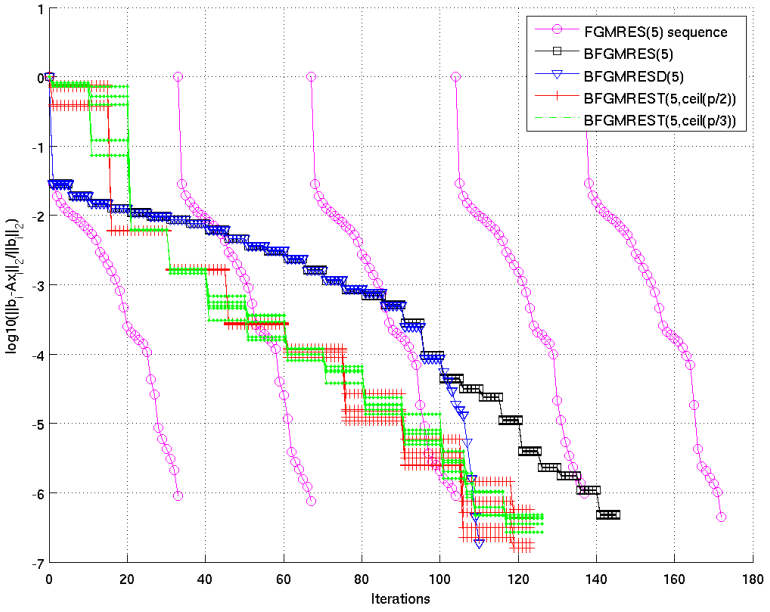

Figure 2.9: Histories of convergence of block methods when solving the convection-diffusion problem for $p = 5$ right-hand sides (Table 2.8).

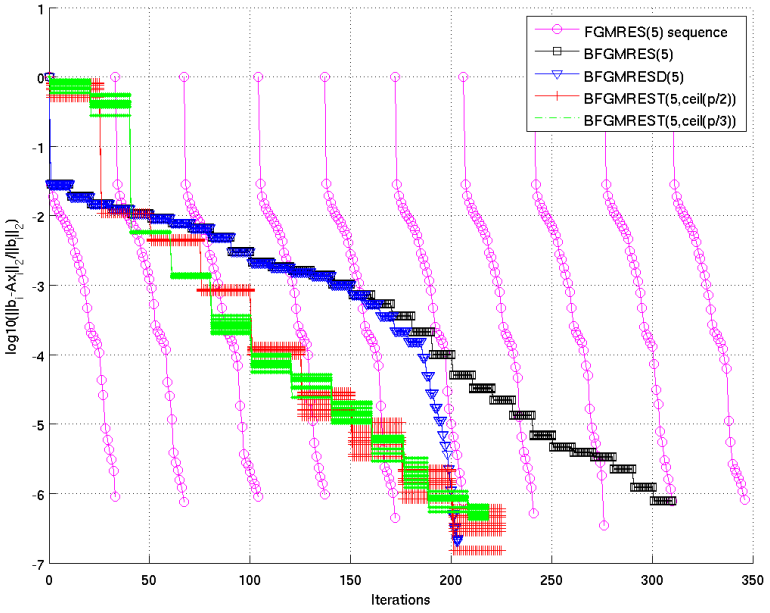

Figure 2.10: Histories of convergence of block methods when solving the convection-diffusion problem for $p = 10$ right-hand sides (Table 2.8).

2.7 Conclusions

In this chapter a flexible variant of GMRES with deflated restarting has been presented. Its principle relies on injecting harmonic Ritz vectors in the Krylov subspace at each restart. Since this method allows the preconditioner to vary from one iteration to the next, the harmonic Ritz vectors are approximate eigenvectors of a different preconditioned matrix from one restart to the next. This method has proven efficient on both academic test cases and real-life applications.

In Section 2.6, we have also illustrated numerically that block methods can greatly improve the convergence of single right-hand side method. A decrease in the number of iterations is observed, especially when useless information for the convergence is removed along the solution phase (BFGMRESD, BFGMREST). However, on the previous examples, the block methods require often more operations than when solving the problem in sequence. This behavior is the consequence of two main features:

- the additional orthogonalization required in the block methods,
- the cheap computational cost of the preconditioning technique ($GMRES(5)$).

Indeed, in this configuration, to reduce significantly the number of iterations does not have a direct impact on the number of operations. Therefore, a favorable situation for block methods would be to use an expensive preconditioner. Nevertheless, in the presented comparisons, no timing or memory estimation were appearing. These quantities are yet of crucial interest in real life applications. Furthermore, considering these quantities when comparing these methods rather than iterations and operations could highlight the interest of block methods. Indeed, on one hand, giving elapsed times could highlight the speed-up obtained when gathering matrix-vector products. On the other hand, in a memory constrained parallel environment, block methods would also appear not as expensive as for sequential experiments, especially when using truncated methods. All these quantities will be analyzed on a real life geophysics application in Chapter 4. Besides, block methods can be numerically improved using spectral information. This would be the purpose of BFGMRESD with deflated restarting ($BFGMRESD-DR$) or BFGMREST with deflated restarting ($BFGMREST-DR$). Such methods will be the object of future works since their derivation and implementation cannot be straightforwardly deduced from Algorithms 11, 12 and 7 respectively. Moreover, another kind of block Arnoldi's process (the Ruhe variant [85]) should be used to guarantee a choice of the parameter k independent of the current block size p_d.

Chapter 3

A three-dimensional geometric two-level method applied to Helmholtz problems

3.1 Introduction

In this chapter, we propose a multigrid preconditioner for the solution of the three-dimensional Helmholtz equation at high wavenumbers with absorbing boundary conditions in a bounded domain Ω:

$$-\Delta u - k^2 u = s$$

where u denotes the wave pressure, k the wavenumber and s a source term. The discretization of such problems is detailed in Appendix A: a second order finite difference discretization scheme is used and the absorbing boundary conditions are formulated with a Perfectly Matched Layer (PML [11, 12]). The finite difference discretization of the Helmholtz problem at high wavenumbers leads to a linear system $Ax = b$ where A is a large sparse matrix. This matrix is complex non-symmetric, indefinite, and generally ill-conditioned. For some years there has been considerable interest in multigrid methods [15, 20, 61, 115] for Helmholtz problems (see also references therein). Nevertheless the indefiniteness of the Helmholtz problem has prevented multigrid methods from being as efficient as they are for symmetric positive-definite problems. Multigrid methods encounter difficulties both in the smoothing procedure and in the coarse grid correction [7, 19, 37, 42, 70]. On the one hand, standard smoothers cannot smooth error components on the intermediate grids. On the other hand, on coarse or very coarse meshes, the approximation of the discrete Helmholtz operator is relatively poor and this creates a difficulty for the coarse grid correction. Remedies have been proposed and analyzed in the case of homogeneous problems [31, 37, 42, 70, 78]. They can be split into three groups:

- In [31, 37, 70], it is advised to use both few grids in the multigrid hierarchy and non-standard smoothers (GMRES) on the coarser levels [37]. However, using few grids in the multigrid hierarchy can be a bottleneck in three dimensions since the coarsest linear system can still be large. Indeed, the solution of the coarse problem could not be affordable in terms of computational resources.

- A second approach is to use a *wave-ray* multigrid algorithm [77] only. It consists in using ray grids in addition of the wave grids to represent the error on the coarser grids of the hierarchy. Thanks to this representation, it is possible to obtain good smoothing properties on the intermediate grids and an efficient coarse grid correction. This method is found efficient for homogeneous problems in both geometric [74, 78] and algebraic multigrid [118]. Notwithstanding, extending this approach to real life applications involves to compute ray functions by possibly solving large eigenvalue problems [122, 123]. Once again, this strategy is expensive in terms of computational resources.

- More recently a third multigrid strategy has been proposed for the numerical solution of the Helmholtz equation [42, 43]. The multigrid method is not directly applied to the discrete Helmholtz operator but to a complex shifted one defined as:

$$-\Delta u - (1 - i\beta)k^2 u$$

where β denotes the shift parameter. Using this shifted operator avoids both the indefiniteness and the coarse grid correction problems [42]. Thus it is possible to build a robust multigrid method with standard multigrid components that is used as a preconditioner for a Krylov subspace method. This solution method has been evaluated on model and realistic geophysical applications involving highly variable coefficients and relatively high wavenumbers. Nevertheless we note that the complexity of the method for pure Helmholtz problems was found to be relatively high at high wavenumbers, see for example the recent analysis on a realistic dataset in geophysics (see [95] in two dimensions and [96] in three dimensions respectively). In [14], the authors apply the shifted strategy to an algebraic multi-level method. Introducing pivoting based on weighted graph matching [35, 103], they perform an incomplete LDL^T factorization on each level of their hierarchy. This multi-level method applied to a shifted Helmholtz operator is then used as a preconditioner for the original Helmholtz problem. This method has been evaluated on realistic geophysical data for both two-dimensional and three-dimensional problems. It has shown efficient to improve the convergence of Krylov methods but its complexity and memory are still relatively high since LDL^T factors must be built and stored. Yet for both geometric and algebraic multilevel preconditioners, an important question is how to determine the shift parameter β. For the algebraic multilevel preconditioner [14], it is advised to use $\beta = 0.1$, this choice is supported by extensive numerical experiments. For the geometric multigrid [42, 43], in two dimensions, it is advised to take $\beta = 0.5$, this choice is led by Fourier analysis [15] and $\beta = 0.4$ when a fourth-order discretization scheme is used for the Helmholtz operator [116]. In three dimensions [96], β is also taken equal to 0.5; the two-dimensional geometric preconditioner [42] is used plane by plane: plane smoothers [88] and semi-coarsening [124] are used. However, when a Fourier analysis is performed in the three-dimensional context, choosing $\beta = 0.6$ would lead to improved results (see Section 3.3.2). Thus, the choice of the shift parameter is really an open question; it depends on the multilevel components and of course on the discretization scheme chosen for the Helmholtz operator.

In the two-dimensional case, the use of a two-grid preconditioner applied to the original Helmholtz operator enables to avoid the choice of a shift parameter [31], and the coarse solution phase of the two-grid algorithm is handled with a sparse direct method. However, this cannot be extended to three dimensions easily; indeed the computational cost of a LU-factorization, even on the coarse grid, may be too severe. Therefore, an iterative method has to be considered on the coarse grid. We are then considering a two-grid cycle with an approximate coarse solution that we call a perturbed two-grid cycle. In this chapter, we will show that a perturbed two-grid cycle can be as efficient as a two-grid method with an exact coarse solution, even when using a really large coarse grid convergence threshold.

Therefore, the purpose of this chapter is to introduce the perturbed two-grid method and to motivate its use for three-dimensional Helmholtz problems. First, we will give some basic information about three-dimensional multigrid and Fourier analysis. We will then perform a smoothing analysis in the Local Fourier Analysis (LFA) sense. Then, a Rigorous Fourier Analysis (RFA) of a perturbed two-grid cycle will be performed. Finally, after a practical smoother selection for the three-dimensional Helmholtz operator with PML, we will analyze the spectrum of this operator preconditioned by one cycle of the perturbed two-level method in the flexible GMRES framework (see Section 2.3.1).

In this chapter, we mainly refer to two monographs related to multigrid: "Multigrid" from U. Trottenberg, C. Oosterlee and A. Schüller [115] and "Multi-Grid Methods and Applications" from W. Hackbusch [61]. Nevertheless, even if they are not cited in the text, we have also found the following books relevant and helpful: "Multigrid methods" from S. F. McCormick [81], "A Multigrid tutorial" from W. L. Briggs and V. E. Henson and S. F. McCormick [20] and "Multigrid methods: fundamental algorithms, model problem analysis and applications" from K. Stüben and U. Trottenberg [113].

3.2 Short introduction to three-dimensional geometric multigrid

The multigrid method is a very efficient multi-scale method for the solution of linear systems arising from the discretization of elliptic partial differential equations. It exploits discretizations with different mesh sizes of a given problem to obtain optimal convergence factor using standard relaxation techniques (Jacobi, Gauss-Seidel...). This method enjoys two main favorable convergence properties for **elliptic** problems:

- the complexity of its algorithm is $O(N)$ where N is the total number of unknowns,
- the convergence factor of a multigrid cycle is essentially independent of the size of the finest grid.

Constant efforts have been made to extend these properties to a larger class of problems. Since we are considering finite difference discretization schemes on structured grids in this chapter, *geometric* multigrid is a natural choice. Therefore coarse grid operators are deduced using the same discretization scheme as for the fine grid operator discretization, considering a coarser mesh size (direct coarse grid approximation). We are using the most standard coarsening: the coarse mesh size is the double of the fine mesh size h ($2h$).

The multigrid method is mainly built on four components: *smoothing*, *restriction*, *prolongation* and *coarse grid solution*. We enumerate their main role in multigrid:

- *Smoothing* enables to avoid solution with high frequency components. Few iterations of a relaxation method, as Jacobi, are used to smooth the high frequency components. Relaxation methods are not efficient in smoothing low-frequency components but these components correspond to high frequency components on a coarse grid. Consequently a hierarchy of grids is used to reduce the low-frequency components efficiently. Thus, transfer operators (restriction, interpolation) are needed to move from a grid to another.
- *Restriction* enables to pass from a fine grid-level to a coarse one.
- *Prolongation* enables to pass from a coarse grid-level to a fine one.
- The final element of multigrid is the solution method on the coarsest grid level. Here, direct or iterative methods can be used to solve this coarse linear system. When several grids are considered in a multigrid hierarchy, to use a direct method is the most natural way to solve the coarse problem as it is of reduced size. However, in the three-dimensional case, when few grids are considered in the multigrid hierarchy, direct methods may be prohibitive in terms of computational resources and iterative methods have to be used. This question will be discussed later in this chapter.

In the next subsections, basic components of a geometric multigrid algorithm in three dimensions are described.

3.2.1 Basic geometric multigrid components

Standard smoothers

As previously said, standard smoothers are often relaxation methods such as Jacobi, forward/back-ward Gauss-Seidel, Red-Black Gauss-Seidel [115], symmetric Gauss-Seidel [93, Section 4.2.6]. The easiest and more general way to write them is first to split the system matrix in the following way:

$$A = D - E - F,$$

where D is the diagonal of A, $-E$ its strictly lower part and $-F$ its strictly upper part respectively. With these notations, we give the expression of a Jacobi, a Gauss-Seidel and a symmetric Gauss-Seidel iteration as in ([101], Chapter 4). Besides, we denote b_h the right-hand side, u the exact solution verifying $Au = b$ and u_h^m the current approximate solution and u_h^{m+1}, the next one.

Jacobi According to the previous notations, a Jacobi iteration can be written as

$$u_h^{m+1} = u_h^m + \omega_r D^{-1}(b_h - Au_h^m),$$

where ω_r denotes a relaxation parameter ($0 < \omega_r < 2$) and its iteration matrix S_h is deduced from

$$u - u_h^{m+1} = S_h(u - u_h^m) = \left(I_h - \omega_r D^{-1}A\right)(u - u_h^m),$$

where I_h denotes the identity matrix.

Forward Gauss-Seidel With the same notations as above, a forward Gauss-Seidel iteration can be written as:

$$x_{k+1} = (D - E)^{-1}(Fx_k + b).$$

Its iteration matrix S_h is then:

$$u - u_h^{m+1} = S_h(u - u_h^m) = (D - E)^{-1}F(u - u_h^m).$$

Backward Gauss-Seidel With the same notations as above, a backward Gauss-Seidel iteration can be written as:

$$x_{k+1} = (D - F)^{-1}(Ex_k + b).$$

Its iteration matrix S_h is

$$u - u_h^{m+1} = S_h(u - u_h^m) = (D - F)^{-1}E(u - u_h^m).$$

Symmetric Gauss-Seidel A symmetric Gauss-Seidel iteration consists of a first iteration of forward Gauss-Seidel and a second iteration of backward Gauss-Seidel. With the same notations as above, a symmetric Gauss-Seidel iteration can be written as:

$$x_{k+1} = (D - F)^{-1}(E((D - E)^{-1}(Fx_k + b)) + b).$$

Its iteration matrix S_h is then

$$u - u_h^{m+1} = S_h(u - u_h^m) = (D - F)^{-1}E(D - E)^{-1}F(u - u_h^m).$$

Example 1. *We consider the matrix A resulting from the discretization of three-dimensional Helmholtz type operators with Dirichlet boundary conditions (discretized with a classical second-order finite difference scheme for a vertex-centered arrangement) also denoted by L_h:*

$$L_h = -\Delta_h - \kappa^2 I_h \text{ with } \kappa \in \mathbb{C}.$$

The stencil of L_h is, using the stencil notation defined in ([115], section 1.3.4),

$$L_h = \frac{1}{h^2}\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6-h^2\kappa^2 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right]_h,$$

where κ denotes a term proportional to the wavenumber k (see Appendix A). If $\kappa = k$, the original three-dimensional Helmholtz operator is obtained. If $\kappa = \sqrt{1-\beta}ik, \beta \in [0,1]$, the shifted three-dimensional Helmholtz operator is obtained ([40], Chapter 7). We deduce the stencils of D, E and F as:

$$D = \frac{1}{h^2}\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 6-h^2\kappa^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right]_h,$$

$$-E = \frac{1}{h^2}\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right]_h,$$

$$-F = \frac{1}{h^2}\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right]_h.$$

It has to be noticed that if the wavenumber κ is taken equal to 0, the different expressions hold for the negative Laplacian operator $-\Delta_h$ with Dirichlet boundary conditions.

Restriction

The aim of restriction is to transfer information from a fine grid to a coarser one. We denote by Ω_ι, $\iota \in \mathbb{R}$, the grid defined by $\Omega_\iota = G_\iota \cap \Omega$ where G_ι denotes the infinite grid:

$$G_\iota = \left\{(x, y, z) | (x, y, z) = (i\iota, j\iota, k\iota); (i, j, k) \in \mathbb{Z}^3\right\}, \tag{3.1}$$

and $\Omega \subset \mathbb{R}^3$ is a closed bounded parallelepiped domain. In geometric multigrid (vertex-centered case), only one out of two points per direction of the fine grid Ω_h will remain on the coarse grid Ω_{2h}. Considering Figure 3.1, the remaining points after restriction are the ones marked with a bullet (•).

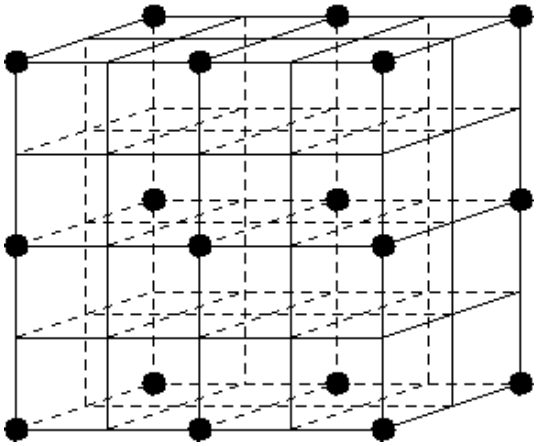

Figure 3.1: A 3D fine grid with standard geometric coarsening (• : coarse grid point).

A natural choice for restriction could be injection. However, this choice is not very practicable in a multigrid context: it implies choosing of a high-order prolongation in order to maintain just convergence. This is debated in Remark (2.7.1) in [115] where a relation between the orders of the prolongation and restriction and the order of the differential operator is given to guarantee the efficiency of multigrid on SPD problems. The order of the prolongation m_P is defined as the highest degree plus one of polynomials that are interpolated exactly [61, Section 3.4.3]. Similarly the order of the restriction m_R is defined as the highest degree plus one of polynomial that are restricted exactly. In order to obtain an efficient multigrid algorithm, the sum of m_R and m_P must be larger than the order of the differential operator denoted by m_{PDE} (higher derivative degree in the partial differential equation (PDE)):

$$m_R + m_P > m_{PDE}.$$

Since the order of the injection is zero, a quadratic interpolation should then be used for a Helmholtz type of PDE.

Thus, a frequent choice for restriction is the *Full weighting (FW)* operator; its order is equal to two ($m_R = 2$). Its principle relies on weighting fine grid coefficients around the neighboring coarse grid points. Considering coordinates $(x, y, z) \in \Omega_{2h}$, the *FW*-restriction function I_h^{2h} applied to a fine grid function r_h in the three-dimensional case is:

$$\begin{aligned}
I_h^{2h}(r_h(x,y,z)) \quad = \quad & \frac{1}{64}(8r_h(x,y,z) + 4r_h(x+h,y,z) + 4r_h(x,y+h,z) + 4r_h(x,y,z+h) \\
& +4r_h(x-h,y,z) + 4r_h(x,y-h,z) + 4r_h(x,y,z-h) \\
& +2r_h(x+h,y+h,z) + 2r_h(x,y+h,z+h) + 2r_h(x+h,y,z+h) \\
& +2r_h(x-h,y-h,z) + 2r_h(x,y-h,z-h) + 2r_h(x-h,y,z-h) \\
& +2r_h(x-h,y+h,z) + 2r_h(x,y-h,z+h) + 2r_h(x-h,y,z+h) \\
& +2r_h(x+h,y-h,z) + 2r_h(x,y+h,z-h) + 2r_h(x+h,y,z-h) \\
& +r_h(x+h,y+h,z+h) + r_h(x+h,y+h,z-h) + r_h(x+h,y-h,z+h) \\
& +r_h(x-h,y+h,z+h) + r_h(x+h,y-h,z-h) + r_h(x-h,y-h,z+h) \\
& +r_h(x-h,y+h,z-h) + r_h(x-h,y-h,z-h)).
\end{aligned}$$

Prolongation

Prolongation transfers information from a coarse ($2h$) to a fine grid h. The prolongation will be based on a trilinear interpolation ($m_P = 2$). Considering Figures 3.2 and 3.3, it can be seen how fine grid points (empty polygons) are deduced from coarse grid points (bullets). Coefficients used for the points on the cube faces are described in Figure 3.3 for the trilinear interpolation case. For the cube center, represented by an empty disk in Figure 3.2, all the coefficients from the eight corners are weighted by a factor of $\frac{1}{8}$. Coarse grid points are remaining with the same associated values. In fact, the trilinear interpolation is the adjoint of the *Full weighting (FW)* restriction.

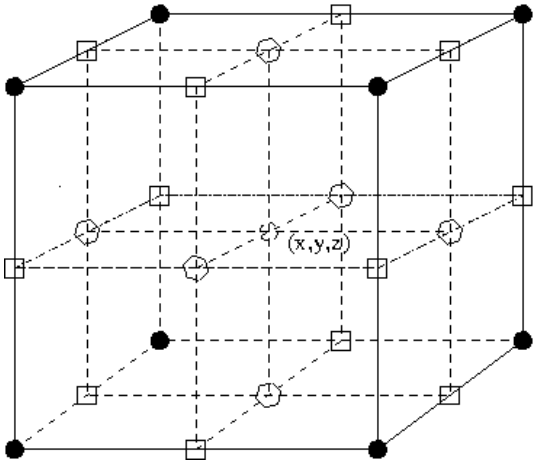

Figure 3.2: Fine grid for a 3D trilinear interpolation (•: coarse grid points).

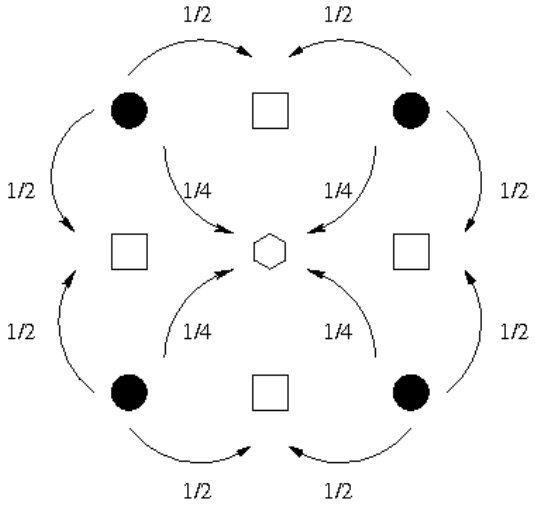

Figure 3.3: Weightings for 3D interpolation on a cube face (•: coarse grid points).

In the next section, we depict how to assemble all these components to obtain a geometric multigrid algorithm.

3.2.2 Geometric multigrid algorithms

In this section, we introduce notations that we will use later in this chapter. We denote by I_h^{2h} the restriction operator, I_{2h}^h the prolongation operator, L_h the fine grid and L_{2h} the coarse grid operators. The vector u is the exact solution satisfying $b_h = L_h u$, u_h the current approximate solution at the fine level, u_{2h} the solution of the current coarse problem, b_{2h} the coarse right-hand side. We denote by ν_1 and ν_2 the number of pre-

and post smoothing iterations respectively and $\mathcal{S}$ the smoothing procedure. Algorithm 13 depicts a classical two-grid cycle.

Algorithm 13 Two-grid cycle $TG(L_h, u_h, b_h)$.

1: Presmoothing: $u_h := \mathcal{S}(L_h, u_h, b_h, \nu_1)$
2: Compute the residual r_h: $r_h = b_h - L_h\, u_h$
3: Restrict the residual: $b_{2h} = I_h^{2h}\, r_h$
4: Solve on Ω_{2h}: $L_{2h} u_{2h} = b_{2h}$
5: Interpolate the coarse solution u_{2h} to obtain a correction of the fine solution u_h: $I_{2h}^h\, u_{2h}$
6: Add this correction to the solution: $u_h := u_h + I_{2h}^h\, u_{2h}$
7: Postsmoothing: $u_h := \mathcal{S}(L_h, u_h, b_h, \nu_2)$

Thus, a two-grid cycle consists first in a smoothing step (presmoothing), the restriction of residual $b_h - L_h u_h$, this last operation gives the coarse right-hand side, the coarse problem is then solved, the coarse solution is interpolated and the obtained correction added to the fine solution. Figure 3.4 represents a V-cycle in the two-grid case which corresponds to Algorithm 13.

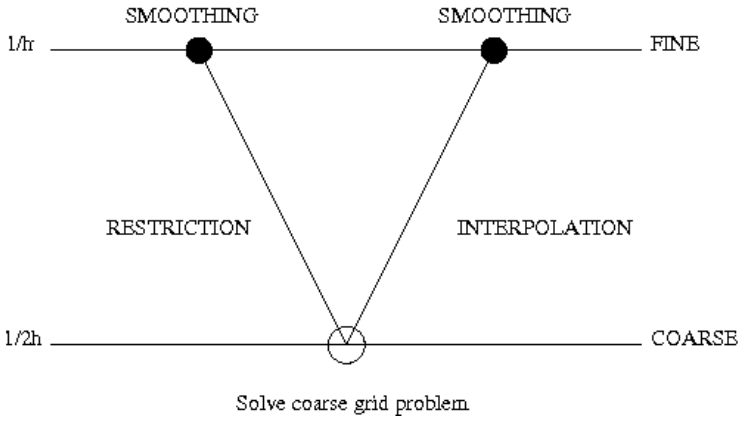

Figure 3.4: Two-grid V-cycle.

The two-grid cycle is the simplest form of a multigrid cycle. This process can be generalized to any number of grid levels, as the next recursive algorithm shows, denoted by $MG(L_h, u_h, b_h)$ in Algorithm 14.

Algorithm 14 Multigrid cycle $MG(L_h, u_h, b_h)$.

1: Presmoothing: $u_h := \mathcal{S}(L_h, u_h, b_h, \nu_1)$
2: Compute the residual r_h: $r_h = b_h - L_h\, u_h$
3: Restrict the residual: $b_{2h} = I_h^{2h}\, r_h$
4: Set $u_{2h} := 0$.
5: **for** it=1:γ **do**
6: $\quad u_{2h} := MG(L_{2h}, u_{2h}, b_{2h})$
7: **end for**
8: Interpolate the coarse solution u_{2h} to obtain a correction of the fine solution u_h: $I_{2h}^h\, u_{2h}$
9: Add this correction to the solution: $u_h := u_h + I_{2h}^h\, u_{2h}$
10: Postsmoothing: $u_h := \mathcal{S}(L_h, u_h, b_h, \nu_2)$

In fact, the shape of the multigrid cycle depends on the γ parameter. The shape of the multigrid cycle can be changed in order to possibly improve the convergence behavior, combining iteratively multigrid components in a different way. A W-cycle is obtained with $\gamma = 2$, a F-cycle with a combination of $\gamma = 1$ and $\gamma = 2$, and a V-cycle with $\gamma = 1$ (Figure 3.5).

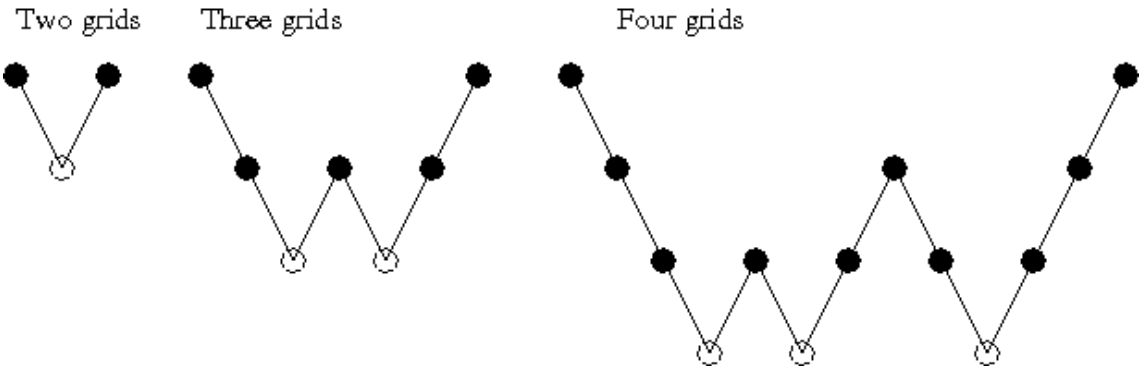

Figure 3.5: F-cycles for two, three and four grids (from left to right).

Notwithstanding, we will only consider a two-grid cycle in the following of this chapter. This is motivated by our application. In the next section, we present a method to analyze the convergence properties of a three-dimensional geometric two-grid cycle. This technique is named Fourier analysis.

3.3 Rigorous and Local Fourier Analysis of a two-grid method

First, we write the iteration matrix M_h of a classical two-grid method with the notations of Section 3.2.2:

$$M_h(u - u_h) = S_h^{\nu_2}(I_h - I_{2h}^h L_{2h}^{-1} I_h^{2h} L_h) S_h^{\nu_1}(u - u_h). \tag{3.2}$$

Fourier analysis aims at obtaining an estimation of the norm and spectral radius of M_h and at analyzing the smoothing behavior of relaxation procedures. In fact, the two-norm of M_h leads the convergence of the two-grid cycle. Indeed, since the $(m+1)th$ iterate u_h^{m+1} satisfy

$$u - u_h^{m+1} = M_h(u - u_h^m),$$

it follows that

$$\frac{\|u - u_h^{m+1}\|_2}{\|u - u_h^m\|_2} \leq \|M_h\|_2 = \sqrt{\rho(M_h^H M_h)}.$$

Furthermore the spectral radius of M_h ($\rho(M_h)$) is equal to the convergence factor of the two-grid cycle. This last quantity plays an important role in the multigrid convergence theory approximating the asymptotic behavior of the two-level cycle.

The Fourier analysis implements techniques to block diagonalize the operator M_h in a Fourier basis [61, p. 25]. This block diagonal representation of M_h enables then to easily deduce the two-norm of M_h.

In this section, we will present two different Fourier analysis. The first one is the Rigorous Fourier analysis (RFA) [115, Section 3.3]: the two-grid convergence factor can be deduced in the situations enumerated in [115, Section 3.4.3]. We will focus on the case where the operator satisfies Dirichlet boundary conditions and where a Jacobi smoother is used. The second one is the Local Fourier analysis (LFA) [115, Chapter 4] or local mode analysis [15]: the influence of boundary conditions is not taken into account and smoothers such as Gauss-Seidel can be analyzed.

3.3.1 Rigorous Fourier Analysis (RFA) of a two-grid method

We now introduce some of the main elements of RFA to study the two-grid convergence. First, we consider the orthogonal basis of the fine grid space $\Omega_h = G_h \cap [0,1]^3$ spanned by the eigenfunctions of L_h:

$$\varphi_h^{l_1,l_2,l_3}(x,y,z) = \sin(l_1\pi x)\,\sin(l_2\pi y)\,\sin(l_3\pi z), \text{ for } l_1,l_2,l_3 = 1,...,n-1 \text{ and } (x,y,z) \in \Omega_h,$$

where n denotes the inverse of the mesh grid size h, $n = 1/h$. These functions are eigenfunctions of the Helmholtz operator with Dirichlet boundary conditions (see Example 1). We introduce then the at most

eight-dimensional spaces of harmonics for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$ [115, Equation (3.4.1)]:

$$\begin{aligned} E_h^{l_1,l_2,l_3} &= span[\varphi_h^{l_1,l_2,l_3}, -\varphi_h^{n-l_1,n-l_2,n-l_3}, -\varphi_h^{n-l_1,l_2,l_3}, \varphi_h^{l_1,n-l_2,n-l_3}, \\ &\quad -\varphi_h^{l_1,n-l_2,l_3}, \varphi_h^{n-l_1,l_2,n-l_3}, -\varphi_h^{l_1,l_2,n-l_3}, \varphi_h^{n-l_1,n-l_2,l_3}], \end{aligned}$$

which allows to block diagonalize [61, p. 25] the two-grid iteration matrix M_h in the Fourier basis $\mathcal{Q}_h$ defined by:

$$\mathcal{Q}_h = \left[\left[\left[\left[E_h^{l_1,l_2,l_3}\right]_{l_1=1,...,n/2}\right]_{l_2=1,...,n/2}\right]_{l_3=1,...,n/2}\right]. \tag{3.3}$$

In fact, M_h leaves the harmonic spaces $E_h^{l_1,l_2,l_3}$ invariant for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$ if e.g. Jacobi or Red-Black Gauss-Seidel is used as a smoother.

The $E_h^{l_1,l_2,l_3}$ spaces are eight-, four-, two- and one-dimensional spaces with respect to the values of l_1, l_2, l_3 respectively:

$$dim(E_h^{l_1,l_2,l_3}) = \begin{cases} 8 & \text{if } l_1 \neq \frac{n}{2} \text{ and } l_2 \neq \frac{n}{2} \text{ and } l_3 \neq \frac{n}{2}, \\ 4 & \text{if } l_1 = \frac{n}{2} \text{ or } l_2 = \frac{n}{2} \text{ or } l_3 = \frac{n}{2}, \\ 2 & \text{if } l_1 = l_3 = \frac{n}{2} \text{ or } l_1 = l_2 = \frac{n}{2} \text{ or } l_2 = l_3 = \frac{n}{2}, \\ 1 & \text{if } l_1 = l_2 = l_3 = \frac{n}{2}. \end{cases}$$

Similarly as on the fine grid, we introduce the eigenfunctions on the coarse grid space $\Omega_{2h} = G_{2h} \cap [0,1]^3$:

$$\varphi_{2h}^{l_1,l_2,l_3}(x,y,z) = \sin(l_1\pi x)\, \sin(l_2\pi y)\, \sin(l_3\pi z), \text{ for } l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1 \text{ and } (x,y,z) \in \Omega_{2h}.$$

On Ω_{2h}, the $E_{2h}^{l_1,l_2,l_3}$ spaces are one-dimensional spaces only. Indeed, the eigenfunctions spanning $E_{2h}^{l_1,l_2,l_3}$ coincide up to their sign on Ω_{2h} for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$:

$$\begin{aligned} \varphi_{2h}^{l_1,l_2,l_3}(x,y,z) &= -\varphi_{2h}^{n-l_1,n-l_2,n-l_3}(x,y,z) = -\varphi_{2h}^{n-l_1,l_2,l_3}(x,y,z) \\ &= \varphi_{2h}^{l_1,n-l_2,n-l_3}(x,y,z) = -\varphi_{2h}^{l_1,n-l_2,l_3}(x,y,z) \\ &= \varphi_{2h}^{n-l_1,l_2,n-l_3}(x,y,z) = -\varphi_{2h}^{l_1,l_2,n-l_3}(x,y,z) \\ &= \varphi_{2h}^{n-l_1,n-l_2,l_3}(x,y,z), \ \forall\ (x,y,z) \in \Omega_{2h}. \end{aligned}$$

Practically, this means that

$$E_{2h}^{l_1,l_2,l_3} = span\left[\varphi_{2h}^{l_1,l_2,l_3}\right].$$

Later in this section, we denote operators written in the Fourier basis $\mathcal{Q}_h$ with a hat. Thus, denoting by Q_h the matrix whose columns span the Fourier basis, $\mathcal{Q}_h = span[Q_h]$, we obtain:

$$M_h = h^3 Q_h \widehat{M_h} Q_h^H.$$

To simplify these notations, we introduce the symbol $\widehat{=}$ and write:

$$M_h \widehat{=} \widehat{M_h}.$$

For each triplet (l_1, l_2, l_3) we have $\widehat{M}_h(l_1, l_2, l_3) \widehat{=} M_h|_{E_h^{l_1,l_2,l_3}}$. Thus, we will have a block diagonal representation of M_h in the Fourier basis:

$$M_h \widehat{=} \widehat{M}_h = \left[\widehat{M}_h(l_1, l_2, l_3)\right]_{l_1,l_2,l_3=1,...,n/2}.$$

In the following, we will deduce a representation of the two-grid iteration matrix (Equation 3.2) with respect to the $E_h^{l_1,l_2,l_3}$ spaces considering a Jacobi smoother. We first give the representation with respect to $E_h^{l_1,l_2,l_3}$ of the three-dimensional Helmholtz operator with Dirichlet boundary conditions (Example 1) both on the fine and the coarse grid. We will then detail the Fourier representation of the trilinear interpolation and the full-weighting restriction denoted by $\widehat{I}_{2h}^h$ and $\widehat{I}_h^{2h}$ respectively. This will enable us to obtain a representation in the Fourier basis of the coarse grid correction operator:

$$K_h^{2h} = I_h - I_{2h}^h L_{2h}^{-1} I_h^{2h} L_h.$$

For that purpose, we introduce ξ, η and γ; these parameters will be used to write more synthetically the different operators in the Fourier basis:

$$\left\{\begin{array}{rcl} \xi & = & \sin^2\left(\dfrac{l_1\pi h}{2}\right), \\ \eta & = & \sin^2\left(\dfrac{l_2\pi h}{2}\right), \\ \gamma & = & \sin^2\left(\dfrac{l_3\pi h}{2}\right). \end{array}\right. \tag{3.4}$$

Lemma 1. *The harmonic spaces $E_h^{l_1,l_2,l_3}$ for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$ are invariant under Helmholtz type operators L_h with Dirichlet boundary conditions:*

$$L_h : E_h^{l_1,l_2,l_3} \longrightarrow E_h^{l_1,l_2,l_3}, \text{ for } l_1, l_2, l_3 = 1, ..., \frac{n}{2}.$$

The operator L_h can be represented in the Fourier basis as a block diagonal matrix. Its representation with respect to the spaces $E_h^{l_1,l_2,l_3}$ consists in diagonal blocks as described below, using notations of Equation 3.4:

- *For $l_1, l_2, l_3 = 1, ..., \dfrac{n}{2} - 1$, we obtain the following 8×8 block*

$$\widehat{L_h}(l_1, l_2, l_3) = diag\left(\left(\begin{array}{c} \frac{4}{h^2}(\xi+\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(3-\xi-\eta-\gamma) - \kappa^2 \\ \frac{4}{h^2}(1-\xi+\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(2+\xi-\eta-\gamma) - \kappa^2 \\ \frac{4}{h^2}(1+\xi-\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(2-\xi+\eta-\gamma) - \kappa^2 \\ \frac{4}{h^2}(1+\xi+\eta-\gamma) - \kappa^2 \\ \frac{4}{h^2}(2-\xi-\eta+\gamma) - \kappa^2 \end{array}\right)\right).$$

- *For $l_1 = \dfrac{n}{2}$, $l_2, l_3 < \dfrac{n}{2}$ or $l_2 = \dfrac{n}{2}$, $l_1, l_3 < \dfrac{n}{2}$ or $l_3 = \frac{n}{2}$, $l_1, l_2 < \dfrac{n}{2}$, we obtain the following 4×4 block*

$$\widehat{L_h}(l_1, l_2, l_3) = diag\left(\left(\begin{array}{c} \frac{4}{h^2}(\xi+\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(3-\xi-\eta-\gamma) - \kappa^2 \\ \frac{4}{h^2}(1-\xi+\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(2+\xi-\eta-\gamma) - \kappa^2 \end{array}\right)\right).$$

- *For $l_1 = l_2 = \dfrac{n}{2}$, $l_3 < \dfrac{n}{2}$ or $l_1 = l_3 = \dfrac{n}{2}$, $l_2 < \dfrac{n}{2}$ or $l_2 = l_3 = \dfrac{n}{2}$, $l_1 < \dfrac{n}{2}$, we obtain the following 2×2 block*

$$\widehat{L_h}(l_1, l_2, l_3) = diag\left(\left(\begin{array}{c} \frac{4}{h^2}(\xi+\eta+\gamma) - \kappa^2 \\ \frac{4}{h^2}(3-\xi-\eta-\gamma) - \kappa^2 \end{array}\right)\right).$$

- *For $l_1 = l_2 = l_3 = \dfrac{n}{2}$, we obtain the following 1×1 block*

$$\widehat{L_h}(l_1, l_2, l_3) = \frac{4}{h^2}(\xi+\eta+\gamma) - \kappa^2.$$

Proof. Obviously, since the eigenfunctions spanning $E_h^{l_1,l_2,l_3}$ are eigenfunctions of L_h, the harmonic spaces are invariant under L_h. The representation of L_h with respect to the harmonic space $E_h^{l_1,l_2,l_3}$ is obtained by calculating the image of each of its basis functions using trigonometric formulas:

$$\begin{aligned}
L_h\,\varphi_h^{l_1,l_2,l_3} &= (\frac{4}{h^2}(\xi+\eta+\gamma)-\kappa^2)\varphi_h^{l_1,l_2,l_3},\\
-L_h\varphi_h^{n-l_1,n-l_2,n-l_3} &= -(\frac{4}{h^2}(3-\xi-\eta-\gamma)-\kappa^2)\varphi_h^{n-l_1,n-l_2,n-l_3},\\
-L_h\varphi_h^{n-l_1,l_2,l_3} &= -(\frac{4}{h^2}(1-\xi+\eta+\gamma)-\kappa^2)\varphi_h^{n-l_1,l_2,l_3},\\
L_h\varphi_h^{l_1,n-l_2,n-l_3} &= (\frac{4}{h^2}(2+\xi-\eta-\gamma)-\kappa^2)\varphi_h^{l_1,n-l_2,n-l_3},\\
-L_h\varphi_h^{l_1,n-l_2,l_3} &= -(\frac{4}{h^2}(1+\xi-\eta+\gamma)-\kappa^2)\varphi_h^{l_1,n-l_2,l_3},\\
L_h\varphi_h^{n-l_1,l_2,n-l_3} &= (\frac{4}{h^2}(2-\xi+\eta-\gamma)-\kappa^2)\varphi_h^{n-l_1,l_2,n-l_3},\\
-L_h\varphi_h^{l_1,l_2,n-l_3} &= -(\frac{4}{h^2}(1+\xi+\eta-\gamma)-\kappa^2)\varphi_h^{l_1,l_2,n-l_3},\\
L_h\varphi_h^{n-l_1,n-l_2,l_3} &= (\frac{4}{h^2}(2-\xi-\eta+\gamma)-\kappa^2)\varphi_h^{n-l_1,n-l_2,l_3}.
\end{aligned}$$

Considering the different values of l_1, l_2, l_3, we obtain the results proposed in Lemma 1 taking into account the dimensions of spaces $E_h^{l_1,l_2,l_3}$.

□

Lemma 2. *On the coarse grid space Ω_{2h}, $E_{2h}^{l_1,l_2,l_3}$ is invariant under the coarse three-dimensional Helmholtz operator L_{2h}:*

$$L_{2h} : E_{2h}^{l_1,l_2,l_3} \longrightarrow E_{2h}^{l_1,l_2,l_3},\ \textit{for}\ l_1,l_2,l_3 = 1,...,\frac{n}{2}$$

and its representation with respect to $E_{2h}^{l_1,l_2,l_3}$ is

$$\widehat{L}_{2h}(l_1,l_2,l_3) = \frac{4}{h^2}((1-\xi)\xi+(1-\eta)\eta+(1-\gamma)\gamma)-\kappa^2.$$

Proof. The proof is similar to that of Lemma 1:

$$L_{2h}\varphi_{2h}^{l_1,l_2,l_3} = [\frac{4}{h^2}((1-\xi)\xi+(1-\eta)\eta+(1-\gamma)\gamma)-\kappa^2]\varphi_{2h}^{l_1,l_2,l_3}.$$

□

We now focus on the grid transfer operators: the full-weighting restriction and the trilinear interpolation. Once the representation in the Fourier basis of the restriction is obtained, the representation of the interpolation is deduced straightforwardly since it is its adjoint [115, Remark 3.3.5].

Lemma 3. *The range of $I_h^{2h}(E_h^{l_1,l_2,l_3})$ coincides with the coarse harmonic space $E_{2h}^{l_1,l_2,l_3}$ for $l_1,l_2,l_3 = 1,...,\frac{n}{2}-1$:*

$$I_h^{2h} : E_h^{l_1,l_2,l_3} \longrightarrow span[\varphi_{2h}^{l_1,l_2,l_3}],\ \textit{for}\ l_1,l_2,l_3 = 1,...,\frac{n}{2}-1.$$

The full-weighting restriction can be block-diagonalized in the Fourier basis and has the following block representation:

- *For* $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$*, we have the following* 8×1 *block*

$$\widehat{I}_h^{2h}(l_1, l_2, l_3) \quad = \quad \begin{bmatrix} (1-\xi)(1-\eta)(1-\gamma) \\ \xi\eta\gamma \\ \xi(1-\eta)(1-\gamma) \\ (1-\xi)\eta\gamma \\ (1-\xi)\eta(1-\gamma) \\ \xi(1-\eta)\gamma \\ (1-\xi)(1-\eta)\gamma \\ \xi\eta(1-\gamma) \end{bmatrix}^T .$$

- *For* $l_1 = \frac{n}{2}$ *or* $l_2 = \frac{n}{2}$ *or* $l_3 = \frac{n}{2}$,

$$\widehat{I}_h^{2h}(l_1, l_2, l_3) \quad = \quad 0.$$

Proof. First, we apply the restriction to the basis functions of $E_h^{l_1,l_2,l_3}$ for $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$ using the full-weighting restriction and trigonometric identities:

$$I_h^{2h} \begin{Bmatrix} \varphi_h^{l_1,l_2,l_3} \\ -\varphi_h^{n-l_1,n-l_2,n-l_3} \\ -\varphi_h^{n-l_1,l_2,l_3} \\ \varphi_h^{l_1,n-l_2,n-l_3} \\ -\varphi_h^{l_1,n-l_2,l_3} \\ \varphi_h^{n-l_1,l_2,n-l_3} \\ -\varphi_h^{l_1,l_2,n-l_3} \\ \varphi_h^{n-l_1,n-l_2,l_3} \end{Bmatrix} \quad = \quad \begin{Bmatrix} (1-\xi)(1-\eta)(1-\gamma) \\ \xi\eta\gamma \\ \xi(1-\eta)(1-\gamma) \\ (1-\xi)\eta\gamma \\ (1-\xi)\eta(1-\gamma) \\ \xi(1-\eta)\gamma \\ (1-\xi)(1-\eta)\gamma \\ \xi\eta(1-\gamma) \end{Bmatrix} \varphi_{2h}^{l_1,l_2,l_3}.$$

These equalities prove that $I_h^{2h}(E_h^{l_1,l_2,l_3}) = span[\varphi_{2h}^{l_1,l_2,l_3}]$ for $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$, and give the block representation of I_h^{2h} in the Fourier basis. Furthermore, if $l_1 = \frac{n}{2}$ or $l_2 = \frac{n}{2}$ or $l_3 = \frac{n}{2}$, the coarse eigenfunctions $\varphi_{2h}^{l_1,l_2,l_3}$ are zero. Indeed, the definition of $\varphi_{2h}^{l_1,l_2,l_3}$ gives, with $(j_1, j_2, j_3) \in \mathbb{N}^3$,

$$\varphi_{2h}^{l_1,l_2,l_3}(j_1 2h, j_2 2h, j_3 2h) = \sin(l_1 \pi j_1 2h)\, \sin(l_2 \pi j_2 2h)\, \sin(l_3 \pi j_2 2h), \text{ for } (j_1 2h, j_2 2h, j_3 2h) \in \Omega_{2h},$$

then, if $l_1 = \frac{n}{2}$, we have $\varphi_{2h}^{\frac{n}{2},l_2,l_3}(j_1 2h, j_2 2h, j_3 2h) = \sin(\frac{n}{2}\pi j_1 2h)\, \sin(l_2 \pi j_2 2h)\, \sin(l_3 \pi j_2 2h)$ and since j_1 is an integer, it follows that

$$\sin(\frac{n}{2}\pi j_1 2h) = \sin(\frac{n}{2}\pi j_1 \frac{2}{n}) = \sin(\pi j_1) = 0.$$

Therefore, we have

$$\varphi_{2h}^{\frac{n}{2},l_2,l_3}(j_1 2h, j_2 2h, j_3 2h) = 0.$$

The proof is similar for $l_2 = \frac{n}{2}$ and $l_3 = \frac{n}{2}$.

□

Lemma 4. *The range of* $I_{2h}^h(\varphi_{2h}^{l_1,l_2,l_3})$ *is* $E_h^{l_1,l_2,l_3}$*, for* $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$*:*

$$I_{2h}^h : span[\varphi_{2h}^{l_1,l_2,l_3}] \longrightarrow E_h^{l_1,l_2,l_3}, \text{ for } l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1.$$

The trilinear interpolation can be block diagonalized in the Fourier basis with the following block representation for $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$*:*

$$\widehat{I}_{2h}^h(l_1, l_2, l_3) = \left(\widehat{I}_h^{2h}(l_1, l_2, l_3)\right)^T .$$

Proof. Using trilinear interpolation and trigonometric identities, it follows that

$$\begin{aligned} I_{2h}^{h}\varphi_{2h}^{l_1,l_2,l_3} &= +(1-\xi)(1-\eta)(1-\gamma)\varphi_h^{l_1,l_2,l_3} \\ &\quad -\xi\eta\gamma\varphi_h^{n-l_1,n-l_2,n-l_3} \\ &\quad -\xi(1-\eta)(1-\gamma)\varphi_h^{n-l_1,l_2,l_3} \\ &\quad +(1-\xi)\eta\gamma\varphi_h^{l_1,n-l_2,n-l_3} \\ &\quad -(1-\xi)\eta(1-\gamma)\varphi_h^{l_1,n-l_2,l_3} \\ &\quad +\xi(1-\eta)\gamma\varphi_h^{n-l_1,l_2,n-l_3} \\ &\quad -(1-\xi)(1-\eta)\gamma\varphi_h^{l_1,l_2,n-l_3} \\ &\quad +\xi\eta(1-\gamma)\varphi_h^{n-l_1,n-l_2,l_3}. \end{aligned}$$

Thus $I_{2h}^{h}\varphi_{2h}^{l_1,l_2,l_3}$ is in $E_h^{l_1,l_2,l_3}$ for $l_1,l_2,l_3 = 1,...,\frac{n}{2}-1$ and, from the representation of $\widehat{I}_h^{2h}(l_1,l_2,l_3)$ obtained in Lemma 3 comes the fact that $\widehat{I}_{2h}^{h}(l_1,l_2,l_3) = (\widehat{I}_h^{2h}(l_1,l_2,l_3))^T$. □

We now give in Theorem 1, the representation of the coarse grid correction operator K_h^{2h} with respect to the harmonic spaces $E_h^{l_1,l_2,l_3}$ for $l_1,l_2,l_3 = 1,...,\frac{n}{2}$.

Theorem 1. *We consider the three-dimensional Helmholtz operators (fine grid L_h, coarse grid L_{2h}) as defined in Example 1, a trilinear interpolation I_{2h}^{h}, its adjoint as restriction I_h^{2h}. With these components, the harmonic spaces $E_h^{l_1,l_2,l_3}$ are invariant under the coarse grid correction operator $K_h^{2h} = I_h - I_{2h}^{h}L_{2h}^{-1}I_h^{2h}L_h$ for $l_1,l_2,l_3 = 1,...,\frac{n}{2}$:*

$$K_h^{2h} : E_h^{l_1,l_2,l_3} \longrightarrow E_h^{l_1,l_2,l_3} \text{ for } l_1,l_2,l_3 = 1,...,\frac{n}{2}.$$

K_h^{2h} can also be block diagonalized in the Fourier basis and its representation with respect to the harmonic spaces $E_h^{l_1,l_2,l_3}$ for $l_1,l_2,l_3 = 1,...,\frac{n}{2}$ is

$$\widehat{K}_h^{2h}(l_1,l_2,l_3) = \begin{cases} I_8 - [b_i\, c_j]_{8,8}/\Lambda \text{ if } l_1,l_2,l_3 < \frac{n}{2} \\ I_4 \text{ if } l_1 = \frac{n}{2} \text{ or } l_2 = \frac{n}{2} \text{ or } l_3 = \frac{n}{2} \\ I_2 \text{ if } l_1 = l_3 = \frac{n}{2} \text{ or } l_1 = l_2 = \frac{n}{2} \text{ or } l_2 = l_3 = \frac{n}{2} \\ I_1 \text{ if } l_1 = l_2 = l_3 = \frac{n}{2} \end{cases}, \tag{3.5}$$

with

$$\begin{cases} I_j \text{ is the } j\times j \text{ identity matrix,} \\ \Lambda = \frac{4}{h^2}(1-\xi)\xi + (1-\eta)\eta + (1-\gamma)\gamma - \kappa^2, \\ b_1 = (1-\xi)(1-\eta)(1-\gamma) \quad c_1 = (1-\xi)(1-\eta)(1-\gamma)\left(\frac{4}{h^2}(\xi+\eta+\gamma) - \kappa^2\right) \\ b_2 = \xi\eta\gamma \quad c_2 = \xi\eta\gamma\left(\frac{4}{h^2}(3-\xi-\eta-\gamma) - \kappa^2\right) \\ b_3 = \xi(1-\eta)(1-\gamma) \quad c_3 = \xi(1-\eta)(1-\gamma)\left(\frac{4}{h^2}(1-\xi+\eta+\gamma) - \kappa^2\right) \\ b_4 = (1-\xi)\eta\gamma \quad c_4 = (1-\xi)\eta\gamma\left(\frac{4}{h^2}(2+\xi-\eta-\gamma) - \kappa^2\right) \\ b_5 = (1-\xi)\eta(1-\gamma) \quad c_5 = (1-\xi)\eta(1-\gamma)\left(\frac{4}{h^2}(1+\xi-\eta+\gamma) - \kappa^2\right) \\ b_6 = \xi(1-\eta)\gamma \quad c_6 = \xi(1-\eta)\gamma\left(\frac{4}{h^2}(2-\xi+\eta-\gamma) - \kappa^2\right) \\ b_7 = (1-\xi)(1-\eta)\gamma \quad c_7 = (1-\xi)(1-\eta)\gamma\left(\frac{4}{h^2}(1+\xi+\eta-\gamma) - \kappa^2\right) \\ b_8 = \xi\eta(1-\gamma) \quad c_8 = \xi\eta(1-\gamma)\left(\frac{4}{h^2}(2-\xi-\eta+\gamma) - \kappa^2\right). \end{cases}$$

Proof. Gathering the results of Lemma 1, 2, 3 and 4 respectively, we first have, for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$,

$$\begin{aligned} L_h : E_h^{l_1,l_2,l_3} &\longrightarrow E_h^{l_1,l_2,l_3}, \\ L_{2h} : span[\varphi_{2h}^{l_1,l_2,l_3}] &\longrightarrow span[\varphi_{2h}^{l_1,l_2,l_3}], \\ I_h^{2h} : E_h^{l_1,l_2,l_3} &\longrightarrow span[\varphi_{2h}^{l_1,l_2,l_3}], \\ I_{2h}^{h} : span[\varphi_{2h}^{l_1,l_2,l_3}] &\longrightarrow E_h^{l_1,l_2,l_3}. \end{aligned}$$

Thus, it follows that

$$K_h^{2h} : E_h^{l_1,l_2,l_3} \longrightarrow E_h^{l_1,l_2,l_3}.$$

Furthermore, combining the representation of L_h, L_{2h}, I_h^{2h} and I_{2h}^h with respect to $E_h^{l_1,l_2,l_3}$, we obtain

$$\left[\widehat{L}_{2h}^{-1}\widehat{I}_h^{2h}\widehat{L}_h\right](l_1,l_2,l_3) = \frac{1}{\Lambda}[c_i]_{i=1...8} \text{ for } l_1,l_2,l_3 = 1,...,\frac{n}{2}-1,$$

and thus, it follows that

$$\left[I_{2h}^h\widehat{L}_{2h}^{-1}\widehat{I}_h^{2h}\widehat{L}_h\right](l_1,l_2,l_3) = \frac{1}{\Lambda}[b_i c_j]_{i,j=1...8} \text{ for } l_1,l_2,l_3 = 1,...,\frac{n}{2}-1.$$

If $l_1 = \frac{n}{2}$ or $l_2 = \frac{n}{2}$ or $l_3 = \frac{n}{2}$, as $\widehat{I}_h^{2h}(l_1,l_2,l_3) = 0$, K_h^{2h} is then reduced to the identity matrix with a dimension corresponding to the dimension of $E_h^{l_1,l_2,l_3}$.

□

The representation of a Jacobi smoother J_h in the Fourier basis is now introduced.

Lemma 5. *The harmonic spaces $E_h^{l_1,l_2,l_3}$ for $l_1, l_2, l_3 = 1, ..., \frac{n}{2} - 1$ are invariant under the Jacobi smoother J_h with damping parameter ω_r (Example 1) for the Helmholtz operator L_h:*

$$J_h : E_h^{l_1,l_2,l_3} \longrightarrow E_h^{l_1,l_2,l_3}, \text{ for } l_1,l_2,l_3 = 1,...,\frac{n}{2}$$

The operator J_h can be represented in the Fourier basis as a diagonal matrix. Its representation with respect to the spaces $E_h^{l_1,l_2,l_3}$ consists in diagonal blocks as described below, for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$:

$$\widehat{J_h}(l_1,l_2,l_3) = 1 - \left(\frac{\omega_r}{\frac{6}{h^2} - \kappa^2}\right)\widehat{L_h}(l_1,l_2,l_3).$$

Proof. We first recall the expression of a Jacobi iteration matrix with a relaxation parameter ω_r:

$$J_h = I_h - \left(\frac{\omega_r}{\frac{6}{h^2} - \kappa^2}\right)L_h.$$

The range $J_h(E_h^{l_1,l_2,l_3})$ is then $E_h^{l_1,l_2,l_3}$ since $E_h^{l_1,l_2,l_3}$ is invariant under L_h. Besides, the representation of J_h in the Fourier basis is, for $l_1, l_2, l_3 = 1, ..., \frac{n}{2}$,

$$\widehat{J_h}(l_1,l_2,l_3) = 1 - \left(\frac{\omega_r}{\frac{6}{h^2} - \kappa^2}\right)\widehat{L_h}(l_1,l_2,l_3).$$

□

We now give the representation of M_h (Equation 3.2) with respect to the harmonic spaces $E_h^{l_1,l_2,l_3}$ assuming that the smoother leaves the spaces of harmonics $E_h^{l_1,l_2,l_3}$ invariant.

Corollary 5. *Considering the three-dimensional fine and coarse grid Helmholtz operators (L_h, L_{2h}), a trilinear interpolation I_{2h}^h, its adjoint as restriction I_h^{2h}, a smoother S_h which leaves $E_h^{l_1,l_2,l_3}$ invariant, the spaces $E_h^{l_1,l_2,l_3}$ are invariant under M_h. This last operator has the following representation in the Fourier basis:*

$$\widehat{M}_h(l_1,l_2,l_3) \quad = \quad [\widehat{S}_h^{\nu_2}(l_1,l_2,l_3)\widehat{K}_h^{2h}(l_1,l_2,l_3)\widehat{S}_h^{\nu_1}(l_1,l_2,l_3)]_{l_1,l_2,l_3=1,\ldots,n/2}.$$

where $\widehat{K}_h^{2h}$ is given in Theorem 1.

Proof. This is a direct consequence of Theorem 1. □

Therefore, the norm of M_h ($\|M_h\|_2$) can be computed thanks to Corollary 5:

$$\|M_h\|_2 \quad = \quad \max\left\{\|\widehat{M}_h(l_1,l_2,l_3)\|_2 \mid 1 \leq \max(l_1,l_2,l_3) \leq \frac{n}{2}\right\}. \tag{3.6}$$

The quantity $\|M_h\|_2$ is obtained by computing numerically $\|\widehat{M}_h(l_1,l_2,l_3)\|_2$ for all $l_1,l_2,l_3 = 1,...,n/2$.

In certain situations, a two-grid cycle can be used as a preconditioner of a Krylov method. Indeed in our application it is found that the two-grid method is not convergent for Helmholtz problems at high wavenumbers. It can be used as a preconditioner of a Krylov method [31]. As said in Chapter 2, the distribution of a preconditioned operator spectrum in the complex plane can influence the convergence of a Krylov method. Moreover, in the symmetric case, the spectrum governs their convergence [101]. The RFA enables to obtain the spectrum of this preconditioned operator [125]. We are then performing this spectrum study in the RFA framework using preconditioning.

As said in Section 2.2.1, a preconditioning matrix M must approximate the inverse of the linear system matrix A. We focus on the case where A is the original Helmholtz matrix $L_h^{(0)} = L_h$ for $\kappa = k \in \mathbb{R}$ (see Example 1) and M the two-grid iteration matrix M_h applied to a possibly shifted Helmholtz operator $L_h^{(\beta)} = L_h$ for $\kappa^2 = (1 - i\beta)k^2$, where β denotes the shift parameter lying in $[0, 1]$. It corresponds to the preconditioners depicted in [31] and [42] in the two-dimensional case. Each preconditioning step requires the solution of the linear system $L_h^{(\beta)} z_h = v_h$. One cycle of a geometric two-grid method is used to approximate the inverse of $L_h^{(\beta)}$. Let $\mathfrak{V}_h^{-1}(\beta)$ denote this approximation. The convergence of the Krylov subspace method is thus related to the spectrum of the matrix $L_h^{(0)}\,\mathfrak{V}_h^{-1}(\beta)$. If only one cycle is performed, the iteration matrix of the preconditioning phase is equal to the iteration matrix of the multigrid procedure, that is:

$$M_h = (I_h - \mathfrak{V}_h^{-1}(\beta)\,L_h^{(\beta)}) \quad \text{or} \quad \mathfrak{V}_h^{-1}(\beta)\,L_h^{(\beta)} = I_h - M_h \tag{3.7}$$

where M_h is the two-grid iteration matrix (see Equation (3.2)). From Equation (3.7) the following relation can be deduced:

$$L_h^{(0)}\,\mathfrak{V}_h^{-1}(\beta) = L_h^{(0)}\,(I_h - M_h)\,(L_h^{(\beta)})^{-1}\,. \tag{3.8}$$

Since all operators in Equation (3.8) are diagonalizable in the Fourier basis (Corollary 5), the spectrum of $L_h^{(0)}\,\mathfrak{V}_h^{-1}$ can be computed solving eigenvalue problems of small size (8×8 at most) only.

Therefore, we compute thanks to RFA the spectrum of $L_h^{(0)}\,\mathfrak{V}_h^{-1}(\beta)$ for two values of β considering the same two-grid method as in Corollary 5 and two Jacobi iteration ($J_h^2(\omega_r)$) as a smoother ($\nu_1 = \nu_2 = 2$).

In Figure 3.6, the spectra of $L_h^{(0)}\,\mathfrak{V}_h^{-1}(0)$ and $L_h^{(0)}\,\mathfrak{V}_h^{-1}(0.6)$ are plotted considering a 64^3 grid for a wavenumber $k = \pi/(6h)$ and relaxation parameters $\omega_r = 0.8$ and $\omega_r = 0.3$ respectively. The choice of the parameters β and ω_r is discussed in Section 3.3.2.

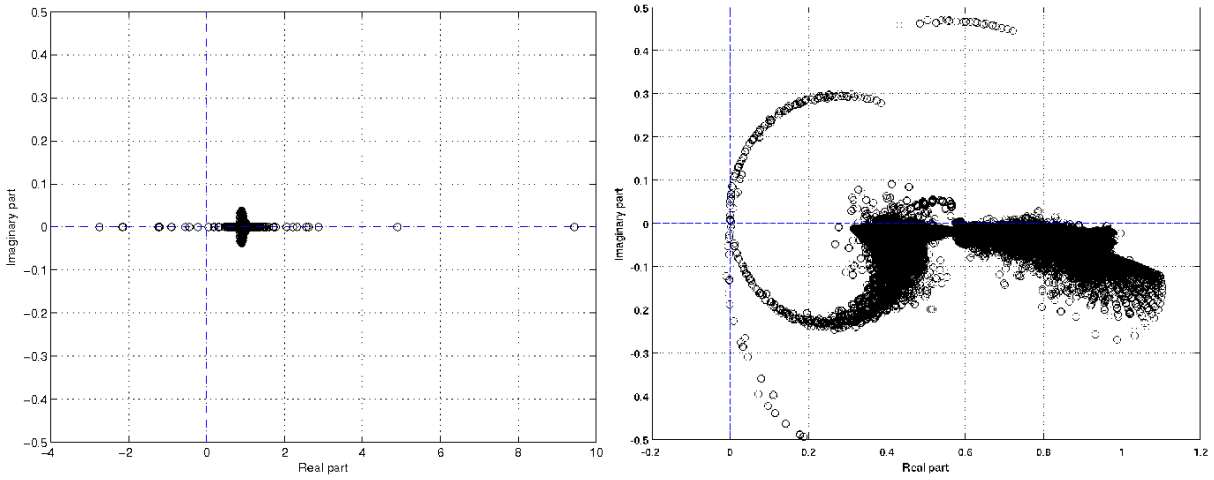

Figure 3.6: Spectra of $L_h^{(0)}\,\mathcal{U}_h^{-1}(\beta)$ for two values of β, ($\beta = 0$, $\omega_r = 0.8$) (left) and ($\beta = 0.6$, $\omega_r = 0.3$) (right), considering a 64^3 grid for a wavenumber $k = \pi/(6h)$.

Both spectra plotted in Figure 3.6 look favorable for the convergence of a Krylov method. Indeed, on one hand, using the two-grid method on the original Helmholtz operator gives a spectrum with a cluster around one with few isolated eigenvalues with positive or negative real parts. On the other hand, when the two-grid method is applied on the shifted Helmholtz operator ($\beta = 0.6$), the spectrum is lying in the positive real part of the complex plane with few eigenvalues close to zero. Moreover, it has to be noticed that the shapes of the spectra are similar as in the two-dimensional case; see Figure 1 in [31] for the original Helmholtz operator and Figure 7 in [42] for the shifted Helmholtz operator.

Nevertheless the lack of generality of the Rigorous Fourier Analysis (RFA) concerning the assumptions on the smoother and on the boundary conditions leads us to investigate the Local Fourier Analysis (LFA). Indeed, LFA enables to analyze general smoothers (see [115, Table 4.4]). Furthermore it does not take into account boundary conditions because it can linearize locally any discrete operator with a constant stencil. We introduce some elements of LFA in Section 3.3.2 before presenting a smoothing analysis for the three-dimensional Helmholtz operator. In fact, this LFA introduction can be seen as an extension to the three-dimensional case of Sections 4.2 and 4.3 in [115].

3.3.2 Local Fourier analysis (LFA) of a two-grid method

We first introduce three-dimensional Helmholtz type operators with periodic boundary conditions in $\Omega_h = G_h \cap [0,1]^3$:

$$\begin{cases} -\Delta u - \kappa^2 u \text{ in } (0,1)^3, \\ u(0,y,z) = u(1,y,z),\ (y,z) \in [0,1]^2, \\ u(x,0,z) = u(x,1,z),\ (x,z) \in [0,1]^2, \\ u(x,y,0) = u(x,y,1),\ (x,y) \in [0,1]^2. \end{cases}$$

We then introduce the eigenfunctions of this operator:

$$\varphi_h^{l_1,l_2,l_3}(x,y,z) = e^{2i\pi l_1 x}\, e^{2i\pi l_2 y}\, e^{2i\pi l_3 z}, \text{ for } -n/2 \le l_1, l_2, l_3 < n/2 \text{ and } (x,y,z) \in \Omega_h.$$

The LFA relies on these grid functions, however instead of the discrete space Θ_h,

$$\Theta_h = \{(2\pi l_1 h, 2\pi l_2 h, 2\pi l_3 h) |\ -n/2 \le l_1, l_2, l_3 < n/2\} \subset [-\pi,\pi)^3,$$

the continuous space $[-\pi,\pi)^3$ is now considered. The LFA then uses the following grid functions:

$$\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z) = e^{i\theta_1 x/h}\, e^{i\theta_2 y/h}\, e^{i\theta_3 z/h}, \text{ for } (\theta_1,\theta_2,\theta_3) \in [-\pi,\pi)^3 \text{ and } (x,y,z) \in G_h.$$

The grid functions $\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z)$ are linearly independent for any $(\theta_1,\theta_2,\theta_3) \in [-\pi,\pi)^3$. Thus, they form a basis of G_h, called once again a Fourier basis.

The infinite grid G_h is considered here in order to obtain a representation of any linear operator with a constant stencil (a stencil which does not depend on (x, y, z)) in this Fourier basis [115, Lemma 4.2.1]. For instance, if the stencil of the three-dimensional Helmholtz operator (Example 1) is considered, the following relation holds

$$L_h \varphi_h^{\theta_1,\theta_2,\theta_3}(x, y, z) = \breve{L}_h(\theta_1, \theta_2, \theta_3)\varphi_h^{\theta_1,\theta_2,\theta_3}(x, y, z)$$

$$\text{where } \breve{L}_h(\theta_1, \theta_2, \theta_3) = \frac{1}{h^2}(6 - e^{i\theta_1} - e^{-i\theta_1} - e^{i\theta_2} - e^{-i\theta_2} - e^{i\theta_3} - e^{-i\theta_3}) - \kappa^2.$$

$\breve{L}_h(\theta_1, \theta_2, \theta_3)$ is named the representation of L_h in the Fourier basis.

Similarly as in Section 3.3.1 (RFA), we assume that the coarse grid is obtained by standard geometric coarsening. This assumption implies that, on the infinite coarse grid G_{2h}, for each $(\theta_1, \theta_2, \theta_3) \in [-\pi/2, \pi/2)^3$, there are seven other values of $(\theta_1^{(j_1)}, \theta_2^{(j_2)}, \theta_3^{(j_3)}) \in [-\pi, \pi)^3$, with $(j_1, j_2, j_3) \in \{0, 1\}^3 \setminus \{(0, 0, 0)\}$, such that

$$\varphi_{2h}^{\theta_1,\theta_2,\theta_3}(x, y, z) = \varphi_{2h}^{\theta_1^{(j_1)},\theta_2^{(j_2)},\theta_3^{(j_3)}}(x, y, z), \text{ for } (x, y, z) \in G_{2h} \text{ and } (j_1, j_2, j_3) \in \{0, 1\}^3 \setminus \{(0, 0, 0)\},$$

where, for $i = 1, 2, 3$, $\theta_i^{(0)}$ and $\theta_i^{(1)}$ are defined as:

$$\begin{aligned} \theta_i^{(0)} &:= \theta_i \\ \theta_i^{(1)} &:= \begin{cases} \theta_i + \pi \text{ if } \theta_i < 0, \\ \theta_i - \pi \text{ if } \theta_i \geq 0. \end{cases} \end{aligned}$$

Thus, only the frequency components $\varphi_{2h}^{\theta_1,\theta_2,\theta_3}$ for $(\theta_1, \theta_2, \theta_3) \in [-\pi/2, \pi/2)^3$ are visible on G_{2h}. This leads us to define low and high frequencies components of $\varphi_h^{\theta_1,\theta_2,\theta_3}$.

Definition 3. *Low and high frequencies components of* $\varphi_h^{\theta_1,\theta_2,\theta_3}$, $(\theta_1, \theta_2, \theta_3) \in [-\pi, \pi)^3$*:*

- $\varphi_h^{\theta_1,\theta_2,\theta_3}$ *is a low frequency component* $\Leftrightarrow (\theta_1, \theta_2, \theta_3) \in \Theta^{low} := [-\pi/2, \pi/2)^3$.
- $\varphi_h^{\theta_1,\theta_2,\theta_3}$ *is a high frequency component* $\Leftrightarrow (\theta_1, \theta_2, \theta_3) \in \Theta^{high} := [-\pi, \pi)^3 \setminus [-\pi/2, \pi/2)^3$.

The coarse level is then only dealing with low frequencies. High frequencies are then only managed on the fine level. In a two-grid algorithm, this means that high frequency components of the error will be managed by smoothing, whereas low frequencies by the coarse grid correction operator. In the next section, we focus on the computation of the smoothing factor.

Smoothing analysis

As said, in the LFA framework it is possible to analyze the smoothing behavior of Gauss-Seidel with lexicographic ordering (Gauss-Seidel-lex). This was not possible in the RFA framework. This is of great interest since Gauss-Seidel-lex is a classical relaxation method that will be used hereafter.

In order to use LFA to analyze the properties of a given smoother, we have to assume that the relaxation method satisfies the following splitting:

$$L_h^+ u_h^{m+1} + L_h^- u_h^m = b_h \text{ with } L_h^+ + L_h^- = L_h \tag{3.9}$$

where, denoting by u_h^m is the previous approximation of u_h (before the smoothing step) and u_h^{m+1} the new approximation of u_h (after the smoothing step).

Remark 7. *Considering the same notations as in Example 1,* $L_h = D - E - F$*, the expressions of* L_h^+ *and* L_h^- *for Jacobi, forward Gauss-Seidel-lex, backward Gauss-Seidel-lex are:*

- *Jacobi:* $\dfrac{D}{\omega_r} u_h^{m+1} + \left(-\dfrac{D}{\omega_r} + L_h\right) u_h^m = b_h$

$$L_h^+ = \frac{D}{\omega_r} \text{ and } L_h^- = -\frac{D}{\omega_r} + L_h.$$

- *forward Gauss-Seidel-lex:* $(-E+D)u_h^{m+1} - Fu_h^m = b_h$

$$L_h^+ = D - E \text{ and } L_h^- = -F.$$

- *backward Gauss-Seidel-lex:* $(-F+D)u_h^{m+1} - Eu_h^m = b_h$

$$L_h^+ = D - F \text{ and } L_h^- = -E.$$

We now define the errors $e_h^{m+1} = u_h - u_h^{m+1}$ and $e_h^m = u_h - u_h^m$ at iterations $(m+1)$ and m respectively denoting by u_h the discrete solution verifying $L_h u_h = b_h$. It follows that:

$$L_h^+ e_h^{m+1} + L_h^- e_h^m = 0.$$

This means that $e_h^{m+1} = S_h e_h^m$ where S_h denotes the smoothing operator (see [115, Lemma 4.3.1]). Thus, since we can write any linear operator with a constant stencil on the Fourier basis $\left\{\varphi_h^{\theta_1,\theta_2,\theta_3}, \text{ for } (\theta_1,\theta_2,\theta_3) \in [-\pi,\pi)^3\right\}$, we use the Fourier representations of L_h^+ and L_h^- to obtain the Fourier representation of the smoothing operator S_h. Indeed, Fourier representations $\breve{L}_h^+(\theta_1,\theta_2,\theta_3)$ and $\breve{L}_h^-(\theta_1,\theta_2,\theta_3)$ can be easily deduced, applying L_h^+ and L_h^- to the basis functions $\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z)$.

Therefore, if a smoother can be expressed as in Equation (3.9), the Fourier representation of the smoothing operator $\breve{S}_h(\theta_1,\theta_2,\theta_3)$ is (assuming that $\breve{L}_h^+(\theta_1,\theta_2,\theta_3) \neq 0 \mid \forall(\theta_1,\theta_2,\theta_3) \in [-\pi,\pi)^3$)

$$\breve{S}_h(\theta_1,\theta_2,\theta_3) = -\frac{\breve{L}_h^-(\theta_1,\theta_2,\theta_3)}{\breve{L}_h^+(\theta_1,\theta_2,\theta_3)}.$$

We now give the representation of a Jacobi iteration in the Fourier basis.

Example 2. *For the Jacobi iteration ($Jac(\omega_r)$), we have*

$$\begin{aligned}
L_h^+\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z) &= \frac{1}{h^2}\frac{6-(h\kappa)^2}{\omega_r}\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z),\\
L_h^-\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z) &= \frac{1}{h^2}\left(\frac{(6-(h\kappa)^2)(\omega_r-1)}{\omega_r} - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3} - e^{i\theta_1} - e^{i\theta_2} - e^{i\theta_3}\right)\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z).
\end{aligned}$$

Therefore, the Fourier representations $\breve{L}_h^+$, $\breve{L}_h^-$ and $\breve{S}_h$ are

$$\begin{aligned}
\breve{L}_h^+(\theta_1,\theta_2,\theta_3) &= \frac{1}{h^2}\frac{6-(h\kappa)^2}{\omega_r},\\
\breve{L}_h^-(\theta_1,\theta_2,\theta_3) &= \frac{1}{h^2}\left(\frac{(6-(h\kappa)^2)(\omega_r-1)}{\omega_r} - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3} - e^{i\theta_1} - e^{i\theta_2} - e^{i\theta_3}\right),\\
\breve{S}_h^{(Jac(\omega_r))}(\theta_1,\theta_2,\theta_3) &= 1 - \frac{\omega_r}{6-(h\kappa)^2}\left(6-(h\kappa)^2 - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3} - e^{i\theta_1} - e^{i\theta_2} - e^{i\theta_3}\right).
\end{aligned}$$

We now give the representation of a forward and backward Gauss-Seidel-lex iteration in the Fourier basis.

Example 3. *The forward Gauss-Seidel-lex iteration (GS-forw), reads*

$$\begin{aligned}
L_h^+\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z) &= \left(\frac{1}{h^2}(6 - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3}) - \kappa^2\right)\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z),\\
L_h^-\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z) &= \left(-\frac{1}{h^2}(e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3})\right)\varphi_h^{\theta_1,\theta_2,\theta_3}(x,y,z).
\end{aligned}$$

Therefore, the Fourier representations $\breve{L}_h^+$, $\breve{L}_h^-$ and $\breve{S}_h$ are

$$\begin{aligned}
\breve{L}_h^+(\theta_1,\theta_2,\theta_3) &= \frac{1}{h^2}(6 - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3}) - \kappa^2,\\
\breve{L}_h^-(\theta_1,\theta_2,\theta_3) &= -\frac{1}{h^2}(e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3}),\\
\breve{S}_h^{(GS-forw)}(\theta_1,\theta_2,\theta_3) &= \frac{e^{i\theta_1} + e^{i\theta_2} + e^{i\theta_3}}{6-(h\kappa)^2 - e^{-i\theta_1} - e^{-i\theta_2} - e^{-i\theta_3}}.
\end{aligned}$$

Remark 8. *We then deduce the Fourier representation of a backward Gauss-Seidel-lex iteration (GS-back):*

$$\begin{aligned}
\check{L}_h^+(\theta_1,\theta_2,\theta_3) &= \frac{1}{h^2}(6-e^{i\theta_1}-e^{i\theta_2}-e^{i\theta_3})-\kappa^2,\\
\check{L}_h^-(\theta_1,\theta_2,\theta_3) &= -\frac{1}{h^2}(e^{-i\theta_1}+e^{-i\theta_2}+e^{-i\theta_3}),\\
\check{S}_h^{(GS-back)}(\theta_1,\theta_2,\theta_3) &= \frac{e^{-i\theta_1}+e^{-i\theta_2}+e^{-i\theta_3}}{6-(h\kappa)^2-e^{i\theta_1}-e^{i\theta_2}-e^{i\theta_3}}.
\end{aligned}$$

A symmetric Gauss-Seidel-lex (GS-sym) iteration consists in one iteration of forward Gauss-Seidel-lex followed by one iteration of backward Gauss-Seidel-lex:

$$S_h^{(GS-sym)} = S_h^{(GS-back)} S_h^{(GS-forw)}.$$

Its Fourier representation can be deduced as:

$$\check{S}_h^{(GS-sym)}(\theta_1,\theta_2,\theta_3) = \check{S}_h^{(GS-back)}(\theta_1,\theta_2,\theta_3)\check{S}_h^{(GS-forw)}(\theta_1,\theta_2,\theta_3).$$

This Fourier representation enables us to define the smoothing factor $\mu_{loc}(S_h)$. It is the supremum of the absolute value of the smoother components in the Fourier basis $\check{S}_h(\theta_1,\theta_2,\theta_3)$ for $(\theta_1,\theta_2,\theta_3)$ in the space of high frequencies Θ_h^{high} (see Definition 3 and [115, Figure 4.1]).

Definition 4. *Smoothing factor* $\mu_{loc}(S_h)$

$$\mu_{loc}(S_h) = \sup_{(\theta_1,\theta_2,\theta_3)\in\Theta_h^{high}} |\check{S}_h(\theta_1,\theta_2,\theta_3)|.$$

Thus, $\mu_{loc}(S_h)$ can be obtained by solving a maximization problem on $(\theta_1,\theta_2,\theta_3)$.

Remark 9. *Considering Definition 4, when the original Helmholtz operator is considered, $\kappa = k \in \mathbb{R}$, the smoothing factor of a symmetric Gauss-Seidel-lex (GS-sym) iteration is equal to the smoothing factor of two iterations of Gauss-Seidel-lex (see Remark 8). Indeed, we have*

$$\check{S}_h^{(GS-sym)}(\theta_1,\theta_2,\theta_3) = |\check{S}_h^{(GS-sym)}(\theta_1,\theta_2,\theta_3)| = |\check{S}_h^{(GS-forw)}(\theta_1,\theta_2,\theta_3)|^2.$$

However, the smoothing behavior of Gauss-Seidel-lex and symmetric Gauss-Seidel-lex can be different on the original Helmholtz operator with PML since it is non-symmetric. This will be noticed in Section 3.4.2.

In Tables 3.1 and 3.2 we present smoothing factors μ_{loc} of the Jacobi smoother $S_h^{(Jac(\omega_r))}$ for the 3D Helmholtz operator and the shifted 3D Helmholtz operator respectively, considering wavenumbers k such that they verify the stability condition $kh = \frac{\pi}{6}$ on the fine level (see Relation A.4 in Appendix A). The smoothing factors $\mu_{loc}((S_h^{(Jac(\omega_r))})^\nu)$ are given on four grids of the multigrid hierarchy and two numbers of iterations: $\nu = 1, 2$ respectively.

The shift parameter β ($\kappa^2 = (1-\beta i)k^2$) and the relaxation parameter ω_r are chosen such that the smoothing factor is smaller than one on the third grid $(1/4h)^3$ and β as small as possible. Extensive computations of the smoothing factor of the shifted operator led us to the following combination of values:

$$\begin{cases} \omega_r = 0.3, \\ 1-\beta i = 1-0.6i. \end{cases}$$

Fine grid $((1/h)^3)$	64^3		128^3		256^3		512^3	
Grid	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$
$(1/h)^3$	0.90	0.82	0.91	0.82	0.91	0.82	0.91	0.82
$(1/2h)^3$	0.93	0.86	0.93	0.87	0.94	0.88	0.94	0.88
$(1/4h)^3$	0.93	0.86	0.95	0.89	0.95	0.91	0.96	0.92
$(1/8h)^3$	0.78	0.61	0.79	0.62	0.79	0.62	0.79	0.62

Table 3.1: Smoothing factors $\mu_{loc}((S_h^{(Jac(\omega_r))})^\nu)$ of the Jacobi smoother $S_h^{(Jac(\omega_r))}$, $\omega_r = 0.3$ for two values of ν and four grid sizes considering the shifted 3D Helmholtz operator ($\beta = 0.6$) for a wavenumber $k = \frac{\pi}{6h}$.

For the original Helmholtz operator ($\beta = 0$), the smoothing factor on the third grid is always larger than one for any value of ω_r as it has been observed in the two-dimensional case [42]. Then, we choose the relaxation parameter such that the smoothing factor on the fine level is as small as possible. We found $\omega_r = 0.8$ numerically.

Fine grid $((1/h)^3)$	64^3		128^3		256^3		512^3	
Grid	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$
$(1/h)^3$	0.74	0.55	0.75	0.56	0.76	0.58	0.76	0.58
$(1/2h)^3$	0.82	0.67	0.84	0.70	0.84	0.71	0.84	0.71
$(1/4h)^3$	(2.72)	(7.39)	(2.76)	(7.62)	(2.77)	(7.68)	(2.77)	(7.70)
$(1/8h)^3$	0.58	0.34	0.61	0.37	0.61	0.38	0.62	0.39

Table 3.2: Smoothing factors $\mu_{loc}((S_h^{(Jac(\omega_r))})^\nu)$ of the Jacobi smoother $S_h^{(Jac(\omega_r))}$, $\omega_r = 0.8$ for two values of ν and four grid sizes considering the original 3D Helmholtz operator ($\beta = 0$) for a wavenumber $k = \frac{\pi}{6h}$. Smoothing factors larger than one are indicated in brackets.

We remark in Tables 3.1 and 3.2 that the smoothing factors are similar on a given level of the hierarchy, when the ratio between the wavenumber k and the mesh grid size h is kept constant. Then, as in the two-dimensional case [42], the Jacobi method is found efficient to smooth high frequencies for the shifted Helmholtz operator on each grid of the multigrid hierarchy, whereas it is not possible to obtain a smoothing factor smaller than one on the third grid for the original Helmholtz operator. Nevertheless, it has to be noticed that the smoothing factors in Table 3.1 are obtained for a larger shift parameter ($\beta = 0.6$) than in the two-dimensional case ($\beta = 0.5$). Furthermore, these smoothing factors are higher than in the two-dimensional case (0.81 in two dimensions and 0.87 in three dimensions for 2 Jacobi iterations). Consequently we deduce that a multigrid cycle on the three-dimensional shifted Helmholtz operator with a Jacobi smoother could not precondition the original Helmholtz operator as efficiently as in the two-dimensional case [42, 43]. In [40, 96], the authors advise to use a *plane* smoother [88] in combination with semi-coarsening [124] to work towards this issue. In the next tables, we show that improved smoothing factors can be obtained for three-dimensional Helmholtz problems at least on the two finest grids.

In Tables 3.3 and 3.4, we present smoothing factors μ_{loc} of the Gauss-Seidel-lex smoother $S_h^{(GS-forw)}$ for the shifted 3D Helmholtz operator and the 3D Helmholtz operator respectively, considering wavenumbers k such that $kh = \frac{\pi}{6}$ (see Relation A.4 in Appendix A). The smoothing factors $\mu_{loc}((S_h^{(GS-forw)})^\nu)$ are given on four grids of the multigrid hierarchy and for two numbers of iterations: $\nu = 1$ and $\nu = 2$ as previously.

For this smoother, the shift parameter does not enable to obtain a smoothing factor smaller than one on the third grid $(1/4h)^3$. The shift parameter β is then once again taken equal to 0.6.

Fine grid $((1/h)^3)$	64^3		128^3		256^3		512^3	
Grid	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$
$(1/h)^3$	0.59	0.36	0.60	0.36	0.61	0.37	0.61	0.37
$(1/2h)^3$	0.75	0.56	0.77	0.60	0.78	0.61	0.79	0.62
$(1/4h)^3$	(5.84)	(34.12)	(7.39)	(54.66)	(8.37)	(70.13)	(8.91)	(79.47)
$(1/8h)^3$	0.24	0.06	0.24	0.06	0.24	0.06	0.24	0.06

Table 3.3: Smoothing factors $\mu_{loc}((S_h^{(GS-forw)})^\nu)$ of the Gauss-Seidel-lex smoother $S_h^{(GS-forw)}$ for two values of ν and four grid sizes considering the shifted 3D Helmholtz operator ($\beta = 0.6$) for a wavenumber $k = \frac{\pi}{6h}$. Smoothing factors larger than one are indicated in brackets.

Fine grid $((1/h)^3)$	64^3		128^3		256^3		512^3	
Grid	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$
$(1/h)^3$	0.58	0.34	0.59	0.35	0.60	0.36	0.60	0.36
$(1/2h)^3$	0.70	0.48	0.72	0.51	0.73	0.53	0.73	0.54
$(1/4h)^3$	(23)	(525)	(61)	(3687)	(794)	(630998)	(794)	(630998)
$(1/8h)^3$	0.34	0.11	0.35	0.12	0.35	0.12	0.35	0.12

Table 3.4: Smoothing factors $\mu_{loc}((S_h^{(GS-forw)})^\nu)$ of the Gauss-Seidel-lex smoother $S_h^{(GS-forw)}$ for two values of ν and four grid sizes considering the original 3D Helmholtz operator ($\beta = 0$) for a wavenumber $k = \dfrac{\pi}{6h}$. Smoothing factors larger than one are indicated in brackets.

Tables 3.3 and 3.4 point out that the lexicographical Gauss-Seidel method is more efficient than the Jacobi method to smooth the high frequency components on the finer grids for both 3D Helmholtz operators (for both $\beta = 0$ and $\beta = 0.6$). However, after extensive experiments we can conclude that this smoother cannot succeed in smoothing on the third grid for any shift parameter.

Therefore, when a shifted Helmholtz operator is considered, a three-dimensional geometric multigrid could be improved by considering Gauss-Seidel on the finer grids ($(1/h)^3$,$(1/2h)^3$) and Jacobi on the coarse ones. Nevertheless, to obtain an efficient coarse Jacobi smoother requires a large shift ($\beta = 0.6$). This can imply a loss of efficiency of a multigrid iteration. Furthermore, as in the two-dimensional case, the relaxation parameter or the shift parameter have to be changed for heterogeneous Helmholtz problems to obtain an efficient multi-level preconditioner. Finally we note that the boundary conditions (PML) can also influence the determination of the shift parameter. A numerical illustration is given in the next section where we analyze spectra and histories of convergence considering the one-dimensional Helmholtz operator with absorbing boundary conditions preconditioned by a two-level method. These absorbing boundary conditions are formulated with a Perfectly Matched Layer (PML) [11].

One-dimensional Helmholtz operator with PML

Regarding the formulation and discretization of the Helmholtz operator, we refer to Section A.2 in Appendix A. The use of a PML formulation implies variable coefficients in the Helmholtz operator $A_h(x, y, z)$, $(x, y, z) \in \Omega_h$. A "frozen" analysis [61, 104] could then be performed to deduce an upper bound of the convergence factor in the Fourier analysis framework. Indeed, since the operator $A_h(x, y, z)$ has variable coefficients, the coefficients of its representation in the Fourier basis $\breve{A}_h(x, y, z)$ depend on the coordinates $(x, y, z) \in \Omega_h$. The "frozen" analysis consists in finding an upper bound of the spectral radius of a multigrid iteration matrix $M_h(x, y, z)$ for $A_h(x, y, z)$ as

$$\rho(M_h(x, y, z)) \leq \max_{(x,y,z)\in\Omega_h} \rho(\breve{M}_h(x, y, z)).$$

Yet it cannot be used to deduce the spectrum of the preconditioned operator since it computes spectra of $A_h(x, y, z)$ for each $(x, y, z) \in \Omega_h$. Besides it is not possible to find the analytic expression of the eigenfunctions of the Helmholtz operator with PML.

The only way to analyze the spectrum of the preconditioned operator $A_h^{(0)}\,\tilde{\mathfrak{V}}_h^{-1}(\beta)$ is its explicitly computation, where $A_h^{(0)}$ denotes the original Helmholtz operator with PML and $\tilde{\mathfrak{V}}_h^{-1}(\beta)$ the operator representing the action of a two-level preconditioner applied to the shifted Helmholtz operator with PML $A_h^{(\beta)}$. Obviously the computation of this spectrum is not affordable in three dimensions. Notwithstanding, since the three-dimensional spectra computed with a RFA (Section 3.3.1) where really similar to the spectra obtained in two dimensions [40, 42] (see Figure 3.6), it is expected that to compute the spectrum of $A_h^{(0)}\,\tilde{\mathfrak{V}}_h^{-1}(\beta)$ in one dimension can provide us some information [41]. Thus, considering two Gauss-Seidel iterations as pre- and post-relaxations ($\nu_1 = \nu_2 = 2$), we compute the spectrum of the preconditioned operator $A_h^{(0)}\,\tilde{\mathfrak{V}}_h^{-1}(\beta)$ in one dimension and report the history of convergence of GMRES(5) using the two-grid preconditioner for different shift parameters.

Considering a one dimensional Helmholtz operator with PML ($1/h = 1024$, $k = \frac{\pi}{6h}$, $n_{PML} = 16$), Figure 3.7 shows histories of convergence of GMRES(5) using the two-grid preconditioner for different values of β and Figure 3.8 shows the corresponding spectra of the preconditioned $A_h^{(0)}\,\tilde{\mathfrak{V}}_h^{-1}(\beta)$ ($1/h = 1024$, $k = \frac{\pi}{6h}$).

In Figure 3.8, when a complex shift is used ($\beta \neq 0$), GMRES(5) stagnates. When the preconditioner is related to the original Helmholtz operator ($\beta = 0$), we observe that the iterative procedure converges. This behavior is related to the eigenvalue distribution. Indeed, in Figure 3.8, the spectra have a similar shape for non-zero values of the shift parameter, eigenvalues lie on an ellipse with few outliers, several eigenvalues on the ellipse close to zero have a negative real part. These spectra are then not favorable to the convergence of GMRES. When no shift is used ($\beta = 0$), the spectrum is clustered around one with few isolated eigenvalues in a half plan of the complex plane.

Therefore, when considering the Helmholtz equation with PML, the shift parameter cannot be used in a similar way as with other boundary conditions (Dirichlet, Robin (of first or second order type) [40]), since convergence cannot be achieved. However, if the shift parameter is set to a negative value, GMRES preconditioned by a two-grid method applied on a shifted Helmholtz operator converges (see Figure 3.9). Indeed, GMRES(5) is converging for each value of β. The spectra for $\beta \neq 0$ do not exhibit isolated eigenvalues and are enclosed in the unit circle centered in one (see Figure 3.10). This is due to the formulation of the shifted Helmholtz equation in the PML. Indeed, its one-dimensional formulation is (see Equation A.1):

$$-\frac{1}{1+i\gamma_x(x)}\frac{\partial}{\partial x}\frac{1}{1+i\gamma_x(x)}\frac{\partial}{\partial x}u(x) - (1-i\beta)k^2u(x) = s, \text{ for } x \in (0,1),$$

where γ_x denotes the one-dimensional PML function (see Equation A.2); it is zero outside the PML layer. If γ_x is set to a fixed value in the PML, say ($\gamma_x = 1$), we obtain the following operator in the PML:

$$\frac{1}{2i}(-\Delta u(x) - (2\beta + 2i)k^2u(x)),$$

Thus, if $\beta \geq 0.5$, this operator will be indefinite in the PML layer at high wavenumbers and it is expected that the preconditioner looses its efficiency. A negative shift handles this difficulty. Moreover, it can be observed that without shift, the operator in the PML layer is a Laplace-type operator, this can be beneficial to the convergence of GMRES.

Remark 10. *The results of the smoothing analysis in Section 3.3.2 also hold for the opposite shift parameter ($-\beta$). Indeed, we first remind the expression of the smoothing factor of a forward lexicographical Gauss-Seidel (Example 3),*

$$\mu_{loc}(\check{S}_h^{(GS-forw)}) = \sup_{(\theta_1,\theta_2,\theta_3)\in\Theta_h^{high}} \frac{|e^{i\theta_1}+e^{i\theta_2}+e^{i\theta_3}|}{|6-(1-i\beta)(hk)^2-e^{-i\theta_1}-e^{-i\theta_2}-e^{-i\theta_3}|}.$$

Since $|z| = |\bar{z}|\ \forall z \in \mathbb{C}$, we have

$$\mu_{loc}(\check{S}_h^{(GS-forw)} = \sup_{(\theta_1,\theta_2,\theta_3)\in\Theta_h^{high}} \frac{|e^{-i\theta_1}+e^{-i\theta_2}+e^{-i\theta_3}|}{|6-(1+i\beta)(hk)^2-e^{i\theta_1}-e^{i\theta_2}-e^{i\theta_3}|}.$$

Since the subspace Θ_h^{high} is symmetric with respect to the origin (Definition 3), we can change $(\theta_1,\theta_2,\theta_3)$ to $(-\theta_1,-\theta_2,-\theta_3)$ in the sup. *It follows that*

$$\mu_{loc}(\check{S}_h^{(GS-forw)}) = \sup_{(\theta_1,\theta_2,\theta_3)\in\Theta_h^{high}} \frac{|e^{i\theta_1}+e^{i\theta_2}+e^{i\theta_3}|}{|6-(1+i\beta)(hk)^2-e^{-i\theta_1}-e^{-i\theta_2}-e^{-i\theta_3}|}.$$

Thus, the smoothing factor of forward lexicographical Gauss-Seidel is the same for a positive shift parameter and its opposite. The same proof can be done for a Jacobi iteration. This has to be kept in mind when choosing the right shift parameter in three dimensions.

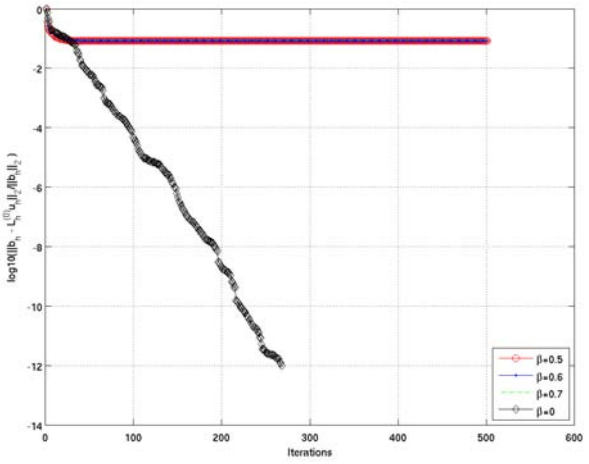

Figure 3.7: History of convergence of GMRES(5) preconditioned by a two-grid cycle using two Gauss-Seidel iterations as pre- and post-relaxations ($\nu_1 = \nu_2 = 2$) to solve a one-dimensional Helmholtz problem with PML ($1/h = 1024$, $k = \frac{\pi}{6h}$) for four values of β (0, 0.5, 0.6, 0.7). Convergence is achieved only in the case $\beta = 0$ here.

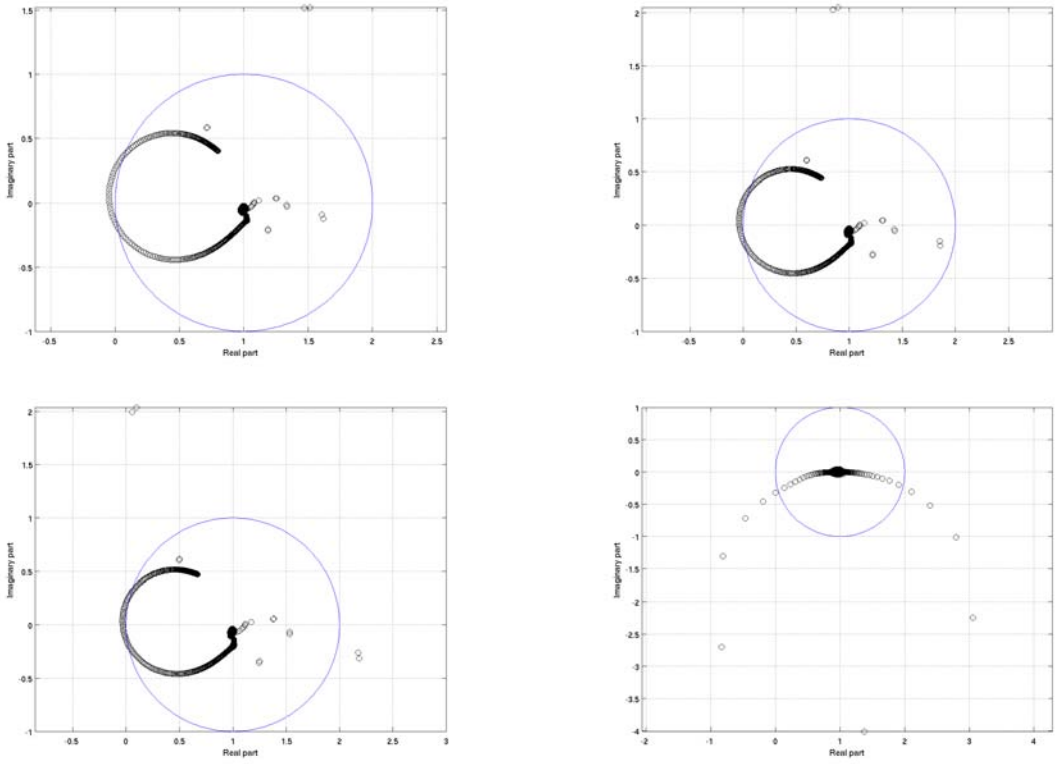

Figure 3.8: Spectra of $A_h^{(0)}\,\tilde{\mathcal{U}}_h^{-1}(\beta)$ ($1/h = 1024$, $k = \frac{\pi}{6h}$) using two Gauss-Seidel iterations as pre- and post-relaxations ($\nu_1 = \nu_2 = 2$) for four values of β, from left to right and from top to bottom, $\beta = 0.5$, $\beta = 0.6$, $\beta = 0.7$ and $\beta = 0$ respectively. The unit circle centered in one (in blue) is used to scale the spectra.

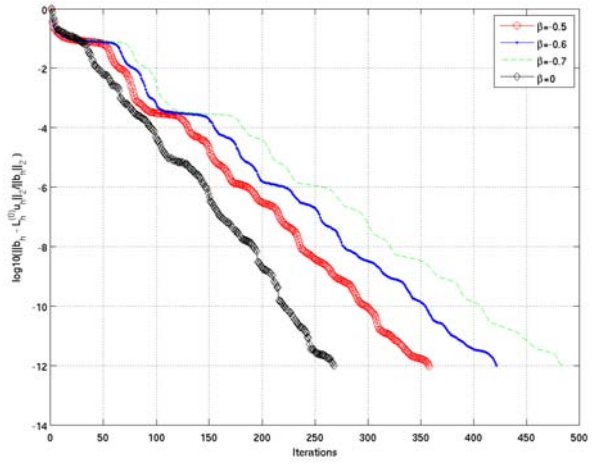

Figure 3.9: History of convergence of GMRES(5) preconditioned by a two-grid cycle using two Gauss-Seidel iterations as pre- and post-relaxations ($\nu_1 = \nu_2 = 2$) to solve a one-dimensional Helmholtz problem with PML ($1/h = 1024$, $k = \frac{\pi}{6h}$) for four values of β (-0.7, -0.6, -0.5, 0).

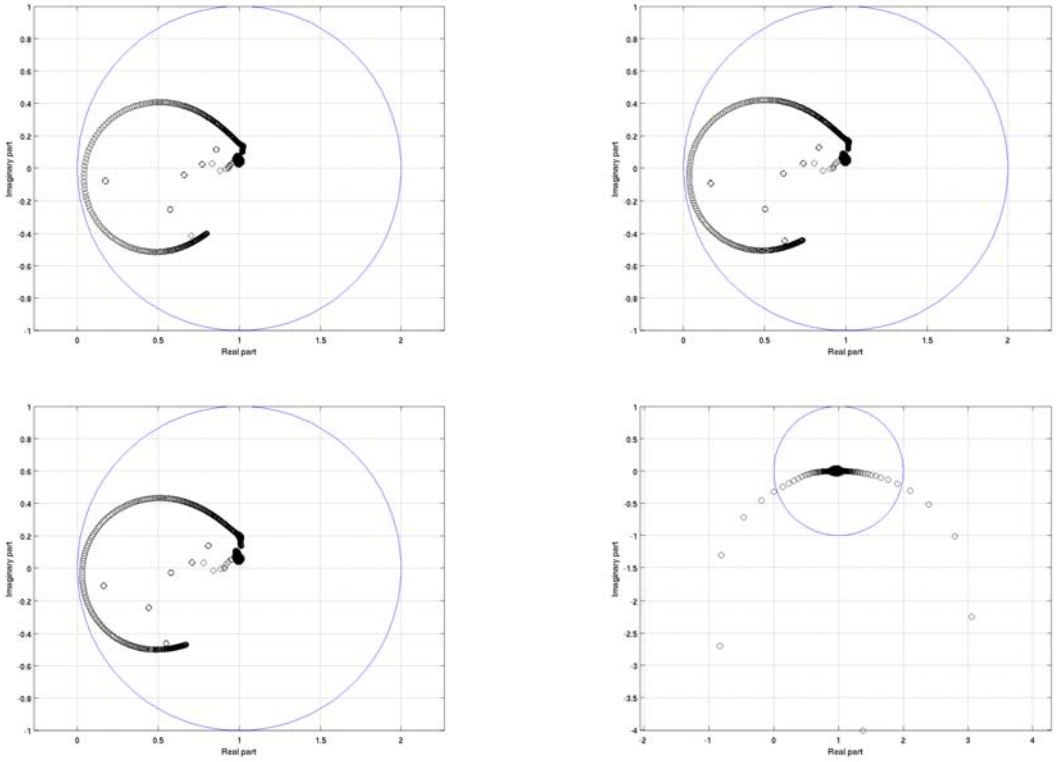

Figure 3.10: Spectra of $A_h^{(0)}\,\tilde{\mho}_h^{-1}(\beta)$ ($1/h = 1024$, $k = \frac{\pi}{6h}$) using two Gauss-Seidel iterations as pre- and post-relaxations ($\nu_1 = \nu_2 = 2$) for four values of β, from left to right and from top to bottom, $\beta = -0.5$, $\beta = -0.6$, $\beta = -0.7$ and $\beta = 0$ respectively. The unit circle centered in one (in blue) is used to scale the spectra.

Therefore, the influence of many parameters makes the choice of the shift β delicate. Indeed, the formulation of the Helmholtz problem, the smoother properties on each grid and the multi-level preconditioner efficiency have to be taken into account to choose the right shift parameter. These dependencies on vari-

ous parameters have led us not to consider a shifted Helmholtz operator. Considering Tables 3.2 and 3.4, we advocate the use of only two grids in the multigrid hierarchy with Gauss-Seidel type smoothers. In two dimensions, a two-level preconditioner on the original Helmholtz operator has proved efficient [31]. However, its efficiency relies on the use of a direct solver (MUMPS [2, 3]) on the coarse level. In three dimensions, even on parallel memory distributed computers, the use of a direct method on the coarse level is prohibitive in terms of computational resources. Indeed, at the beginning of this thesis, the largest three-dimensional case that we could solve in core with MUMPS 4.7.3 [4] was of size 128^3 on 80 cores of an IBM JS21 machine (two GigaBytes per core). Even if this size is already large, it is still too small to solve the three-dimensional Helmholtz equation at large wavenumbers using a direct method on the coarse level of a two-level preconditioner. Thus, a three-dimensional two-level preconditioner necessarily implies to use an iterative method on the coarse level.

We call this scheme a perturbed two-level method. Consequently, a coarse stopping criterion has to be chosen for the coarse iterative solver. In the next section, we will show that a perturbed two-level method is an efficient preconditioner even when a large tolerance on the coarse linear system is chosen.

3.4 A perturbed two-level preconditioner

We focus on the design of a two-level preconditioner for Helmholtz problems with absorbing boundary conditions of PML type [12] at high wavenumbers. The formulation and discretization of this problem are discussed in detail in Appendix A.

We have first considered a three-level preconditioner. Numerical tests confirmed the results of the smoothing analysis of Section 3.4.2: three levels with geometric coarsening are found inefficient for three-dimensional problems (see Figure 3.11). We are then considering a perturbed two-level cycle as a preconditioner where an iterative method is used on the coarse level (Algorithm 15). This involves that the dominant component in terms of computational work of the two-level method will be the solution method of the coarse problem. Nevertheless, a convergence criterion must be chosen to stop iterative methods. Therefore, we have to select a convergence threshold for the solution of the coarse level problem in order to minimize the computational cost of the coarse solution phase without damaging the preconditioning properties of the two-grid cycle. In Section 3.4.1, we show that, when using a nonlinear coarse solver (for instance preconditioned GMRES), a large coarse tolerance can ensure a convergence factor close to the one obtained when the coarse problem is solved exactly. This will be proved thanks to a Rigorous Fourier Analysis and corresponds to the main new result of this chapter.

Notwithstanding, even if coarse problems are solved within a large tolerance, this remains still the most expensive part of the perturbed two-grid method. It is then of great interest to select the other components of the two-level cycle to reduce the number of required iterations. A way to improve a two-grid cycle is to improve the smoothers. It often needs to perform a few smoothing iterations more to really improve the cycle. In Section 3.4.2, we select the smoother according to some numerical experiments and show their smoothing effect on the three-dimensional Helmholtz operator with PML. This selection is performed with numerical experiments because a traditional Local Fourier Analysis can only provide the smoothing factors reported in Table 3.4. Indeed, boundary conditions are not taken into account in LFA and non-standard smoothers, for instance Krylov methods, cannot be analyzed in this framework. Besides, we use the two-grid method as a preconditioner and not as a solver, this does not enable us to make a Fourier analysis without including some random parameters [125].

Prolongation and restriction could also be selected to improve the two-level operator choosing them depending on the matrix [127] or of higher-order (cubic, quadratic) [61, Section 3.4.3]. Yet to choose matrix-dependent transfer operators implies a higher cost in memory and operations than trilinear interpolation and full-weighting restriction. Furthermore the implementation of high-order transfer operators in a parallel environment is not straightforward; it requires neighboring points at a sometimes large distance. Thus, we do only consider full-weighting restriction and its adjoint as an interpolation in this work.

3.4.1 Approximation of the convergence factor of a perturbed two-grid method

In this section, we first consider a two-grid cycle used as a solver on the three-dimensional Helmholtz operator with Dirichlet boundary conditions at small wavenumbers. The discretization of the Helmholtz operator is still handled with a second order finite difference scheme for a vertex-centered grid arrangement.

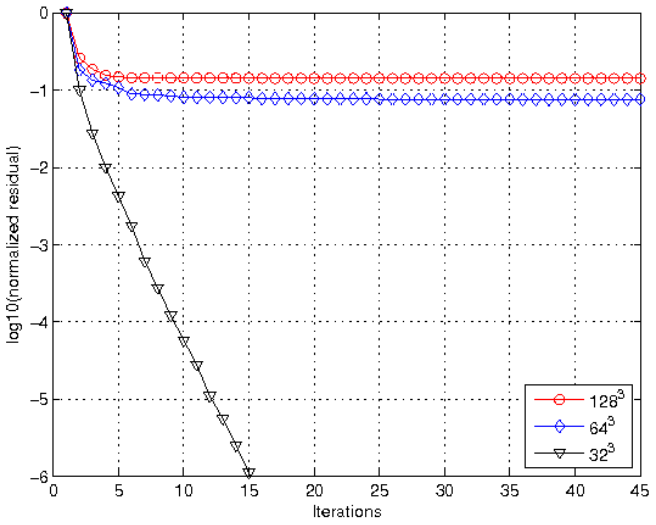

Figure 3.11: Histories of convergence of FGMRES(5) preconditioned by a three-grid V-cycle with two iterations of lexicographical forward Gauss-Seidel as pre- and post-smoother ($\nu_1 = \nu_2 = 2$) for a wavenumber $k = \dfrac{\pi}{6h}$.

We choose both these boundary conditions and wavenumbers to be able to use the elements of RFA theory introduced in Section 3.3.1 and to obtain a general idea on the influence of the accuracy required for the coarse solution. Through this study, we consider a two-grid cycle described in Algorithm 15 (its main components have been chosen as in Algorithm 13). Algorithm 15 describes a classical two-grid cycle when $\varepsilon_{2h} = 0$ (i.e. when the coarse problem is solved exactly).

Algorithm 15 Perturbed two-grid cycle to solve $L_h u_h = b_h$

1: Presmoothing: $u_h := \mathcal{S}_h(L_h, u_h, b_h, \nu_1)$
2: Compute the residual r_h: $r_h = b_h - L_h\, u_h$
3: Restrict the residual: $b_{2h} = I_h^{2h}\, r_h$
4: Set $u_{2h} := 0$
5: Solve approximately $L_{2h}u_{2h} = b_{2h}$ on Ω_{2h} such that $\dfrac{\|b_{2h} - L_{2h}u_{2h}\|_2}{\|b_{2h}\|_2} \leq \varepsilon_{2h}$.
6: Interpolate the coarse solution u_{2h} to obtain a correction of the fine solution u_h: $I_{2h}^h\, u_{2h}$
7: Add this correction to the solution: $u_h := u_h + I_{2h}^h\, u_{2h}$
8: Postsmoothing: $u_h := \mathcal{S}_h(L_h, u_h, b_h, \nu_2)$

We would like to determine an estimation of the convergence factor of a two-grid cycle with an approximate coarse grid solution denoted later by T_h. In the SPD case, it is known that the coarse tolerance (ε_{2h}) is not required to be very tight to obtain a good convergence factor for a two-grid cycle [115, p. 45]. We denote by P_h the perturbed two-grid iteration matrix where the coarse problem is solved inexactly:

$$P_h(u - u_h) \quad = \quad S_h^{\nu_2}(I_h - I_{2h}^h C_{2h} I_h^{2h} L_h) S_h^{\nu_1}(u - u_h),$$

denoting by C_{2h} the iteration matrix of the coarse solution method. According to [87, Relation 3.2] - if a symmetric multigrid scheme is considered [87, section 1] - an upper bound of the spectral radius of P_h can be found with respect to the spectral radii of both M_h and C_{2h}:

$$\begin{aligned} \rho(P_h) &\leq 1 - (1 - \rho(M_h))(1 - \rho(C_{2h})), \\ \rho(P_h) &\leq \rho(M_h) + \rho(C_{2h})(1 - \rho(M_h)). \end{aligned}$$

Therefore, if above $\rho(C_{2h}) = 0.1$ and $\rho(M_h) = 0.5$, $\rho(P_h)$ is bounded by 0.6 which remains attractive. The estimation of $\rho(P_h)$ can be even more rigorous considering its representation in the Fourier basis. Thus, if C_{2h} can be written in the Fourier basis, $\rho(P_h)$ can be explicitly computed. However this analysis does not cover the case of a Krylov method for the coarse problem. This is due to the non-linearity of Krylov methods. Indeed the solution can be expressed as a polynomial of the matrix $\mathcal{A}_h$ applied to the initial error $(u - u_h^{(0)})$:

$$u - (u_h)_m = \sum_{k=0}^{m-1} \alpha_k \mathcal{A}_h^k (u - u_h^{(0)}),$$

the coefficients α_k of the minimization polynomial depend nonlinearly on the operator $\mathcal{A}_h$ and the projected residual. Despite this nonlinearity, we propose a simplified analysis to obtain an estimation of the convergence factor of a perturbed two-level cycle depending on the coarse tolerance, ε_{2h}. This approach consists in injecting in the Fourier representation of the coarse grid operator (see Theorem 1) a perturbation term corresponding to the approximate coarse problem solution for each $(l_1, l_2, l_3) \mid l_1, l_2, l_3 < \frac{n}{2}$.

First, we consider the following obvious statement. Solving a linear system $Ax = b$ with an iterative method such that $\dfrac{\|b - A\widetilde{x}\|_2}{\|b\|_2} \leq \varepsilon$, is equivalent to solve exactly the following linear system with a perturbed right-hand side $b + \Delta b$ such that:

$$A\widetilde{x} = b + \Delta b \text{ with } \|\Delta b\|_2 \leq \varepsilon \|b\|_2 \Leftrightarrow \frac{\|b - A\widetilde{x}\|_2}{\|b\|_2} \leq \varepsilon. \tag{3.10}$$

In this case Δb is nothing else than the opposite of the residual: $\Delta b = -(b - A\widetilde{x})$. Thus, we will consider the effect of the inaccuracy of the coarse solution when considering a perturbed right-hand side on the coarse level $[b_{2h} + \Delta b_{2h}]$ instead of the coarse right-hand side $b_{2h} = I_h^{2h} L_h S_h^{\nu_1}(u - u_h)$ only. With these notations, the perturbed two-grid operator T_h implemented in Algorithm 15 can be written using Equation (3.2):

$$T_h(u - u_h) \quad = \quad S_h^{\nu_2}\left(S_h^{\nu_1}(u - u_h) - I_{2h}^h L_{2h}^{-1}(b_{2h} + \Delta b_{2h})\right). \tag{3.11}$$

The following proposition enables us to block diagonalize this perturbed two-grid operator in the Fourier basis using some reasonable assumptions.

Proposition 9. *With the same notations as in Algorithm 15 and Corollary 5, we consider one cycle of the perturbed two-grid operator T_h (Equation (3.11)). This operator has the following representation in the Fourier basis:*

$$T_h \quad \widehat{=} \quad [\widehat{S_h}^{\nu_2}(l_1, l_2, l_3)\widehat{\Upsilon}_h^{2h}(l_1, l_2, l_3)\widehat{S_h}^{\nu_1}(l_1, l_2, l_3)]_{l_1,l_2,l_3=1,\ldots,n/2},$$

$$\text{with} \begin{cases} \widehat{\Upsilon}_h^{2h}(l_1, l_2, l_3) = \begin{cases} I_8 - (1 + \varepsilon_{2h}^{l_1,l_2,l_3})[b_i\, c_j]_{8,8}/\Lambda \text{ if } l_1, l_2, l_3 < \frac{n}{2} \\ I_4 \text{ if } l_1 = \frac{n}{2} \text{ or } l_2 = \frac{n}{2} \text{ or } l_3 = \frac{n}{2} \\ I_2 \text{ if } l_1 = l_2 = \frac{n}{2} \text{ or } l_1 = l_3 = \frac{n}{2} \text{ or } l_2 = l_3 = \frac{n}{2} \\ I_1 \text{ if } l_1 = l_2 = l_3 = \frac{n}{2} \end{cases} \\ b_{2h} \widehat{=} \widehat{I}_h^{2h}\widehat{L}_h\widehat{S_h}^{\nu_1}(\widehat{u} - \widehat{u_h}) = [\alpha_{2h}^{l_1,l_2,l_3}]_{l_1,l_2,l_3=1,\ldots,n/2-1}, \\ \Delta b_{2h} \widehat{=} [\varepsilon_{2h}^{l_1,l_2,l_3}\alpha_{2h}^{l_1,l_2,l_3}]_{l_1,l_2,l_3=1,\ldots,n/2-1}, \text{ with } \varepsilon_{2h}^{l_1,l_2,l_3} \in \mathbb{R}, \ \forall\, l_1, l_2, l_3 = 1, \ldots, n/2 - 1. \end{cases}$$

Proof. As discussed in Section 3.3.1, on the coarse grid space Ω_{2h}, spaces of harmonics $E_{2h}^{l_1,l_2,l_3}$ are reduced to one-dimensional spaces $span[\varphi_{2h}^{l_1,l_2,l_3}]$. Therefore, we can write in the coarse Fourier basis $(\varphi_{2h}^{l_1,l_2,l_3}, l_1, l_2, l_3 = 1, \ldots, n/2 - 1)$, the components of the coarse right-hand side perturbation as a collinear perturbation of the coarse right-hand side for each (l_1, l_2, l_3), $l_1, l_2, l_3 \leq 1, \ldots, n/2 - 1$, i.e.:

$$\begin{aligned} b_{2h} \quad &\widehat{=} \quad \left[\alpha_{2h}^{l_1,l_2,l_3}\right]_{l_1,l_2,l_3=1,\ldots,n/2-1}, \\ \Delta b_{2h} \quad &\widehat{=} \quad \left[\varepsilon_{2h}^{l_1,l_2,l_3}\alpha_{2h}^{l_1,l_2,l_3}\right]_{l_1,l_2,l_3=1,\ldots,n/2-1}. \end{aligned}$$

We show how the expression of $\widehat{\Upsilon}_h^{2h}$ can be deduced from the expression of T_h:

$$\begin{aligned} T_h(u-u_h) &= S_h^{\nu_2}\left(S_h^{\nu_1}(u-u_h) - I_{2h}^h L_{2h}^{-1}[b_{2h}+\Delta b_{2h}]\right), \\ &\widehat{=} \widehat{S}_h^{\nu_2}(\widehat{S}_h^{\nu_1}(\widehat{u}-\widehat{u}_h) - \widehat{I}_{2h}^h \widehat{L}_{2h}^{-1}[(1+\varepsilon_{2h}^{l_1,l_2,l_3})\alpha_{2h}^{l_1,l_2,l_3}]_{l_1,l_2,l_3=1,\dots,n/2-1}). \end{aligned}$$

Since the coarse right-hand side has the following expression in the Fourier basis $b_{2h} \widehat{=} \widehat{I}_h^{2h}\widehat{L}_h\widehat{S}_h^{\nu_1}(\widehat{u}-\widehat{u}_h)$, we have:

$$\begin{aligned} T_h(u-u_h) &\widehat{=} \widehat{S}_h^{\nu_2}(\widehat{S}_h^{\nu_1}(\widehat{u}-\widehat{u}_h) - \widehat{I}_{2h}^h \widehat{L}_{2h}^{-1} diag([1+\varepsilon_{2h}^{l_1,l_2,l_3}]_{l_1,l_2,l_3=1,\dots,n/2-1})\widehat{I}_h^{2h}\widehat{L}_h\widehat{S}_h^{\nu_1}(\widehat{u}-\widehat{u}_h)) \\ &\widehat{=} \widehat{S}_h^{\nu_2}(I_h - \widehat{I}_{2h}^h \widehat{L}_{2h}^{-1} diag([1+\varepsilon_{2h}^{l_1,l_2,l_3}]_{l_1,l_2,l_3=1,\dots,n/2-1})\widehat{I}_h^{2h}\widehat{L}_h)\widehat{S}_h^{\nu_1}(\widehat{u}-\widehat{u}_h). \end{aligned}$$

It follows that:

$$T_h \widehat{=} \begin{cases} \left[\widehat{S}_h^{\nu_2}(l_1,l_2,l_3)\left(I_8 - (1+\varepsilon_{2h}^{l_1,l_2,l_3})\right)\widehat{\Xi}(l_1,l_2,l_3)\widehat{S}_h^{\nu_1}(l_1,l_2,l_3)\right]_{l_1,l_2,l_3=1,\dots,n/2-1} \\ \\ \widehat{S}_h^{\nu_2+\nu_1}(l_1,l_2,l_3) \text{ if } k=\frac{n}{2} \text{ or } l=\frac{n}{2} \text{ or } m=\frac{n}{2}. \end{cases}$$

where $\widehat{\Xi}(l_1,l_2,l_3) = \widehat{I}_{2h}^h(l_1,l_2,l_3)\widehat{L}_{2h}^{-1}(l_1,l_2,l_3)\widehat{I}_h^{2h}(l_1,l_2,l_3)\widehat{L}_h(l_1,l_2,l_3)$ for $l_1,l_2,l_3 = 1,...,n/2-1$.

Therefore we have

$$\widehat{\Upsilon}_h^{2h} = \begin{cases} [I_8 - (1+\varepsilon_{2h}^{l_1,l_2,l_3})\widehat{I}_{2h}^h(l_1,l_2,l_3)\widehat{L}_{2h}^{-1}(l_1,l_2,l_3)\widehat{I}_h^{2h}(l_1,l_2,l_3)\widehat{L}_h(l_1,l_2,l_3)]_{l_1,l_2,l_3=1,\dots,n/2-1} \\ I_4 \text{ if } k=\frac{n}{2} \text{ or } l=\frac{n}{2} \text{ or } m=\frac{n}{2} \\ I_2 \text{ if } k=m=\frac{n}{2} \text{ or } k=l=\frac{n}{2} \text{ or } l=m=\frac{n}{2} \\ I_1 \text{ if } k=l=m=\frac{n}{2}. \end{cases}$$

The expression of the coarse grid correction operator K_h^{2h} in Theorem 1 gives the final explicit expression of $\widehat{\Upsilon}_h^{2h}$.

□

We now focus on a specific perturbation Δb_{2h} such that $\|\Delta b_{2h}\|_2 \leq \varepsilon_{2h}\|b_{2h}\|_2$, which means that the coarse problem is solved with a normalized error below ε_{2h} (see Relation 3.10). This hypothesis on Δb_{2h} adds a constraint on its components in the Fourier basis. Using notations of Proposition 9, the relation $\|\Delta b_{2h}\|_2^2 \leq \varepsilon_{2h}^2\|b_{2h}\|_2^2$ becomes:

$$\sum_{l_1,l_2,l_3=1}^{n/2-1} (\varepsilon_{2h}^{l_1,l_2,l_3}\alpha_{2h}^{l_1,l_2,l_3})^2 \leq \varepsilon_{2h}^2 \sum_{l_1,l_2,l_3=1}^{n/2-1} (\alpha_{2h}^{l_1,l_2,l_3})^2. \tag{3.12}$$

Therefore, to perform a rigorous Fourier analysis with a coarse perturbation satisfying $\|\Delta b_{2h}\|_2 \leq \varepsilon_{2h}\|b_{2h}\|_2$, we need to select its Fourier components such that relation (3.12) is satisfied. In practice, we cannot verify (3.12) on the $\varepsilon_{2h}^{l_1,l_2,l_3}$: it would involve the coarse right-hand-side coefficients $\alpha_{2h}^{l_1,l_2,l_3}$. The Fourier analysis could then be possible only if these coefficients were accessed at each iteration of the two-grid cycle; this kind of analysis would be pointless. We then focus on a subset of the set spanned by the hypothesis $\|\Delta b_{2h}\|_2 \leq \varepsilon_{2h}\|b_{2h}\|_2$:

$$S_{\varepsilon_{2h}}^{l_1,l_2,l_3} = \left\{\varepsilon_{2h}^{l_1,l_2,l_3} \in \mathbb{R} \,|\, |\varepsilon_{2h}^{l_1,l_2,l_3}| \leq \varepsilon_{2h}\right\}.$$

Choosing this subset clearly implies a loss of generality in our study. However, if a relaxation method is used as a precondtioner in the coarse solver, it is reasonable to think that the coarse residual Δb_{2h} will be smooth (see Section 3.4.2 for a graphical illustration). Furthermore this subset is found to be relevant in practice (see Table 3.5) and allows to describe well the perturbed two-grid behavior. Practically, we select few values (10, say) for $\varepsilon_{2h}^{l_1,l_2,l_3}$ in $[-\varepsilon_{2h},\varepsilon_{2h}]$ and compute the corresponding spectral radii of $\widehat{T}_h(l_1,l_2,l_3,\varepsilon_{2h}^{l_1,l_2,l_3})$ for each triplet (l_1,l_2,l_3) with $l_1,l_2,l_3 = 1,\dots,n-1$. Finally, we obtain an estimation $(\tilde{\rho}(T_h))$ of $\rho(T_h)$ as:

$$\tilde{\rho}(T_h) = \max_{l_1,l_2,l_3=1,\dots,n-1} \max_{\varepsilon_{2h}^{l_1,l_2,l_3} \in S_{\varepsilon_{2h}}^{l_1,l_2,l_3}} \rho(\widehat{T}_h(l_1,l_2,l_3,\varepsilon_{2h}^{l_1,l_2,l_3})). \tag{3.13}$$

In Table 3.5 we compare $\tilde{\rho}(T_h)$ with the experimental convergence factor $\rho_{Exp}(T_h)$, Algorithm 15 with a preconditioned Krylov solver on the coarse level. The experimental convergence factor $\rho_{Exp}(T_h)$ is obtained by computing the ratio between the two last errors in the history of convergence.

We perform one Jacobi iteration ($\nu_1 = \nu_2 = 1$) with a relaxation parameter $\omega = 6/7$. We select the largest wavenumbers k on the original Helmholtz operator for which the classical two-grid method has still a good convergence factor, typically 0.5.

ε_{2h}	64^3, $k = 15$		128^3, $k = 19$		256^3, $k = 30$		512^3, $k = 36$	
	$\tilde{\rho}(T_h)$	$\rho_{Exp}(T_h)$	$\tilde{\rho}(T_h)$	$\rho_{Exp}(T_h)$	$\tilde{\rho}(T_h)$	$\rho_{Exp}(T_h)$	$\tilde{\rho}(T_h)$	$\rho_{Exp}(T_h)$
1	1.31	0.99	1.33	0.99	1.23	0.99	1.38	0.99
0.9	1.20	0.92	1.22	0.88	1.12	0.89	1.26	0.93
0.8	1.09	0.80	1.11	0.79	1.01	0.86	1.14	0.82
0.7	0.98	0.70	0.99	0.69	0.90	0.83	1.03	0.73
0.6	0.87	0.63	0.88	0.61	0.79	0.74	0.92	0.63
0.5	0.76	0.53	0.77	0.51	0.69	0.69	0.80	0.53
0.4	0.65	0.45	0.66	0.46	0.60	0.62	0.69	0.43
0.3	0.58	0.48	0.58	0.50	0.57	0.53	0.58	0.41
0.2	0.56	0.49	0.55	0.50	0.55	0.45	0.55	0.43
0.1	0.54	0.49	0.53	0.50	0.53	0.41	0.53	0.46
10^{-12}	0.52	0.49	0.51	0.50	0.51	0.50	0.51	-

Table 3.5: Theoretical estimation of the convergence factor ($\tilde{\rho}(T_h)$) and experimental convergence factors $\rho_{Exp}(T_h)$ for several coarse tolerances ε_{2h}.

Both theoretical and experimental convergence factors in Table 3.5 confirm that a really large tolerance on the coarse problem can lead to the same convergence factor as in the case of an exact coarse solution, $\varepsilon_{2h} = 10^{-12}$ (consider the last two rows of Table 3.5). Indeed, it can first be noticed that when the coarse tolerance is decreasing, the convergence factor is decreasing as well for both theoretical and experimental computations. Then, for all grid sizes and wavenumbers considered, these convergence factors ($\tilde{\rho}(T_h)$ and $\rho_{Exp}(T_h)$) are close to 0.5 when ε_{2h} is less or equal to 0.2. Table 3.5 also shows that the theoretical convergence factor $\tilde{\rho}(T_h)$ estimates quite well $\rho(T_h)$. Thus, a large coarse tolerance ε_{2h} can provide a two-grid cycle as efficient as a two-grid cycle with an exact coarse solution on this model problem. Indeed, really large coarse tolerance (about 0.1) can provide convergence factors that are similar.

However only small wavenumbers were considered in this simplified analysis. Therefore since at high wavenumbers the two-grid method does not converge on the original Helmholtz operator, we propose to analyze spectrum as in Section 3.3.1. We no longer consider the spectrum of the preconditioned operator $\mho_h^{-1}$ (see Equation 3.7) depending on β but on the coarse tolerance ε_{2h}, $\mho_h^{-1}(\varepsilon_{2h})$. Thus, we compute the spectrum of the following operator:

$$L_h^{(0)}\,\mho_h^{-1}(\varepsilon_{2h}) = L_h^{(0)}\,(I_h - T_h(\varepsilon_{2h}))\,(L_h^{(0)})^{-1}\,. \tag{3.14}$$

Since a representation of $T_h(\varepsilon_{2h})$ in the Fourier basis (Proposition 9) is available, we plot the spectrum of the following operator:

$$\left[\widehat{L}_h^{(0)}(l_1,l_2,l_3)\,(\widehat{I}_h(l_1,l_2,l_3) - \widehat{T}_h(l_1,l_2,l_3,\widehat{\varepsilon}_{2h}^{l_1,l_2,l_3}))\,(\widehat{L}_h^{(0)})^{-1}(l_1,l_2,l_3)\right]_{l_1,l_2,l_3=1,\ldots,n/2-1},$$

where $\widehat{\varepsilon}_{2h}^{l_1,l_2,l_3} = \text{argmax}\left\{\rho(\widehat{T}_h(l_1,l_2,l_3,\varepsilon_{2h}^{l_1,l_2,l_3})) \mid \varepsilon_{2h}^{l_1,l_2,l_3} \in S_{\varepsilon_{2h}}^{l_1,l_2,l_3}\right\}$.

Figure 3.12 shows the spectra of $L_h^{(0)}\,\mho_h^{-1}(\varepsilon_{2h})$ for two values of ε_{2h} ($\varepsilon_{2h} = 0$ and $\varepsilon_{2h} = 0.1$ respectively).

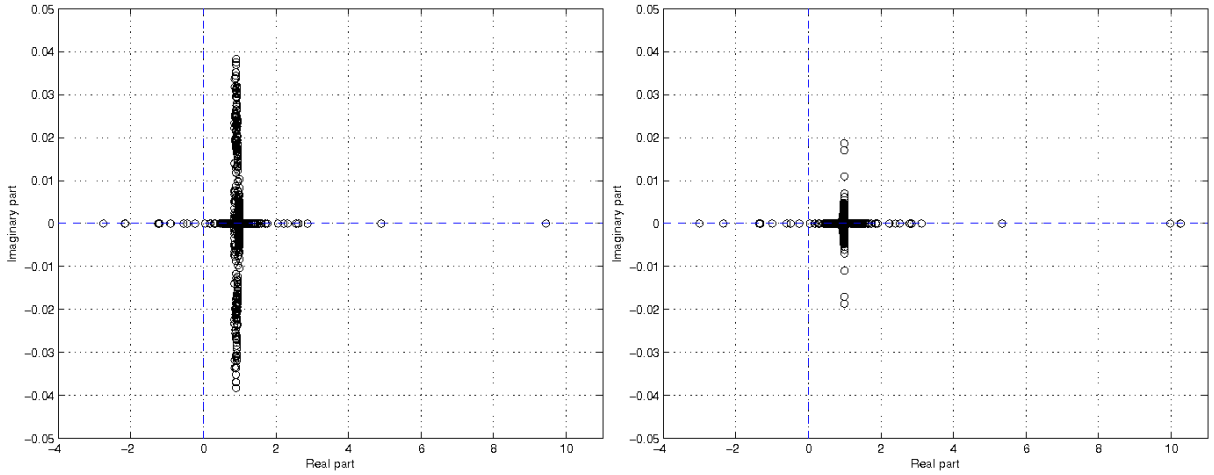

Figure 3.12: Spectra of $L_h^{(0)}\,\mathfrak{V}_h^{-1}(\varepsilon_{2h})$ for two values of ε_{2h} ($\varepsilon_{2h} = 0$ (left) and $\varepsilon_{2h} = 0.1$ (right)), considering Helmholtz problems with Dirichlet boundary conditions with a 64^3 grid for a wavenumber $k = \pi/(6h)$ and two iterations of Jacobi as a smoother ($\nu_1 = \nu_2 = 1$) with relaxation parameter $\omega_r = 0.4$.

We note that Figure 3.12, left corresponds to the same spectrum (with a different scale) as in the left part of Figure 3.6. The spectra shown in Figure 3.12 are very similar, eigenvalues are clustered around one whereas few eigenvalues are isolated. Therefore the perturbed two-level method may be a preconditioner as efficient as the exact two-level method for the original Helmholtz problem. Nevertheless, we shall later investigate whether this result holds with absorbing boundary conditions of PML type. We address this important topic in Section 3.5. Numerical examples show that this property still holds for the original Helmholtz operator with PML at large wavenumbers. Beforehand, we discuss how to choose the smoother in practice.

3.4.2 Smoother selection

We propose then to select the smoother thanks to numerical experiments, considering an exact coarse solver. For this selection, we consider a classical two-grid method (Algorithm 13) as a preconditioner of FGMRES(5) (Algorithm 3). The choice of a flexible method is motivated by the possible use of a Krylov method as a smoother as it is advised in [37]. This study is done for several grids (and their corresponding wavenumbers) in a parallel setting. We refer to [47] and [48] for the parallel implementation of GMRES and FGMRES respectively. For the implementation of the two-level preconditioner on structured grids, we refer to [115, Section 6]. We select the number of cores so that the local problems have the same size on each grid. We will deal only with the same number of iterations for pre- and post- smoothing: $\nu_1 = \nu_2 = \nu$. Since a parallel environment is chosen, we use local relaxation methods. As noticed in Section 3.3.2, the smoothing analysis shows that lexicographic Gauss-Seidel type methods behave well for the Helmholtz problem on the fine level. We are then focusing on lexicographic Gauss-Seidel smoothers. We consider the local lexicographic forward Gauss-Seidel method [115, Remark 6.2.5], instead of the lexicographic forward Gauss-Seidel method. We denote by GS_{LEX} the local lexicographic Gauss-Seidel , GS_{SYM} the local symmetric lexicographic forward Gauss-Seidel and $GMRES(\nu)/GS_{SYM}(1)$, ν iterations of GMRES preconditioned by one iteration of GS_{SYM}. The coarse problem is solved with a restarted GMRES(10) preconditioned by one iteration of GS_{SYM} so that the coarse normalized residual is below 10^{-12}. The initial solution of FGMRES is set to zero and the convergence threshold of the method is set at 10^{-6}. Numerical experiments are reported in Table 3.6.

We first remark in Table 3.6 that the number of iterations is increasing with the size of the problem. This is due to the fact that the wavenumber is coupled to the grid size (see relation A.4 in Appendix A), implying its increase with respect to the inverse of h (the grid size) and thus the increasing indefiniteness of the problem. Concerning the smoothers, reading the table from left to right, a standard GS_{LEX} is first used and improved by increasing the number of both pre- and post-smoothing iterations. This method can be once more improved considering as a second iteration a backward Gauss-Seidel iteration (GS_{SYM}) but increasing the GS_{SYM} number of iterations has nearly no effect. GMRES alone is not a good smoother

		$GS_{LEX}(\nu)$		$GS_{SYM}(\nu)$		$GMRES(\nu)$		$GMRES(\nu)/GS_{SYM}(1)$	
Grid	#Cores	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$
64^3	2	14	11	8	8	25	12	7	6
128^3	16	17	14	11	11	28	14	10	9
256^3	128	27	20	18	18	44	24	17	16
512^3	1024	68	47	44	43	121	63	42	40

Table 3.6: Number of iterations needed to reach 10^{-6} for FGMRES(5) preconditioned by a two-grid cycle considering several smoothers and grids ($1/h^3$) at wavenumbers $k = \frac{\pi}{6h}$.

but using GS_{SYM} as a preconditioner for GMRES gives the best results when two GMRES iterations are performed with respect to the number of iterations. Since the coarse problem is solved at each iteration and it is the dominant component of the two-grid preconditioner in terms of computational resources, we select the smoother which minimizes the number of iterations. We will then use as a smoother a $GMRES(2)$ preconditioned by one GS_{SYM} iteration in our perturbed two grid algorithm.

A graphical study in Matlab is then provided to further analyze the results of Table 3.6. In fact, by plotting a slice of a three-dimensional random error after smoothing, we want to point out how smoothers are handling the high frequency components of the error for the Helmholtz equation with PML. Therefore, we choose a random error vector with the Matlab random number generator rand('seed',0) and emulate parallelism for GS_{LEX} ans GS_{SYM}. Smoothing is then performed on this error for each method of Table 3.6 and a slice of the smoothed error is shown. This slice is located in the vertical plane (x, z) of the unit cube $\Omega = [0, 1]^3$ for $y = 0.5$. Since this is a Matlab program, we consider the smallest grid size 64^3 ($k = \frac{\pi}{6h}$) considered in Table 3.6 and emulate parallelism on two processors. The smoothers are parallelized partitioning the three-dimensional cubic physical domain in smaller parallelepipeds. In our case, since we have only 2 processors, the physical domain is divided in two parallelepiped boxes along the z-direction. The error slice is plotted in Figure 3.13, the error slices after application of selected smoothers in Table 3.6 are plotted in Figures 3.14, 3.15, 3.16 and 3.17 respectively.

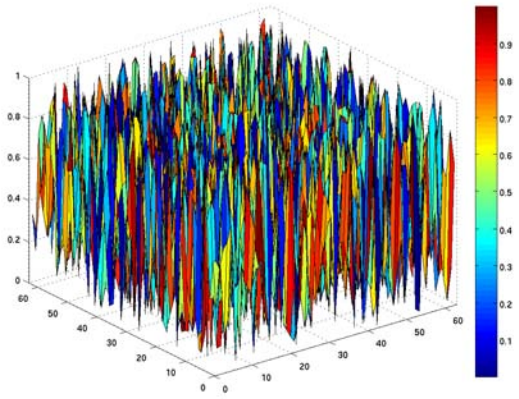

Figure 3.13: Slice of the initial error ($y = 0.5$) in the plane (x, z) for the 64^3 grid built with the Matlab random number generator rand('seed',0).

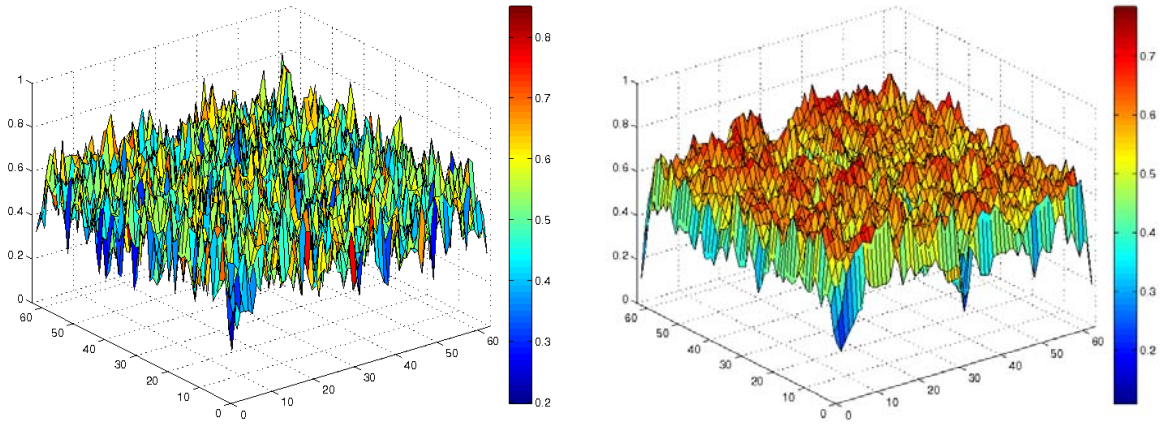

Figure 3.14: Slices of the error ($y = 0.5$) in the plane (x, z) after one iteration of Gauss-Seidel ($GS_{LEX}(1)$, left) and two iterations of Gauss-Seidel ($GS_{LEX}(2)$, right) for the 64^3 grid ($k = 33.51$) on two processors.

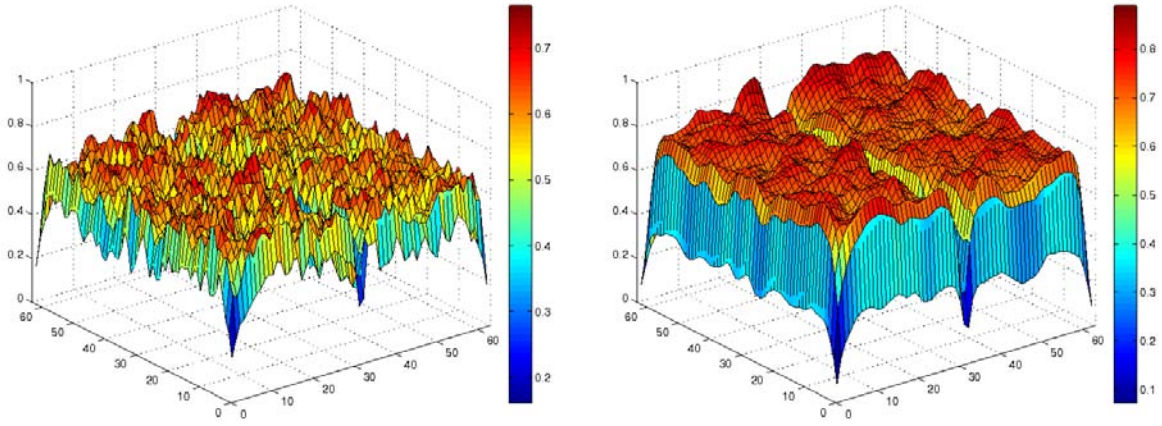

Figure 3.15: Slices of the error ($y = 0.5$) in the plane (x, z) after one iteration of Symmetric Gauss-Seidel ($GS_{SYM}(1)$, left) and two iterations of Symmetric Gauss-Seidel ($GS_{SYM}(2)$, right) for the 64^3 grid ($k = 33.51$) on two processors.

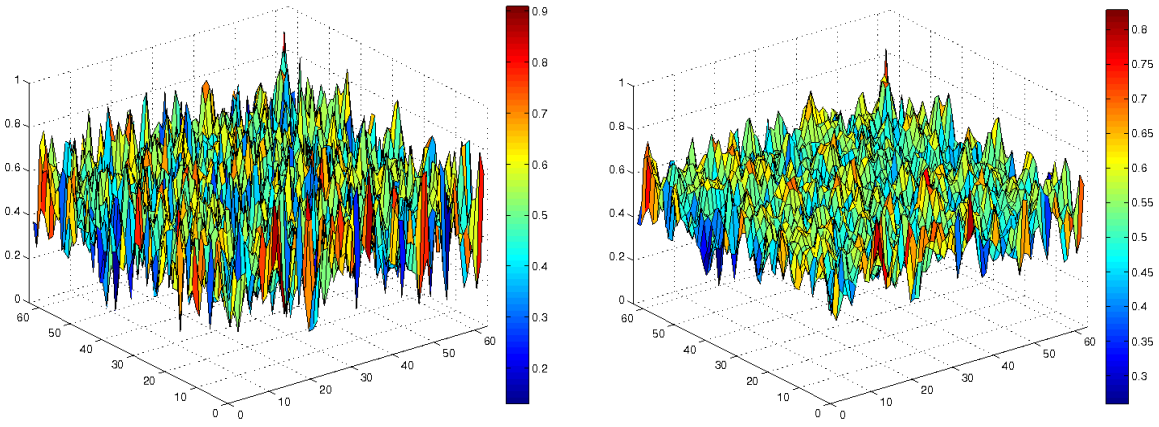

Figure 3.16: Slices of the error ($y = 0.5$) in the plane (x, z) after one iteration of GMRES ($GMRES(1)$, left) and two iterations of GMRES ($GMRES(2)$, right) for the 64^3 grid ($k = 33.51$) on two processors.

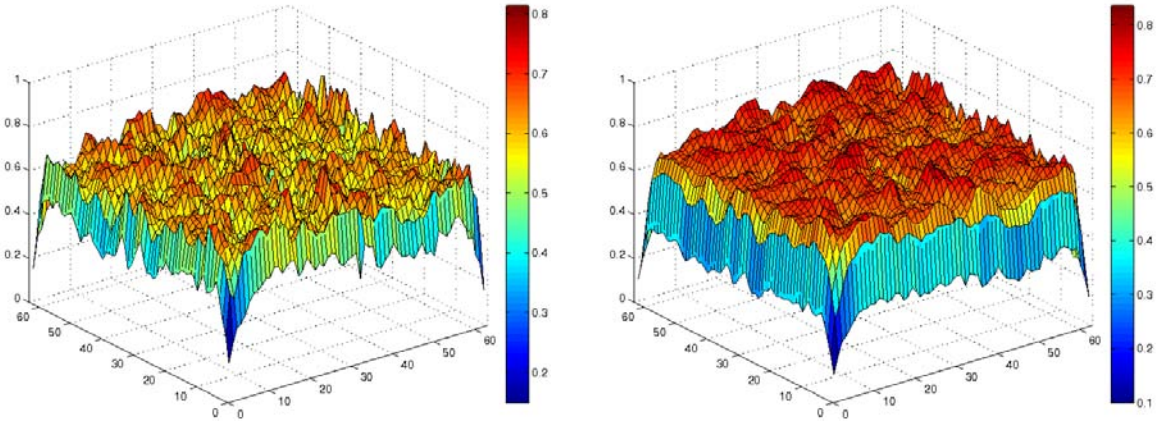

Figure 3.17: Slices of the error ($y = 0.5$) in the plane (x, z) after one iteration of GMRES preconditioned by one iteration of symmetric Gauss-Seidel ($GMRES(\nu)/GS_{SYM}(1)$, left) and two iterations of GMRES preconditioned by one iteration of symmetric Gauss-Seidel ($GMRES(\nu)/GS_{SYM}(2)$, right) for the 64^3 grid ($k = 33.51$) on two processors.

We first notice that there is a relation between the shape of the error, considered as a surface, and the number of iterations in Table 3.6. In fact, the smoother the surface, the lesser the need of preconditioning steps. $GMRES$ has nearly no smoothing effect (Figure 3.16) whereas using it in combination with GS_{SYM} gives the smoothest errors. However, the effect of parallelism on smoothing is remarkable on these plots, especially on the right plot of Figure 3.15. Indeed, one can see the physical domain splitting on the z axis in this figure, smoothing is performed independently on each subdomain. This is a consequence of local Gauss-Seidel definition, it is acting only locally. Yet it can be seen in Figure 3.17 that $GMRES$ enables GS_{SYM} to smooth uniformly all the components of the error. Therefore, even if $GMRES$ is a bad smoother, it can be used efficiently with standard local relaxation method as a preconditioner to balance their lack of parallelisation.
In Section 3.5, we analyze the perturbed two-grid preconditioner according to the coarse tolerance with the spectrum analysis presented in Section 2.3.1.

3.5 Spectrum analysis of the perturbed two-level method in the Flexible GMRES framework

Before considering the spectrum analysis, we first summarize the findings of Section 3.4 in Algorithm 16.

3.5.1 Algorithm of the perturbed two-level preconditioner for three-dimensional Helmholtz problem

Algorithm 16 Perturbed two-grid cycle to solve approximately $L_h z_h = v_h$

1: Pre-smoothing: ν_1 iterations of preconditioned GMRES(m_s): $z_h := K(L_h, v_h, z_h, m_s)$.
2: Restriction of the residual to obtain the coarse right-hand side: $v_{2h} = I_h^{2h}(v_h - Az_h)$
3: Solve only approximately the coarse problem $L_{2h}z_{2h} = v_{2h}$ such that $\frac{\|v_{2h} - L_{2h}z_{2h}\|_2}{\|v_{2h}\|_2} \leq \varepsilon_{2h}$ thanks to a preconditioned GMRES(m_c).
4: Interpolation of the coarse solution z_{2h}: $I_{2h}^h z_{2h}$.
5: Add this correction to z_h: $z_h := z_h + I_{2h}^h z_{2h}$
6: Post-smoothing: ν_2 iterations of preconditioned GMRES(m_s): $z_h = K(L_h, v_h, z_h, m_s)$.

All Krylov methods are preconditioned by one reverse symmetric Gauss-Seidel iteration.
Coarse problem: Take zero as an initial guess.
Notations: z_h and z_{2h} the fine and the coarse grid solutions, v_h and v_{2h} the fine and the coarse grid right-hand sides, ε_{2h} the coarse tolerance, m_s is the smoother restart size, m_c is the coarse solver restart size.

The main differences between Algorithm 16 and Algorithm 13 are on one hand the use of a preconditioned Krylov method as a smoother and on the other hand the approximate solution of the coarse problem. This last point is the topic of this section: which stopping criterion should be used on the coarse level when the perturbed two-grid is used as a preconditioner? Since we are using nonlinear methods as smoother and coarse solver in our two-grid preconditioner, a traditional Fourier analysis (either local or rigorous) cannot help us to answer this question. However, the result obtained on an academic case in Section 3.4.1 tends to yield that a really large tolerance ε_{2h} can provide an efficient preconditioner. Indeed, we are using the same transfer operators and a discretization scheme of the same order as in the example in Section 3.4.1. Nevertheless, the boundary conditions (PML) were not taken into account, we then evaluate numerically the two-grid preconditioner according to its coarse tolerance. Moreover, in order to better understand the results of the next section, we will perform a Hessenberg spectrum analysis in the FGMRES framework (see Section 2.3.1).

3.5.2 Influence of the approximate coarse solution on the convergence of the Krylov method

In this section, we focus on the behavior of FGMRES(5) preconditioned by our perturbed two-grid method (Algorithm 16) according to the coarse convergence threshold ε_{2h}. We have then performed extensive numerical experiments for different coarse tolerances, using a restarted $GMRES(m)$ preconditioned by a local reverse symmetric Gauss-Seidel cycle as both a coarse solver ($m_c = 10$) and as a smoother ($m_s = 2$). The coarse solution method is obtained as soon as $\frac{\|b_{2h} - L_{2h}x_{2h}\|_2}{\|b_{2h}\|_2} \leq \varepsilon_{2h}$ is satisfied (Algorithm 16 line 3). The total number of iterations (number of applications of the preconditioner) needed by FGMRES(5) to converge to 10^{-6} for four grid sizes is reported in Table 3.7.

We first notice in Table 3.7 that the number of iterations is increasing with the problem size as in Table 3.6. Then, it can be noticed that the number of iterations has a general trend to decrease when the coarse tolerance is decreasing too. Finally, we remark that at a certain coarse tolerance, the number of iterations stabilizes. Indeed, for the smallest grids (64^3, 128^3), the numbers of iterations are the same for $\varepsilon_{2h} = 0.1$ and $\varepsilon_{2h} = 10^{-12}$, respectively 6 and 9. For the case of 256^3, one can see that the number of iterations is stabilizing to 17 when the coarse tolerance is below 0.4 (which is really close to the 16 iterations for $\varepsilon_{2h} = 10^{-12}$). For the largest grid (512^3), the number of iterations required no longer behaves in a monotonic way. In fact, the number of iterations first decreases when $\varepsilon_{2h} \in [0.6, 1]$, and then increases when $\varepsilon_{2h} \in [0.3, 0.5]$, and finally stabilizes around 40 iterations. Therefore, it can be seen on these examples that highly accurate

Grid	# Cores	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	10^{-12}
64^3	2	41	38	32	24	18	15	11	9	7	6	6
128^3	16	117	101	61	37	25	17	13	11	10	9	9
256^3	128	265	191	93	44	25	19	17	17	17	17	16
512^3	1024	628	367	146	45	33	35	37	42	41	41	40

Table 3.7: Number of iterations (It) of FGMRES(5) with respect to the coarse problem normalized tolerance (ε_{2h}) for wavenumbers $k = \dfrac{\pi}{6h}$.

coarse solution is not needed to reach convergence. Even worse, a more accurate solution can deteriorate (hopefully not too much) the convergence of FGMRES when large grids are considered.

We make a spectrum analysis to better understand this behavior.

3.5.3 Spectrum analysis in the flexible GMRES framework for three-dimensional homogeneous Helmholtz problems

In this section, we make a spectrum analysis as in Section 2.3.1 for the three-dimensional Helmholtz problem. The matrix H_{m+1} still denotes the augmented Hessenberg matrix:

$$H_{m+1} = \begin{bmatrix} \bar{H}_m & 0_{(m+1)\times(n-m-1)} \\ 0_{(n-m-1)\times(m+1)} & I_{n-m-1} \end{bmatrix}.$$

We compute the eigenvalues of H_{m+1} at the end of each restart for the same coarse grid tolerances as presented in Table 3.7 and superpose them on the same plot. The parameter guiding the quality of the preconditioner is then the coarse grid tolerance ε_{2h}. We focus on a test case where the restart parameter m is equal to 5 on a 512^3 grid (which seems to be the more interesting test case according to Table 3.7). We then compute the eigenspectra of H_{m+1} at each FGMRES restart for several coarse grid tolerances. We denote by $H_{m+1}^{(i)}$ the Hessenberg matrix corresponding to the *ith* restart and by $\lambda(H_{m+1}^{(i)}(\varepsilon_{2h}))$ the eigenspectrum of $H_{m+1}^{(i)}$ corresponding to the coarse tolerance ε_{2h}. Finally, we denote by $\Lambda(H_{m+1}(\varepsilon_{2h}))$ the union of the $\lambda(H_{m+1}^{(i)}(\varepsilon_{2h}))$ on the restart parameter i:

$$\Lambda(H_{m+1}(\varepsilon_{2h})) = \cup_i \lambda(H_{m+1}^{(i)}(\varepsilon_{2h})).$$

We plot $\Lambda(H_{m+1}(\varepsilon_{2h}))$ for $m = 5$ in Figure 3.18. The eigenspectra are distributed with respect to the coarse grid tolerance in the first five plots and we plot in the bottom-right corner three spectra for relevant values of ε_{2h}: 1, 0.6 and 10^{-12} to present an overview of the evolution of the spectrum. In the upper-left corner, it can be seen that $\Lambda(H_{m+1}(\varepsilon_{2h}))$ is very similar when $\varepsilon_{2h} = 1$ and $\varepsilon_{2h} = 0.9$: several eigenvalues are close to zero enclosed in an ellipse lying in a half plane of the complex plane. In the upper-right corner plot, $\Lambda(H_{m+1}(0.8))$ is located approximately in the same ellipse as the previous eigenspectra whereas it can be noticed that $\Lambda(H_{m+1}(0.7))$ is more farther from zero. Looking at Table 3.7, it appears that the number of iterations is greatly decreasing when $\varepsilon_{2h} \leq 0.7$. In fact, the number of iterations of FGMRES(5) seems to be related to the location of $\Lambda(H_{m+1}(\varepsilon_{2h}))$ in the complex plane. Indeed, for $\varepsilon_{2h} \geq 0.6$, the real parts of $\Lambda(H_{m+1}(\varepsilon_{2h}))$ move away from zero and then get closer to zero again when $\varepsilon_{2h} \leq 0.5$. A related behavior is recorded in Table 3.7, the number of iterations is varying with respect to the number of eigenvalues close or not to zero. Thus, this spectrum study gives extra information about the behavior of the method. It also confirms that the efficiency of the flexible preconditioner does not depend monotonically on the convergence of its coarse grid problem. Nevertheless, we can only deduce thanks to this analysis the optimal ε_{2h} parameter for one grid size only. In Table 3.7, the best ε_{2h} is 0.6 on a 512^3 grid, 0.4 on a 256^3 grid. Moreover, to converge at each iteration of FGMRES to a certain ε_{2h} on the coarse level can be very expensive in computational resources.

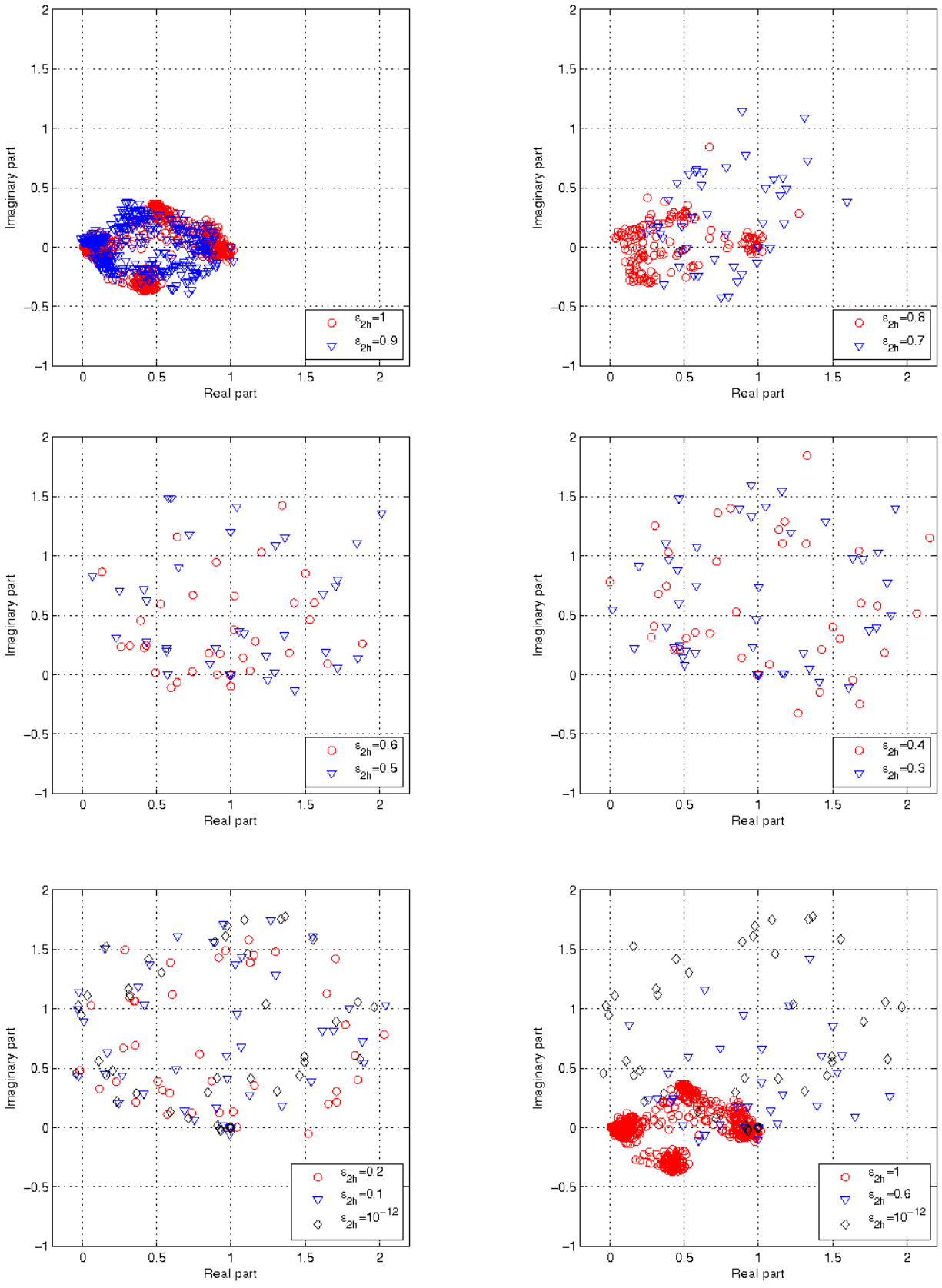

Figure 3.18: From right to left: $\Lambda(H_{m+1})$ for different coarse tolerance ε_{2h}, $m = 5$ on a 512^3 grid with $k = \dfrac{\pi}{6h}$ and PML.

We then decide to fix the number of coarse iterations per preconditioning cycle to save some computational time in the solution of the linear system. Indeed, we plot in Figure 3.19, the numbers of coarse iterations needed to get a normalized residual below 0.6 with respect to the FGMRES iterations. It can be seen that the number of coarse iteration is oscillating between 50 and 300. In average, 95 coarse iterations are performed for each FGMRES iteration. Then we fix the number of coarse iterations to 100 per FGMRES iterations. It can be seen in Table 3.8 that the number of iterations needed to converge for FGMRES is in the range of the best one of Tables 3.7 while a constant number of iterations on the coarse level is done. This is also confirmed when plotting the eigenvalues of H_{m+1}, on the 512^3 grid $\Lambda(H_{m+1})$ real parts are greater than 0.2 and and on the 256^3 grid $\Lambda(H_{m+1})$ real parts are greater than 0.5. Therefore, we will follow this fixed coarse iteration strategy in order to yield a cycle with fixed computational work per preconditioning step.

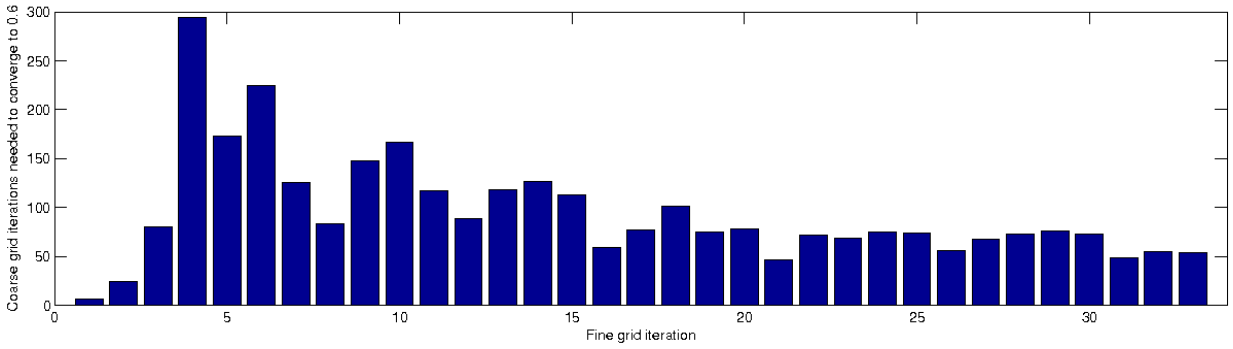

Figure 3.19: Number of iterations needed by GMRES(10) preconditioned by a reverse symmetric Gauss-Seidel cycle to converge to 0.6 with respect to the FGMRES(5) current iteration.

Grid	k	# Cores	$m = 5$
512^3	268.08	1024	32
256^3	134.04	128	18

Table 3.8: Number of iterations of FGMRES(5) required to reach 10^{-6} performing 100 iterations of preconditioned GMRES(10) on the coarse level at each iteration of FGMRES(5) for two wavenumbers.

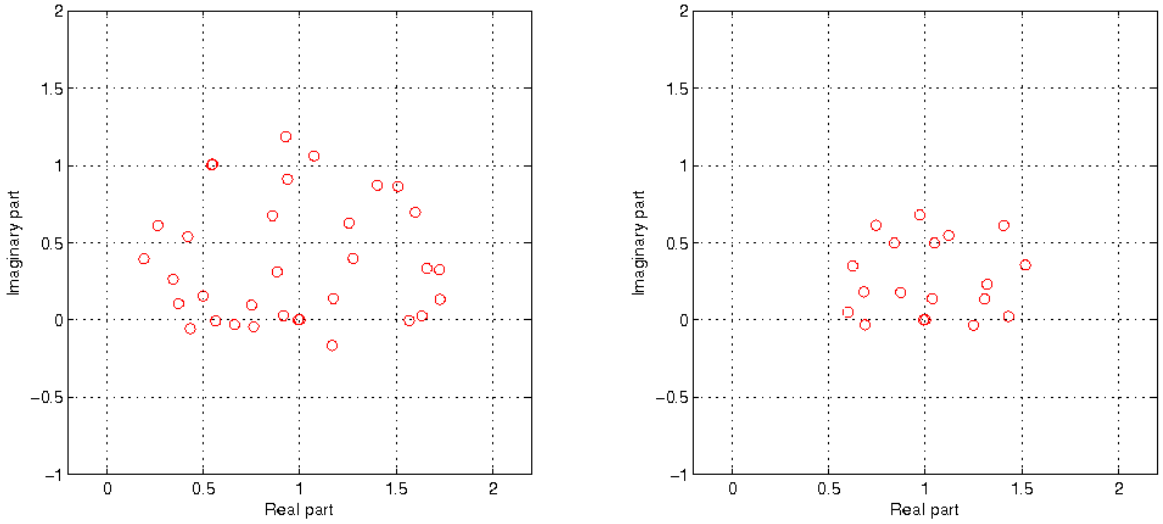

Figure 3.20: From right to left: $\Lambda(H_{m+1})$ spectrum using 100 coarse iterations of GMRES(10) preconditioned by a reverse symmetric Gauss cycle for a 512^3 grid ($k = 268.08$) and a 256^3 grid ($k = 134.04$) to converge to 10^{-6} with FGMRES(5).

3.6 Conclusions

In this chapter, we proposed a three-dimensional multilevel preconditioner for the solution of the three-dimensional Helmholtz equation with PML. Keeping in mind [42], we tried to extend to three dimensions a multigrid preconditioner acting on a shifted Helmholtz operator. This preconditioner is excepted to work provided that the right smoother acts in three dimensions. Yet the choice of the shift parameter is an open question that can only be solved, from our point of view, by a trial and error procedure. Since this choice strongly depends on the multigrid operator components, the discretization of the Helmholtz operator itself and its homogeneity/heterogeneity (constant/variable velocity in the physical domain), we have decided to focus on a preconditioner acting directly on the original Helmholtz operator. This strategy implies to use a restricted number of levels (two).

We have then designed a perturbed two-grid preconditioner for three-dimensional Helmholtz problems. Its principle relies on two remarkable phenomena:

- in a standard two-grid cycle (Algorithm 15), the coarse solution is not required to be exact to obtain a two-grid method as efficient as when the coarse solution is exact. Moreover, the two-grid method behaves well even if the convergence threshold is very large, say about 0.1.
- Gauss-Seidel type methods are efficient to smooth error on the fine grid for three-dimensional Helmholtz problems at high wavenumbers. Furthermore, in a parallel environment, local Gauss-Seidel methods can be further improved by a Krylov accelerator such as GMRES.

According to the spectrum analysis of Section 3.5, using the two-level method as a preconditioner for the original Helmholtz operator is relevant. According to Table 3.7, the two-level preconditioner combined with FGMRES requires a reasonable number of iterations even at large wavenumbers. However, its numerical efficiency may not imply its computational efficiency. The next chapter is devoted to numerical experiments on parallel computers. The computational efficiency of our perturbed two-grid preconditioner will be shown both for homogeneous test cases (constant propagation velocity in the physical domain) and heterogeneous ones (variable propagation velocity).

Chapter 4

Numerical experiments - Applications to geophysics

4.1 Introduction

In this chapter we evaluate the efficiency of the perturbed two-level preconditioner proposed in Chapter 3 for solving three-dimensional Helmholtz problems occurring in geophysics. This evaluation will be performed for both homogeneous and heterogeneous media in a single and multiple right-hand side situation. Since the linear systems arising from the discretization of this Helmholtz operator are very large (see Appendix A), the methods presented in Chapters 2 and 3 have to be implemented in a parallel memory distributed environment. We refer to [47] and [48] for the parallel implementation of GMRES and FGMRES respectively and to Chapter 6 in [115] for the parallel implementation of the perturbed two-level preconditioner on structured grids.

Denoting by $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$, $X \in \mathbb{C}^{n\times p}$, the matrix of the linear system, the right-hand side and the solution respectively, we focus on preconditioned iterative methods for the solution of

$$AX = B,$$

with a zero initial iterate. The iterative procedures are stopped when the Euclidean norm of each column of the block residual normalized by the Euclidean norm of the corresponding right-hand side satisfies the following relation in the 2-norm:

$$\frac{\|B(:,l) - AX(:,l)\|_2}{\|B(:,l)\|_2} \leq 10^{-5}, \ \forall\, l = 1,\ldots,p, \tag{4.1}$$

The tolerance is set to 10^{-5} so as to use the same stopping criterion in both single and double precision arithmetic.

First we present numerical experiments for the single right-hand side situation ($p = 1$). The FGMRES(5) method (Algorithm 3 of Chapter 2) preconditioned by one cycle of the perturbed two-grid method (Algorithm 16 of Chapter 3) is used on both homogeneous and heterogeneous problems. Several wavenumbers - from moderate to huge - will be considered. When variable velocity fields are considered, the resulting pressure fields are plotted versus the considered frequencies.

Secondly we consider the case of multiple right-hand sides ($p > 1$). Once again one cycle of the perturbed two-grid algorithm (Algorithm 16) is used as a preconditioner. The flexible block methods presented in Section 2.6 will be evaluated only in the case of heterogeneous media. Several numerical tests on public domain model problems will help us to determine the best strategy when solving such linear systems.

4.2 Three-dimensional homogeneous Helmholtz problems with a single right-hand side

In this section we present numerical experiments that we have performed during the PRACE (Partnership for Advanced Computing in Europe) petascale summer school in Stockholm, Sweden during the last week of August 2008[1]. Two parallel computers were available: a Cray XT4 located in Espoo (Finland) and a IBM Blue Gene/P located in Jülich (Germany).

We are focusing on two sets of experiments. The first set called *weak scalability* experiments consists in increasing the global size of the problem proportionally to the number of cores keeping the size of the local problem on each core fixed. The second set called *strong scalability* experiments consists in increasing the number of cores keeping the size of the global problem fixed. We are also interested in investigating the behavior of the algorithms in single and double precision arithmetic. Indeed geophysical computations are often performed in single precision [89]. For all these experiments the algorithm used is FGMRES(5) preconditioned by a two-grid cycle (Algorithm 16) with GMRES(2) preconditioned by a local symmetric Gauss-Seidel iteration as a smoother and 100 iterations of GMRES(10) preconditioned by a local symmetric Gauss-Seidel cycle as an approximate coarse solver. For all these experiments the algorithm is stopped when Relation 4.1 is satisfied with $p = 1$. The right-hand side b, representing the wave source S, is resulting from the discretization of a Kronecker function $\delta_{(x_\delta, y_\delta, z_\delta)}$ where the source position is located at:

$$b = \delta_{(n_x/2, n_y/2, n_{PML}+1)},$$

where n_x, n_y are the number of points in the $x-$ and $y-$directions respectively and n_{PML} the number of points in the absorbing layer ($n_{PML} = 16$, see Appendix A). The PML is located inside the physical grid and the wavenumbers k are selected such that they satisfy the stability condition (Relation A.4):

$$k = \frac{\pi}{6h}.$$

The actual memory M allocated in our code is given with the following formula (in Gigabytes (GB)):

$$M = \sum_{c=1}^{\#Cores} n_{loc}(c) \times \left[6 + (2m_f + 1) + (m_s + 2) + \frac{(m_g + 2) + 3}{8} \right] \times \frac{\vartheta}{1024^3} \tag{4.2}$$

where $n_{loc}(c)$ is the local problem size on the core c, m_f the restart parameter of FGMRES, m_s the restart parameter of the smoother, m_g the restart parameter of the coarse solver and ϑ the memory required to store a number in the considered arithmetic precision. It has to be noticed that the numbers 6 and $3/8$ in Equation 4.2 are related to the storage of the solution, right-hand side, work arrays and the diagonal of the Helmholtz matrix on the fine and coarse grids respectively. The other matrix diagonals do not need to be stored in our matrix-free implementation of matrix-vector products and Gauss-Seidel procedures. First we present numerical experiments performed on a Cray XT4 related to weak and strong scalability experiments and a comparison between single and double precision algorithms. Then we will present results on larger problems performed on the Blue Gene/P focusing on the single precision arithmetic only. In the following tables h denotes the mesh grid size, k the wavenumber, Grid the number of points of the problem and their repartition per direction, # Cores the total number of cores, Partition the repartition of the cores per direction, T and It the total elapsed time and the number of iterations respectively, T/It the time per iteration and M the total memory cost of the algorithm in GB (see Relation 4.2).

4.2.1 PRACE experiments: Cray XT4 at Espoo (Finland)

Cray XT4 Louhi

The Cray XT4 Louhi [2] consists of 1012 quad-core AMD Opteron Barcelona processors with 1 GB of memory per core. The clock rate of these processors is 2.3 GHz. Each node (a single quad-core processor) has 4 GB memory (1 GB/core). The Cray SeaStar interconnect system directly connects all nodes in a

[1] http://www.pdc.kth.se/education/historical/2008/PRACE-P2S2
[2] http://www.csc.fi/english/pages/louhi_guide/index_html

three-dimensional torus topology using Hyper-Transport links of Opteron processors. On this machine, our Fortran 90 code has been compiled with the Portland compiler suite with "-O3 -fastsse" options and linked with the ACML library (AMD Core Math Library).

Weak scalability experiments in single and double precision arithmetic

Cray XT4 Louhi											
Weak scalability experiments in single and double precision arithmetic											
h	k	Grid	# Cores	T(s)		It		T/It		M(GB)	
				dp	sp	dp	sp	dp	sp	dp	sp
1/256	134.04	256^3	8	418	250	13	13	32.13	19.26	5.60	2.80
1/512	268.08	512^3	64	888	546	25	26	35.53	21.00	44.78	22.39
1/1024	536.16	1024^3	512	1600	1129	48	48	33.33	23.52	358.28	179.14

Table 4.1: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the homogeneous model problem with wavenumber k such that $k\,h = \pi/6$. The results are shown for both single precision (sp) and double precision (dp) arithmetic.

In Table 4.1, it is found that the number of iterations behaves linearly with the wavenumber k in both single and double precision arithmetic; this is illustrated in Figure 4.1. This behavior has also been observed in the literature for other multilevel strategies [14, 96], when addressing smaller problem sizes although.

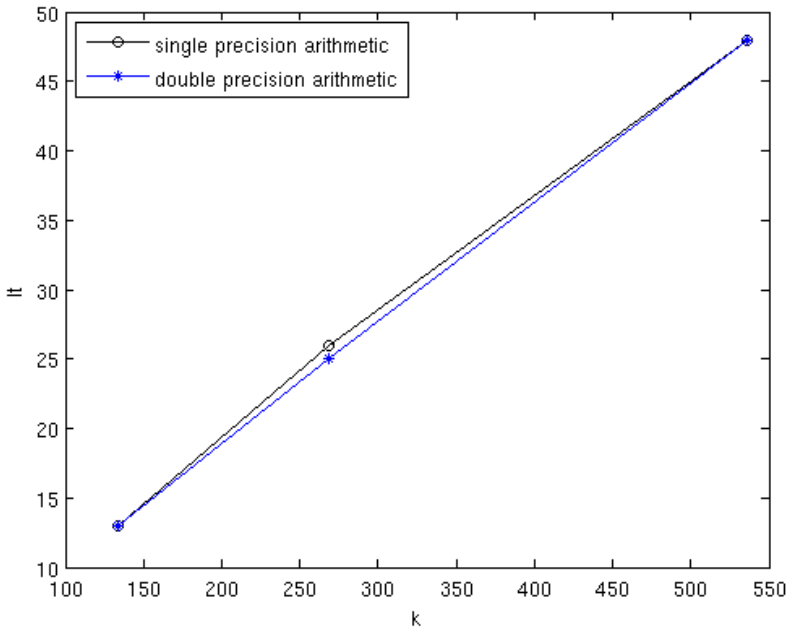

Figure 4.1: Number of iterations (It) of Table 4.1 for both single and double precision arithmetic with respect to the wavenumber k.

As expected, using single precision arithmetic leads to a reduction by a factor of two in the memory requirements, as shown in Table 4.1. The time per iteration is found to be almost constant indicating a good load-balancing property.

Strong scalability experiments in single precision arithmetic

Cray XT4 Louhi									
Strong scalability experiments in single precision arithmetic									
h	k	Grid	# Cores	Partition	T(s)	It	T/It	τ	M(GB)
1/1024	536.16	1024^3	256	$4 \times 8 \times 8$	2175	47	46.28	1.00	178
1/1024	536.16	1024^3	512	$8 \times 8 \times 8$	1171	48	24.40	0.93	179
1/1024	536.16	1024^3	1024	$8 \times 8 \times 16$	447	48	9.31	1.23	182

Table 4.2: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the homogeneous model problem with wavenumber k such that $k\,h = \pi/6$. $\tau = \frac{T_{ref}}{T} / \frac{P}{P_{ref}}$ is a scaled speed-up where T, P denote the elapsed time and number of cores on a given experiment respectively.

Since it has been remarked in the previous section that the codes in single and double precision arithmetic behave numerically similarly, we only focus on single precision arithmetic for these strong scalability experiments. In this table the global problem size (1024^3) is kept constant, whereas the number of cores is multiplied by a factor of two from one row to the other. The number of required iterations is found to be quite constant. The differences can be explained by the fact that the preconditioning method used in both the smoother and the coarse solver are local and thus depends on the number of cores and their repartition. We note that the memory requirement depends on the number of cores too (indeed when more cores are used, more local boundaries and overlapping zones have to be stored). The τ parameter is an indicator of the scalability of the algorithm: if $\tau = 1$, the code perfectly scales. Taking as T_{ref} and P_{ref}, the time and the number of cores corresponding to the first experiment (2175 seconds and 256 cores respectively), it appears that the code scales quite well. The behavior $\tau = 1.23$ for the last experiment is probably due to cache effects and would deserve further investigation.

4.2.2 PRACE experiments: IBM Blue Gene/P at Jülich (Germany)

IBM Blue Gene/P Jugene

The IBM Blue Gene/P Jugene [3] consists of 72 racks, each one containing 1024 nodes with 2 GB of memory per node. A node is made of 4 computing cores running at 850 MHz (32-bit PowerPC 450). The interconnect system directly connects all nodes in a three-dimensional torus topology.

On this machine, following the constructor recommandation, our Fortran 90 code has been compiled with IBM native compilers with "-O3 -g -qmaxmem=-1 -qarch=450 -qtune=450" options and linked with the ESSL library. The virtual node execution mode has been chosen (4 MPI processes per node with 512 MB as maximum memory per MPI process). The mapping used is of MESH type.

Thanks to the availability of this machine, we have been allowed to run large test cases, using up to 65536 cores (the whole machine at that time, August 2008) to solve a linear system of size larger than 68 billion. All the numerical experiments have been performed in single precision arithmetic and are summarized in Tables 4.3 and 4.4 .

Weak scalability experiments in single precision arithmetic

The behavior of the first two experiments (see the first two rows) is similar as in Table 4.1, the number of iterations doubles when the wavenumber is multiplied by a factor of two. However, when $k = 2144.66$, the number of iterations is nearly three times the number of iterations when $k = 1072.33$. The problem size (4096^3) and the large value of the wavenumber could explain this behavior. Despite this increase in iterations, the algorithm still scales quite well: the time per iteration is found to be constant (about 29 seconds) when the ratio between the size of the global problem and the number of cores is kept constant.

[3] http://www.fz-juelich.de/jsc/jugene

Blue Gene/P Jugene								
Weak scalability experiments in single precision arithmetic								
h	k	Grid	# Cores	Partition	T(s)	It	T/It	M(GB)
1/1024	536.16	1024^3	1024	$8 \times 8 \times 16$	1396	48	29.08	182
1/2048	1072.33	2048^3	8192	$16 \times 16 \times 32$	2987	102	29.28	1455
1/4096	2144.66	4096^3	65536	$32 \times 32 \times 64$	8308	282	29.46	11641

Table 4.3: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the homogeneous model problem with wavenumber k such that $k\,h = \pi/6$.

Strong scalability experiments in single precision arithmetic

Blue Gene/P Jugene									
Strong scalability experiments in single precision arithmetic									
h	k	Grid	# Cores	Partition	T(s)	It	T/It	τ	M(GB)
1/2048	1072.33	2048^3	4096	$16 \times 16 \times 16$	6321	105	60.20	1.00	1433
1/2048	1072.33	2048^3	8192	$16 \times 16 \times 32$	2987	102	29.28	1.06	1455
1/2048	1072.33	2048^3	16384	$16 \times 32 \times 32$	1427	103	13.85	1.11	1478
1/2048	1072.33	2048^3	32768	$32 \times 32 \times 32$	650	105	6.19	1.22	1500

Table 4.4: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the homogeneous model problem with wavenumber k such that $k\,h = \pi/6$. $\tau = \frac{T_{ref}}{T} / \frac{P}{P_{ref}}$ is a scaled speed-up where T, P denote the elapsed time and corresponding number of cores on a given experiment respectively.

Most of the comments related to Table 4.2 do apply to Table 4.4 as well. The number of iterations is slightly varying because of the local nature of both the smoothers and the preconditioner used in the coarse solver. It can be noticed that the elapsed time is divided by more than a factor of two from one row to the next. Indeed the τ parameter is always larger than one and increases with the number of cores. It seems to be due to cache effects once again.

4.3 Three-dimensional heterogeneous Helmholtz problems with a single right-hand side

In this section we present numerical experiments for two variable velocity fields that are publicly available: the SEG/EAGE Salt dome and the SEG/EAGE Overthrust models [5] defined in a domain of size $L_x \times L_y \times L_z$ (m^3). The solution method is the same as in the homogeneous case (Section 4.2). The source is also located at the center of the (x, y) plane below the PML layer. Contrary to the homogeneous case, we now fix the frequency f in Hz and deduce the mesh grid size h in m according to Relation A.3:

$$h = \frac{\min_{(x,y,z)\in\Omega_h} v(x,y,z)}{12\,f}. \tag{4.3}$$

Furthermore, the PML layers (see Appendix A) are added around the physical domain ($n_{PML} = 16$). This implies the following grid partition:

$$\left[\frac{12\,f\,L_x}{\min_{(x,y,z)\in\Omega_h} v(x,y,z)} + 32, \frac{12\,f\,L_y}{\min_{(x,y,z)\in\Omega_h} v(x,y,z)} + 32, \frac{12\,f\,L_z}{\min_{(x,y,z)\in\Omega_h} v(x,y,z)} + 32 \right].$$

Consequently the ratio between the number of unknowns will not be proportional to the ratio of corresponding frequencies indicated later by the "Grid ratio" value.

Yet we are still focusing on strong and weak scalability experiments in single precision arithmetic with a number of cores proportional to the frequency f. First we present experiments for the SEG/EAGE Salt

dome velocity field. In the following tables, h is the mesh grid size in m, f the frequency in Hz, Grid the number of points and their repartition per direction ($n_x \times n_y \times n_z$), Grid ratio the ratio between the grid size in the current line and the grid size in the preceding line, # Cores the number of cores, Partition the repartition of the cores per direction, T and It the elapsed time and the number of iterations, T/It the time per iteration and M the total memory requested in GB.

IBM Blue Gene/P Babel

All the numerical experiments in this section have been performed on the IBM Blue Gene/P Babel at IDRIS in Orsay (France) [4]. The Babel machine is a IBM Blue Gene/P system. It consists of 10 racks, each one containing 1024 nodes with 2 GB of memory per node. A node has 4 computing cores running at 850 MHz (32-bit PowerPC 450). The interconnect system directly connects all nodes in a three-dimensional torus topology.

On this machine, our Fortran 90 code has been compiled with IBM native compiler with "-O3 -qhot -qarch=450 -qtune=450" options and linked with the ESSL library. The virtual node execution mode has been chosen (4 MPI processes per node with 512 MB as maximum memory per MPI process). The mapping used is of MESH type.

The iterative procedure is stopped when Relation (4.1) is satisfied for $p = 1$.

4.3.1 SEG/EAGE Salt dome model problem

The SEG/EAGE Salt dome model [5] is a velocity field containing a salt dome in a sedimentary embankment. It is a parallelepiped domain of size $13.5 \times 13.5 \times 4$ km^3. The minimum value of the velocity is 1500 $m.s^{-1}$ and its maximum value is 4481 $m.s^{-1}$ respectively. The whole velocity field has been considered here.

Weak scalability experiments in single precision arithmetic

Salt dome - Blue Gene/P Babel									
Weak scalability experiments in single precision arithmetic									
h(m)	f(Hz)	Grid	Grid ratio	# Cores	Partition	T(s)	It	T/It	M(GB)
50	2.5	$301 \times 301 \times 115$	1	32	$4 \times 4 \times 2$	70	11	6.36	1.8
25	5	$571 \times 571 \times 199$	6.22	256	$8 \times 8 \times 4$	119	25	4.76	11
12.5	10	$1112 \times 1112 \times 367$	6.99	2048	$16 \times 16 \times 8$	270	62	4.35	80
6.25	20	$2200 \times 2200 \times 709$	7.56	16384	$32 \times 32 \times 16$	1081	257	4.20	605

Table 4.5: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model with mesh grid size h such that $h = \min_{(x,y,z)\in\Omega_h} v(x,y,z)/(12\,f)$. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

In Table 4.5 we remark that as in the homogeneous case, in the first three rows, when the frequency is multiplied by a factor of two from one line to the next, the number of FGMRES(5) iterations is also multiplied by a factor close to two. However, in the case of $f = 20$ Hz, the number of iterations greatly increases; it is about four times the number of iterations required to solve the Helmholtz problem at $f = 10$ Hz. We also remark that the time per iteration decreases when the frequency increases. Indeed the Grid ratio is always smaller than the ratio between the numbers of cores (8).

The real part of the solutions and the velocity fields at these four frequencies are plotted in Figures 4.2, 4.3, 4.4 and 4.5 respectively. Two different plots are shown: first the contour of the real part of the solution is plotted next to the contour of the velocity field, then a section of the real part of the solution in the plane (x, y) for $y = hn_y/2$ is plotted next to the corresponding section of the velocity field. In these figures we

[4] http://www.idris.fr/

observe the propagation of the wave and the position of the source. We also note that the variations in the pressure field due to the heterogeneity of the media clearly appear.

Strong scalability experiments in single precision arithmetic

Salt dome - Blue Gene/P Babel									
Strong scalability experiments in single precision arithmetic									
h(m)	f(Hz)	Grid	# Cores	Partition	T(s)	It	T/It	τ	M(GB)
12.5	10	$1112 \times 1112 \times 367$	256	$4 \times 8 \times 8$	2017	51	39.55	1.00	76
12.5	10	$1112 \times 1112 \times 367$	512	$8 \times 8 \times 8$	932	53	17.58	1.08	78
12.5	10	$1112 \times 1112 \times 367$	1024	$16 \times 8 \times 8$	469	57	8.22	1.08	79
12.5	10	$1112 \times 1112 \times 367$	2048	$16 \times 16 \times 8$	270	62	4.35	0.93	80
12.5	10	$1112 \times 1112 \times 367$	4096	$16 \times 16 \times 16$	171	87	1.97	0.74	83
12.5	10	$1112 \times 1112 \times 367$	8192	$16 \times 32 \times 16$	129	117	1.10	0.49	85
12.5	10	$1112 \times 1112 \times 367$	16384	$32 \times 32 \times 16$	78	136	0.57	0.40	88

Table 4.6: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model with mesh grid size h such that $h = \min_{(x,y,z)\in\Omega_h} v(x,y,z)/(12\,f)$. The parameter T denotes the total computational time, It the number of preconditioner applications and M the memory. $\tau = \frac{T_{ref}}{T} / \frac{P}{P_{ref}}$ is a scaled speed-up where T, P denote the elapsed time and corresponding number of cores on a given experiment respectively.

First it can be noticed in Table 4.6 that the method does not scale as well as in the homogeneous case. In fact, from one row to the next, the number of iterations increases especially when large numbers of cores (8192, 16384) are considered. This phenomenon is probably due to the nature of the preconditioner of both the coarse solution method and smoothers; the heterogeneity of the medium badly influences its efficiency. Indeed, when more iterations are performed on the coarse level (see Table 4.7 where 200 coarse iterations are imposed instead of 100), the number of outer iterations decreases and the method scales up to 2048 cores. In Table 4.7 it has to be noticed that in the case of 4096 cores, the number of iterations is significantly reduced compared to the results shown in Table 4.6. However it is still larger than the number of iterations on smaller numbers of cores (see first rows of Table 4.7).

Salt dome - Blue Gene/P Babel									
Strong scalability experiments in single precision arithmetic									
h(m)	f(Hz)	Grid	# Cores	Partition	T(s)	It	T/It	τ	M(GB)
12.5	10	$1112 \times 1112 \times 367$	256	$4 \times 8 \times 8$	2331	35	66.60	1.00	76
12.5	10	$1112 \times 1112 \times 367$	512	$8 \times 8 \times 8$	1028	35	29.37	1.13	78
12.5	10	$1112 \times 1112 \times 367$	1024	$16 \times 8 \times 8$	516	39	13.43	1.13	79
12.5	10	$1112 \times 1112 \times 367$	2048	$16 \times 16 \times 8$	284	40	7.10	1.03	80
12.5	10	$1112 \times 1112 \times 367$	4096	$16 \times 16 \times 16$	227	70	3.24	0.64	83

Table 4.7: Two-grid preconditioned Flexible GMRES(5) performing 200 coarse iterations per cycle for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model with mesh grid size h such that $h = \min_{(x,y,z)\in\Omega_h} v(x,y,z)/(12\,f)$. The parameter T denotes the total computational time, It the number of preconditioner applications and M the memory. $\tau = \frac{T_{ref}}{T} / \frac{P}{P_{ref}}$ is a scaled speed-up where T, P denote the elapsed time and corresponding number of cores on a given experiment respectively.

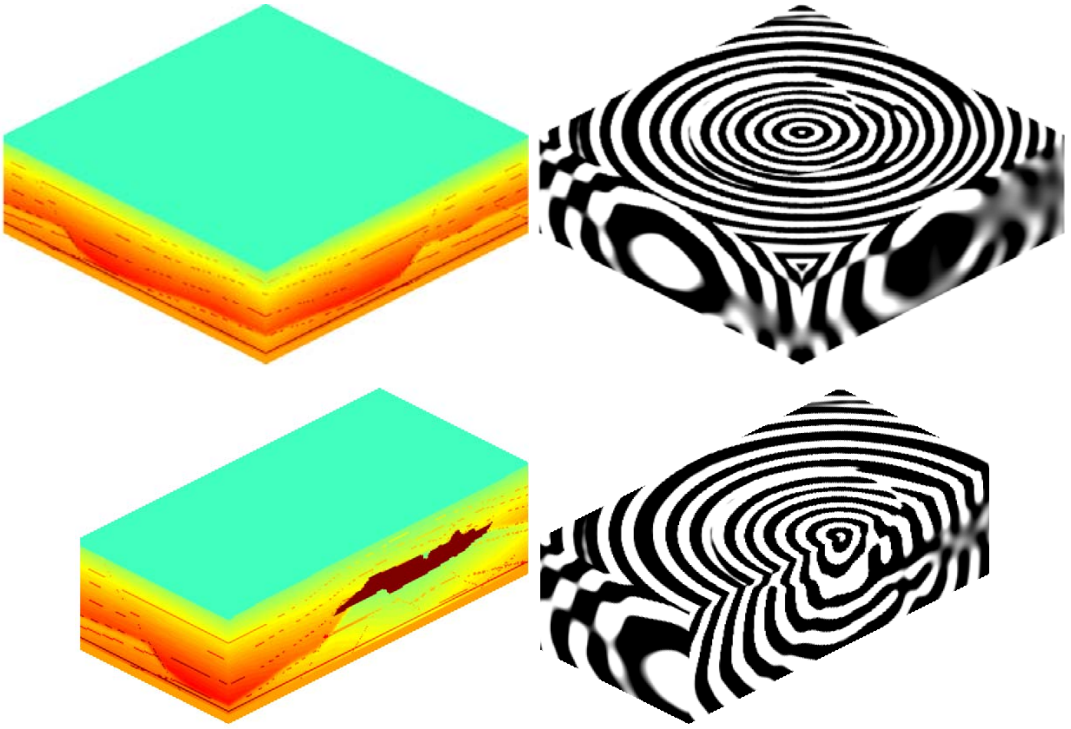

Figure 4.2: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 2.5Hz$ (right) and the SEG/EAGE Salt dome - velocity field (left).

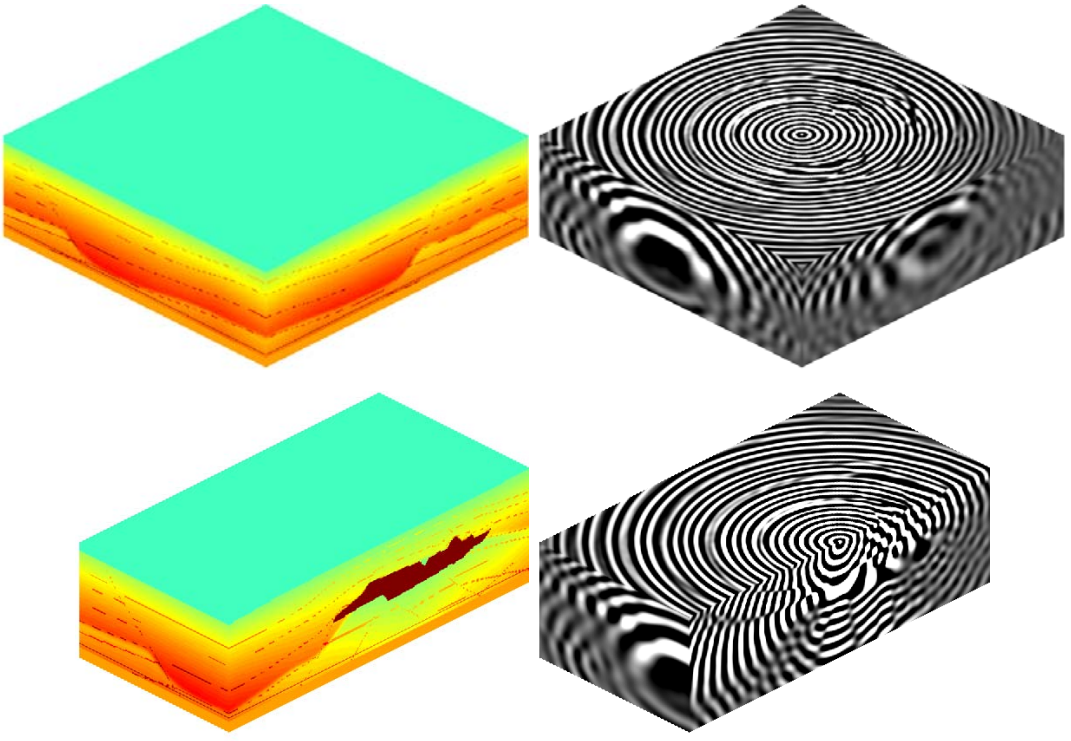

Figure 4.3: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 5Hz$ (right) and the SEG/EAGE Salt dome - velocity field (left).

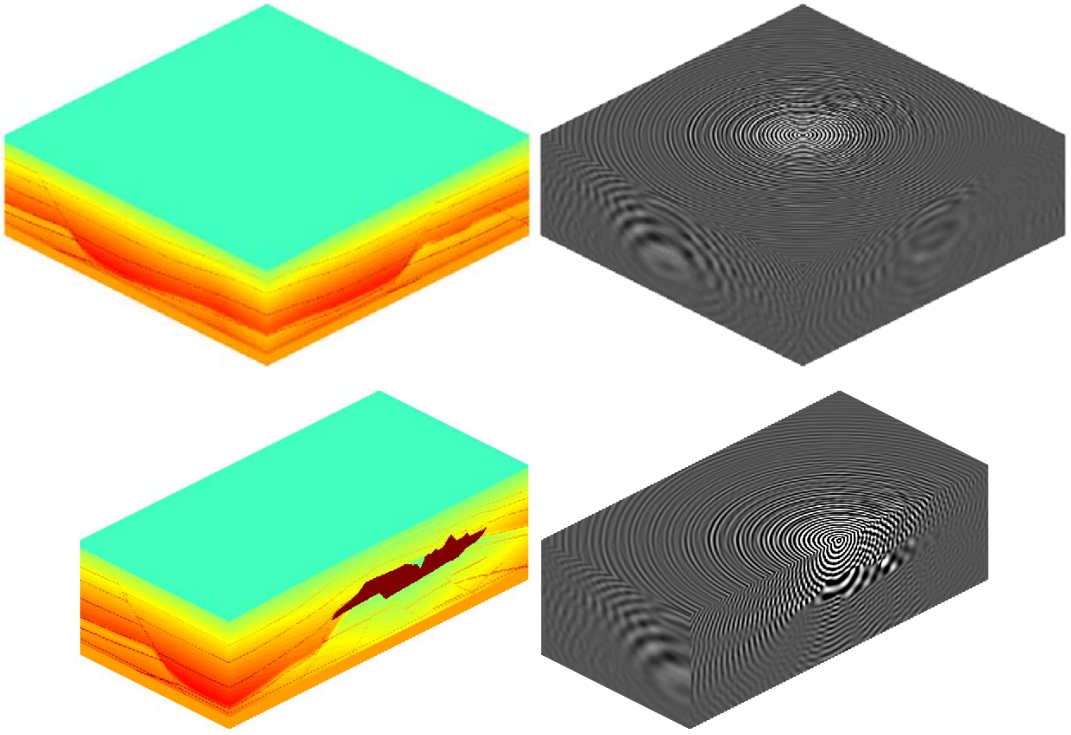

Figure 4.4: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 10Hz$ (right) and the SEG/EAGE Salt dome velocity field (left).

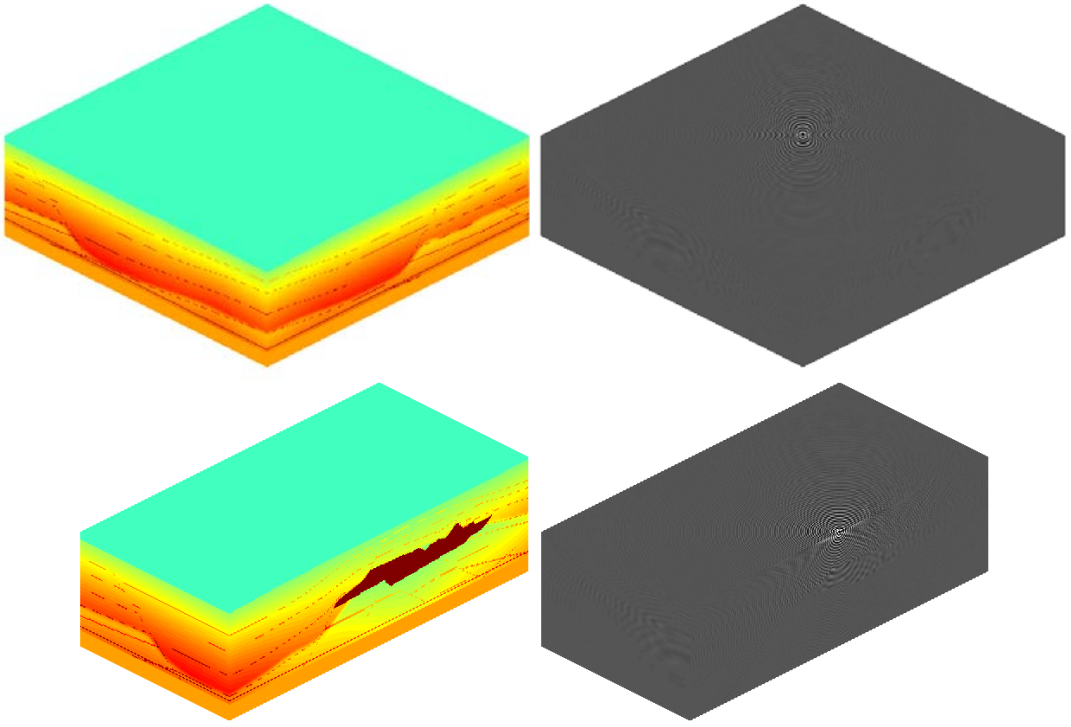

Figure 4.5: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 20Hz$ (right) and the SEG/EAGE Salt dome velocity field (left).

4.3.2 SEG/EAGE Overthrust model problem

The SEG/EAGE Overthrust model [5] is a synthetic velocity field. It is a parallelepiped domain of size $20\times20\times4.65\ km^3$. The minimum value of the velocity is 2179 $m.s^{-1}$ and the maximum value is 6000 $m.s^{-1}$ respectively. The whole velocity field has been considered here.

Weak scalability experiments in single precision arithmetic

Overthrust - Blue Gene/P Babel									
Weak scalability experiments in single precision arithmetic									
h(m)	f(Hz)	Grid	Grid ratio	# Cores	Partition	T(s)	It	T/It	M(GB)
48.42	3.75	$446\times446\times130$	1	32	$4\times4\times2$	218	14	15.57	4
24.21	7.5	$863\times863\times231$	6.65	256	$8\times8\times4$	422	31	13.61	29
12.11	15	$1690\times1690\times426$	7.07	2048	$16\times16\times8$	1637	137	11.95	209
6.05	30	$3356\times3356\times829$	7.67	16384	$32\times32\times16$	6453	558	11.56	1604

Table 4.8: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the SEG/EAGE Overthrust model with mesh grid size h such that $h = \min_{(x,y,z)\in\Omega_h} v(x,y,z)/(12\,f)$. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

In Table 4.8 FGMRES(5) behaves in a very similar way as in Table 4.5. Initially the number of iterations of the method is doubling when the frequency is doubling. Then the number of iterations is greatly increasing when high frequencies are considered (see last two rows of Table 4.8). Once again, we observe that the method is not as efficient as in the case of homogeneous problems. However, it still converges even at a very high frequency (30 Hz).

The real parts of the solutions and the corresponding velocity fields at these four frequencies are plotted in Figures 4.6, 4.7, 4.8 and 4.9 respectively.

Strong scalability experiments in single precision arithmetic

Overthrust - Blue Gene/P Babel									
Strong scalability experiments in single precision arithmetic									
h(m)	f(Hz)	Grid	# Cores	Partition	T(s)	It	T/It	τ	M(GB)
12.11	15	$1690\times1690\times426$	512	$8\times8\times8$	4542	88	51.61	1.00	205
12.11	15	$1690\times1690\times426$	1024	$8\times16\times8$	2827	110	25.70	0.80	207
12.11	15	$1690\times1690\times426$	2048	$16\times16\times8$	1637	137	11.95	0.69	209
12.11	15	$1690\times1690\times426$	4096	$16\times16\times16$	852	144	5.91	0.67	216
12.11	15	$1690\times1690\times426$	8192	$16\times32\times16$	472	162	2.91	0.60	220
12.11	15	$1690\times1690\times426$	16384	$32\times32\times16$	260	183	1.42	0.55	224

Table 4.9: Two-grid preconditioned Flexible GMRES(5) for the solution of the Helmholtz equation for the SEG/EAGE Overthrust model with mesh grid size h such that $h = \min_{(x,y,z)\in\Omega_h} v(x,y,z)/(12\,f)$. The parameter T denotes the total computational time, It the number of preconditioner applications and M the memory. $\tau = \frac{T_{ref}}{T} / \frac{P}{P_{ref}}$ is a scaled speed-up where T denotes a computational time.

In Table 4.9 we remark that the method does not exhibit good scaling properties at this high frequency ($f = 15\ Hz$). Indeed as in Table 4.6, the method encounters difficulties due to the heterogeneous nature of the media; the convergence is damaged by the local nature of the preconditioner in both the smoother and the coarse grid solver. However, even if the number of iterations increases with respect to the number of cores, it is still reasonable for such a high frequency (about 150 iterations for a problem of size 1.2×10^9) .

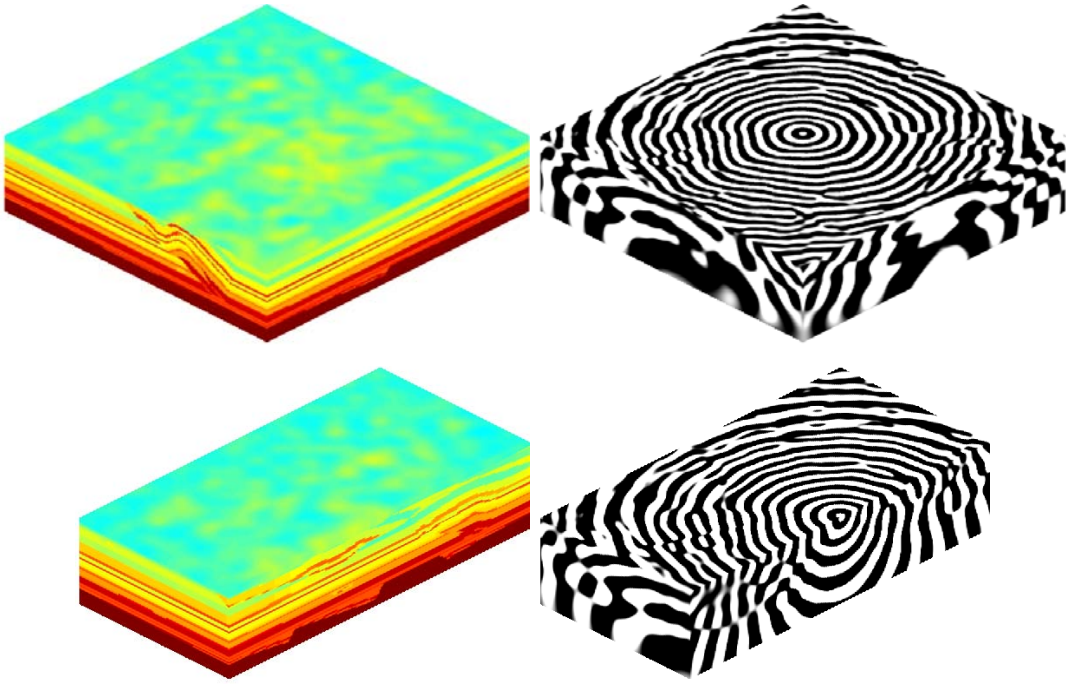

Figure 4.6: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 3.75Hz$ (right) and the SEG/EAGE Overthrust velocity field (left).

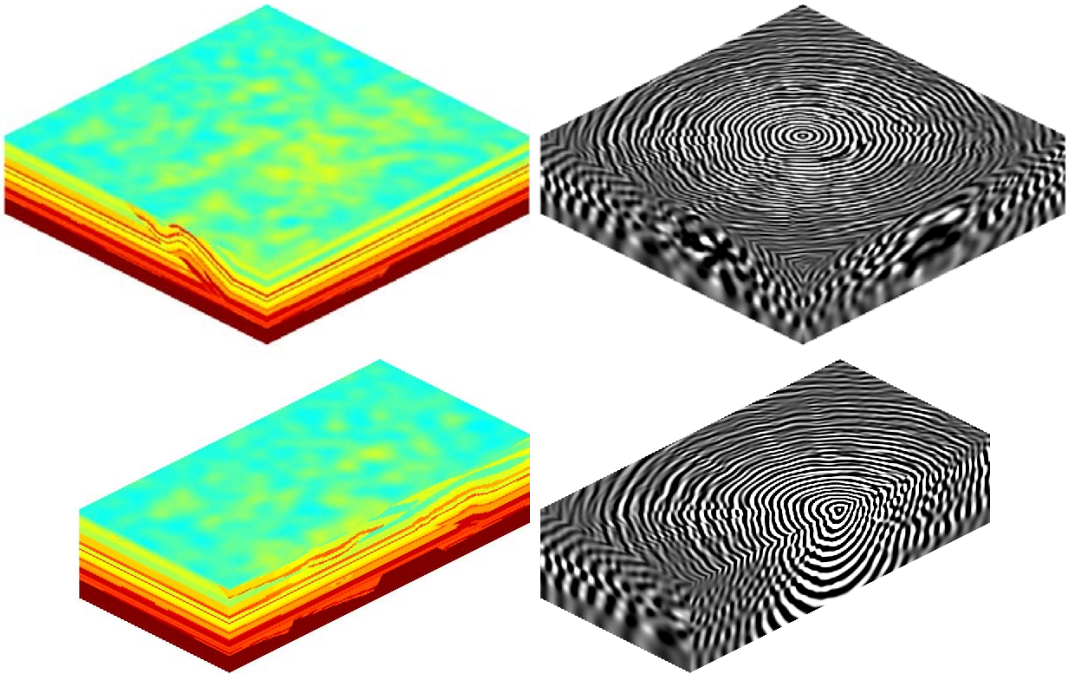

Figure 4.7: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 7.5Hz$ (right) and the SEG/EAGE Salt dome velocity field (left).

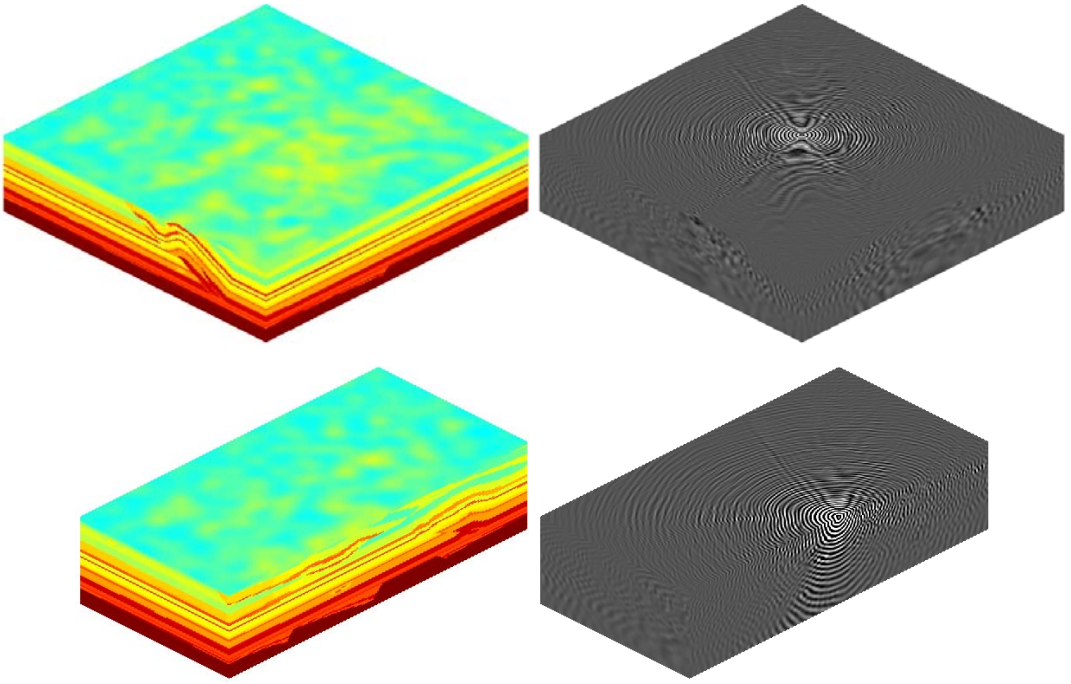

Figure 4.8: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 15Hz$ (right) and the SEG/EAGE Overthrust velocity field (left).

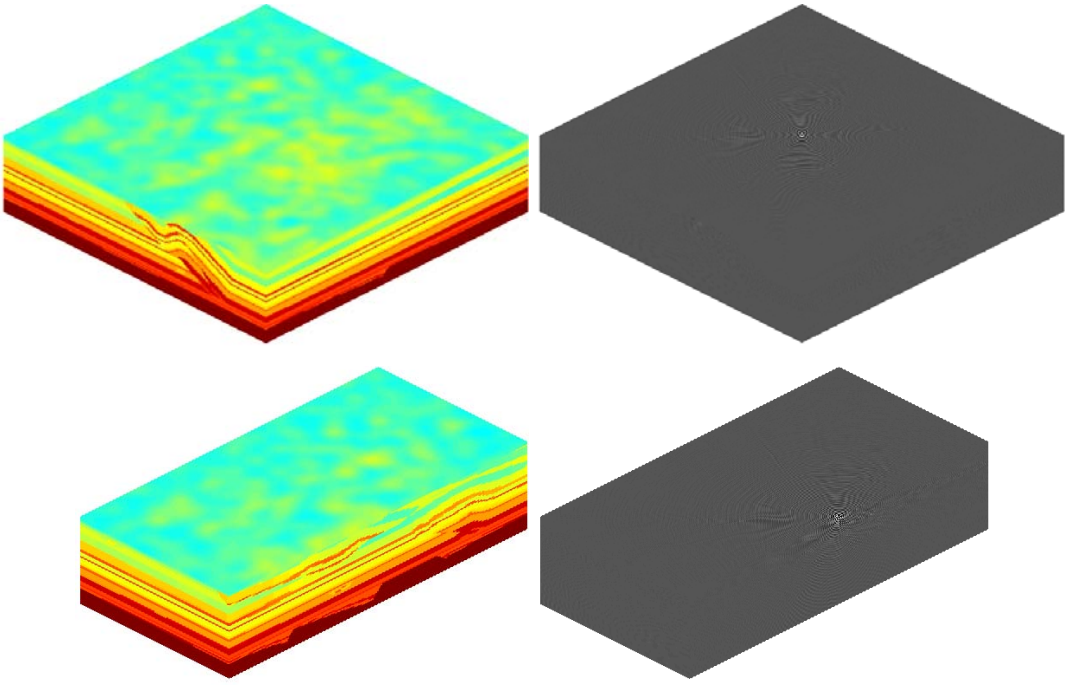

Figure 4.9: Contours and sections of the solution of the three-dimensional Helmholtz problem at $f = 30Hz$ (right) and the SEG/EAGE Overthrust velocity field (left).

Summary

In Sections 4.2 and 4.3 we have shown that the perturbed two-level preconditioner in combination with FGMRES(5) is able to solve Helmholtz problems at high wavenumbers on a large number of cores in both homogeneous and heterogeneous media. Nevertheless we noticed that the number of required iterations is increasing, first linearly with the wavenumber and, then more significantly when very high wavenumbers are considered. The method does then not scale in the sense of the weak scalability. Regarding the strong scalability, the method scales for homogeneous problems whereas it has difficulties to scale in the heterogeneous case. Indeed, when variable velocity fields are considered, a too large number of cores is deteriorating the efficiency of the method. It is due to the local nature of the preconditioner used in the coarse solver and smoothers (local lexicographical symmetric Gauss-Seidel). However the method starts to loose the strong scalability property when only a very large number of cores is considered. We note that such a situation has to be considered because the IBM Blue Gene/P machine has only 512 MB per core. If a computer with more memory per core were considered, the method would need less cores and it would scale better in the strong sense. This will be investigated in the near future.

In Section 4.4 we present next some numerical experiments in the multiple right-hand side context for both SEG/EAGE Salt dome and SEG/EAGE Overthrust model problems.

4.4 Three-dimensional heterogeneous Helmholtz problems with multiple right-hand sides

In this section we focus on solving Helmholtz problems for multiple sources given at once. In fact to manage multiple sources amounts to solve a linear system with multiple right-hand sides. The most simple strategy to address this issue is to solve the linear systems for each right-hand side one after the other. However, in Section 2.6, we developed block Krylov methods allowing to solve linear systems with several right-hand sides simultaneously taking advantage of this situation. Thus we compare these methods when solving Helmholtz problems at few frequencies and for different numbers of right-hand sides ($p \neq 1$). We use the perturbed two-level preconditioner in block methods as in the single right-hand side situation; preconditioning is now applied on each right-hand side independently.

We consider five solution methods altogether. The first one, FGMRES(5) sequence, consists in solving the linear systems one after the other always with a zero initial guess. The second method, FGMRES(5) simultaneous, aims at solving the linear systems gathering matrix-vector products, dot products and MPI communications, applying FGMRES(5) to each linear system simultaneously but independently. This method aims at benefiting from possible computational speed-up obtained by gathering operations and minimizing memory transfers. The third method is Block FGMRES(5) (BFGMRES(5) Algorithm 10), the fourth one is Block FGMRES(5) with SVD based deflation (BFGMRESD(5) Algorithm 11) and the fifth one is Block FGMRES(5) with SVD based truncation (BFGMREST(5) Algorithm 12). In this last method we consider two fixed block sizes p_f ($p_f = p/2$ and $p_f = p/4$) where p denotes the total number of right-hand sides. All these methods are compared according to the number of iterations (applications of the preconditioner on a single vector) required to satisfy the stopping criterion (Relation 4.1). The stopping criterion used in block methods and FGMRES(5) simultaneous is similar (see Relation 4.1). Notwithstanding, the stopping criterion of FGMRES(5) sequence is the same as in Section 4.2 and 4.3 (Relation 4.1 for $p = 1$). Therefore, it should be not surprising if FGMRES(5) simultaneous converges using more iterations (application of the preconditioner) than FGMRES(5) sequence. Indeed, when FGMRES(5) simultaneous has converged for a specific right-hand side, the method is still considering all right-hand sides until the convergence is reached.

The Helmholtz operator is discretized as in Section 4.3. However, contrary to the single right-hand side case, we fix the mesh grid size h in m and deduce the frequency f in Hz according to Relation A.3:

$$f = \frac{\min_{(x,y,z)\in\Omega_h} v(x,y,z)}{12\,h}. \tag{4.4}$$

This choice is led by the fact that we want to locate sources each 50 meters, to fix the mesh grid h is then convenient. Thus the p sources are located below the PML layer on the line $y = hn_y/2$ each 50 meters along the x axis starting from $x = (n_{PML} + 1)h$:

$$B(:,l) = \delta_{\left(n_{PML}+1+(l-1)\frac{50}{h},\ \frac{n_y}{2},\ n_{PML}+1\right)},\ \forall\, l = 1,\ldots,p.$$

An estimation of the total memory cost M in these block methods can be obtained in GB with the following formula:

$$M = \sum_{c=1}^{\#Cores} n_{loc}(c) \times \left[1 + 2p + 3p_f + (2m_f + 1)p_f + (m_s + 2)p_f + \frac{(m_g + 4)p_f + 1}{8} \right] \times \frac{\vartheta}{1024^3} \tag{4.5}$$

where $n_{loc}(c)$ is the local problem size on the core c, m_f the restart parameter of the outer block method, m_s the restart parameter of the smoother, m_g the restart parameter of the coarse solver, p the total number of right-hand sides, p_f the size of the blocks in the outer Krylov method and ϑ the memory required to store a number in the considered arithmetic precision. Note that when FGMRES(5) simultaneous, BFGMRES(5) and BFGMRESD(5) are considered, we have $m_f = 5$ and $p_f = p$. Concerning FGMRES(5) simultaneous, we always have $p_f = p = 1$, and Equation (4.5) becomes Equation (4.2).

4.4.1 SEG/EAGE Salt dome model problem

In Tables 4.10, 4.11 and 4.12 respectively, six methods have been compared on fixed mesh grid sizes (50 m, 25 m, 12.5 m) and with different numbers of right-hand sides respectively. In fact, in each table, we fix the mesh grid size and consider three different numbers of right-hand sides ($p = 8$, 16, 32 respectively). Since doubling the number of right-hand sides nearly doubles the memory requirement of the block methods (see Equation 4.5), we also multiply the number of cores by a factor of two with respect to the number of right-hand sides. This aims at imposing the same memory constraint on each core for all the numerical experiments.

Salt dome - Blue Gene/P Babel									
Grid : $301 \times 301 \times 115$, $h = 50\ m$, $f = 2.5\ Hz$									
Number of right-hand sides (p)	$p = 8$, #Cores=32			$p = 16$, #Cores=64			$p = 32$, #Cores=128		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	91	533	1.8	194	555	1.9	490	679	1.9
FGMRES(5) simultaneous	112	656	14	224	627	28	608	820	58
BFGMRES(5)	104	622	14	224	651	28	544	788	58
BFGMRESD(5)	70	423	14	130	**384**	28	280	409	58
BFGMREST(5,p/2)	80	480	7.6	150	432	16	320	450	32
BFGMREST(5,p/4)	84	508	4.5	156	448	9.28	336	473	19

Table 4.10: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model and $f = 2.5\ Hz$ ($h = 50\ m$), with 8, 16 and 32 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

Salt dome - Blue Gene/P Babel									
Grid : $571 \times 571 \times 199$, $h = 25\ m$, $f = 5\ Hz$									
Number of right-hand sides (p)	$p = 8$, #Cores=256			$p = 16$, #Cores=512			$p = 32$, #Cores=1024		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	237	1097	11	501	1104	12	1305	1445	12
FGMRES(5) simultaneous	272	1266	87	592	1289	179	1536	1636	368
BFGMRES(5)	256	1218	87	544	1235	179	1376	1583	368
BFGMRESD(5)	155	753	87	285	**659**	179	635	742	368
BFGMREST(5,p/2)	172	804	48	345	768	99	715	794	202
BFGMREST(5,p/4)	196	932	28	388	863	59	856	948	120

Table 4.11: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model and $f = 5\ Hz$ ($h = 25\ m$), with 8, 16 and 32 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

Salt dome - Blue Gene/P Babel									
Grid : $1112 \times 1112 \times 367$, $h = 12.5\ m$, $f = 10\ Hz$									
Number of right-hand sides (p)	$p = 8$, #Cores=2048			$p = 16$, #Cores=4096			$p = 32$, #Cores=8192		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	653	2841	80	1785	3534	83	5255	5065	85
FGMRES(5) simultaneous	688	2900	609	1920	3674	1265	6080	5531	2596
BFGMRES(5)	680	2927	609	1840	3675	1265	5536	5457	2596
BFGMRESD(5)	480	**2116**	609	1195	2445	1265	3180	3159	2596
BFGMREST(5,p/2)	536	2263	337	1320	2583	696	3590	3382	1428
BFGMREST(5,p/4)	564	2480	200	1504	2948	413	4440	4232	844

Table 4.12: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Salt dome model and $f = 10\ Hz$ ($h = 12.5\ m$), with 8, 16 and 32 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

In these three tables, whatever the number of right-hand sides p, it is noticeable that FGMRES(5) simultaneous never performs better than FGMRES(5) sequence in terms of iterations and elapsed times. This is due to the convergence detection used in FGMRES(5) simultaneous: it still iterates on a converged solution until all solutions have converged. Besides no significant computational speed-up can be obtained by gathering operations (matrix-vector products, dot products) on this problem. We also observe that BFGMRES(5) always needs more preconditioner applications to converge than FGMRES(5) sequence and requires of course an increased computational time since more operations are performed. However using more sophisticated block methods improves both the number of iterations and elapsed times. Indeed, BFGMRESD(5) always delivers the best numbers of iterations and the best elapsed times. Truncated methods, BFGMREST(5, $p/2$) and BFGMREST(5, $p/4$), also perform well; their elapsed times are close to those of BFGMRESD(5) and the corresponding numbers of iterations are always much smaller than those of FGMRES(5) sequence.

Comparing these methods according to the number of right-hand sides p, it is remarkable that the best result for each method is not obtained for the larger value of p. In fact, in Tables 4.10 and 4.11, (at $f = 2.5\ Hz$ and $f = 5\ Hz$ respectively), the best results are obtained for $p = 16$ whereas, in Table 4.12 ($f = 10\ Hz$) it is obtained for $p = 8$. Considering the number of iterations of FGMRES(5) sequence versus the parameter p, this phenomenon can be explained by the fact that FGMRES(5) does not require the same number of iterations for each p and number of cores. This is due to the lack of scalability of the method for a single right-hand side in a heterogeneous media that has been already observed (see Table 4.6). Nevertheless, it can be noticed that the deflated and truncated methods diminish the influence of the number of cores on the method. Indeed, the corresponding number of iterations and elapsed times are increasing from one column to the next with a smaller factor than those of FGMRES(5) sequence.

The previous tables do not put in light the qualities of block truncated methods. Indeed the purpose of *BFGMRES T* is to use significantly less memory when solving problems with several right-hand sides simultaneously. In Table 4.13, different truncated strategies are presented for a fixed number of right-hand sides (16). All these truncated methods have a fixed block size in their block Krylov subspaces ($p_f = 4$). We consider then three different strategies to solve such Helmholtz problems for a total number of 16 right-hand sides:

- four applications of BFGMRESD(5) (*BFGMREST(5,p)*) with $p = p_f = 4$,
- two applications of *BFGMREST*$(5, p/2)$ *with* $p = 8$ *and* $p_f = p/2 = 4$,
- *one application of BFGMREST*$(5, p/4)$ *with* $p = 16$ *and* $p_f = p/4 = 4$.

Salt dome - Blue Gene/P Babel											
Total number of right-hand sides: 16											
Frequency			$f = 2.5\ Hz$			$f = 5\ Hz$			$f = 10\ Hz$		
# Cores			32			256			2048		
p	# runs	Strategy	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
4	4	BFGMREST(5,p)	172	1038	7.0	400	1901	44	1135	4922	307
8	2	BFGMREST(5,p/2)	**156**	**930**	7.6	358	1678	48	1092	**4603**	337
16	1	BFGMREST(5,p/4)	**156**	943	9.0	**352**	**1667**	56	**1088**	4622	396

Table 4.13: Different strategies using BFGMREST preconditioned by a perturbed two-grid method in order to solve the Helmholtz equation for the SEG/EAGE Salt dome model with 16 right-hand sides at three frequencies. The parameter p denotes the number of right-hand side taken at once, # *runs* the number of times BFGMREST is used, T the computational time, It the number of iterations and M the requested memory.

In Table 4.13 the best strategy at each frequency is the truncation. However it is nearly equivalent to deal with all the right-hand sides at once or to split the right-hand sides into two groups. Thus, processing many right-hand sides at once may not be more efficient than processing less right-hand sides at once. Nevertheless, truncated methods are efficient in a memory constrained environment.

4.4.2 SEG/EAGE Overthrust model problem

In this section numerical experiments are conducted similarly as in Section 4.4.1. Indeed, in Tables 4.14, 4.15 and 4.16 respectively, six methods are compared on fixed mesh grid sizes (50 m, 25 m, 12.5 m) with different numbers of right-hand sides. In each table, we fix the mesh grid size and consider three different numbers of right-hand sides ($p = 4, 8, 16$ respectively).

Overthrust - Blue Gene/P Babel									
Grid : $446 \times 446 \times 130$, $h = 50\ m$, $f = 3.64\ Hz$									
Number of right-hand sides (p)	$p = 4$, #Cores=32			$p = 8$, #Cores=64			$p = 16$, #Cores=128		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	58	819	4.4	114	734	4.5	226	758	4.6
FGMRES(5) simultaneous	64	885	17	128	839	35	256	852	70
BFGMRES(5)	60	842	17	128	858	35	256	886	70
BFGMRESD(5)	45	645	17	77	526	35	134	**476**	70
BFGMREST(5,p/2)	52	743	9.4	86	576	19	161	554	39
BFGMREST(5,p/4)	53	767	5.6	94	633	11	168	567	23

Table 4.14: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Overthrust model and $f = 3.64\ Hz$ ($h = 50\ m$), with 4, 8 and 16 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

Overthrust - Blue Gene/P Babel									
Grid : $836 \times 836 \times 224$,$h = 25\ m$, $f = 7.27\ Hz$									
Number of right-hand sides (p)	$p = 4$, #Cores=256			$p = 8$, #Cores=512			$p = 16$, #Cores=1024		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	131	1569	29	248	1446	30	493	1482	31
FGMRES(5) simultaneous	140	1715	113	296	1734	232	608	1794	471
BFGMRES(5)	144	1784	113	296	1769	232	608	1861	471
BFGMRESD(5)	100	1278	113	165	1029	232	260	**855**	471
BFGMREST(5,p/2)	110	1361	63	182	1075	128	300	915	259
BFGMREST(5,p/4)	115	1411	38	204	1187	76	360	1060	154

Table 4.15: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Overthrust model and $f = 7.27\ Hz$ ($h = 25\ m$), with 4, 8 and 16 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

Overthrust - Blue Gene/P Babel									
Grid : $1637 \times 1637 \times 413$, $h = 12.5\ m$, $f = 14.53\ Hz$									
Number of right-hand sides (p)	$p = 4$, #Cores=2048			$p = 8$, #Cores=4096			$p = 16$, #Cores=8192		
Method	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
FGMRES(5) sequence	414	4214	209	975	4554	216	2000	4553	220
FGMRES(5) simultaneous	440	4782	802	1056	5208	1651	2192	5103	3351
BFGMRES(5)	440	4844	802	1048	5288	1651	2160	5248	3351
BFGMRESD(5)	325	3651	802	620	3254	1651	1105	**2802**	3351
BFGMREST(5,p/2)	348	3871	446	735	3641	912	1276	3027	1846
BFGMREST(5,p/4)	349	3666	267	772	3784	543	1580	3686	1094

Table 4.16: Perturbed two-grid preconditioned block methods for the solution of the Helmholtz equation for the SEG/EAGE Overthrust model and $f = 14.53\ Hz$ ($h = 12.5\ m$), with 4, 8 and 16 right-hand sides at once. The parameter T denotes the total computational time, It the number of preconditioner applications and M the requested memory.

Most of the comments related to Tables 4.10, 4.11 and 4.12 still remain available. The use of FGMRES(5) simultaneous and BFGMRES(5) neither improves the number of iterations nor the computational times, whereas *BFGMRESD* and *BFGMREST* perform well. In fact, once again, *BFGMRESD* delivers both the best number of iterations and computational times (see bold values). Moreover the results of *BFGMREST* methods are closer to those of *BFGMRESD* than in Section 4.4.1. Yet comparing the results from one column to the next, we observe that the best times for the deflated and truncated block methods are obtained when $p = 16$ whatever the frequency. Concerning FGMRES(5) sequence, the total number of iterations is growing nearly linearly with respect to the parameter p. Therefore the parallelism does not deteriorate the convergence of the method contrary to the previous model problem (Section 4.4.1).

As in Section 4.4.1 we evaluate now the *BFGMREST* method. We would like to determine the best strategy to solve such Helmholtz problems with a total of 8 right-hand sides with a fixed size of block in the block Krylov subspaces ($p_f = 2$). Numerical results are reported in Table 4.17.

Overthrust - Blue Gene/P Babel											
Total number of right-hand sides: 8											
Frequency			f = 3.64 Hz			f = 7.27 Hz			f = 14.53 Hz		
# Cores			32			256			2048		
p	# runs	Strategy	It	T(s)	M(GB)	It	T(s)	M(GB)	It	T(s)	M(GB)
2	4	BFGMREST(5,p)	112	1502	8.6	240	2840	57	760	8432	406
4	2	BFGMREST(5,p/2)	98	1315	9.4	218	2543	63	696	7579	446
8	1	BFGMREST(5,p/4)	**92**	**1247**	11.03	**198**	**2316**	74	**642**	**6984**	524

Table 4.17: Different strategies using BFGMREST preconditioned by a perturbed two-grid method in order to solve the Helmholtz equation for the SEG/EAGE Overthrust model with 8 right-hand sides at three frequencies. The parameter p denotes the number of right-hand side taken at once, # *runs* the number of times BFGMREST is launched, T the computational time, It the number of preconditioner applications and M the requested memory.

As in Table 4.13, truncated methods perform well. However in this test case, the best strategy is to handle all sources at once (BFGMREST(5, $p/4$)) (see bold values). Then truncated methods prove efficient for diminishing both the memory requirements and the computational times.

4.5 Conclusions

In this chapter we have shown computationally the relevance of the perturbed two-level preconditioner for the solution of three-dimensional Helmholtz problems in both homogeneous and heterogeneous media. This preconditioner was found efficient in the case of both single and multiple right-hand sides. In the homogeneous case the method scales in the strong sense up to 32768 cores on a 2048^3 grid. Up to this size the number of iterations is growing linearly with respect to the wavenumber k. Moreover it enables to solve huge homogeneous problems (up to 4096^3 unknowns) in a truly massively parallel environment (65536 cores). Nevertheless when heterogeneous media are considered, the method does not scale in the strong sense for a number of cores larger than 2048. The number of iterations of the method is significantly increasing at large frequencies (more than 10 Hz). However block Krylov methods partly overcome these issues. Indeed both BFGMRESD and BFGMREST (Algorithms 11 and 12 respectively) significantly improve the total number of iterations even for large numbers of cores at high frequencies. Besides the memory requirement of both the perturbed two-level preconditioners and Krylov methods can be explicitly evaluated. This is of great interest in a memory constraint environment.

Notwithstanding, we have mainly considered IBM Blue Gene/P computers (Jugene and Babel) which architecture consists of a huge number of cores together with a small amount of memory per core. Due to the dependency of the perturbed two-level preconditioner on the number of cores, especially in the heterogeneous case, the use of a computing platform with a larger amount of memory per core should be profitable and will be investigated in a near future.

Chapter 5

Conclusions

Solving large and indefinite linear systems, possibly with multiple right-hand sides, stemming from three-dimensional physical applications is a great challenge for preconditioned Krylov subspace methods. Even if the memory requirement of the Krylov method in itself is generally low and monitorable, they sometimes encounter difficulties to reach convergence in a reasonable number of iterations and therefore efficient preconditioners are required. Furthermore, an active research is currently performed in the iterative methods community for handling efficiently multiple right-hand sides problems, meaning that there is no standard approach to tackle this question. Seismic migration in geophysics gathers all these complications: it is a three-dimensional physical application whose solution requires to cope with large indefinite problems (Helmholtz problems) with multiple right-hand sides. Thus, since this was our target application, the aims of this thesis were originally twofold: to design an efficient preconditioner for three-dimensional heterogeneous Helmholtz problems and to extend a Krylov method to both a deflated restarting version and to an efficient block version.

Regarding the design of a good preconditioner for three-dimensional Helmholtz problems, we have focused on multi-level methods. The main difficulty we had then to face was the adaptation of multigrid techniques to indefinite problems. To overcome this difficulty, we have chosen to consider in our hierarchy a limited number of grids where the infinite dimensional problem is well represented. We have then designed a perturbed geometric two-level preconditioner where the coarse problem is solved only approximately. We managed to prove with a Fourier analysis that the coarse solution is not required to be exact to obtain an efficient two-grid preconditioner. Besides, when considering as a smoother few iterations of GMRES preconditioned by one symmetric lexicographical Gauss-Seidel iteration, this preconditioner has shown experimentally robust at high wavenumbers, the number of iterations grows linearly with respect to the wavenumber. Furthermore, by investigating a spectrum study in the flexible GMRES context, we have shown that a fixed number of iterations on the coarse level of the two-level preconditioner can be satisfactory enough for the convergence of flexible GMRES: such a preconditioner is efficient when solving Helmholtz problems at very high wavenumbers on parallel distributed memory computers. It has enabled us to solve homogeneous Helmholtz problems at wavenumbers up to two thousands, requiring the solution of a linear system of size larger than 68 billions on 65500 BlueGene/P cores. This method has also shown efficient for heterogeneous problems, even if it does not scale as well as in the homogeneous case. Indeed, on a fixed problem size, the number of iterations required by flexible GMRES to reach convergence is increasing with respect to the number of cores. However, we were able to obtain solutions in a still reasonable number of iterations and times at high frequencies (up to 30 Hz,) considering two public domain velocity models (SEG/EAGE Salt dome and Overthrust).

Concerning Krylov method enhancement, we have extended the flexible GMRES method to either a version with deflated restarting or to a block version. The purpose of the spectral deflation implemented in flexible GMRES with deflated restarting, is to improve the convergence rate injecting at the restart harmonic Ritz vectors in the Krylov subspace. This method has shown a good potential on both academic test cases and real-life applications. Concerning the block FMGRES methods, we have shown that their convergence can benefit from the multiple right-hand side situation. Indeed, despite the failure of block flexible GMRES to improve both the numbers of iterations and operations, the deflation of the residual at the restart can significantly reduce the total number of iterations and the total computational time. The deflation of the residual is based on a SVD of the block residual and on the use of the leading singular vectors defined by

a tolerance. The truncation technique is also based on a SVD of the block residual at each restart but uses as an initial block residual a fixed number of singular vectors. The truncation of the residual at the restart also improves the convergence and definitively lowers the memory requirements of the method, a nice feature when handling very large problems with many right-hand sides in a constrained memory environment. These methods made it possible solving linear systems with a billion of unknowns and multiple right-hand sides in our geophysical application. Furthermore, considering heterogeneous velocity fields, they tend to reduce the influence of parallelism on the total number of iterations.

Therefore we consider that these results are encouraging for the numerical solution of three-dimensional heterogeneous Helmholtz problems in the frequency domain. They are important milestones also in the context of seismic migration inverse problems. Nevertheless, further investigations are required to obtain a more robust and faster preconditioner. Indeed, even if the perturbed two-level method combined with flexible GMRES leads to a reduced number of iterations, the most expensive part of the solution phase is related to the approximate solution of the coarse problems. An effort should then be done to design an efficient preconditioner for the coarse problem. In this context, the use of a multilevel preconditioner on a shifted coarse operator should be investigated. Another way to improve the properties of the two-level preconditioner properties could be to use a Galerkin formulation for the coarse operator. Indeed, this was used in the two-dimensional case [31, 42]. Although the Galerkin formulation of the coarse operator is rather tedious to implement in a parallel environment, we plan to investigate its effect on the properties of our two-level perturbed preconditioner in three dimensions.

In this thesis, we have only considered a simple second-order finite difference discretization scheme with a number of points per wavelength fixed to 12. This implied to solve very large linear systems because of the stability condition of our discretization scheme. This issue could be overcome by considering more sophisticated discretization schemes. Indeed, in [89], a 27 point finite difference scheme has been designed in order to decrease the dispersion error with only 4 points per wavelength. A major consequence is that, at a given wavenumber, this discretization scheme leads to a linear system 27 times smaller than the one obtained for the second-order finite difference discretization scheme. Considering such an improved discretization scheme could then reduce dramatically the memory requirements of the solution method. However, it does not mean that the problem would be easier to solve with preconditioned iterative methods [62]. Indeed, in the two-level geometric method context, a direct coarse approximation discretization cannot be deduced without further investigation on the discretization scheme itself. Once again, in this context, a Galerkin formulation on the coarse level would be particularly relevant. The use of algebraic multigrid [16, 17, 18], Appendix A in [115], should also be considered for other discretization schemes of the Helmholtz operator such as finite element techniques or spectral elements [71].

Furthermore, another interesting aspect of this 27 point stencil discretization scheme could be to use a fixed mesh grid size on a given frequency range. Since the matrix size would be much smaller than that obtained with a 7 point stencil, it would be possible to do all the computations for all frequencies using the same mesh grid size determined for the highest frequency. Consequently numerical problems would be then easier for small and middle range frequencies since the mesh grid size h would be much smaller than necessary. This idea leads us to a formulation with multiple left- and right-hand sides where matrices and vectors are belonging to the same matrix spaces and vector spaces respectively. Solving these systems in sequence with preconditioned flexible subspace method is thus a natural route. In this context, rather than discarding the subspace generated when solving a given linear system, we plan to investigate subspace recycling [22, 92] to hopefully reduce the number of iterations required for the next system.

Appendix A

Three-dimensional Helmholtz equation in the frequency domain with a PML formulation

A.1 Continuous formulation

We present the formulation of the three-dimensional Helmholtz equation with some absorbing boundary conditions that has been reatined in this thesis. These absorbing conditions are modeled with a Perfectly Matched Layer technique (PML [11, 12]). It consists in adding an artificial layer around the domain which absorbs the reflection of the waves. As we will see, the use of these functions makes the system complex-valued in the two- and three-dimensional cases in this layer. A formulation of the visco-acoustic wave equation is given in [65] and [89] respectively. This equation is a more general expression of the Helmholtz equation. We follow the developments proposed in the two-dimensional case in [65] to deduce the three-dimensional Helmholtz equation in the frequency domain. The three-dimensional frequency-domain acoustic wave equation in the time domain with PML can be formulated as a first-order hyperbolic system [119], involving the wave pressure u, and the propagation velocity v:

$$
\left\{
\begin{array}{rcl}
\dfrac{\partial}{\partial t} u_x(x,y,z,t) + \gamma_x(x)\, u_x(x,y,z,t) &=& K(x,y,z) \dfrac{\partial}{\partial x} v_x(x,y,z,t) + S(x,y,z,t), \\
\dfrac{\partial}{\partial t} u_y(x,y,z,t) + \gamma_y(y)\, u_y(x,y,z,t) &=& K(x,y,z) \dfrac{\partial}{\partial y} v_y(x,y,z,t), \\
\dfrac{\partial}{\partial t} u_z(x,y,z,t) + \gamma_z(z)\, u_z(x,y,z,t) &=& K(x,y,z) \dfrac{\partial}{\partial z} v_z(x,y,z,t), \\
\dfrac{\partial}{\partial t} v_x(x,y,z,t) + \gamma_x(x)\, v_x(x,y,z,t) &=& b(x,y,z) \dfrac{\partial}{\partial x} u(x,y,z,t), \\
\dfrac{\partial}{\partial t} v_y(x,y,z,t) + \gamma_y(y)\, v_y(x,y,z,t) &=& b(x,y,z) \dfrac{\partial}{\partial y} u(x,y,z,t), \\
\dfrac{\partial}{\partial t} v_z(x,y,z,t) + \gamma_z(z)\, v_z(x,y,z,t) &=& b(x,y,z) \dfrac{\partial}{\partial z} u(x,y,z,t),
\end{array}
\right.
$$

where v_x, v_y and v_z are the components of the velocity v on the Cartesian grid ($v = v_x + v_y + v_z$) and u_x, u_y and u_z are the non-physical components of the wave pressure u ($u = u_x + u_y + u_z$). This splitting allows to introduce directional derivatives and then to take into account the one-dimensional nature of the PML functions γ_x, γ_y, γ_z. These functions are zero outside the PML layer. The quantity $b(x,y,z)$ is the buoyancy of the media (the inverse of the density) and $K(x,y,z)$ is the bulk modulus, defined by $K(x,y,z) = \frac{v(x,y,z)^2}{b(x,y,z)}$.

A Fourier transform of this hyperbolic system is performed in order to obtain a system of equations in

the frequency domain:

$$\begin{cases} \dfrac{-i\omega\xi_x(x)}{K(x,y,z)}u_x(x,y,z,\omega) &= \dfrac{\partial}{\partial x}v_x(x,y,z,\omega) + S(x,y,z,\omega), \\ \dfrac{-i\omega\xi_y(y)}{K(x,y,z)}u_y(x,y,z,\omega) &= \dfrac{\partial}{\partial y}v_y(x,y,z,\omega), \\ \dfrac{-i\omega\xi_z(z)}{K(x,y,z)}u_z(x,y,z,\omega) &= \dfrac{\partial}{\partial z}v_z(x,y,z,\omega), \\ v_x(x,y,z,\omega) &= -\dfrac{b(x,y,z)}{-i\omega\xi_x(x)}\dfrac{\partial}{\partial x}u(x,y,z,\omega), \\ v_y(x,y,z,\omega) &= -\dfrac{b(x,y,z)}{i\omega\xi_y(y)}\dfrac{\partial}{\partial y}u(x,y,z,\omega), \\ v_z(x,y,z,\omega) &= -\dfrac{b(x,y,z)}{i\omega\xi_z(z)}\dfrac{\partial}{\partial z}u(x,y,z,\omega). \end{cases}$$

The parameter ω denotes the angular frequency ($\omega = 2\pi f$) and the one-dimensional damping functions ξ_x, ξ_y, ξ_z are defined as follows:

$$\begin{cases} \xi_x(x) &= 1 + i\gamma_x(x), \\ \xi_y(y) &= 1 + i\gamma_y(y), \\ \xi_z(z) &= 1 + i\gamma_z(z). \end{cases}$$

Replacing the expression of v_x, v_y, v_z in the first three equations, it follows that

$$\begin{cases} -\dfrac{\omega^2\xi_x(x)}{K(x,y,z)}u_x(x,y,z,\omega) &= \dfrac{\partial}{\partial x}\dfrac{b(x,y,z)}{\xi_x(x)}\dfrac{\partial}{\partial x}u(x,y,z,\omega) + S(x,y,z,\omega), \\ -\dfrac{\omega^2\xi_y(y)}{K(x,y,z)}u_y(x,y,z,\omega) &= \dfrac{\partial}{\partial y}\dfrac{b(x,y,z)}{\xi_y(y)}\dfrac{\partial}{\partial y}u(x,y,z,\omega), \\ -\dfrac{\omega^2\xi_z(z)}{K(x,y,z)}u_z(x,y,z,\omega) &= \dfrac{\partial}{\partial z}\dfrac{b(x,y,z)}{\xi_z(z)}\dfrac{\partial}{\partial z}u(x,y,z,\omega). \end{cases}$$

Adding these three equations, we obtain the three-dimensional frequency-domain visco-acoustic wave equation:

$$\left[-\frac{\omega^2}{K(x,y,z)} - \frac{1}{\xi_x(x)}\frac{\partial}{\partial x}\frac{b(x,y,z)}{\xi_x(x)}\frac{\partial}{\partial x} - \frac{1}{\xi_y(y)}\frac{\partial}{\partial y}\frac{b(x,y,z)}{\xi_y(y)}\frac{\partial}{\partial y} - \frac{1}{\xi_z(z)}\frac{\partial}{\partial z}\frac{b(x,y,z)}{\xi_z(z)}\frac{\partial}{\partial z}\right]u(x,y,z,\omega)$$

$$= S(x,y,z,\omega).$$

However, we want to focus on the three-dimensional frequency-domain acoustic wave equation. To remove the viscosity, we assume that the gradient of the density is infinitesimally small as in [40]. It is equivalent to assume that the density $\dfrac{1}{b(x,y,z)}$ is constant. Thus, the bulk modulus $K(x,y,z)$ reads $\dfrac{v(x,y,z)^2}{b(x,y,z)} = \dfrac{v(x,y,z)^2}{b}$. The Helmholtz equation, scaling the source term $S(x,y,z,\omega)$ by b ($s(x,y,z,\omega) := S(x,y,z,\omega)/b$), can then be written as

$$\left[-\frac{1}{\xi_x(x)}\frac{\partial}{\partial x}\frac{1}{\xi_x(x)}\frac{\partial}{\partial x} - \frac{1}{\xi_y(y)}\frac{\partial}{\partial y}\frac{1}{\xi_y(y)}\frac{\partial}{\partial y} - \frac{1}{\xi_z(z)}\frac{\partial}{\partial z}\frac{1}{\xi_z(z)}\frac{\partial}{\partial z} - \frac{\omega^2}{v^2(x,y,z)}\right]u(x,y,z,\omega) = s(x,y,z,\omega). \quad \text{(A.1)}$$

Since the damping functions ξ_x, ξ_y, ξ_z are equal to one outside the PML layer, denoting by Ω a parallelepiped domain, Γ its border and Ω_{PML} the PML layer (see Figure A.1), the Helmholtz equation can be

split into three equations:

$$\begin{cases} -\Delta u - k^2 u = g \text{ in } \Omega \backslash (\Omega_{PML} \cup \Gamma), \\ \left[-\frac{1}{\xi_x(x)} \frac{\partial}{\partial x} \frac{1}{\xi_x(x)} \frac{\partial}{\partial x} - \frac{1}{\xi_y(y)} \frac{\partial}{\partial y} \frac{1}{\xi_y(y)} \frac{\partial}{\partial y} - \frac{1}{\xi_z(z)} \frac{\partial}{\partial z} \frac{1}{\xi_z(z)} \frac{\partial}{\partial z} - k(x,y,z)^2 \right] u = s \text{ in } \Omega_{PML} \backslash \Gamma, \\ u = 0 \text{ on } \Gamma, \end{cases}$$

where $k(x, y, z)$ is called the wavenumber $\frac{\omega}{v(x, y, z)}$.

We now present the PML functions that will be used when discretizing the Helmholtz equation. We choose the same PML functions as in [89] for each direction x, y, z. Denoting by L_{PML} the width of the PML, the PML function can be described for the direction x in $\Omega = [0, 1]^3$:

$$\gamma_x(x) = \begin{cases} -cos\left(\frac{\pi x}{2L_{PML}}\right) & \text{if } 0 \le x \le L_{PML}, \\ 0 & \text{if } L_{PML} < x < 1 - L_{PML}, \\ -cos\left(\frac{\pi(1-x)}{2L_{PML}}\right) & \text{if } 1 - L_{PML} \le x \le 1. \end{cases} \tag{A.2}$$

We note that the width of the PML must be at least as long as one wavelength.

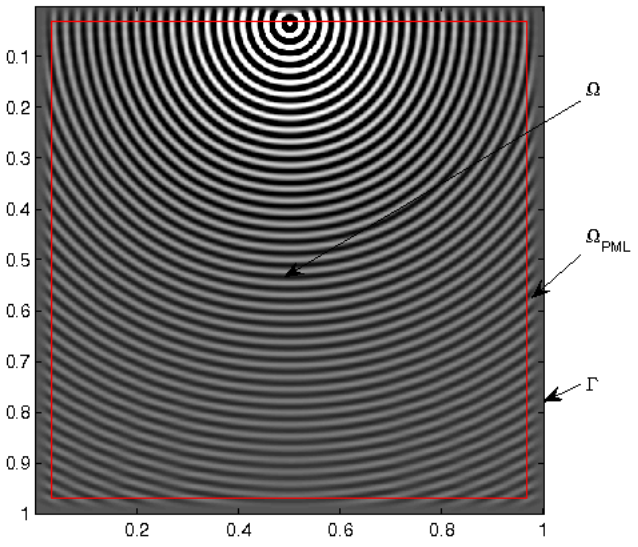

Figure A.1: Slice of a three-dimensional solution ($\Omega = [0, 1]^3$, $h = 1/512$, $k = \frac{\pi}{6h}$), The source term is located at $(\frac{1}{2}, \frac{1}{2}, L_{PML} + h)$. Red lines represent the interface between the interior and the PML zone.

A.2 Discrete formulation

We will consider a second-order finite difference scheme in three dimensions to discretize the Helmholtz equation with PML. This discretization scheme requires for the Helmholtz equation a stability condition to hold, relating both the wavenumber $k(x, y, z)$ and the mesh grid size h [25]:

$$k(x, y, z)h \le \frac{2\pi}{n_\lambda}, \ \forall \ (x, y, z) \in \Omega_h$$

where n_λ is the number of points per wavelength. This condition will be satisfied if:

$$\frac{\omega}{\min_{(x,y,z)\in\Omega_h} v(x,y,z)} h = \frac{2\pi}{n_\lambda}, \ \forall\ (x,y,z) \in \Omega_h. \tag{A.3}$$

The quantity n_λ is depending on the discretization scheme, for a second order discretization scheme we usually select $10 \leq n_\lambda \leq 12$ and take $n_\lambda = 12$. When academic model problems are considered in the unit cube $\Omega_h = (0,1)^3$, the propagation velocity in the domain is set to one $v(i,j,k) = 1, \ \forall\ (i,j,k) \in \Omega_h$ and the mesh grid size h corresponds to the inverse of the number of points per direction. These model problems are then adimensional. For this kind of problems, the wavenumber is taken such that Relation A.4 is satisfied:

$$k = \frac{2\pi}{n_\lambda h} = \frac{\pi}{6h}. \tag{A.4}$$

Wavenumbers corresponding to $h = \frac{1}{2^p}, p \in \mathbb{N}$ are reported in Table A.1.

Grid	64^3	128^3	256^3	512^3	1024^3	2048^3	4096^3
k	33.51	67.02	134.04	268.08	536.16	1072.32	2148.64

Table A.1: Wavenumbers corresponding to $h = \frac{1}{2^p}, p \in \mathbb{N}$ for adimensional model problems.

Considering such large grids (few billions of unknowns) is of crucial interest for the geophysical application dealing with the solution of Helmholtz problems in the frequency domain [89]. In fact, when considering an academic velocity field such as SEG/EAGE Salt dome [5], solving the Helmholtz equation for frequencies up to 20 Hz involves to consider grids with few billions of points. The grid sizes for different frequencies satisfying the stability condition (A.3) are presented in Table A.2.

$f\ (Hz)$	2.5	5	10	20
$h\ (m)$	50	25	12.5	6.25
Grid	$301 \times 301 \times 115$	$571 \times 571 \times 199$	$1112 \times 1112 \times 367$	$2200 \times 2200 \times 709$

Table A.2: Grid sizes for different frequencies such that they verify the stability condition (A.3) for the SEG/EAGE Salt dome velocity field with minimum velocity 1500 $m.s^{-1}$ and size $13.5 \times 13.5 \times 4$ km^3, taking 16 points in the PML layer on each side of the physical domain.

Thus, we have to solve large, complex-valued and non-Hermitian problems due to both the stability condition on the discretization and the absorbing boundary conditions (PML). Furthermore, when large wavenumbers are considered, the problem is indefinite. As previously said, a second order discretization scheme is used to discretize the three-dimensional Helmholtz equation. A classical Cartesian stencil in three dimensions (7 points stencil, Figure A.2) is then used.

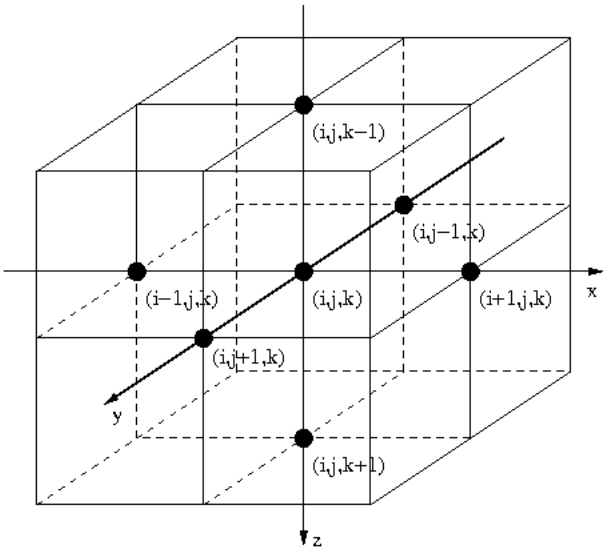

Figure A.2: Cartesian stencil (7 points) for a Laplacian-like operator.

Thus, the matrix obtained after discretization has a sparse structure with seven bands (see Figure A.3).

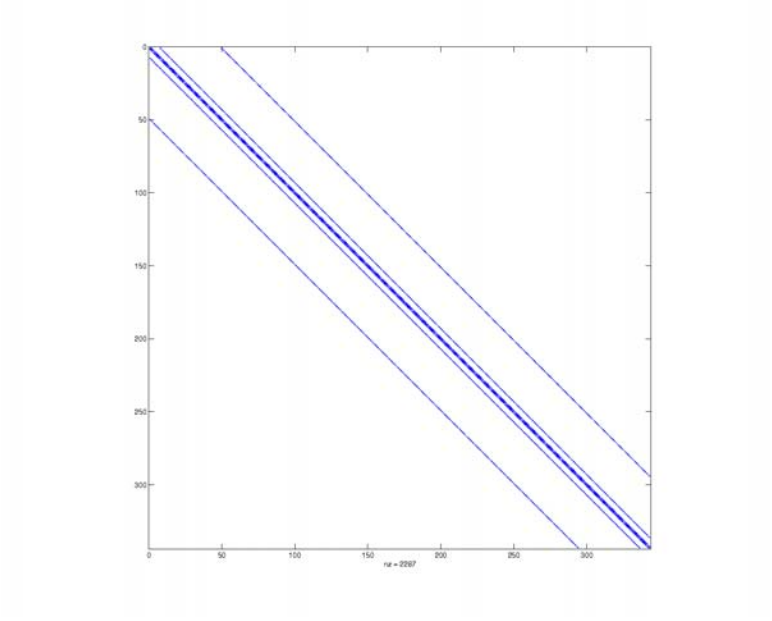

Figure A.3: Pattern of the Helmholtz matrix with a lexicographical ordering of the unknowns.

For each grid point, the stencil coefficients are obtained generalizing to the three dimensional case the formulas in [65]. First, the Laplacian is discretized for each Cartesian direction x, y, z at the point (i, j, k):

$$\begin{aligned} s_{i,j,k} \quad = \quad & \frac{-\omega^2}{v_{i,j,k}^2} u_{i,j,k} - \frac{1}{\xi_i}\frac{1}{h^2}\left[\frac{(u_{i+1,j,k}-u_{i,j,k})}{\xi_{i+1/2}} + \frac{(u_{i,j,k}-u_{i-1,j,k})}{\xi_{i-1/2}}\right] \\ & -\frac{1}{\xi_j}\frac{1}{h^2}\left[\frac{(u_{i,j+1,k}-u_{i,j,k})}{\xi_{j+1/2}} + \frac{(u_{i,j,k}-u_{i,j-1,k})}{\xi_{j-1/2}}\right] \\ & -\frac{1}{\xi_k}\frac{1}{h^2}\left[\frac{(u_{i,j,k+1}-u_{i,j,k})}{\xi_{k+1/2}} + \frac{(u_{i,j,k}-u_{i,j,k-1})}{\xi_{k-1/2}}\right], \end{aligned} \tag{A.5}$$

where ξ_s denotes a discrete PML function for the direction s, $\xi_{s-1/2} = \frac{1}{2}(\xi_s + \xi_{s-1})$ and $\xi_{s+1/2} = \frac{1}{2}(\xi_s + \xi_{s+1})$. Thus the coefficients at each point of the stencil (Figure A.2) become:

$$
\begin{aligned}
s_{i,j,k} \quad = \quad & \left[\frac{-\omega^2}{v_{i,j,k}^2} + \frac{1}{h^2}\left(\frac{1}{\xi_i \xi_{i+1/2}} + \frac{1}{\xi_i \xi_{i-1/2}}\right) + \frac{1}{h^2}\left(\frac{1}{\xi_j \xi_{j+1/2}} + \frac{1}{\xi_j \xi_{j-1/2}}\right) + \frac{1}{h^2}\left(\frac{1}{\xi_k \xi_{k+1/2}} + \frac{1}{\xi_k \xi_{k-1/2}}\right) \right] u_{i,j,k} \\
& - \frac{1}{h^2}\frac{1}{\xi_i \xi_{i+1/2}} u_{i+1,j,k} - \frac{1}{h^2}\frac{1}{\xi_i \xi_{i-1/2}} u_{i-1,j,k} \\
& - \frac{1}{h^2}\frac{1}{\xi_j \xi_{j+1/2}} u_{i,j+1,k} - \frac{1}{h^2}\frac{1}{\xi_j \xi_{j-1/2}} u_{i,j-1,k} \\
& - \frac{1}{h^2}\frac{1}{\xi_k \xi_{k+1/2}} u_{i,j,k+1} - \frac{1}{h^2}\frac{1}{\xi_k \xi_{k-1/2}} u_{i,j,k-1}.
\end{aligned}
$$

Other discretization schemes are available in the three-dimensional case, they have been developed in order to minimize simultaneously the dispersion error and the required number of points per wavelength. This has been notably done in [89] where several finite difference schemes are combined and weighted in order to obtain a 27 point stencil requiring only 4 points per wavelength. The use of this stencil leads to a matrix which is not as sparse as the one from the Cartesian stencil. Its main feature is the reduced size of the linear system to treat the same frequency. We have also implemented this combined scheme and plan to use it in a future work.

Discrete right-hand side The right-hand side S, representing the source, is resulting from the discretization of a Kronecker function $\delta_{(x_\delta, y_\delta, z_\delta)}$ where $(x_\delta, y_\delta, z_\delta)$ denotes the source position. We locate its non-zero entry at $(n_x/2, n_y/2, n_{PML} + 1)$. In fact, we have

$$s = \delta_{(n_x/2, n_y/2, n_{PML}+1)},$$

where n_x, n_y are the number of points in $x-$ and $y-$directions respectively and n_{PML} the number of point in the PML. Since the PML width must correspond to at least one wavelength and the number of points per wavelength n_λ is 12, we fix n_{PML} to 16 independently of the wavenumber.

Appendix B

Résumé en Français

Cette thèse a été réalisée dans le cadre d'une collaboration entre le CERFACS et le groupe industriel TOTAL. Elle s'incrit donc dans une thématique liée à l'exploitation pétrolière dont la migration profondeur est un sujet de recherche phare [24]. En effet, elle permet d'affiner numériquement la connaissance du sous-sol d'une zone à partir de données préliminaires issues d'une campagne géophysique. Durant une campagne géophysique, qu'elle soit réalisée en mer ou sur terre, des ondes sont envoyées dans le sous-sol à partir de plusieurs points sources situés à la surface. Ces ondes se propagent alors sous Terre se réfractant, se réflechissant ou encore s'atténuant au gré des différentes couches géologiques s'y trouvant. La réponse de ces ondes réfléchies est enregistrée par plusieurs recepteurs lorsqu'elles atteignent la surface. Ainsi, une première image de la structure du sous-sol est obtenue après interprétation de ces données. Ce processus peut être reproduit numériquement afin d'obtenir une image plus précise. Les phénomènes de propagation d'ondes sont généralement simulés dans le domaine temporel. Toutefois une transformée de Fourier inverse sur les équations modélisant ces phénomènes permet également de travailler dans le domaine fréquentiel. Une approche fréquentielle offre de nombreux avantages, elle permet d'avoir une image locale dans le domaine physique et offre une formulation intéressante du problème inverse en imagerie. Toutefois, une formulation fréquentielle du problème de propagation d'onde implique une résolution de problèmes d'Helmholtz en fréquence pour plusieurs nombres d'onde et termes source, ce qui peut alourdir nettement le coût de la méthode. Les problèmes d'Helmholtz peuvent être écrits ainsi:

$$-\Delta u - k^2 u = s,$$

où u est la pression de l'onde, k le nombre d'onde et s un terme source. De manière à simuler un domaine physique infini, des conditions aux limites de type absorbantes sont utilisées lors de la discretisation de cette équation. Une fois discrétisée, que ce soit avec des éléments finis ou des différences finies, un système linéaire à second membres multiples doit être résolu pour chaque fréquence:

$$AX = B,$$

où $A \in \mathbb{C}^{n\times n}$, $X \in \mathbb{C}^{n\times p}$ et $B \in \mathbb{C}^{n\times p}$ où p est le nombre de sources. La matrice A est carrée, creuse, de grande dimension, indéfinie pour de grands nombres d'onde et dans la plupart des cas non-hermitienne, non-symétrique, le second membre B contient l'information liée aux p sources. Ainsi, le principal défi lancé par une formulation fréquentielle de problèmes de propagation d'ondes reste la confection d'une méthode de résolution robuste et efficace pour des problèmes d'Helmholtz en fréquence à grand nombre d'onde et de nombreux termes source.

Ces systèmes linéaires, de grande taille et indéfinis, peuvent être résolus par des méthodes directes pour matrices creuses [29, 30], en utilisant par exemple une factorisation Gaussienne LU de la matrice. En effet, de récents développements ont permis, grâce à plusieurs opérations de pré-traitement (permutations et mise à l'échelle), de rendre ces méthodes efficaces dans le cas indéfini non-symétrique [34]. En deux dimensions, l'efficacité de ces méthodes est indéniable du fait de la stabilité de la résolution et de la réutilisation de la factorisation de la matrice pour toutes les sources à fréquence donnée. Cependant, le coût mémoire de telles méthodes croît rapidemment avec la taille du problème, ce qui rend leur utilisation délicate en trois dimensions pour de grands nombres d'ondes. En effet, dans [89], les auteurs rapportent que le coût mémoire

d'une factorisation LU, pour un schéma de discrétisation en différences finies pour un stencil compact à 27 points, est de l'ordre de $O(35n^{4/3})$ tandis que les nombres d'opération effectuées lors de la factorisation et des phases de résolutions sont de l'ordre de $O(n^2)$ et $O(n^{4/3})$ respectivement. Néanmoins, malgré ce coût en mémoire et en opérations, les auteurs ont montré que cette approche permettait de résoudre des problèmes d'Helmholtz à haute fréquence. En utilisant MUMPS [2, 3, 4], ils ont réussi à résoudre un problème d'Helmholtz à une fréquence de 10 Hz pour le modèle de vitesse SEG/EAGE Overthrust [5], ce qui correspond à une grille physique de taille $409 \times 409 \times 102$, en allouant toutefois plus de 450 GB de mémoire.

Ainsi, pour pallier la consommation mémoire de ces méthodes, l'utilisation de méthodes itératives s'impose. Se pose alors la question du préconditionnement, élément essentiel à l'obtention d'une convergence rapide. De même, le choix de la méthode de Krylov peut influencer le nombre de phase de préconditionnement pour converger, surtout quand plusieurs systèmes linéaires sont résolus en même temps (situation à multiple second-membres). Nous présentons dans un premier temps differentes méthodes de préconditionnement que l'on retrouve dans la littérature pour les problèmes d'Helmholtz.

Les méthodes de factorisation incomplète (ILU) [8] sont des techniques de préconditionnement très répandues et utilisées. Dans le cas indéfini, leurs facteurs incomplets peuvent néanmoins posséder un mauvais conditionnement et donc mener à une matrice préconditionnée, elle aussi mal conditionnée. Bien sûr, plusieurs idées ont été proposées pour contrer ces difficultés pour les problèmes d'Helmholtz en fréquence. Une première approche consiste à pratiquer une factorisation incomplète analytique de la matrice (AILU) [52]. Toutefois cette technique s'appuie sur des propriétés analytiques du problème initial, ce qui rend difficile son adaptation au cas hétérogène. Une factorisation incomplète avec seuil (ILUT [100]) peut aussi être utilisée lorsque la matrice est obtenue par une discretisation en élément finis [68] (moindre carrés de Galerkin (GLS)). Enfin, une autre approche consiste à réaliser une factorisation incomplète non pas du problème d'Helmholtz originel mais d'un problème décallé, qui est alors complexe [80, 90] (voir équation B.1 et détails ci-après). Cependant, il faut bien avoir à l'esprit que la convergence de méthodes de Krylov préconditionnées par ces techniques ILU est généralement lente pour de grands nombres d'onde et que le stockage des facteurs, même incomplés, peut être problématique. En outre, il est assez difficile de paralléliser ce genre de préconditionnement [8, 63].

Une autre famille de méthodes de préconditionnement repose sur les techniques de décomposition de domaines [94, 111, 114]. Le principe de ces méthodes est de subdiviser le domaine physique en sous-domaines, ainsi les problèmes locaux associés à ces sous-domaines sont résolus par une méthode directe et ces solutions locales sont utilisées pour résoudre le problème originel sur tout le domaine. Dans le cas de problèmes elliptiques, la convergence de méthodes de Krylov préconditionnées par de telles techniques est indépendante du nombre de sous-domaines à la condition près qu'une correction d'espace grossier soit effectuée. Bien entendu, du fait du caractère indéfini des problèmes d'Helmholtz pour de grands nombres d'ondes, ces méthodes doivent être modifiées en conséquence pour pouvoir être utilisées en pratique (voir la section (11.5.2) dans [114]). En effet, pour avoir un préconditionnement efficace, l'espace de correction grossière doit être discrétisé plutôt finement et par conséquent, les problèmes locaux sont de taille plus importante. De plus, utiliser des conditions aux limites de type Dirichlet ou Neumann dans les sous-domaines peut rendre les problèmes locaux singuliers.

Dans le cas de méthodes de décomposition de domaines sans recouvrement, l'utilisation de conditions de Sommerfeld permet de s'affranchir, dans une certaine mesure, de cette dernière difficulté [10, 26, 51]. La méthode FETI-H est un exemple de méthode de décomposition de domaine efficace pour les problèmes d'Helmholtz, elle fait appel à un problème grossier auxiliaire en décomposant le problème grossier original sur des ondes planes. Cette méthode a été améliorée [44, 45], donnant lieu à une variante duale primale de FETI (FETI-DPH) qui permet de résoudre des problèmes d'Helmholtz de dispersion pour des fréquences de rang intermédiaire sur un grand nombre de processeurs [44]. Le plus récent résultat théorique [76], à notre connaissance, concernant les méthodes de décomposition de domaines appliquées à des problèmes d'Helmholtz avec des conditions aux limites de Dirichlet, donne une borne supérieure du conditionnement de l'opérateur préconditionné $A\ M^{-1}$ si les problèmes locaux sont résolus exactement:

$$\kappa(A\ M^{-1}) \leq C(1+k^2)(1+k^2H^2)\left(1+log\left(\frac{H}{h}\right)\right)^2.$$

où C est une constante positive indépendante du diamètre des éléments h et du diamètre maximal des sous-

domaines H. Dans cette majoration, il apparaît clairement que le conditionnement $\kappa(A\ M^{-1})$ croît proportionnellement avec k^4. Ceci illustre bien la difficulté d'utiliser un tel préconditionement pour des nombres d'onde élevés. Une alternative algébrique de ce type de préconditionnement a pourtant été proposée pour des problèmes d'Helmholtz [62, 120]. En utilisant un préconditionnement algébrique de type Schwarz additif, des problèmes d'Helmholtz pour de hautes fréquences ont été résolus en des temps raisonnables. Le coût mémoire d'une telle méthode reste tout de même élevé; pour résoudre l'équation d'Helmholtz à 12 Hz pour le modèle de vitesse réaliste SEG/EAGE Saltdom [5], pas moins de 2000 processeurs de BlueGene/P sont nécessaires [1].

Les méthodes multigrille [15, 20, 61, 115] peuvent être elles aussi employées pour résoudre des problèmes d'Helmholtz. Cependant, une fois encore, l'utilisation de ce type de méthodes pour des problèmes indéfinis n'est pas des plus aisée. En effet, les composantes classiques d'un cycle multigrille, lissage standard et correction de grille grossière, se révèlent inopérantes sur les problèmes d'Helmholtz [7, 19, 37, 42, 70]. D'une part, l'emploi de méthodes de relaxation (Jacobi, Gauss-Seidel...) ne permet pas de lisser l'erreur sur les grilles intermédiaires de la hiérarchie multigrille. D'autre part, l'implication du nombre d'onde dans l'opérateur d'Helmholtz rend ses approximations sur les grilles grossières peu en rapport avec celles du niveau fin, compromettant ainsi la pertinence de la correction de grille grossière. Immanquablement, une adaptation des techniques multigrille aux problèmes d'Helmholtz doit être effectuée pour obtenir des méthodes de résolution efficaces. Ainsi dans [31, 37, 42, 70, 78], les auteurs nous proposent, dans ce contexte, différentes manières d'utiliser ces méthodes de façon judicieuse.

Une première stratégie consiste à préconditionner l'opérateur d'Helmholtz par un cycle multigrille avec peu de grilles dans sa hiérarchie [31, 37, 70]. De la sorte, la correction de grille grossière joue son rôle sur les grilles considérées. Si plus de deux grilles sont utilisées, le recours à des lisseurs plus exotiques, telle une méthode de Krylov comme GMRES [102], peut contrer les effets de bord inhérents à l'utilisation de lisseurs standards sur les grilles intermédiaires de la hiérarchie multigrille [37]. Toutefois, une hiérarchie de grille réduite entraîne une taille de problème grossier importante qui peut, en trois dimensions, interdire l'utilisation d'une méthode directe pour des raisons de ressources informatiques.

Une deuxième idée fait appel à deux représentations de l'erreur sur les grilles grossières (sur les fonctions d'"ondes" et de "raie" [77]). De ce fait, le multigrille peut être utilisé en tant que méthode de résolution sans passer par une méthode de Krylov; cette double représentation grossière rendant leurs rôles aux lisseurs et corrections de grille grossière. Même si cette méthode a un comportement parfait (passage à l'échelle au sens du multigrille) sur des cas homogènes [74, 78, 118], elle peine à être étendue au cas hétérogène. En effet, les fonctions de "raie" doivent alors être explicitement calculées [122, 123], ce qui entraîne une résolution de plusieurs problèmes de valeurs propres et donc un surcoût en opérations de la méthode.

Plus récemment, un troisième préconditionnement multigrille a été présenté dans [42, 43]. Il est considéré comme une avancée considérable dans la résolution des problèmes d'Helmholtz dans le domaine fréquentiel. Son principe tient à utiliser un opérateur d'Helmholtz décalé ("shifted" en Anglais) dans le problème de préconditionnement, le paramètre de décalage, β, étant complexe:

$$-\Delta u - (1 - i\beta)k^2 u. \tag{B.1}$$

Le paramètre β rend alors efficaces les ingrédients classiques du multigrille sur le problème de préconditionnement susnommé. En effet, une telle stratégie a permis à ces élaborateurs de résoudre des problèmes d'Helmholtz à des nombres d'ondes relativemment élevés et pour des milieux hétérogènes [42, 95, 96]. Toutefois le nombre d'opérations à effectuer pour atteindre la convergence demeure élevé autant en deux dimensions [95] qu'en trois dimensions [96]. Dans la continuité de cette idée, un preconditionneur algébrique multi-niveaux a été proposé dans [14]. Cette fois-ci, une factorisation LDL^H incomplète est effectuée sur plusieurs niveaux tirant avantage des récentes avancées des méthodes directes pour les systèmes linéaires symétriques et creux [35, 103]. Encore une fois, malgré l'efficacité d'une telle stratégie sur des problèmes d'Helmholtz en deux et trois dimensions, la complexité de cette technique reste élevée, particulièrement en ce qui concerne son coût mémoire. De plus, l'efficacité de telles méthodes dépend essentiellement d'un choix judicieux de β. Ce choix sera influencé par les composantes de la méthode multi-niveaux [14, 42] mais aussi par le schéma de discrétisation de l'opérateur d'Helmholtz [116]. Ainsi, il n'est pas aisé de répondre à

[1] http://www.idris.fr/docs/docu/projets-Babel/SEISCOPE/CR-projet-SEISCOPE.html

cette question sans avoir recours à de nombreuses expérimentations ou une analyse de Fourier détaillée [15].

En deux dimensions, la question de ce choix peut être évité. En effet, un préconditionnement à deux grilles classique appliqué à l'opérateur d'Helmholtz originel le permet [31]. Cependant l'utilisation d'une méthode directe pour la résolution du problème de grille grossière rend difficile son extension au cas tridimensionel. Même si le problème grossier est bien plus petit que le problème fin, le coût informatique d'une factorisation de type LU n'en permet pas son utilisation. Vient alors l'idée naturelle d'employer une méthode itérative afin de résoudre le problème grossier. Assurément cette option diminue nettement le coût mémoire de notre méthode. L'efficacité d'une telle méthode s'assiéra donc sur son faible coût mémoire mais aussi sur ses bonnes propriétés de préconditionnement et ce, malgré un critère de convergence élevé pour le problème grossier. Ce dernier point fera l'objet d'une étude détaillée à l'aide d'une analyse de Fourier, d'expérimentations numériques et d'une analyse de spectres.

Une méthode de Krylov préconditionnée sera utilisée au niveau grossier. Ce dernier choix a pour conséquence la non-constance du précondtionnement à deux grilles d'une itération à la suivante. Ceci nous conduit à utiliser une méthode GMRES flexible (FGMRES [99]) préconditionnée par la méthode perturbée à deux niveaux. Nous avons d'ailleurs étendu la méthode GMRES-DR [85] au cas flexible dans cette optique (FGMRES-DR [53]). Le principe de ces méthodes repose sur l'ajout d'information spectrale dans le sous-espace de Krylov de GMRES ou FGMRES. Outre le fait que cette information peut influencer positivement la convergence, ces méthodes bénéficient surtout du faible coût de l'ajout de ces directions dans le sous-espace de Krylov (calcul de vecteurs harmoniques de Ritz). Du reste, cette méthode s'est révélée efficace pour résoudre des problèmes d'Helmholtz bidimensionnels avec des conditions aux limites de type Dirichlet.

L'application géophysique ayant trait aux problèmes d'Helmholtz pose aussi le difficile problème du traitement de milliers de source pour une fréquence donnée. Ceci revient à résoudre des systèmes linéaires avec plusieurs seconds membres et nous mènent à travailler sur la conception d'une méthode de Krylov par bloc efficace. Ainsi, partant de maintes références sur GMRES par bloc (BGMRES) [59, 72, 73, 79, 97, 112, 121], nous avons construit des variantes de GMRES flexible par bloc (BFGMRES) mettant en oeuvre la déflation du résidu par bloc au recommencement de la méthode: GMRES flexible par bloc avec déflation basée sur une décomposition en valeurs singulières (BFGMRESD) et GMRES flexible par bloc avec troncation basée sur une décomposition en valeurs singulières (BFGMREST). Ces deux methodes effectuent une décomposition en valeurs singulières du résidu par bloc au début de chaque recommencement ($R = U\Sigma W^H$ [54]). La méthode BFGMRESD ne garde alors comme bloc initial de vecteurs que les vecteurs singuliers de U correspondant à des valeurs singulières plus grandes qu'un certain seuil tandis que la méthode BFGMREST n'en garde qu'un nombre fixe.

Toutes ces méthodes ont été évaluées sur des machines massivement parallèles. Ces tests ont montré l'efficacité et la robustesse du préconditionnement deux-grilles perturbé pour des problèmes d'Helmholtz homogènes et héterogènes et ce, pour des milliers de processeurs. Enfin, les méthodes par bloc BFGMRESD et BFGMREST se sont montrées très efficaces pour résoudre des systèmes d'équations à plusieurs seconds membres.

Le plan de la thèse est tel qu'il suit:

- Dans l'introduction, nous présentons tout d'abord l'application choisie en géophysique. Nous passons ensuite brièvement en revue les différentes méthodes numériques retenues pour la résolution de l'équation d'Helmholtz dans le domaine fréquentiel.

- Dans le deuxième chapitre, nous présentons des méthodes de Krylov destinées à la résolution de systèmes linéaires comportant un ou plusieurs seconds membres. En premier lieu, nous survolons brièvement les méthodes GMRES et Flexible GMRES (FGMRES), et donnons un outil permettant d'effectuer une analyse spectrale dans le cadre flexible. L'algorithme de la méthode "FGMRES with Deflated Restarting" (FGMRES-DR) est ensuite décrit. Ce chapitre se clôt sur la description de méthodes par blocs. Ces méthodes tirent avantage de la présence simultanée de plusieurs seconds membres. Deux stratégies sont présentées: la déflation et la troncation du bloc de résidus.

- Dans le troisième chapitre, nous nous intéressons aux méthodes à plusieurs niveaux afin de les utiliser pour préconditionner des problèmes d'Helmholtz en trois dimensions. Ainsi, nous commençons par énoncer quelques rappels sur les techniques multigrille géométriques et l'analyse de Fourier en trois dimensions. L'analyse de Fourier nous permet alors de réaliser, dans un premier temps, une étude de lissage de l'opérateur d'Helmholtz original et décalé ("shifted"), puis, dans un second temps, l'analyse d'un cycle deux grilles mettant en jeu une méthode de Krylov au niveau grossier. Nous appelons ce type de cycle une méthode perturbée à deux niveaux. Cette analyse montre que le taux de convergence d'une telle méthode est à peu près similaire, que le problème grossier soit résolu exactement ou très approximativement. Ce phénomène est encore constaté lorsque la méthode perturbée à deux niveaux est utilisée en tant que préconditionnement permettant ainsi d'obtenir une méthode économique en mémoire et en temps de calcul. Nous terminons ce chapitre par une analyse spectrale de l'opérateur d'Helmholtz préconditionné en fonction de plusieurs tolérances grossières.

- Le quatrième chapitre est consacré aux expérimentations numériques sur des machines parallèles à mémoire distribuée. Dans un premier temps, des problèmes d'Helmholtz homogènes en trois dimensions sont étudiés. La propriété d'extensibilité au sens fort (en Anglais "strong scalability") est obtenue pour FGMRES préconditionnée par la méthode deux niveaux du troisième chapitre et ce jusqu'à plus de 65000 coeurs. Concernant l'extensibilité au sens faible, le nombre d'itérations de FGMRES augmente linéairement avec le terme fréquentiel –résultat comparable à d'autres approches dans la littérature– toutefois pour des tailles de problèmes allant jusqu'à 2048^3. Ensuite des problèmes hétérogènes disponibles dans le domaine public sont considérés: SEG/EAGE Saltdom et SEG/EAGE Overthrust. Le préconditionnement à deux niveaux se révèle encore efficace pour les problèmes hétérogènes, même si ses propriétés d'extensibilité peuvent être dégradées. Enfin, nous utilisons les méthodes blocs du deuxième chapitre pour résoudre des problèmes d'Helmholtz hétérogènes avec plusieurs seconds membres. Nous montrons qu'il est possible de tirer avantage de la combinaison de ces méthodes blocs et du préconditionement perturbé à deux niveaux pour obtenir une méthode permettant de résoudre des problèmes réalistes à plus d'un milliard d'inconnues.

Bibliography

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández. A Lanczos-type method for multiple starting vectors. *Mathematics of Computation*, 69:1577–1601, 2000.

[2] P. R. Amestoy, I. S. Duff, J. Koster, and J. Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Analysis and Applications*, 23 (1):15–41, 2001.

[3] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods Appl. Mech. Eng.*, 184:501–520, 2000.

[4] P. R. Amestoy, A. Guermouche, J. Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[5] F. Aminzadeh, J. Brac, and T. Kunz. 3D Salt and Overthrust model. Modeling series i, Society of Exploration Geophysicists, 1997.

[6] A. H. Baker, J. M. Dennis, and E. R. Jessup. An efficient block variant of GMRES. *SIAM J. Scientific Computing*, 27:1608–1626, 2006.

[7] R. E. Bank. A comparison of two multilevel iterative methods for nonsymmetric and indefinite elliptic finite element equations. *SIAM J. Numerical Analysis*, 18:724–743, 1981.

[8] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[9] C. Beattie. Harmonic Ritz and Lehmann bounds. *Electronic Transactions on Numerical Analysis*, 7:18–39, 1998.

[10] J.-D. Benamou and B. Desprès. A domain decomposition method for the Helmholtz equation and related optimal control problems. *J. Comp. Phys.*, 136:68–82, 1997.

[11] J.-P. Berenger. A perfectly matched layer for absorption of electromagnetic waves. *J. Comp. Phys.*, 114:185–200, 1994.

[12] J.-P. Berenger. Three-dimensional perfectly matched layer for absorption of electromagnetic waves. *J. Comp. Phys.*, 127:363–379, 1996.

[13] R. F. Boisvert, R. Pozo, K. Remington, R. Barrett, and J. J. Dongarra. The Matrix Market: A web resource for test matrix collections. In Ronald F. Boisvert, editor, *Quality of Numerical Software, Assessment and Enhancement*, pages 125–137, London, 1997. Chapman & Hall.

[14] M. Bollhöfer, M. J. Grote, and O. Schenk. Algebraic multilevel preconditioner for the solution of the Helmholtz equation in heterogeneous media. *SIAM J. Scientific Computing*, 31:3781–3805, 2009.

[15] A. Brandt. A multi-level adaptative solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

[16] A. Brandt. Algebraic multigrid theory: the symmetric case. *Appl. Math. Comp.*, 19:23–56, 1986.

[17] A. Brandt, S. F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. Technical report, Institution for computational studies, Fort Collins, Colorado, 1982.

[18] A. Brandt, S. F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its Applications*, pages 257–284. Cambridge University Press, 1984.

[19] A. Brandt and S. Ta'asan. Multigrid method for nearly singular and slightly indefinite problems. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods II*, pages 99–121. Springer-Verlag, 1986.

[20] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[21] P. A. Businger and G. Golub. Linear least squares solutions by Householder transformations. *Numerische Mathematik*, 7:269–276, 1965.

[22] L. M. Carvalho, S. Gratton, R. Lago, and X. Vasseur. A Flexible Generalized Conjugate Residual Method with Inner Orthogonalization and Deflated Restarting. Technical Report TR/PA/10/10, CERFACS, Toulouse, France, 2010.

[23] A. Chapman and Y. Saad. Deflated and augmented Krylov subspace techniques. *Numerical Linear Algebra with Applications*, 4:43–66, 1996.

[24] J. F. Claerbout. *Imaging the earths interior*. Blackwell Scientific Publications, Inc, 1985.

[25] G. Cohen. *Higher-order numerical methods for transient wave equations*. Springer, 2002.

[26] F. Collino, S. Ghanemi, and P. Joly. Domain decomposition method for harmonic wave propagation : a general presentation. *Comput. Methods Appl. Mech. Engrg.*, 184:171–211, 2000.

[27] J. Cullum and T. Zhang. Two-sided Arnoldi and non-symmetric Lanczos algorithms. *SIAM J. Matrix Analysis and Applications*, 24:303–319, 2002.

[28] J. Cullum and T. Zhang. Modified Gram-Schmidt (MGS), least squares and backward stability of MGS-GMRES. *SIAM J. Matrix Analysis and Applications*, 28:264–284, 2006.

[29] T. A. Davis. *Direct methods for sparse linear systems*. SIAM, Philadelphia, 2006.

[30] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Oxford University Press, 1989.

[31] I. S. Duff, S. Gratton, X. Pinel, and X. Vasseur. Multigrid based preconditioners for the numerical solution of two-dimensional heterogeneous problems in geophysics. *International Journal of Computer Mathematics*, 84-88:1167–1181, 2007.

[32] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Trans. Math. Softw.*, 15:1–14, 1989.

[33] I. S. Duff, R. G. Grimes, and J. G. Lewis. Users' guide for the Harwell-Boeing sparse matrix collection (Release I). Technical Report TR/PA/92/86, CERFACS, 1992.

[34] I. S. Duff and J. Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM J. Matrix Analysis and Applications*, 20:889–901, 1999.

[35] I. S. Duff and S. Pralet. Strategies for scaling and pivoting for sparse symmetric indefinite problems. *SIAM J. Matrix Analysis and Applications*, 27:313–340, 2005.

[36] L. Elbouyahyaoui, A. Messaoudi, and H. Sadok. Algebraic properties of the block GMRES and block Arnoldi methods. *Electronic Transactions on Numerical Analysis*, 33:207–220, 2009.

[37] H. C. Elman, O. G. Ernst, and D. P. O'Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Scientific Computing*, 23:1291–1315, 2001.

[38] H. C. Elman and D. P. O'Leary. Eigenanalysis of some preconditioned Helmholtz problems. *Numerische Mathematik*, 83:231–257, 1999.

[39] M. Embree. The Tortoise and the Hare restart GMRES. *SIAM Review*, 45:259–266, 2003.

[40] Y. A. Erlangga. *A Robust and Efficient Iterative Method for the Numerical Solution of the Helmholtz Equation*. PhD thesis, Delt University of Technology, Netherlands, 2005.

[41] Y. A. Erlangga. Advances in iterative methods and preconditioners for the Helmholtz equation. *Archives of Computational Methods in Engineering*, 15:37–66, 2008.

[42] Y. A. Erlangga, C. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Scientific Computing*, 27:1471–1492, 2006.

[43] Y. A. Erlangga, C. Vuik, and C. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Num. Math.*, 50:409–425, 2004.

[44] C. Farhat, P. Avery, R. Tezaur, and J. Li. FETI-DPH: A dual-primal domain decomposition method for acoustic scattering. *J. Comp. Acoustics*, 13:499–524, 2005.

[45] C. Farhat and J. Li. An iterative domain decomposition method for the solution of a class of indefinite problems in computational structural dynamics. *Appl. Num. Math.*, 54:150–166, 2005.

[46] C. Farhat, A. Macedo, and M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numerische Mathematik*, 85:283–308, 2000.

[47] V. Frayssé, L. Giraud, and S. Gratton. Algorithm 881: A set of flexible GMRES routines for real and complex arithmetics on high performance computers. *ACM Trans. Math. Softw.*, 35-2:1–12, 2009.

[48] V. Frayssé, L. Giraud, S. Gratton, and J. Langou. Algorithm 842: A set of GMRES routines for real and complex arithmetics on high performance computers. *ACM Trans. Math. Softw.*, 31-2:228–238, 2005.

[49] R. W. Freund and M. Malhotra. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides. *Linear Algebra and its Applications*, 254:119–157, 1997.

[50] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60:315–339, 1991.

[51] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Scientific Computing*, 24:38–60, 2002.

[52] M. J. Gander and F. Nataf. AILU for Helmholtz problems: A new preconditioner based on the analytic parabolic factorization. *J. Comp. Acoustics*, 9:1499–1509, 2001.

[53] L. Giraud, S. Gratton, X. Pinel, and X. Vasseur. Flexible GMRES with deflated restarting. Technical Report TR/PA/09/111, CERFACS, 2009. Accepted for publication in SIAM J. Scientific Computing.

[54] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996. Third edition.

[55] S. Goossens and D. Roose. Ritz and harmonic Ritz values and the convergence of FOM and GMRES. *NLAA*, 6(4):281–293, 1999.

[56] A. Greenbaum, V. Ptak, and Z. Strakos. Any nonincreasing convergence curve is possible for GMRES. *SIAM J. Matrix Analysis and Applications*, 17:465–469, 1996.

[57] G.-D. Gu and Z.H. Cao. A block GMRES method augmented with eigenvectors. *Applied mathematics and computation*, 121:271–289, 2001.

[58] A. El Guennouni, K. Jbilou, and H. Sadok. A block version of BICGSTAB for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis*, 16:129–142, 2003.

[59] M. H. Gutknecht. Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. In A.H. Siddiqi, I.S. Duff, and O. Christensen, editors, *Modern Mathematical Models, Methods and Algorithms for Real World Systems*, pages 420–447, New Delhi, India, 2006. Anamaya Publishers.

[60] M. H. Gutknecht and T. Schmelzer. The block grade of a block Krylov space. *Linear Algebra and its Applications*, 430:174–185, 2009.

[61] W. Hackbusch. *Multi-Grid methods and Applications*. Springer, 2003. Second edition.

[62] A. Haidar. *On the parallel scalability of hybrid linear solvers for large 3D problems*. PhD thesis, CERFACS, 2009.

[63] P. Hénon and Y. Saad. A parallel multilevel ILU factorization based on a hierarchical graph decomposition. *SIAM J. Scientific Computing*, 28:2266–2293, 2006.

[64] Y. P. Hong and C. T. Pan. Rank revealing QR factorizations and the singular value decomposition. *Mathematics of Computation*, 58:213–232, 1992.

[65] B. Hustedt, S. Operto, and J. Virieux. Mixed-grid and staggered-grid finite difference methods for frequency-domain acoustic wave modelling. *Geophys. J. Int.*, 157:1269–1296, 2004.

[66] I .M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapounov equations. *SIAM J. Numerical Analysis*, 31:227–251, 1994.

[67] K. Jbilou, A. Messaoudi, and H. Sadok. Global FOM and GMRES algorithms for matrix equations. *Appl. Num. Math.*, 31(1):49–63, 1999.

[68] R. Kechroud, A. Soulaimani, Y. Saad, and S. Gowda. Preconditioning techniques for the solution of the Helmholtz equation by the finite element method. *Math. Comput. Simul.*, 65(4-5):303–321, 2004.

[69] A. Khabou. Solveur itératif haute performance pour les systèmes linéaires avec seconds membres multiples, 2009. Master thesis report, ENSEIRB/INRIA Bordeaux Sud-Ouest.

[70] S. Kim and S. Kim. Multigrid simulations for high-frequency solutions of the Helmholtz problem in heterogeneous media. *SIAM J. Scientific Computing*, 24:359–392, 2002.

[71] D. Komatitsch and J. Tromp. Introduction to the spectral-element method for 3-D seismic wave propagation. *Geophys. J. Int.*, 139:806–822, 1999.

[72] J. Langou. *Iterative methods for solving linear systems with multiple right-hand sides*. PhD thesis, CERFACS, 2003.

[73] J. Langou. For a few iterations less. In *Copper Mountain Conference on Iterative Methods, Copper Mountain (CO)*, 2004.

[74] B. Lee, T. .A Manteuffel, S. F. McCormick, and J. Ruge. First-order system least-squares for the Helmholtz equation. *SIAM J. Scientific Computing*, 21:1927–1949, 2000.

[75] G. Li. A block variant of the GMRES method on massively parallel processors. *Parallel Computing*, 23(8):1005–1019, 1997.

[76] J. Li and X. Tu. A balancing domain decomposition method by constraints for advection-diffusion problems. *Numerical Linear Algebra with Applications*, 16:745–773, 2009.

[77] I. Livshits and A. Brandt. Wave-ray multigrid method for standing wave equations. *Electronic Transactions on Numerical Analysis*, 6:162–181, 1997.

[78] I. Livshits and A. Brandt. Accuracy properties of the wave-ray multigrid algorithm for Helmholtz equations. *SIAM J. Scientific Computing*, 28:1228–1251, 2006.

[79] D. Loher. *Reliable Nonsymmetric Block Lanczos Algorithms*. PhD thesis, Swiss Federal Institute of Technology Zurich (ETHZ), Switzerland, 2005.

[80] M. M. M. Made. Incomplete factorization-based preconditionings for solving the Helmholtz equation. *Int J. Numerical Methods in Engineering*, 50:1077 – 1101, 2001.

[81] S. F. McCormick. *Multigrid methods*. SIAM, 1987.

[82] R. B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Analysis and Applications*, 16:1154–1171, 1995.

[83] R. B. Morgan. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Analysis and Applications*, 21(4):1112–1135, 2000.

[84] R. B. Morgan. GMRES with deflated restarting. *SIAM J. Scientific Computing*, 24(1):20–37, 2002.

[85] R. B. Morgan. Restarted block GMRES with deflated restarting. *Appl. Num. Math.*, 54:222–236, 2005.

[86] A. A. Nikishin and A. Yu. Yeremin. Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, i: General iterative scheme. *SIAM J. Matrix Analysis and Applications*, 16(4):1135–1153, 1995.

[87] Y. Notay. Convergence analysis of perturbed two-grid and multigrid methods. *SIAM J. Numerical Analysis*, 45:1035–1044, 2007.

[88] C. W. Oosterlee. A GMRES-based plane smoother in multigrid to solve 3D anisotropic fluid flow problems. *J. Comp. Phys.*, 130:41–53, 1997.

[89] S. Operto, J. Virieux, P. R. Amestoy, J.-Y. L'Excellent, L. Giraud, and H. Ben Hadj Ali. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. *Geophysics*, 72-5:195–211, 2007.

[90] D. Osei-Kuffuor and Y. Saad. Preconditioning Helmholtz linear systems. Technical Report UMSI-2009-30, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, 2009. Accepted for publication APNUM - 09/10/09.

[91] C. C. Paige, B. N. Parlett, and H. A. Van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2:115–134, 1995.

[92] M. Parks, E. de Sturler, G. Mackey, D.D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Scientific Computing*, 28(5):1651–1674, 2006.

[93] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*. Springer-Verlag, 2000.

[94] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 1999.

[95] C. D. Riyanti, Y. A. Erlangga, R.-E. Plessix, W. A. Mulder, C. Vuik, and C. Oosterlee. A new iterative solver for the time-harmonic wave equation. *Geophysics*, 71:57–63, 2006.

[96] C. D. Riyanti, A. Kononov, Y. A. Erlangga, R.-E. Plessix, W. A. Mulder, C. Vuik, and C. Oosterlee. A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *J. Comp. Phys.*, 224:431–448, 2007.

[97] M. Robbé and M. Sadkane. Exact and inexact breakdowns in the block GMRES method. *Linear Algebra and its Applications*, 419:265–285, 2006.

[98] S. Röllin and W. Fichtner. Improving the accuracy of GMRes with deflated restarting. *SIAM J. Scientific Computing*, 30(1):232–245, 2007.

[99] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Scientific and Statistical Computing*, 14:461–469, 1993.

[100] Y. Saad. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications*, 1:387–402, 1994.

[101] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, 2003. Second edition.

[102] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Scientific and Statistical Computing*, 7:856–869, 1986.

[103] O. Schenk and K. Gärtner. On fast factorization pivoting methods for symmetric indefinite systems. *Electronic Transactions on Numerical Analysis*, 23:158–179, 2006.

[104] B. Seynaeve, E. Rosseel, N. Bart, and S. Vandewalle. Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *J. Comp. Phys.*, 224:132–149, 2007.

[105] V. Simoncini and E. Gallopoulos. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides. *J. Comput. Appl. Math.*, 66:457–469, 1995.

[106] V. Simoncini and E. Gallopoulos. An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM J. Scientific Computing*, 16:917–933, 1995.

[107] V. Simoncini and E. Gallopoulos. Convergence properties of block GMRES and matrix polynomials. *Linear Algebra and its Applications*, 247:97–119, 1996.

[108] V. Simoncini and D. B. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numerical Analysis*, 40:2219–2239, 2003.

[109] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14:1–59, 2007.

[110] G. L. G. Sleijpen and H. A. Van der Vorst. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Analysis and Applications*, 17(2):401–425, 1996.

[111] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[112] P. Soudais. Iterative solution methods of a 3-D scattering problem from arbitrary shaped multidielectric and multiconducting bodies. *IEEE Trans. on Antennas and Propagation*, 42 (7):954–959, 1994.

[113] K. Stüben and U. Trottenberg. Multigrid methods: fundamental algorithms, model problem analysis and applications. In W. Hackbusch and U. Trottenberg, editors, *Multigrid methods, Koeln-Porz, 1981, Lecture Notes in Mathematics, volume 960*. Springer-Verlag, 1982.

[114] A. Toselli and O. Widlund. *Domain Decomposition methods - Algorithms and Theory*. Springer Series on Computational Mathematics, Springer, 34, 2004.

[115] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press Inc., 2001.

[116] N. Umetani, S. P. MacLachlan, and C. W. Oosterlee. A multigrid-based shifted Laplacian preconditioner for fourth-order Helmholtz discretization. *Numerical Linear Algebra with Applications*, 16:603–626, 2009.

[117] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Scientific Computing*, 13:631–644, 1992.

[118] P. Vanek, J. Mandel, and M. Brezina. Two-level algebraic multigrid for the Helmholtz problem. *Contemp. Math.*, 218:349–356, 1998.

[119] J. Virieux. SH wave propagation in heterogeneous media, velocity-stress finite difference method. *Geophysics*, 49:1259–1266, 1984.

[120] J. Virieux, S. Operto, H. Ben Hadj Ali, R. Brossier, V. Etienne, F. Sourbier, L. Giraud, and A. Haidar. Seismic wave modeling for seismic imaging. *The Leading Edge*, 25(8):538–544, 2009.

[121] B. Vital. *Etude de quelques méthodes de résolution de problème linéaire de grande taille sur multiprocesseur*. PhD thesis, Université de Rennes, 1990.

[122] H. Waisman, J. Fish, R. Tuminaro, and J. Shadid. The generalized global basis (GGB) method. *Int J. Numerical Methods in Engineering*, 61:1243–1269, 2004.

[123] H. Waisman, J. Fish, R. Tuminaro, and J. Shadid. Acceleration of the generalized global basis (GGB) method for nonlinear problems. *J. Comp. Phys.*, 210:274–291, 2005.

[124] T. Washio and C. W. Oosterlee. Flexible multiple semicoarsening for three-dimensional singularly perturbed problems. *SIAM J. Scientific Computing*, 19:1646–1666, 1998.

[125] R. Wienands, C. W. Oosterlee, and T. Washio. Fourier analysis of GMRES(m) preconditioned by multigrid. *SIAM J. Scientific Computing*, 22:582–603, 2000.

[126] I. Zavorin, D. P. O'Leary, and H. Elman. Stagnation of GMRES. *Linear Algebra and its Applications*, 367:165–183, 2003.

[127] P. M. De Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comput. Appl. Math.*, 33:1–27, 1990.