Arch Comput Methods Eng (2008) 15: 37–66
DOI 10.1007/s11831-007-9013-7

Advances in Iterative Methods and Preconditioners for the Helmholtz Equation

Yogi A. Erlangga

Received: 9 February 2007 / Accepted: 9 February 2007 / Published online: 6 December 2007

Abstract In this paper we survey the development of fast iterative solvers aimed at solving 2D/3D Helmholtz problems. In the first half of the paper, a survey on some recently developed methods is given. The second half of the paper focuses on the development of the shifted Laplacian preconditioner used to accelerate the convergence of Krylov subspace methods applied to the Helmholtz equation. Numerical examples are given for some difficult problems, which had not been solved iteratively before.

1 Introduction

The linear system arising from a discretization of the Helmholtz equation in a 2D or 3D domain is typically characterized by indefiniteness of the eigenvalues of the corresponding coefficient matrix. With such a property, an iterative method—either basic or advanced—encounters convergence problems. The method usually converges very slowly or diverges. While the Helmholtz equation finds applications in many important fields, e.g., in aeroacoustics, under-water acoustics, seismic inversion and electromagnetics, the ability to solve the Helmholtz equation efficiently may limit its potential applications. For high frequency problems, e.g. in seismic survey, and in 3D, the number of gridpoints grows very rapidly in order to maintain accuracy; the error is proportional to $k^{p+1}h^p$, if p is the order of discretization and h the grid size [11, 73, 74]. The linear system becomes extremely large and highly indefinite. This makes the problem even harder to solve. Direct methods [58] easily suffer from excessive fill-in (especially in 3D, due to the large bandwidth) and inacceptable computational work (which grows like N^3, if N is the number of unknowns). While parallel direct solvers for sparse linear systems have also been quite advanced nowadays (see, e.g., PARDISO [107, 108]), cheaper alternatives—less memory requirement and hopefully faster—are still sought.

The traditional counterpart for direct methods is iterative methods. Known for its low memory requirement, an iterative method usually consists only of a few (usually one or two) matrix-vector multiplications, some inner products of two vectors and vector updates per iteration. Among modern iterative methods are Krylov subspace iterative methods, multigrid and domain decomposition methods.

Iterative solutions for the Helmholtz equation have been an active research field since the 1980s (see, e.g., [10] and [62]). Since then many attempts have been spent to develop a powerful iterative method for solving it. So far, no standard, well-known generic iterative methods work efficiently for the Helmholtz equation.

For Krylov subspace methods, the main difficulty is to find a good preconditioner for the *indefinite* Helmholtz equation. Even a Krylov subspace method with a preconditioner particularly designed for the indefinite linear systems, e.g. [53], mostly fails to result in a satisfactory convergence.

For multigrid, indefiniteness arises difficulties in having both good smoothing property and constructing appropriate coarse-grid approximations of the problem, which are responsible for further reduction of low frequency errors.

In the recent years, however, some advances have been made within the standard iterative method framework. This paper reviews and highlights some advances in iterative methods for the Helmholtz equation, in 2D and 3D. This is the first part of this paper. In the second part, we particularly

Y.A. Erlangga (✉)
TU Berlin, Institut für Mathematik, MA 3-3, Strasse des 17. Juni 136, 10623 Berlin, Germany
e-mail: erlangga@math.tu-berlin.de

Reprinted with permission from Archives of Computational Methods in Engineering, 15, 37–66.

focus on Krylov subspace methods and the shifted Laplacian preconditioner, a classification introduced in [41] but with the original work going back to [10]. We give some of the theories behind the shifted Laplacian preconditioner, motivate its gradual evolution to the current version and present some extremely difficult Helmholtz problems which are successfully solved by this method.

In Sect. 2 we briefly discuss the Helmholtz equation, boundary conditions required for computations and discretization methods. Krylov subspace, multigrid and domain decomposition methods are revisited in Sect. 3. Related to Krylov subspace methods, we discuss the development of some modern *ad hoc* preconditioners in Sect. 4, which will be focused on incomplete LU decompositions. Section 5 is then devoted to the shifted Laplacian preconditioner. Multigrid as the preconditioner solver within the context of the Helmholtz equation and the shifted Laplacian preconditioner is discussed in Sect. 6. In Sect. 7 numerical results are presented for realistic problems. We end the paper with concluding remarks in Sect. 8.

Complementary to this survey is a survey paper by Turkel [124], to which the reader is also referred.

2 The Helmholtz Equation

In this section we discuss briefly the Helmholtz equation, the computational boundary conditions and their discretization.

2.1 Derivation

The Helmholtz equation can be derived from the general time-dependent two-way wave equation [17, 30, 31, 60] in $\Omega \subset \mathbb{R}^d$, $d = 1, \ldots, 3$:

$$\frac{\partial^2 p(\boldsymbol{x},t)}{\partial t^2} - \frac{1}{c^2}\nabla \cdot \nabla p(\boldsymbol{x},t) = 0, \quad \boldsymbol{x} = (x_1, x_2, x_3), \tag{1}$$

by assuming time-harmonic waves. This assumption allows the time-dependent pressure p to be decomposed as

$$p(\boldsymbol{x},t) = u(\boldsymbol{x}) \exp(-\hat{j}\omega_w t), \tag{2}$$

where $\omega_w > 0$ and $\hat{j} = \sqrt{-1}$ denote the angular frequency and the imaginary unit, respectively. Substitution of (2) into (1) yields

$$-\nabla \cdot \nabla u(\boldsymbol{x}) - k^2 u(\boldsymbol{x}) = 0, \quad k = \frac{\omega}{c}, \tag{3}$$

where $\omega = 2\pi f$ is the angular frequency, f is the frequency in hertz, and c the local speed of sound (in m/s). c in this case may vary in space. Equation (3) is called the Helmholtz equation.

In a more general formulation, one can also include a forcing term in the right-hand side of (1). If this term is also assumed to be time-harmonic, the resultant Helmholtz equation reads

$$-\nabla \cdot \nabla u(\boldsymbol{x}) - k^2 u(\boldsymbol{x}) = g(\boldsymbol{x}), \tag{4}$$

with $g(\boldsymbol{x})$ the forcing term.

For a scaled problem, i.e. problem defined in a unit domain $\Omega = (0,1)^d$, the Helmholtz equation can be made dimensionless by introducing a characteristic length $\bar{l}$. The nondimensional length is then determined as $\widetilde{x}_1 = x_1/\bar{l}$ and so on. Substitution of the nondimensional length into (4) yields

$$-\widetilde{\nabla} \cdot \widetilde{\nabla} u(\widetilde{\boldsymbol{x}}) - \bar{k}^2 u(\widetilde{\boldsymbol{x}}) = g(\widetilde{\boldsymbol{x}}). \tag{5}$$

Here, $\bar{k}$ is the wavenumber in the nondimensional domain and can be related to the physical quantities in the physical domain by the relation

$$\bar{k} = 2\pi f \bar{l}/c. \tag{6}$$

$\bar{k}$ is sometimes also referred to as the reduced frequency.

In the discussion to follow we will use the notation k for wavenumbers regardless the domain we are considering. However, the meaning should be clear from the context. If we consider a unit domain, the wavenumber should be dimensionless.

2.2 Boundary Conditions and Perfectly Matched Layer

In order to solve the Helmholtz equation (4) boundary conditions have to be imposed on $\Gamma = \partial\Omega$. If one is concerned with waves propagating to infinite distance, the so-called *radiation* or Sommerfeld condition is needed. See [60] for the derivation of such a condition.

In practice one needs to limit the size of the computational domain. In this case the Sommerfeld condition is no longer satisfied at finite distance. In order to mimic the out-going waves and to ensure that there is no non-physical reflection of the outgoing waves from this artificial boundary the non-reflecting (absorbing) boundary conditions must be used. There exist several formulations for the absorbing boundary conditions [8, 9, 28, 40]. The first order absorbing boundary condition reads as follows [40], defined for 3D case:

$$\mathcal{B}_1 u|_{\text{face}} := \left(\frac{\partial}{\partial \eta} - \hat{j}k\right) u = 0, \tag{7}$$

where η is the direction normal to the boundary, pointing outwardly.

Less reflections from the computational boundary can be produced by the second order absorption boundary conditions, defined for 3D problems [40, 71]:

- Faces:

$$\mathcal{B}_2 u|_{\text{face}} := \pm\frac{\partial u}{\partial x_i} - \hat{j}ku - \frac{\hat{j}}{2k}\sum_{j=1,j\neq i}^{3}\frac{\partial^2 u}{\partial x_j^2} = 0, \quad i = 1,\ldots,3. \tag{8}$$

  Here x_i is the coordinate perpendicular to the face.
- Edges:

$$\mathcal{B}_2 u|_{\text{edge}} := -\frac{3}{2}k^2 u - \hat{j}k\sum_{j=1,j\neq i}^{3}\left(\pm\frac{\partial u}{\partial x_j}\right) - \frac{1}{2}\frac{\partial^2 u}{\partial x_i^2} = 0, \quad i = 1,\ldots,3, \tag{9}$$

  with x_i the coordinate parallel to the edge.
- Corners:

$$\mathcal{B}_2 u|_{\text{corner}} := -2\hat{j}ku + \sum_{i=1}^{3}\pm\frac{\partial u}{\partial x_i} = 0. \tag{10}$$

In (8–10) the $\pm$ sign is determined such that for out going waves the non-reflecting condition is satisfied.

We note here that the inclusion of some boundary conditions may lead to a nonsymmetric system (e.g., the BGT condition [9]). In such a case, from an iterative method point of view both nonsymmetry and indefiniteness of the system may lead to even harder linear systems to solve.

Damping/Sponge Layer To further reduce the non-physical reflections at the boundaries, one can add extra region Ω_d surrounding the original computational domain Ω as illustrated in Fig. 1. In Ω_d the waves are modeled by the *damped* Helmholtz equation:

$$-\Delta u(\boldsymbol{x}) - (1 - \alpha\hat{j})k^2(\boldsymbol{x})u(\boldsymbol{x}) = g(\boldsymbol{x}), \tag{11}$$

with $0 \leq \alpha \ll 1$ indicating the fraction of damping in the medium. In geophysical applications, for example, α is related to the quality factor $Q = 1/\alpha$, which is usually $20 < Q < 100$. Therefore, α can be set as high as 0.05. This additional layer is sometimes called as "the damping layer" or "the sponge layer". In this case, however, α is set equal to zero at the interface and increases quadratically outwardly. So, the outgoing waves are gradually damped in the damping layer, and so are the non-physical reflections if any.

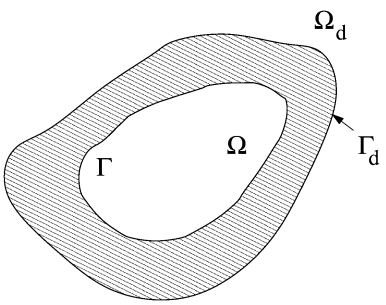

Fig. 1 Absorption layer

Perfectly Matched Layer Similar to the damping layer, a more advanced technique to reduce the non-physical reflections is achieved by using Perfectly Matched Layers (PML) [1, 2, 15, 16, 123]. The PML Helmholtz equation reads, which is due to Tsynkov and Turkel [123] and shown for 2D,

$$-\frac{\partial}{\partial x_1}\left(\frac{s_2}{s_1}\frac{\partial u}{\partial x_1}\right) - \frac{\partial}{\partial x_2}\left(\frac{s_1}{s_2}\frac{\partial u}{\partial x_2}\right) - \frac{\omega^2}{c^2}s_1 s_2 u = g, \quad \text{in } \Omega_p = \Omega \cup \Omega_d, \tag{12}$$

with

$$s_1 = 1 + \frac{\sigma_1}{\hat{j}\omega}, \qquad s_2 = 1 + \frac{\sigma_2}{\hat{j}\omega}. \tag{13}$$

In (13), σ_1 and σ_2 are real and non-negative, and are represented by piecewise smooth functions depending only on x_1 and x_2 respectively. In Ω, σ_1 and σ_2 are equal to zero, and (12) reduces to the standard Helmholtz equation. In Ω_d, a linear function is often sufficient for σ_1 and σ_2, for example

$$\sigma_1 = \alpha_1 x_1, \qquad \sigma_2 = \alpha_2 x_2, \qquad \alpha_1, \alpha_2 \in \mathbb{R}_+. \tag{14}$$

Furthermore, the value of $k = \omega/c$ in Ω_d is set equal to the value of k on $\Gamma = \partial\Omega$. In addition, the Dirichlet boundary condition $u = 0$ is imposed on $\partial\Omega_d$.

Discretization of the damped Helmholtz equation or the PML Helmholtz equation eventually leads to a symmetric system; this is more favorable than the accurate radiation condition (which is possibly nonsymmetry and non-local). This, however, must be done in expense of extra work to resolve solutions in this additional but only artificial layers.

2.3 Discretization

Either finite differences or finite elements can be used to discretize the Helmholtz equation. Finite elements for the Helmholtz equation are, among others, discussed in references [5–7, 34, 73–75], and a survey paper [69]. In this paper, we mostly base our numerical examples on finite difference discretizations of the Helmholtz equation. This is not however the restriction of iterative methods explained here.

A general discussion on finite difference methods for partial differential equations can be found, e.g. in [113]. The development of a high order finite difference methods is given, e.g. in [12, 87, 88, 117]. There exist different finite difference approximations for the Helmholtz equation (see e.g. [70, 76, 109]).

The prototype of a (compact) finite difference scheme is usually written in the (nine-point) stencil notation:

$$A_{h,\delta} \triangleq \begin{bmatrix} a_7 & a_8 & a_9 \\ a_4 & a_5 & a_6 \\ a_1 & a_2 & a_3 \end{bmatrix}, \tag{15}$$

where the exact value of the entries $a_i, i = 1, \ldots, 9$ depends on the type of discretization. A second order finite difference scheme for second order elliptic operators has a stencil representation with $a_1 = a_3 = a_7 = a_9 = 0$, leading to the five-point stencil. For higher order schemes the entries a_i are nonzero.

In particular to the Helmholtz equation, a good scheme not only has to be high order accurate to reduce the required minimum number of grid point per wavelength, but also has to lead to minimal dispersion and anisotropy. For some of the 9-point stencil finite difference schemes, see, e.g., [70] and [109], or [76], which is not so often cited by mathematicians but widely used in geophysical community. Studies on a sixth-order accurate finite difference scheme which is also applicable to the PML equation can be found in [123] and [110].

For the PML Helmholtz equation, and as well the Helmholtz equation with damping layers, however, it is not necessary to apply a high order scheme in both Ω and Ω_d, since the additional layers are only artificial, and a very accurate solution in those layers are not needed. So, one can for example use a nine-point stencil discretization in Ω and the five-point stencil in Ω_d.

3 Iterative Methods for the Helmholtz Equation

Discretization of the Helmholtz equation results in the linear system

$$Au = g, \quad A \in \mathbb{C}^{N\times N}, \tag{16}$$

where N is the number of unknowns. The matrix A is complex-valued because of the boundary conditions. Furthermore, A is mostly indefinite. Only in a few cases which are often not of practical interest A is definite. By "indefinite" we means that the real part of the eigenvalues of A lie in both positive and negative half plane in the complex plane. We should also note here that the dimension of A increases if the frequency/wave number is increased. But, since we use finite differences or finite elements A is sparse.

In the sequel we discuss briefly some iterative methods meant for solving a linear system like (16).

3.1 Basic Iterative Methods

Basic iterative methods are fixed-point iterations which are based on the splitting

$$A = F - G, \quad F, G \in \mathbb{C}^{N\times N}.$$

After substitution into (16), we have

$$(F - G)u = g \iff Fu = g + Gu. \tag{17}$$

For u^{j-1} the approximate solution after the $(j-1)$-th iteration, the new approximation u^j can be computed as

$$Fu^j = g + Gu^{j-1} \Longrightarrow u^j = F^{-1}(g + Gu^{j-1}). \tag{18}$$

Thus,

$$\begin{aligned} u^j &= F^{-1}g + (I - F^{-1}A)u^{j-1} \\ &= u^{j-1} + F^{-1}r^{j-1}, \end{aligned} \tag{19}$$

with $r^{j-1} := g - Au^{j-1}$ the residual after the $(j-1)$-th iteration, and I the identity matrix. Equation (19) is called the *basic iterative* method, and it is convergent if $u^j = u^{j-1}$, meaning that $r^{j-1} = 0$.

The basic iteration is distinguished by the way the splitting is chosen. If the splitting is defined by $A = D - E$, where $D = \mathrm{diag}(A)$, the *Jacobi* iteration results, namely

$$u_h^j = u_h^{j-1} + D_h^{-1}r_h^{j-1}. \tag{20}$$

The *Gauss-Seidel* iteration is obtained from the splitting $A = L - U$, where L and U are lower and upper triangular matrices, respectively. This iteration is written as

$$u^j = u^{j-1} + L^{-1}r^{j-1}. \tag{21}$$

To improve the convergence of the fixed point iterations, a relaxation factor ω can be introduced in (20) and (21). For Jacobi iterations, for example, we then have

$$u^j = u^{j-1} + \frac{1}{\omega_r}D^{-1}r^{j-1}. \tag{22}$$

A standard approach to investigate the convergence of this type of methods is by analyzing the amplitude of Fourier modes between two successive iterations. For indefinite Helmholtz problems, it can be shown that there exists no ω_r such that the error is reduced during the iterations. This means that the Jacobi iteration always diverges if applied to the *indefinite* Helmholtz equation [68]. Similar result holds for the Gauss-Seidel iteration.

A convergent Jacobi iteration can be designed by using two stages Jacobi iteration with different relaxation factors. This method is proposed by Hadley in [68]. In this case, we have fixed point iteration like

$$u^{j-\frac{1}{2}} = u^{j-1} + \frac{1}{\omega_{r,1}}D^{-1}r^{j-1}, \tag{23}$$

$$u^j = u^{j-\frac{1}{2}} + \frac{1}{\omega_{r,2}}D^{-1}r^{j-\frac{1}{2}}, \tag{24}$$

where $\omega_{r,1} \neq \omega_{r,2} \in \mathbb{C}$. From Fourier analysis in order to have definite eigenvalues of the Helmholtz equation, we must set $\omega_{r,1} = -\omega_{r,2}^*$. Furthermore, to obtain a convergent method it is required that $\mathrm{Im}(\omega_{r,1}) < 0$. The largest convergence rate is reached if $\omega_{r,1} = \sqrt{3} - \hat{j}$. While the resulting algorithm is very simple, the convergence, however, is still considered somewhat slow; see numerical examples in [68, 132, 133].

3.2 Krylov Subspace Methods

The Krylov subspace iteration methods are developed based on a construction of consecutive iterants in a Krylov subspace, i.e. a subspace of the form

$$\mathcal{K}^j(A, r^0) = \mathrm{span}\{r^0, Ar^0, A^2r^0, \ldots, A^{j-1}r^0\}, \tag{25}$$

where $r^0 := g - Au^0$ is the initial residual related to the initial guess u^0. The dimension of $\mathcal{K}^j$ is equal to j and increases by one at each step of the approximation process.

The idea of Krylov subspace methods can be outlined as follows [63, 105, 127]. For an initial solution u^0, approximations u^j to the solution u are computed every step by iterants u^j of the form

$$u^j \in u^0 + \mathcal{K}^j(A, r^0), \quad j > 1. \tag{26}$$

The Krylov subspace $\mathcal{K}^j$ is constructed by the basis $v^1, v^2, \ldots, v^j$, where

$$V^j = [v^1, v^2, \ldots, v^j] \in \mathcal{K}^j. \tag{27}$$

With residual $r^j = g - Au^j$, (26) gives an expression for the residual at the j-th step

$$r^j = r^0 - AV^j y^j, \tag{28}$$

where $y^j \in \mathbb{C}^N$ and $u^j = u^0 + V^j y^j$. From (28) we observe that Krylov subspace methods rely on constructing the basis of V^j and the vector y^j. In general we identify two methods that can be used for constructing the basis of V^j: Arnoldi's method [4] and Lanczos' method [83, 84]. The vector y^j can be constructed by a residual projection or by a residual norm minimization method.

Among methods which are based on construction the Krylov subspace methods are Conjugate Gradient (CG) [72], COCG [128], GMRES [106], CGS [112], Bi-CGSTAB [126] and QMR [54].

CG can be derived from the symmetric analogue of Arnoldi's method. The resultant process is characterized by short-recurrences and can be realized in an algorithm consisting only of one multiplication of a vector with the matrix A. The residuals produced during the iterations satisfy the orthogonality condition based on the true inner product, i.e.,

$$(r^{j+1}, r^j) = (r^j)^* r^j = 0, \tag{29}$$

where x^* means the transpose conjugate of a vector x. During the iteration, CG minimizes the A-norm of the error for A a symmetric positive definite (SPD) matrix. In fact, for symmetric positive definite A we have the following bound:

$$\|u - u^{j+1}\|_A \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^j \|u - u^j\|_A, \tag{30}$$

for u the exact solution of $Au = g$, κ the condition number (of A), and $\|x\|_A = (Ax, x)^{1/2}$. In case A is indefinite the method does not converge.

We notice, however, that A^*A is symmetric positive definite, where A^* is the transpose conjugate of A. Thus we can apply CG on A^*A, and an algorithm called CGNR results. CGNR, however, requires two matrix-vector multiplications (which is one more than required by CG), and in general converges too slowly. Therefore, CGNR is not used so widely in practice. Only for few problems does CGNR find its applications.

A short recurrence algorithm can be constructed for symmetric but non-Hermitian systems. This is achieved, for example, by replacing the true inner product (29) by $(x, y) = x^*y = \overline{x}^*y$ by $(\overline{x}, y) = x^T y$. The new orthogonality condition is defined according to this inner product:

$$(\overline{r}^{j+1}, r^j) = (r^j)^T r^j = 0. \tag{31}$$

An algorithm called conjugate orthogonal-conjugate gradient (COCG) [128] results. COCG is very similar to CG. COCG, however, does not have a minimization property. It is typical for COCG that the convergence is characterized by an erratic behavior of the residuals. One can smooth the convergence by, e.g., applying residual smoothing [142].

For nonsymmetric linear systems, the Krylov subspace can be built from Arnoldi's process, which leads to, e.g., GMRES [106]. GMRES is an optimal method; it reduces the 2-norm of the residual at every iteration. GMRES, however, requires long recurrences, which is usually limited by the available memory. A remedy is by restarting, which sometimes leads to slow convergence or stagnation.

Some algorithms for solving nonsymmetric systems can be constructed from the Lanczos bi-orthogonalization process. The original version is called BiCG [50]. In addition to a multiplication of a vector with A, BiCG requires another one vector multiplication with A^T. If a solution related to A^T is also needed, then BiCG may be a good method, because the algorithm solve the linear system $A^T x = y$ as well. Sonneveld [112] observes that if p_m is any polynomial, then $(p_m(A)x, p_m(A^T)y) = (p_m^2(A)x, y)$. This leads to a transpose-free version of BiCG called CGS. Both Bi-CG and CGS have no optimality property, and convergence is often characterized by an erratic behavior. If BiCG converges, then the convergence of CGS is faster than BiCG. Introducing stabilization in CGS results in Bi-CGSTAB [127].

Bi-CGSTAB is a short recurrence algorithm, but, likewise CGS, it requires two matrix-vector multiplications with A, i.e. one multiplication more than needed by GMRES. Furthermore, it is lack of optimality property. Nevertheless, Bi-CGSTAB is often considered as an alternative to GMRES. Methods based on Lanczos process however are susceptible to break down. One cure for breakdown is by implementing look-ahead-strategies [25, 65].

To have a rather smooth convergence and at the same time to maintain short recurrences, the Lanczos process can be replaced by another sequence of approximate solutions with associated with residuals satisfying quasi-minimal property. This leads to a process called QMR [54].

Numerical Example To show convergence of various Krylov subspace methods applied to the Helmholtz equation, we consider the 2D Helmholtz equation defined in $\Omega = (0, 1)^2$, with the first order absorbing conditions (7) on $\Gamma = \partial\Omega$. We compute the approximate solution with CGNR, COCG, Bi-CGSTAB, QMR and GMRES, starting with an initial guess. The iteration is terminated at the j-th iteration if the relative residual is less than 10^{-7}.

Convergence results are presented in Table 1. Notice that with k increasing, the performance of all methods deteriorates. In this case, COCG seems to be the method of choice (short recurrence and with comparable convergence as GMRES). It also appears that the convergence of GMRES is slightly faster than QMR.

Convergence history for $k = 20$ is displayed in Fig. 2. An erratic convergence behavior typical for COCG is observed. Superlinear convergence [129] is clearly seen for GMRES and CGNR. A long, slow converging part of CGNR convergence is due to many small eigenvalues related to the normal equations A^*A.

The effect of increasing the number of gridpoints per wavelength on the number of iterations for $k = 20$ is presented in Table 2. For all algorithms, the convergence slows down if h is decreased, with CGNR the most sensitive algorithm in this regard. Typically similar convergence behaviors are observed in GMRES and QMR.

Since the matrix A is complex symmetric, one can replace QMR by the symmetric QMR (SQMR) as described in [52].

Efficient Krylov subspace methods can be obtained if preconditioning is included. This is discussed in Sect. 4.

Table 1 Number of iterations of several Krylov subspace methods to reach convergence for the Helmholtz equation with constant wavenumbers. $kh = 0.625$ ($\sim$10 gridpoints per wavelength)

k	CGNR	COCG	Bi-CGSTAB	QMR	GMRES
10	49	33	35	32	32
20	208	80	136	80	79
30	492	154	429	154	143
40	943	255	816	255	241

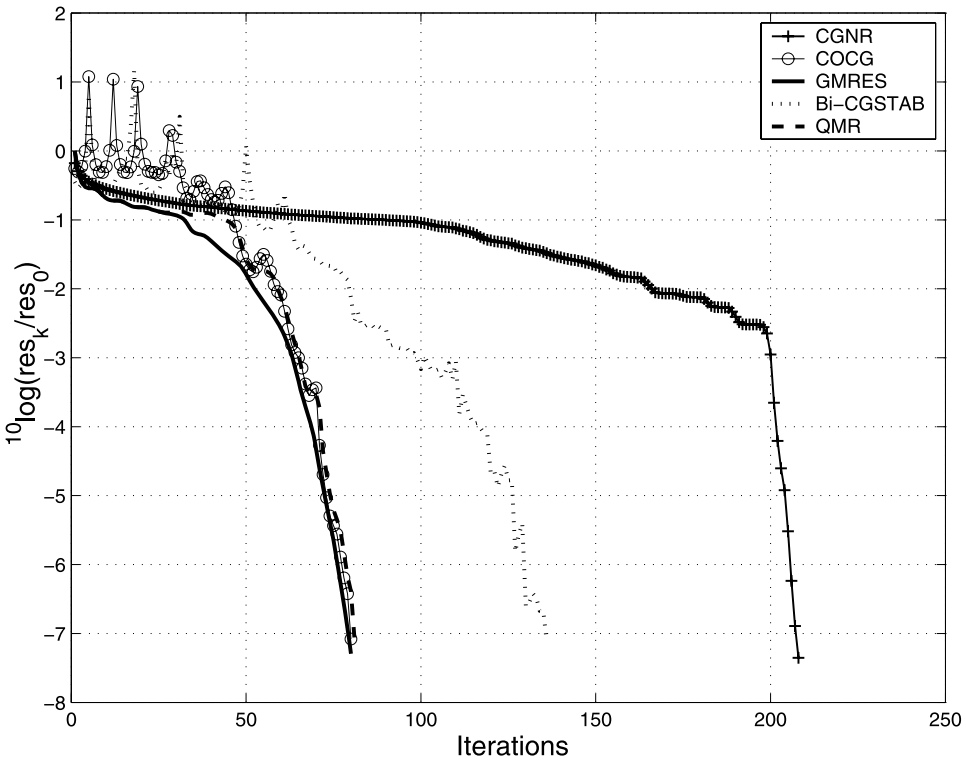

Fig. 2 Convergence history for several Krylov subspace methods. $k = 20$, and $kh = 0.625$

Table 2 Number of iterations of several Krylov subspace methods for different numbers of gridpoints per wavelength. $k = 20$

kh	CGNR	COCG	Bi-CGSTAB	QMR	GMRES
0.6250	208	80	136	80	79
0.3750	775	165	356	165	161
0.1875	1907	260	627	261	252

3.3 Multigrid Methods

Multigrid is known for its efficiency and robustness when it is applied to elliptic equations with self-adjoint operators [21, 26, 67, 122, 139]. A multigrid method exhibits fast convergence independent of grid size, h, and with a complexity of $O(N \log N)$. To achieve such a convergence, however, requires a careful design of all components in a multigrid method.

The efficiency of multigrid methods to solve a linear system arising from an elliptic partial differential equation comes from two facts. First of all, many classical iterative methods have a strong smoothing effect on the error of any approximation. The error is not necessarily small, but smooth. This is called *smoothing*, which is the first principle of multigrid. The second principle is the so called *coarse-grid correction* (CGC). Any smooth quantity on a fine grid can be well approximated on a coarse grid by a suitable procedure.

A multigrid method can be explained by considering a *two-grid* method, with a grid sequence Ω_h, Ω_H, where Ω_h is the fine grid and Ω_H the corresponding coarse grid obtained by doubling the grid size, i.e. $H = 2h$, which is called *standard coarsening* of Ω_h. (When we discuss multigrid, we will use the subscript $_h$ and $_H$ to indicate vectors and matrices related to the fine and coarse grid, respectively.)

To see how the two multigrid principles work on two grid levels, we consider the Poisson equation: $-\Delta u = g$, with Dirichlet boundary conditions, discretized by the five-point finite difference stencil. This leads to a linear system $L_h u_h = g_h$, with $L_h = -\Delta_h$, the discrete negative Laplacian. The domain is the unit square. If u_h^j is an approximation of u_h after the j-th iteration, the error between these two quantities can be written as

$$v_h^j := u_h - u_h^j = \sum_{\ell_1,\ell_2=1}^{n-1} \alpha_{\ell_1,\ell_2} \sin(\pi \ell_1 x_1) \sin(\pi \ell_2 x_2). \quad (32)$$

For $(x_1, x_2) = \Omega_h$, the discrete operator Δ_h has eigenfunctions

$$\phi_h^{\ell_1,\ell_2}(x_1, x_2) = \sin(\pi \ell_1 x_1) \sin(\pi \ell_2 x_2), \quad \ell_1, \ell_2 = 1, 2, \ldots, \sqrt{N} - 1, \quad (33)$$

with N the total unknowns. We consider four eigenfunctions

$$\phi_h^{\ell_1,\ell_2}, \quad \phi_h^{\sqrt{N}-\ell_1,n-\ell_2}, \quad \phi_h^{\sqrt{N}-\ell_1,\ell_2}, \quad \phi_h^{\sqrt{N}-\ell_1,\ell_2}. \quad (34)$$

On the coarse grid Ω_H, we observe that

$$\phi_H^{\ell_1,\ell_2} = -\phi_H^{\sqrt{N}-\ell_1,\ell_2} = -\phi_H^{\ell-1,\sqrt{N}-\ell_2} = \phi_H^{\sqrt{N}-\ell_1,\sqrt{N}-\ell_2}. \quad (35)$$

This means that the four eigenfunctions (34) cannot be distinguished on Ω_H. Since the high frequencies coincide with the low frequencies on Ω_H, only low frequencies are visible on Ω_H. An iterative method with good smoothing properties is one that annihilates the high frequency components quickly.

Now the residual is determined by

$$r_h^j := g_h - L_h u_h^j, \quad (36)$$

which is equivalent to the defect equation:

$$L_h v_h^j = r_h^j. \quad (37)$$

To solve the defect equation, the two-grid method benefits from the smooth error and uses the approximation of this smooth error on the coarse grid to solve this defect equation *approximately*. Thus, instead of solving (37), in the two-grid method one solves

$$L_H \widehat{v}_H^j = r_H^j. \quad (38)$$

The size of the matrix L_H is substantially smaller than the size of the matrix L_h. Therefore, the solution of the defect equation is cheap to obtain. Since r_H^j and $\widehat{v}_H^j$ are functions defined on the coarse grid Ω_H, two transfer operators are required to relate the fine-to-coarse grid functions. The first transfer operator is used to *restrict* r_h^j to Ω_H, such that

$$r_H^j := I_h^H r_h^j, \qquad I_h^H : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_H). \quad (39)$$

The second operator is used to interpolate (*prolongate*) the correction $\widehat{v}_H^j$ back to Ω_h:

$$\widehat{v}_h^j := I_H^h \widehat{v}_H^j, \qquad I_H^h : \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_h). \quad (40)$$

Using this correction, the new approximation can then be computed:

$$u_h^{j+1} = u_h^j + \widehat{v}_h^j. \quad (41)$$

This is called the coarse-grid correction. The coarse-grid correction by itself is, however, not a converging method, as high frequency errors are not annihilated.

A two-grid cycle is a cycle consisting of presmoothing, coarse-grid correction and postsmoothing. In pre- and postsmoothing, a classical iterative method with a smoothing property is used. In the coarse-grid correction, the defect equation is solved on Ω_H.

In multigrid methods we consider more than two grid levels given in a sequence of grids $\Omega_h, \Omega_{2h}, \Omega_{4h}, \ldots, \Omega_{h_0}$, with Ω_{h_0} is the coarsest grid, and recursively apply the same principles as in the two-grid case. Here, the solution of the defect equation on Ω_{2h} is approximately computed by the two-grid method, and so on. On Ω_{h_0} exact solution of the problem can take place. This in general leads to a very efficient iterative for some classes of problems.

Multigrid is, however, not an effective method for the *indefinite* Helmholtz equation. To see this, we consider the eigenvalues of the discrete Helmholtz operator (with homogeneous Dirichlet boundary conditions) [23, 39, 43]:

$$\begin{aligned}\lambda_h^{\ell_1,\ell_2} &= \mu_h^{\ell_1,\ell_2} - k^2 \\ &\equiv \frac{2}{h^2}(2 - \cos\pi\ell_1 h - \cos\pi\ell_2 h) - k^2, \\ \ell_1, \ell_2 &= 1, \ldots, \sqrt{N} - 1. \end{aligned} \tag{42}$$

These eigenvalues are not equal to zero as long as k^2 is not equal to any of the eigenvalues of the corresponding discrete Laplace operator $\mu_h^{\ell_1,\ell_2}$. Otherwise, the matrix is singular and its null-space is spanned by the eigenfunctions

$$\widetilde{u}_h^{\ell_1\ell_2} = \sin(\pi\ell_1 x_1)\sin(\pi\ell_2 x_2), \tag{43}$$

with ℓ_1, ℓ_2 for which $\lambda_h^{\ell_1,\ell_2} = 0$. Denote by $\mu_h^{1,1}$ the minimum eigenvalue of the Laplace operator. For $k^2 > \mu_h^{1,1}$, the matrix has both positive and negative eigenvalues. Pointwise Jacobi iteration with underrelaxation does not *converge* in that case, but since its smoothing properties are satisfactory, the multigrid convergence will deteriorate only gradually for k^2 increasing. By the time k^2 approaches the 6th eigenvalue $\mu_h^{\ell_1,\ell_2}$ ($k^2 \approx 150$), standard multigrid diverges. The Jacobi relaxation now diverges for smooth eigenfrequencies $\widetilde{u}_h^{\ell_1,\ell_2}$ with $\mu_h^{\ell_1,\ell_2} < k^2$. Consequently, multigrid will still converge as long as the coarsest level used is fine enough to represent these smooth eigenfrequencies sufficiently. So, the coarsest level chosen limits the convergence. See also the similar observation in [61, 66]. When k^2 gets larger more variables need to be represented on the coarsest level for standard multigrid convergence. Eventually, this does not result in an $O(N \log N)$ iterative method.

Furthermore, eigenvalues close to the origin may undergo a sign change after discretization on a coarser grid. If a sign change occurs the coarse-grid solution does not give a convergence acceleration to the finer grid problem, but a severe convergence degradation (or even divergence) instead.

Elman et al. in [39] propose a remedy for the coarse-grid correction related to these problematic eigenvalues. The remedy includes GMRES preconditioned by multigrid, and GMRES for reducing errors at the intermediate levels. The convergence results achieved are impressive, but in order to achieve that quite a number of GMRES iterations are needed at the intermediate levels.

Standard multigrid will also fail for k^2-values very close to eigenvalues. In that case subspace correction techniques should be employed [24].

An advanced multigrid based solution method for the Helmholtz equation is the wave-ray multigrid method [23], which has been developed for Helmholtz problems with constant or smoothly varying wavenumbers. The method has been adapted for a first-order system least-squares version of the Helmholtz equation in [86], using coarse-grid basis functions derived from plane waves, and in [135] within the context of smoothed aggregations [134] in algebraic multigrid methods.

Fish and Qu [48, 49] proposed a method called the global basis (GB) two-level method which is used to identify eigenvalues of the smoothing iteration matrix outside the region of convergence. The coarse-grid approximation is then constructed based on the corresponding eigenvectors. The generalized global basis (GBB) method attempts to stabilize the multilevel procedure [137]. GBB constructs additional coarse-grid corrections spanned by the unresolved eigenmodes by filtering the slow convergence modes.

3.4 Domain Decomposition Methods

For large problems, methods suitable for parallel machines are necessary. One of methods which are suitable and mainly developed by fully considering parallel machine architectures is domain decomposition methods [100, 111, 121]. In domain decomposition methods, the computational domain Ω is decomposed or partitioned into m subdomains Ω_i, $i = 1, \ldots, m$, which may be overlapping. The basic idea of domain decomposition methods (DDM) is to find a solution in Ω by solving the local (subdomain) problems in Ω_j and then exchanging solutions in the interface between two neighboring domains. Several classes of DDM exist, e.g. multiplicative and additive (and as well hybrid) Schwarz methods, Schur complement methods and FETI methods. In practice, DDM can be used both as a solver or as a preconditioner for Krylov subspace methods.

Some of the early work on the domain decomposition method (DDM) to solve the Helmholtz equation are due to Despres [37], which provide convergence analysis for nonoverlapping DDM on the differential level. The well-posedness is ensured by incorporating the consistency condition in the form of the complex Robin condition at the

interfaces:

$$\frac{\partial u_i}{\partial \eta_i} + \hat{\jmath} k u_i = -\frac{\partial u_j}{\partial \eta_j} + \hat{\jmath} k u_j, \quad i \neq j. \tag{44}$$

Convergence analysis on discrete level, e.g. is given by Kim in [78] for the second order accurate discretization and in [80] for finite element discretizations. This method, however, converges slowly. Enhancing the convergence of the DDM can be done via a generalization of the Robin condition by replacing k with an arbitrary constant θ. The constant θ is chosen such that the spectral radius of the iteration matrix related to this DDM is much less than one, which can be done automatically [78, 79]. See also [13, 29, 59, 115] for similar work.

Larsson [85] approaches the problem by constructing local problems via Schur complement of the related Helmholtz matrix, after reordering the unknowns based on the interior points and boundary points. The local subdomain problem is solved by using a fast Poisson preconditioner [97]. Having identical problem on the subdomain level, DDM can be solved combined with iterative methods.

Another class of domain decomposition methods is called FETI methods, with extension to the Helmholtz equation called FETI-H method [47, 119]. The main concept of the FETI method is based on expressing the discrete elliptic operator in a discrete Lagrange multipliers form. This allows the splitting of the complete formulae into two parts: the classical Lagrange function and the interface quantity.

This method is favorable not only in terms of computational performance but also from a numerical scalability point of view. The method has been proved to be scalable both with respect to mesh size and subdomain size. Literature [47] provides a concise discussion about the FETI method and its extension to the FETI-H method.

4 Preconditioners for the Helmholtz Equation

The convergence of Krylov subspace methods can be improved by incorporating preconditioners. By preconditioning we solve the following equivalent system:

$$M_1^{-1} A M_2^{-1} \widetilde{u} = M_1^{-1} g, \quad \text{where } \widetilde{u} = M_2 u_2. \tag{45}$$

Usually, $M = M_1 M_2$, but this is not necessary. For positive definite matrices, M_1 and M_2 are chosen such that $\kappa(M_1^{-1} A M_2^{-1}) \ll \kappa(A)$. For general matrices, M_1 and M_2 are chosen such that $M_1^{-1} A M_2^{-1}$ is close to identity. Note that M_1 and M_2 must be easy to invert. Beside these two requirements, there is no specific rule in choosing M_1 and M_2, and there exist many ways of doing this.

One way to do this is by considering the product $M_1 M_2$ as a sort of approximation to A. The simplest way is by setting $M_2 = I$, and $M_1 = \text{diag}(A)$; i.e., M is a matrix obtained by dropping entries of A except those in the diagonal. This preconditioner is referred to as the *diagonal scaling*. This preconditioner gives only a very rough approximation to A, and is not effective for the Helmholtz equation.

Among popular preconditioners are those based on incomplete LU factorizations [91, 92], and sparse approximate inverse factorizations, e.g. [64]. It is worth mentioning here that for some classes of problems, multigrid and domain decomposition methods can be used as effective preconditioners for a Krylov subspace method.

In the sequel we discuss ILU preconditioners. A special preconditioner for the Helmholtz equation, called the shifted Laplacian preconditioner, is discussed in Sect. 5.

4.1 ILU Preconditioners

We distinguish two incomplete LU (ILU) preconditioners: algebraic and analytical ILU.

4.1.1 Algebraic ILU

A better approximation to A can be obtained if M_1 and M_2 are constructed based on an incomplete LU factorization of A [27, 32, 38, 46, 89, 104, 105]. The incomplete LU factors are obtained by performing Gaussian elimination and then dropping some elements. Thus, in this case M_1 and M_2 are the lower and upper triangular matrices resulted from such a process.

The degree of approximation of $LU = M_1 M_2$ depends on the number of fill-in elements allowed in the LU factors. The simplest one is the so-called ILU(0), wherein the same non-zero structure as A is retained in ILU.

A more accurate approximation can be obtained by increasing the level of fill-in. Two scenarios exist. The first is more structure-oriented, and is done by adding more off-diagonals in the LU factors. We denote this as ILU(n_{lev}), which $n_{\text{lev}} > 0$ is a reasonably small integer, indicating the level of fill-in. This scenario will result in a structured linear system for L and U. The second scenario is related to a drop tolerance of fill-in. So, this scenario is more value-oriented. If during an LU factorization the value of an element falls below a prescribed tolerance ϵ, which is small, this element is set to zero. We denote this incomplete LU decomposition as ILUT(ϵ), with ϵ the drop tolerance. An ILUT(ϵ) process often leads to unstructured L and U matrices. There are several others variants of ILU, for example, by including pivoting (ILUTP [104, 105]).

Recall that the matrix A is indefinite. For indefinite matrices straightforward LU factorizations and as well incomplete LU factorizations may not be stable. This may eventually lead to LU factors which are not good approximations to A.

Typical convergence results are shown in Tables 3 and 4 for COCG, Bi-CGSTAB, QMR and GMRES combined with ILU(0) and ILUT(0.01) preconditioners. We again consider the same problem as used in Sect. 3.2.

Compared to the unpreconditioned case, ILU preconditioners improve the convergence significantly. With more fill-in, a faster convergence can be achieved for small wavenumber. With ILU preconditioners, COCG turns out to be the worst method shown in this example. For $k = 20$ and with ILUT(0.01), COCG does not converge after 2,000 iterations; see Table 4.

GMRES and Bi-CGSTAB result in a comparable convergence in terms of the number of iterations. Note that Bi-CGSTAB requires two matrix-vector products and four preconditioner solves (due to split form (45)). Thus, the overall performance of Bi-CGSTAB is not better than GMRES. With increasing k the performance of QMR deteriorates faster than Bi-CGSTAB and GMRES.

In Table 4 we also measure the number of nonzero elements in the $L = M_1$ and $U = M_2$ matrices for ILUT(0.01). For increasing k, and thus N, the number of nonzero elements becomes unacceptably large. ILU(0), which results in far less nonzero elements (similar to the nonzero elements of A) in the L and U matrices compared to ILUT(0.01) seems to be a more effective preconditioner for the Helmholtz equation in this case.

Recently some more sophisticated approaches have been proposed to construct more efficient ILU preconditioners. This usually involves special preprocessing on the matrix prior to implementing incomplete LU factorization strategy. Benzi et al. [14], for example, apply a nonsymmetric reordering strategy and scaling in order to obtain an equivalent diagonal dominant matrix, which is more stable with respect to incomplete LU factorizations. Stability of incomplete LU factorizations can be improved further by taking into account the inverse triangular factors [18].

Table 3 Number of iterations of Krylov subspace methods for a 2D Helmholtz problem. The preconditioner is ILU(0). $kh = 0.625$ (∼10 gridpoints per wavelength)

k	COCG	Bi-CGSTAB	QMR	GMRES
10	25	22	25	24
20	73	102	75	63
30	155	194	150	124
40	267	296	267	199

4.1.2 Analytical ILU

An incomplete LU factorization can also be constructed analytically based on the underlying partial differential equation. This approach is proposed by Gander and Nataf called analytic ILU or AILU [55] (see as well similar work by Plessix and Mulder [98]). Applications of AILU on the Helmholtz equation are presented in e.g. [56] and [57]. In this case, the Helmholtz operator $\mathcal{L}(u)$ is factorized into two operators:

$$\mathcal{L}(u) = -(\partial_{x_1} + \Lambda_1)(\partial_{x_1} - \Lambda_2) \tag{46}$$

with Λ_1 and Λ_2 positive, nonlocal operators. These non local operators can be viewed as fill-in in the L and U matrix in LU factorization analogue.

To obtain AILU, the non local operators are, in Fourier space, then approximated by local operators depending on the wavenumber k and some constants $c_1, c_2 \in \mathbb{C}$, $\mathrm{Re}(c_2) > 0$, i.e. $\Lambda_{1,2}^{\text{approx}} = \mathcal{F}_{x_2}^{-1}(c_1 + c_2\widehat{k}^2)$, with $\widehat{k}$ the Fourier frequency. Notice that

$$\mathcal{L}^{\text{approx}}(u) + \mathcal{L}(u) = \mathcal{L}^{\text{approx}}(u) + f, \tag{47}$$

or, in the view of fixed point iteration,

$$\mathcal{L}^{\text{approx}}(u^{j+1}) = (\mathcal{L}^{\text{approx}} - \mathcal{L})(u^j) + f. \tag{48}$$

The constants c_1 and c_2 are determined such that the convergence factor of the fixed point iteration (48) is as small as possible. Since $\rho = \rho(c_1, c_2, k, \widehat{k}, k_{x_1})$ and $c_1, c_2 \in \mathbb{C}$ this requires optimization of four real parameters for a given k.

Because of (48) AILU can be used as an iterative solver. This is a convergent method, but as reported in [57] the convergence is not satisfactory. As a preconditioner AILU is more superior than ILU(0) in terms of convergence speed up and computational work. The convergence however depends rather strongly on k and h.

Table 4 Number of iterations of Krylov subspace methods for a 2D Helmholtz problem. The preconditioner is ILUT(0.01). $kh = 0.625$ (∼10 gridpoints per wavelength)

k	$nz(A)$	$nz(L)$	$nz(U)$	COCG	Bi-CGSTAB	QMR	GMRES
10	1065	2023	2008	30	6	10	10
20	4681	10430	10224	>2000	11	19	18
30	10857	25355	24806	–	23	34	31
40	19593	46762	45700	–	53	65	48

5 Shifted Laplacian Preconditioner for the Helmholtz Equation

The first prototype of the shifted Laplacian preconditioner for the Helmholtz equation was proposed by Bayliss, Goldstein and Turkel in [10] in the 1980s. It did not attract so much attentions until the work of Giles and Laird [82] appeared in 2002. Erlangga, Vuik and Oosterlee [41, 43, 44] extended and generalized this work to a more robust and efficient preconditioner within this class. An analysis for general setting (e.g., with possible finite element discretizations, heterogeneity and damping layers) is described by van Gijzen et al. in [131]. The generalized version is used to solve large and difficult realistic 2D and 3D problems in [81, 101, 102], in either sequential or parallel machines.

This class of preconditioners is constructed from a discretization of the following operator:

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\nabla\cdot\nabla - (\beta_1 - \hat{\jmath}\beta_2)k^2, \quad \beta_1,\beta_2 \in \mathbb{R}, \tag{49}$$

which is called "shifted Laplace operator". The preconditioners used in [10] and [82] belong to this class of preconditioners, which can be recovered from (49) by setting $(\beta_1,\beta_2) = (0,0)$ (the Laplacian) and $(\beta_1,\beta_2) = (-1,0)$ (definite Helmholtz), respectively.

5.1 One-Dimensional Case

The motivation of the development of the shifted Laplace preconditioners can be seen from a simple 1D Helmholtz problem with constant wavenumber. The results of the analysis, however, generally holds for nonconstant wavenumbers and in 2D or 3D. We first start with a *real* shift, by setting $\beta_2 = 0$ in the next subsection.

5.1.1 Real Shift

We consider a 1D Helmholtz equation in a unit domain $\Omega = (0,1)$:

$$-\frac{d^2u}{dx_1^2} - k^2u = 0, \quad k = \text{const.}, \tag{50}$$

with Dirichlet boundary conditions $u(0) = u(1) = 0$. The related *continuous* eigenvalue problem is

$$-\left(\frac{d^2}{dx_1^2} + k^2\right)\widetilde{u} = \lambda\widetilde{u}, \tag{51}$$

with the general solution $\widetilde{u} = \sin(ax_1)$, satisfying the conditions at $x_1 = 0$ and $x_1 = 1$. By substituting this solution in (51) we arrive at the following relation:

$$\begin{aligned}&(k_{\ell_1}^2 - k^2)\sin(\pi\ell_1 x_1)\\ &\quad = \lambda\sin(\pi\ell_1 x_1) \rightarrow \lambda^{\ell_1} = k_{\ell_1}^2 - k^2,\end{aligned} \tag{52}$$

where $k_{\ell_1} = \pi\ell_1,\ \ell_1 \in \mathbb{N}/\{0\}$. Thus, for large wavenumbers k the eigenvalues change sign, indicating the indefiniteness of the problem.

In 1D the preconditioning operator (49) reads

$$\mathcal{M}_{\beta_1} := -\frac{d^2}{dx_1^2} - \beta_1 k^2. \tag{53}$$

Here, we assume that $\beta_1 \le 0$. Later we will relax this assumption. So matrices related to operator (53) are symmetric positive definite. Then, we have the following preconditioned (generalized) eigenvalue problem:

$$\left(-\frac{d^2}{dx_1^2} - k^2\right)\widetilde{u}^{\ell_1} = \lambda_r^{\ell_1}\left(-\frac{d^2}{dx_1^2} - \beta_1 k^2\right)\widetilde{u}^{\ell_1}. \tag{54}$$

By assuming a solution of the form $\widetilde{u} = \sin(ax_1)$, the eigenvalues are found to be

$$\lambda_r^{\ell_1} = \frac{k_{\ell_1}^2 - k^2}{k_{\ell_1}^2 - \beta_1 k^2} = \frac{1 - (k/k_{\ell_1})^2}{1 - \beta_1(k/k_{\ell_1})^2}, \tag{55}$$

where $k_{\ell_1} = \pi\ell_1,\ \ell_1 \in \mathbb{N}/\{0\}$. For $\ell_1 \to \infty$, $\lambda_r^{\ell_1} \to 1$, i.e., the eigenvalues are bounded above by one. For $\ell_1 \to 0$, the low eigenmodes, we have $\lambda_r^{\ell_1} \to 1/\beta_1$. The modulus of this eigenvalue remains bounded unless $-1 \le \beta_1 \le 0$. The maximum eigenvalue can therefore be written as

$$|(\lambda_r^{\ell_1})_{\max}| = \max\left(\left|\frac{1}{\beta_1}\right|, 1\right). \tag{56}$$

The smallest eigenvalue is estimated as follows. Assume that the minimum eigenvalue is very close (but not equal) to zero. From (55) this implies that $k_{\ell_1} = k + \epsilon$, $0 < \epsilon \ll 1$. Substituting of this relation into (55), we find that

$$(\lambda_r^{\ell_1})_{\min} = \frac{2}{1-\beta_1}\left(\frac{\epsilon}{k}\right). \tag{57}$$

From (57), the minimum eigenvalue can be very close to zero as β_1 goes to infinity. The condition number of the preconditioned Helmholtz operator now reads

$$\kappa = \begin{cases} \frac{1}{2}(1-\beta_1)k/\epsilon & \text{if } \beta_1 \le -1,\\ \frac{1}{2|\beta_1|}(1-\beta_1)k/\epsilon & \text{if } -1 \le \beta_1 \le 0. \end{cases} \tag{58}$$

In the limit we find from (58) that

$$\lim_{\beta_1\downarrow -1}\kappa = \lim_{\beta_2\uparrow -1}\kappa = k/\epsilon, \tag{59}$$

which is the minimum value of κ for $\beta_1 \le 0 \in \mathbb{R}$. The relation (58) tells us that taking the leading part of the Helmholtz equation (i.e. the Laplacian, $\beta_1 = 0$) is generally advisable for high wavenumbers, as advocated in [90]. However, this

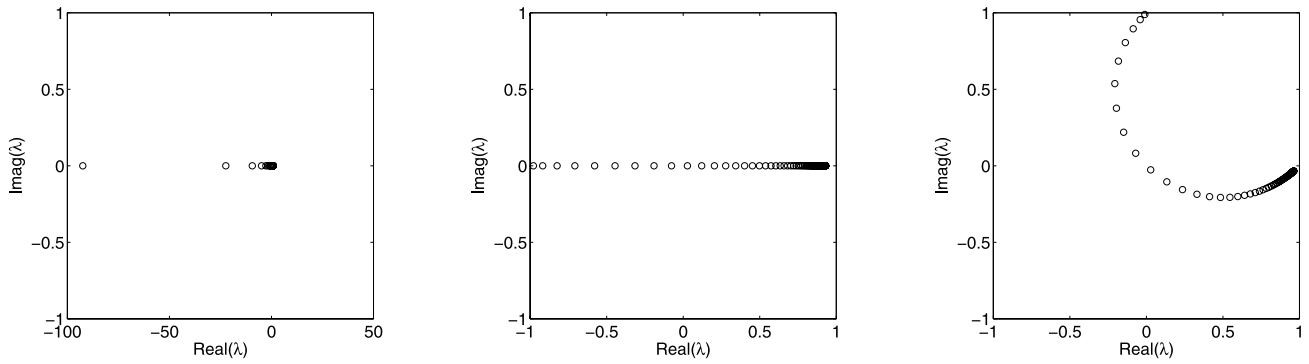

Fig. 3 Eigenvalues of the preconditioned 1D Helmholtz equation, $k = 30$. *Left*: $\mathcal{M}_{(0,0)}$, *Mid*: $\mathcal{M}_{(-1,0)}$, *Right*: $\mathcal{M}_{(0,1)}$

is no longer true for small wavenumbers, for which the condition number of the preconditioned linear system may become very large. From the convergence bound of CG (30) the result (58) also gives an indication that setting $\beta_1 = -1$ in (53) leads to a fast converging preconditioned iterative method for the Helmholtz equation.

5.1.2 Generalization to Complex Shift

In order to improve the minimum eigenvalue but at the same time keep the upper bound, a complex shift is introduced [41]. For that purpose, we consider the complex shifted Laplace operator (49), which is written for 1D as

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\frac{d^2}{dx^2} - (\beta_1 - \hat{\jmath}\beta_2)k^2, \quad \beta_1 \le 0, \beta_2 \in \mathbb{R}. \tag{60}$$

In this case by setting $\beta_1 \le 0$ (as in the previous section) we ensure that the real part of the discrete representation of $\mathcal{M}_{(\beta_1,\beta_2)}$ is positive.

Eigenvalues of the premultiplied equation, denoted by $\lambda_c^{\ell_1}$, are

$$\begin{aligned}\lambda_c^{\ell_1} &= \frac{k_{\ell_1}^2 - k^2}{k_{\ell_1}^2 - (\beta_1 - \hat{\jmath}\beta_2)k^2} \Rightarrow |\lambda_c^{\ell_1}|^2 \\ &= \frac{(k_{\ell_1}^2 - k^2)^2}{(k_{\ell_1}^2 - \beta_1 k^2)^2 + \beta_2^2 k^4}.\end{aligned} \tag{61}$$

Evaluating $(\lambda_c^{\ell_1})_{\max}$ and $(\lambda_c^{\ell_1})_{\min}$ as in (56) and (57) one finds

$$\begin{aligned}|(\lambda_c^{\ell_1})_{\max}|^2 &= \max\left(\frac{1}{\beta_1^2 + \beta_2^2}, 1\right), \\ |(\lambda_c^{\ell_1})_{\min}|^2 &= \frac{4}{(1-\beta_1)^2 + \beta_2^2}\left(\frac{\epsilon}{k}\right)^2.\end{aligned} \tag{62}$$

These results give the following condition numbers

$$\kappa^2 = \begin{cases} \frac{1}{4}\left(1 + \frac{1-2\beta_1}{\beta_1^2+\beta_2^2}\right)(k/\epsilon)^2, & \beta_1^2 + \beta_2^2 \le 1,\ \beta_1 \le 0, \\ \frac{1}{4}\left((1+\beta_1)^2 + \beta_2^2\right)(k/\epsilon)^2, & \beta_1^2 + \beta_2^2 \ge 1. \end{cases} \tag{63}$$

By evaluating (63) [41], it can be concluded that κ^2 is minimal if $\beta_1 = 0$ and $\beta_2 = \pm 1$.

Figure 3 shows the spectra of the 1D Helmholtz problem (50) preconditioned by operators $\mathcal{M}_{(\beta_1=0,\beta_2=0)}$, $\mathcal{M}_{(\beta_1=-1,\beta_2=0)}$, and $\mathcal{M}_{(\beta_1=0,\beta_2=+1)}$. For simplicity, we denote these preconditioning operators by $\mathcal{M}_{(0,0)}$, $\mathcal{M}_{(-1,0)}$, and $\mathcal{M}_{(0,1)}$, respectively.

Figure 3 shows that $\mathcal{M}_{(0,1)}$ clusters the eigenvalues stronger than the other two and pushes the eigenvalues in the negative real plane towards the imaginary axis. This clustering may improve the performance of the preconditioned iterative methods. Note that with $\mathcal{M}_{(0,1)}$ there is still a possibility that some eigenvalues lie very close to zero, causing an unsatisfactory numerical performance. But in this case we have the estimate:

$$\mathrm{Re}((\lambda_{(0,1)}^{\ell_1})_{\min}) = \epsilon/k, \tag{64}$$

which is the same as the estimate for $\mathcal{M}_{(-1,0)}$ and smaller than that for $\mathcal{M}_{(0,0)}$. However, the modulus $|(\lambda_{(0,1)}^{\ell_1})_{\min}| = \sqrt{2}(\epsilon/k) > |(\lambda_{(-1,0)}^{\ell_1})_{\min}| = \epsilon/k$ because of the imaginary shift. Because of the same upper bound as $\mathcal{M}_{(-1,0)}$, we can expect that $\mathcal{M}_{(0,1)}$ will perform better than $\mathcal{M}_{(0,0)}$ and $\mathcal{M}_{(-1,0)}$.

The above analysis implies that if CGNR is applied to $M_h^{-1}A_h$, with M_h coming from either $\mathcal{M}_{(0,0)}$, $\mathcal{M}_{(-1,0)}$ $\mathcal{M}_{(0,1)}$, the preconditioner related to $\mathcal{M}_{(0,1)}$ may lead to a faster convergence than the others.

5.2 Spectral Properties and CGNR Convergence

We extend the analysis for constant wavenumbers to the discrete formulation of (50) and relate the results to the convergence of CGNR.

Discretization of (50) with the second-order finite difference scheme results in the linear system

$$Au = (L - K^2) = g, \quad K = \mathrm{diag}(k_i), \quad i = 1, \ldots, N. \tag{65}$$

At this moment we assume Dirichlet or Neumann conditions at the boundaries. In this case the matrix A is real-valued. We discretize the operator $\mathcal{M}_{\beta_1,\beta_2}$ with the *same* boundary conditions as (50) and obtain

$$M_{(\beta_1,\beta_2)} = L - (\beta_1 - \hat{\jmath}\beta_2)k^2 I_h.$$

This gives the preconditioned system:

$$\begin{aligned}&\big(L - (\beta_1 - \hat{\jmath}\beta_2)k^2 I\big)^{-1}(L - k^2 I)u \\ &\quad = \big(L - (\beta_1 - \hat{\jmath}\beta_2)k^2 I\big)^{-1} g.\end{aligned} \tag{66}$$

The generalized eigenvalue problem related to (66) is accordingly

$$(L - k^2 I)\widehat{u} = \widehat{\lambda}\big(L - (\beta_1 - \hat{\jmath}\beta_2)k^2 I\big)\widehat{u}, \tag{67}$$

with $\widehat{u}$ the eigenvector corresponding to the discrete eigenvalue $\widehat{\lambda}$. Since both (65) and (66) are indefinite for $k^2 > \lambda_{\min}(L)$, convergence properties of (66) are difficult to estimate. Therefore, the analysis will be based on a normal equations formulation of the preconditioned matrix system (as in [82]).

Suppose that the eigenvalues of L are ordered increasingly as $0 < \mu_1 \leq \cdots \leq \mu_N$. Furthermore, denote the normal equations A^*A, $(M_{(0,0)}^{-1}A)^*(M_{(0,0)}^{-1}A)$, $(M_{(-1,0)}^{-1}A)^*(M_{(-1,0)}^{-1}A)$ and $(M_{(0,1)}^{-1}A)^*(M_{(0,1)}^{-1}A)$ by Q_A, $Q_{(0,0)}$, $Q_{(-1,0)}$ and $Q_{(0,1)}$, respectively. We find the eigenvalues in the four following cases as:

$$\lambda(Q_A) = (\mu_m - k^2)^2, \tag{68}$$

$$\lambda(Q_{(0,0)}) = \left(1 - \frac{k^2}{\mu_m}\right)^2, \tag{69}$$

$$\lambda(Q_{(-1,0)}) = \left(1 - \frac{2k^2}{\mu_m + k^2}\right)^2, \tag{70}$$

$$\lambda(Q_{(0,1)}) = 1 - \frac{2\mu_m k^2}{\mu_m^2 + k^4}. \tag{71}$$

We have several possibilities.

5.2.1 *Cases* $0 < k^2 < \mu_1$

In this case, by using (68–71) the minimum and maximum eigenvalues for all cases are given in Table 5.

Using Table 5 we conclude the following:

- For $k^2/\mu_1 < 1$, $\lambda(Q_{(0,0)})_{\min} > \lambda(Q_{(-1,0)})_{\min}$
- Also, $\lambda(Q_{(0,0)})_{\min} > \lambda(Q_{(0,1)})_{\min}$

Table 5 Minimum and maximum eigenvalues for $0 < k^2 < \mu_1$

	Q_A	$Q_{(0,0)}$	$Q_{(-1,0)}$	$Q_{(0,1)}$
$\lambda_{\min}$	$(\mu_1 - k^2)^2$	$(1 - \frac{k^2}{\mu_1})^2$	$(1 - \frac{2k^2}{\mu_1 + k^2})^2$	$1 - \frac{2\mu_1 k^2}{\mu_1^2 + k^4}$
$\lambda_{\max}$	$(\mu_N - k^2)^2$	$(1 - \frac{k^2}{\mu_N})^2$	$(1 - \frac{2k^2}{\mu_N + k^2})^2$	$1 - \frac{2\mu_N k^2}{\mu_N^2 + k^4}$

- $\lim_{\mu_N \to \infty} \lambda(Q_{(0,0)})_{\max} = \lim_{\mu_N \to \infty} \lambda(Q_{(-1,0)})_{\max} = \lim_{\mu_N \to \infty} \lambda(Q_{(0,1)})_{\max} = 1$, for $m \to \infty$

As the consequence, for $k < \sqrt{\mu_1}$, $\kappa(Q_{0,0}) < \kappa(Q_{-1,0})$ and $\kappa(Q_{0,0}) < \kappa(Q_{0,1})$. Thus, For k small, $M_{(0,0)}$ is more effective than $M_{(-1,0)}$ and $M_{(0,1)}$.

5.2.2 *Cases* $\mu_1 < k^2 < \mu_N$

We have for Q_A:

$$\begin{aligned}&\lambda(Q_A)_{\min} = (\mu_{m_1} - k^2)^2, \\ &\quad \text{where } |\mu_{m_1} - k^2| \leq |\mu_m - k^2|, \ \forall m, \\ &\lambda(Q_A)_{\max} = (\mu_N - k^2)^2,\end{aligned} \tag{72}$$

and the eigenvalues are unbounded either for large μ_N or large k.

For $Q_{(0,0)}$, we have

$$\begin{aligned}\lambda(Q_{(0,0)})_{\min} &= \left(\frac{\mu_{m_2} - k^2}{\mu_{m_2}}\right)^2, \quad \text{where } \left|\frac{\mu_{m_2} - k^2}{\mu_{m_2}}\right| \\ &\leq \left|\frac{\mu_m - k^2}{\mu_m}\right|, \quad \forall m, \\ \lambda(Q_{(0,0)})_{\max} &= \max\left(\left(\frac{\mu_N - k^2}{\mu_N}\right)^2, \left(\frac{\mu_1 - k^2}{\mu_1}\right)^2\right).\end{aligned} \tag{73}$$

In this case, for $\mu_N \to \infty$, $\lambda_N = 1$ as long as k is finite (because $\lim_{k\to\infty}((\mu_m - k^2)/(\mu_m))^2 = \infty$). Furthermore, $\lim_{\mu_1 \to 0}((\mu_1 - k^2)/(\mu_1))^2 = \infty$. Therefore, $\lambda_{\max}$ can become extremely large, which makes $M_{(0,0)}$ less favorable for preconditioning.

For $Q_{(-1,0)}$, we have

$$\begin{aligned}\lambda(Q_{(-1,0)})_{\min} &= \left(\frac{\mu_{m_3} - k^2}{\mu_{m_3} + k^2}\right)^2, \quad \text{where } \left|\frac{\mu_{m_3} - k^2}{\mu_{m_3} + k^2}\right| \\ &\leq \left|\frac{\mu_m - k^2}{\mu_m + \mu_{m_3}}\right|, \quad \forall m, \\ \lambda(Q_{(-1,0)})_{\max} &= \max\left(\left(\frac{\mu_N - k^2}{\mu_N + k^2}\right)^2, \left(\frac{\mu_1 - k^2}{\mu_1 + k^2}\right)^2\right),\end{aligned} \tag{74}$$

which lead to

$$\lim_{\mu_N\to\infty}\left(\frac{\mu_N-k^2}{\mu_N+k^2}\right)^2=\lim_{\mu_1\to 0}\left(\frac{\mu_1-k^2}{\mu_1+k^2}\right)^2$$
$$=\lim_{k\to\infty}\left(\frac{\mu_m-k^2}{\mu_m+k^2}\right)^2=1.$$

The preconditioned system $M_{(-1,0)}^{-1}A$ is always bounded above by one, i.e. the eigenvalues are always clustered. Moreover, $M_{(-1,0)}$ provides a better condition number than $M_{(0,0)}$. For large k, $M_{(-1,0)}$ is more effective than $M_{(0,0)}$.

Finally, for $Q_{(0,1)}$, we have

$$\lambda(Q_{(0,1)})_{\min}=\frac{(\mu_{m_4}-k^2)^2}{\mu_{m_4}^2+k^4},\quad \text{where } \left|\frac{(\mu_{m_4}-k^2)^2}{\mu_{m_4}^2+k^4}\right|$$
$$\le\left|\frac{(\mu_m-k^2)^2}{\mu_m^2+k^4}\right|,\quad \forall m, \tag{75}$$
$$\lambda(Q_{(0,1)})_{\max}=\max\left(1-\frac{2\mu_1k^2}{\mu_1^2+k^4},1-\frac{2\mu_Nk^2}{\mu_N^2+k^4}\right),$$

which lead to

$$\lim_{\mu_N\to\infty}\lambda(Q_{(0,1)})_{\max}=\lim_{\mu_1\to 0}\lambda(Q_{(0,1)})_{\max}$$
$$=\lim_{k\to\infty}\lambda(Q_{(0,1)})_{\max}=1.$$

Hence, the eigenvalues of $Q_{(0,1)}$ are bounded above by one. Typically, preconditioning with $M_{(0,1)}$ gives a better condition number than with $M_{(0,0)}$.

The lower bound of the eigenvalues can be estimated by assuming that $\lambda_{\min}\approx 0$, implying $\mu_m=k^2+\epsilon$, $\epsilon>0$. This leads to

$$\lambda(Q_{(0,1)})_{\min}=\frac{1}{2}\frac{\epsilon^2}{k^4}\quad\text{and}\quad\lambda(Q_{(-1,0)})_{\min}=\frac{1}{4}\frac{\epsilon^2}{k^4}. \tag{76}$$

With respect to the condition number, we then have

$$\kappa(Q_{(0,1)})=2\left(\frac{k^4}{\epsilon^2}\right)\quad\text{and}\quad\kappa(Q_{(-1,0)})=4\left(\frac{k^4}{\epsilon^2}\right).$$

Thus, we can expect that $M_{(0,1)}$ is more effective as the preconditioner than $M_{(-1,0)}$.

5.2.3 Numerical Experiments

To show the implication of the analysis in the preceding subsections, we solve the following 2D Helmholtz problem:

$$\begin{cases}(-\nabla\cdot\nabla-k^2)u=(5\pi^2-k^2)\sin(\pi x_1)\sin(2\pi x_2),\\ \quad\text{in }\Omega=(0,1)^2,\\ u=0,\quad\text{on }\Gamma=\partial\Omega.\end{cases} \tag{77}$$

The exact solution of (77) is $u=\sin(\pi x_1)\sin(2\pi x_2)$. The numerical solutions are obtained by GMRES and are shown in Table 6 for various wavenumbers. A mesh of 10 gridpoints per wavelength ($kh=0.625$) is used. Note that in this case M is *inverted exactly*.

Table 6 Performance of GMRES to reduce the relative residual by 7 orders for Problem (77). The preconditioners are inverted by using a direct solver

k	$M_{(0,0)}$		$M_{(-1,0)}$		$M_{(0,1)}$	
	Iter	Time (s)	Iter	Time (s)	Iter	Time (s)
5	8	0.01	9	0.01	9	0.03
10	13	0.03	16	0.04	15	0.11
20	28	0.35	30	0.38	26	1.13
30	56	1.77	58	1.84	47	6.54
40	106	7.36	102	7.19	82	26.44
50	149	19.31	142	18.15	114	67.95

Table 7 Number of iteration of GMRES, Bi-CGSTAB and QMR to reduce the relative residual by 7 orders for Problem (77). The preconditioner $M_{(0,1)}$ is inverted by using a direct solver

	GMRES	QMR	Bi-CGSTAB
10	15	15	11
20	26	26	24
40	82	90	94

For low frequencies, all preconditioners show a very satisfactorily comparable performance. $M_{(0,0)}$ becomes less effective for increasing values of k, where the number of iterations increases somewhat faster than for $M_{(-1,0)}$ or $M_{(0,1)}$. For large k, preconditioning with $M_{(0,1)}$ gives the fastest convergence. This behavior is in accordance with the theory. The effectiveness of $M_{(0,1)}$ for large k makes this choice more attractive than $M_{(-1,0)}$ or $M_{(0,1)}$.

In Table 6 we however observe that the computation using $M_{(0,1)}$ requires more work than the others. This is due to complex arithmetics involved in $M_{(0,1)}$, which is not necessary for Problem (77). For this problem, A and as well $M_{(0,0)}$ and $M_{(-1,0)}$ are real valued.

A comparison of convergence of the same problem for GMRES, QMR and Bi-CGSTAB is shown in Table 7, only for $M_{(0,1)}$. For high wavenumbers, it appears that GMRES converges faster than QMR and Bi-CGSTAB. With memory becoming very demanding with an increase in k, restarting GMRES to overcome the memory limitation only slows down the convergence.

5.3 Radiation Conditions and GMRES Convergence Bound

The effect of the inclusion of absorbing boundary conditions on the spectrum of the preconditioned system can be analyzed as follows.

To be more general, we consider the 2D Helmholtz equation. We only need to evaluate $\mathcal{M}_{(0,1)}$ since the analysis in Sect. 5.2 reveals that $\mathcal{M}_{(0,1)}$ is more superior than $\mathcal{M}_{(0,0)}$ and $\mathcal{M}_{(-1,0)}$.

First, we need an explicit expression for eigenvalues of the preconditioned eigenvalue problem in 2D. Following a similar analysis as in Sect. 5.2, we get

$$\lambda_p^{\ell_1,\ell_2} = \frac{k_{\ell_1}^2 + k_{\ell_2}^2 - k^2}{k_{\ell_1}^2 + k_{\ell_2}^2 + \hat{\jmath}k^2},$$
$$k_{\ell_1} = \ell_1\pi,\ k_{\ell_2} = \ell_2\pi,\ \ell_1,\ell_2 \in \mathbb{N}. \tag{78}$$

Adding up the real and the imaginary part of (78) yields

$$\mathrm{Re}(\lambda_p^{\ell_1,\ell_2}) + \mathrm{Im}(\lambda_p^{\ell_1,\ell_2}) = (k_{\ell_1,\ell_2}^2 - k^2)^2/(k_{\ell_1,\ell_2}^4 + k^4). \tag{79}$$

We have the following Lemma.

Lemma 5.1 *Let the 2D Helmholtz equation with homogeneous Dirichlet boundary conditions be preconditioned by $\mathcal{M}_{(0,1)}$. If resonance does not occur, then for all $k^2 \neq k_{\ell_1,\ell_2}^2$ the spectrum then completely lies above the line* $\mathrm{Re}(z) + \mathrm{Im}(z) = 0$.

Proof From (79) it is clear that $\mathrm{Re}(\lambda_p) + \mathrm{Im}(\lambda_p) > 0$. □

Notice that if we rotate the eigenvalues by an angle $\theta = -\pi/4$ all eigenvalues are in the positive half plane. In fact we have the following Lemma.

Lemma 5.2 *Let the 2D Helmholtz problem with homogeneous Dirichlet boundary conditions be preconditioned by $\mathcal{M}_{(0,1)}$ and assume that resonance does not occur. Furthermore, let the spectrum be rotated by an angle θ. For $\theta = -\pi/4$, the spectrum has the following properties*:

(i) *All eigenvalues lie on a circle with center $z_{c,-\frac{\pi}{4}} = \frac{1}{2}\sqrt{2}$ and radius $r = \frac{1}{2}\sqrt{2}$. There are no zero eigenvalues.*
(ii) *This circle is independent of wavenumber k.*

Proof Denote by $\lambda_{p,\theta} = \lambda_p \exp(\hat{\jmath}\theta)$ the eigenvalue obtained by rotating λ_p by an angle θ. From (78), we have that

$$\mathrm{Re}(\lambda_{p,\theta}^{\ell_1,\ell_2}) = \left(\frac{k_{\ell_1,\ell_2}^2 - k^2}{k_{\ell_1,\ell_2}^4 + k^4}\right)\left(k_{\ell_1,\ell_2}^2 \cos\theta + k^2 \sin\theta\right), \tag{80}$$

$$\mathrm{Im}(\lambda_{p,\theta}^{\ell_1,\ell_2}) = \left(\frac{k_{\ell_1,\ell_2}^2 - k^2}{k_{\ell_1,\ell_2}^4 + k^4}\right)\left(k_{\ell_1,\ell_2}^2 \sin\theta - k^2 \cos\theta\right), \tag{81}$$

where $k_{\ell_1,\ell_2}^4 = (k_{\ell_1}^2 + k_{\ell_2}^2)^2$. Substitution of $\theta = -\pi/4$ gives

$$\mathrm{Re}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) = \frac{1}{2}\sqrt{2}\frac{(k_{\ell_1,\ell_2}^2 - k^2)^2}{k_{\ell_1,\ell_2}^4 + k^4} > 0, \quad \forall k_{\ell_1,\ell_2}, k, \tag{82}$$

$$\mathrm{Im}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) = -\frac{1}{2}\sqrt{2}\frac{k_{\ell_1,\ell_2}^4 - k^4}{k_{\ell_1,\ell_2}^4 + k^4}. \tag{83}$$

Note that $\mathrm{Im}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) > 0$ if $k_{\ell_1\ell_2}^4 < k^4$.

Elimination of $k_{\ell_1\ell_2}$ from both equations yields

$$\left(\mathrm{Re}\left(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}\right) - \frac{1}{2}\sqrt{2}\right)^2 + \mathrm{Im}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1\ell_2})^2 = \frac{1}{2}, \tag{84}$$

or

$$\left|\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2} - \frac{1}{2}\sqrt{2}\right| = \frac{1}{2}\sqrt{2}, \tag{85}$$

which proves the lemma. □

This rotation is beneficial only for theoretical purposes. In practice, this is equivalent to solving the system $PM_{(0,1)}^{-1}Au = PM_{(0,1)}^{-1}g$ with $P = \frac{1}{2}\sqrt{2}\mathrm{diag}(1+\hat{\jmath})$. The rotation is not necessary if Krylov subspace algorithms like GMRES or Bi-CGSTAB are used.

From Lemma 5.2, we get the following corollary which can be proved by using a $\theta = \pi/4$ rotation.

Corollary 5.3 *For the Helmholtz problem with Dirichlet boundary conditions and preconditioned by operator $\mathcal{M}_{(0,1)}$, the spectrum lies on the circle $|z - z_c| = \frac{1}{2}\sqrt{2}$, with $z_c = \frac{1}{2}(1+\hat{\jmath})$ the center of circle.*

Due to the absorbing boundary conditions, the matrix A is now complex-valued. We consider a special case of complex-valued matrices, where the real part and the imaginary part of the matrix are both symmetric positive definite. We call this matrix a complex SPD (or CSPD) matrix. The next lemma holds for a CSPD matrix [41].

Lemma 5.4 *Let B be any CSPD matrix and let $\lambda_B \in \mathbb{C}$ be an eigenvalue. Then* $\mathrm{Re}(\lambda_B), \mathrm{Im}(\lambda_B) > 0$.

Consider again the matrix A and the splitting:

$$A = L - K^2 \Rightarrow C := K^{-1}LK^{-1} = K^{-1}AK^{-1} + I, \tag{86}$$

where L is a CSPD matrix related to discretization of the Laplacian *and* the radiation boundary condition. Observe that C is also CSPD. By Lemma 5.4, the eigenvalue of C, denoted by λ_C, satisfies $\mathrm{Re}(\lambda_C), \mathrm{Im}(\lambda_C) > 0$.

Let $M_{(0,1)}$ be split in the similar way, and consider the eigenvalue problem

$$M_{(0,1)}^{-1}Av = \lambda_{M_{(0,1)}^{-1}A}v.$$

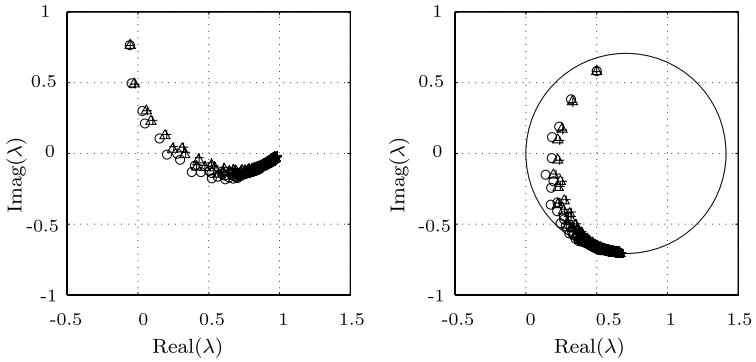

Fig. 4 Spectrum of the 2D Helmholtz problem ($k = 10$) with radiation boundary conditions, preconditioned by $M_{(0,1)}$ before (*left*) and after (*right*) rotation. Number of gridpoints: $10^2(\circ)$, $20^2(\triangle)$, $30^2(+)$

It can be shown that

$$\lambda_{M_{(0,1)}^{-1}A} = \frac{\lambda_C - 1}{\lambda_C + \hat{j}}. \tag{87}$$

We then have the following lemma.

Theorem 5.5 *Let $\lambda_{M_{(0,1)}^{-1}A}$ be an eigenvalue of $M_{(0,1)}^{-1}A$ with boundary condition (7). Let $|z - z_c| = \frac{1}{2}\sqrt{2}$ with $z_c = \frac{1}{2}(1 + \hat{j})$ be the circle corresponding to all eigenvalues of the "closed-off" problem (as described in Corollary 5.3). Then, $\lambda_{M_{(0,1)}^{-1}A}$ is enclosed by this circle.*

Proof By using (87) and Corollary 5.3 we have that

$$\begin{aligned}\lambda_{M_{(0,1)}^{-1}A} - z_c &= \frac{\lambda_C - 1}{\lambda_C + \hat{j}} - \frac{1}{2}(1 + \hat{j}) \\ &= \frac{1}{2}\frac{\lambda_C - 1 - \hat{j}(\lambda_C + 1)}{\lambda_C + \hat{j}} \\ &= \frac{1}{2}\frac{(\lambda_C - 1 - \hat{j}(\lambda_C + 1))(\overline{\lambda}_C - \hat{j})}{(\lambda_C + \hat{j})(\overline{\lambda}_C - \hat{j})}.\end{aligned}$$

With $|\lambda_{M_{(0,1)}^{-1}A} - z_c|^2 = (\lambda_{M_{(0,1)}^{-1}A} - z_c)\overline{(\lambda_{M_{(0,1)}^{-1}A} - z_c)}$, we find that

$$\begin{aligned}|\lambda_{M_{(0,1)}^{-1}A} - z_c| &= \frac{1}{2}\sqrt{2}\sqrt{\frac{\lambda_C - \hat{j}}{\overline{\lambda}_C - \hat{j}} \cdot \frac{\overline{\lambda}_C + \hat{j}}{\lambda_C + \hat{j}}} \\ &= \frac{1}{2}\sqrt{2}\sqrt{\frac{\lambda_C\overline{\lambda}_C - 2\mathrm{Im}(\lambda_C) + 1}{\lambda_C\overline{\lambda}_C + 2\mathrm{Im}(\lambda_C) + 1}} < \frac{1}{2}\sqrt{2}\end{aligned}$$

for every λ_C because of Lemma 5.4. Therefore, the eigenvalue $\lambda_{M_{(0,1)}^{-1}A}$ lies inside the circle. □

Theorem 5.5 tells us that:

- with radiation boundary conditions, the spectrum of preconditioned system is more clustered than the spectrum with Dirichlet boundary conditions (see Fig. 4),
- since this spectrum does not touch the circle, one can in principle choose an ellipse such that the origin is not inside or on this ellipse. By using the well-known convergence bound of GMRES [105, 106], GMRES applied to the preconditioned system will converge.

Table 8 Number of iterations of GMRES, QMR and Bi-CGSTAB to reduce the relative residual by 7 orders for a 2D Helmholtz problem with absorbing conditions. The preconditioner $M_{(0,1)}$ is inverted by using a direct solver

	GMRES	QMR	Bi-CGSTAB
10	13	13	9
20	26	27	23
40	69	85	68
80	220	296	161

In practice, we observe a faster convergence if radiation boundary conditions are used, than if Dirichlet or Neumann conditions are used, as seen in Table 8 (refer to Table 7 for comparison). From Table 8, it reveals that Bi-CGSTAB converges faster than QMR. The performance of QMR appears to quickly deteriorate as k increases.

5.4 h-Independence of Convergence

To have insights about the convergence dependence of the gridsize h, we consider again the 1D Helmholtz equation with boundary conditions $u(0) = 1$ and $u(1) = 0$.

For $k = 0$ (the Laplace equation) the eigenvalues are $\mu_{\ell_1}^c = (\ell_1\pi)^2$, $\ell_1 = 1, 2, \ldots$. Using the central difference scheme for the Laplace operator, with $N + 1$ gridpoints and $h = 1/N$, the eigenvalues of the discrete Laplacian operator are given as

$$\mu_{\ell_1} = \frac{4}{h^2}\left(\sin\frac{\pi h \ell_1}{2}\right)^2, \quad \ell_1 = 1, \ldots, N. \tag{88}$$

For some of the eigenvalues, say $\widetilde{\ell}_1$, such that $\pi h\widetilde{\ell}_1 \ll 1$ it follows that $|\mu_{\ell_1} - \mu_{\ell_1}^c| = O(h^2)$ for $\ell_1 < \widetilde{\ell}_1$. So, the smallest eigenvalues of the matrix L_h are good approximations

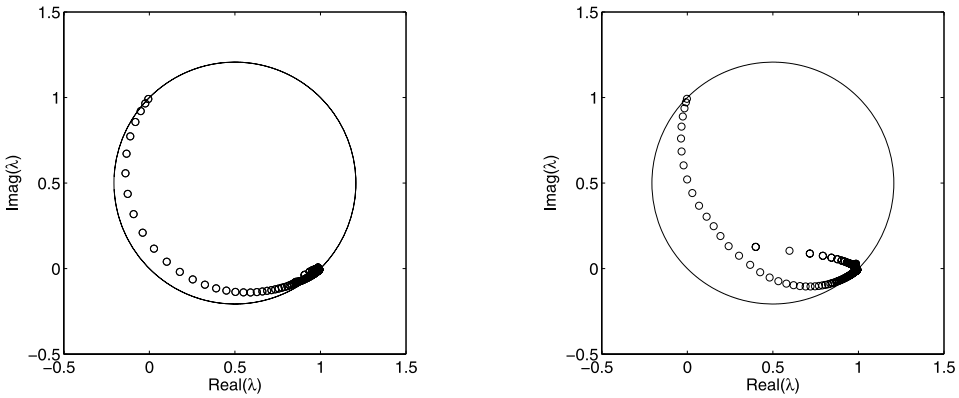

Fig. 5 Spectrum of 1D PML Helmholtz equation for $k = 30$. The thickness of PML is 0.1 (*left*) and 0.2 (*right*) of the physical domain

Table 9 Computational performance of GMRES (number of iterations) to solve Problem (77) with different grid resolutions

k	$M_{(0,0)}$			$M_{(-1,0)}$			$M_{(0,1)}$		
	h^{-1}			h^{-1}			h^{-1}		
	50	100	150	50	100	150	50	100	150
10	13	14	14	17	17	17	15	15	16
40	99	96	98	96	96	99	79	78	80

of the eigenvalues of the continuous problem. Suppose now that $k^2 \neq \mu^c_{\ell_1} \neq 0$ for all ℓ_1. Then we have that

$$\lim_{h\to 0} \min_{\ell_1} |\mu_{\ell_1} - k^2| = |\mu^c_m - k^2| \neq 0, \tag{89}$$

where $|\mu^c_m - k^2| = \min_{\ell_1} |\mu^c_{\ell_1} - k^2|$. Combining this limit with the analysis given in Sect. 5.2, for $M_{h,(0,1)}$ we can show that [42, 136]

$$\lim_{h\to 0} \lambda_{\min}(Q_{(0,1)}) = \frac{|\mu^c_m - k^2|^2}{2k^4}. \tag{90}$$

Since the maximal eigenvalues of $Q_{(0,1)}$ are bounded by 1, we conclude that the convergence behavior of this preconditioners is independent of h (see also [90]). Only initially there can be dependence of the smallest eigenvalue on h. In a similar way, for $M_{(-1,0)}$ we find that

$$\lim_{h\to 0} \lambda_{\min}(Q_{(1,0)}) = \frac{|\mu^c_m - k^2|^2}{4k^4}, \tag{91}$$

which also indicates an h-independent convergence.

Table 9 presents convergence results of GMRES for the 2D Helmholtz equation with Dirichlet boundary conditions, shown for several k and h. As seen in the table, the convergence is independent of the gridsize h. Similar convergence results also hold for Bi-CGSTAB and QMR.

5.5 More General Problems: Heterogeneous Media, PML

In [131] a similar but more general analysis is presented from a more algebraic point of view. The results discussed in the preceding section generally holds for general problem, namely, in the case absorbing-type boundary conditions, the spectrum can be enclosed by a circle related to the location of eigenvalues of the preconditioned Helmholtz problem with Dirichlet boundary conditions in the complex plane. See Theorem 3.5 in [131].

Figure 5 is an example of spectrum for a 1D PML Helmholtz problem with constant $k = 30$, shown for different PML thicknesses. The preconditioner is the shifted Laplacian with imaginary shift, which in the case of PML has the form:

$$\mathcal{M}_{\text{PML}} := -\frac{\partial}{\partial x_1}\left(\frac{s_2}{s_1}\frac{\partial}{\partial x_1}\right) + \hat{\jmath}\frac{\omega^2}{c^2}s_1 s_2, \tag{92}$$

with s_1 and s_2 defined as in (13).

The spectrum is again enclosed by the circle defined in Theorem 5.5. It appears that an increase in PML thickness leads to a spectrum which is far from the origin. This may be good for convergence, but will requires more work due to the enlarged computational domain. Especially for thick PML, one notices a branch in the spectrum, which is most probably due to PML.

Figure 6 shows spectra for a 1D PML Helmholtz problem with heterogeneity. In this case we set the wavenumber as follows:

$$k = \begin{cases} k_{\text{ref}}, & 0 \le x < 0.5, \\ 0.5k_{\text{ref}}, & 0.5 \le x \le 1. \end{cases} \tag{93}$$

In this case, we set $k_{\text{ref}} = 30$. The spectra are shown for different PML thickness. Again, the spectra are enclosed by the same circle defined in Theorem 5.5.

An example of spectra for a 2D case is shown in Fig. 7 for $k = 5$ and different PML thickness. The spectra are again enclosed by the same circle as in the 1D case (Theorem 5.5).

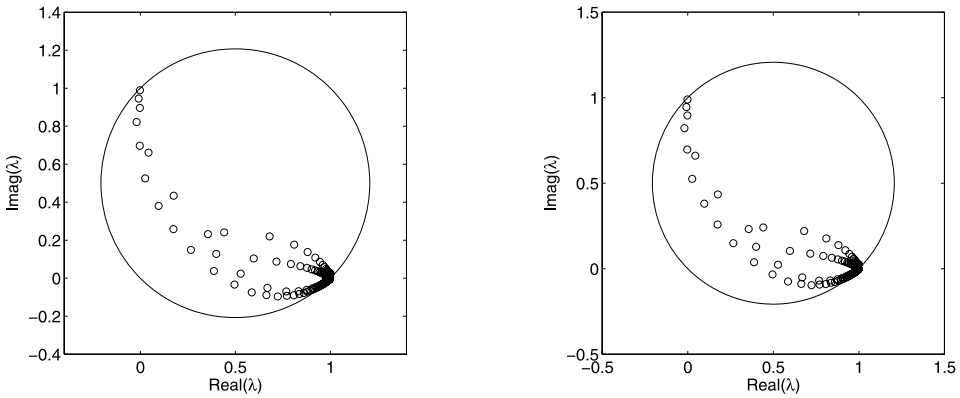

Fig. 6 Spectrum of 1D PML Helmholtz equation for $k_{\text{ref}} = 30$. The thickness of PML is 0.1 (*left*) and 0.2 (*right*) of the physical domain

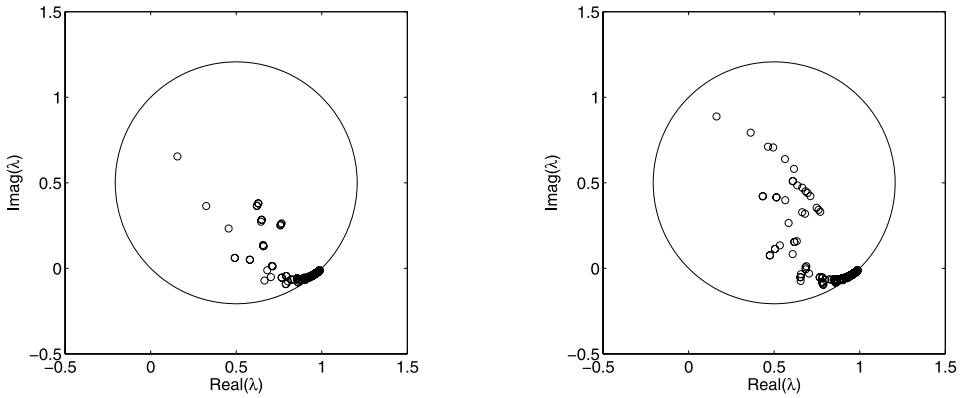

Fig. 7 Spectrum of 2D PML Helmholtz equation for $k = 5$. The thickness of PML is 0.1 (*left*) and 0.2 (*right*) of the physical domain

The similarity between spectral properties of the preconditioned PML equation and spectral properties of the preconditioned Helmholtz equation suggests that the convergence of both cases are expected to be similar.

6 Preconditioner Solves

So far, we only solved the preconditioner exactly; I.e. we applied a sparse direct method to invert the preconditioning matrix exactly. In this section, multigrid is employed to invert it only approximately. Furthermore, by using some multigrid analysis, we can generalize the shift and choose a better shift suitable for both multigrid approximation and Krylov subspace acceleration.

6.1 Multigrid for Solving the Preconditioner

In this case, multigrid is not used to solve the Helmholtz equation, but to approximately solve a preconditioning matrix [10, 43, 62, 82]. We again use standard notations for multigrid analysis: the subscripts $_h$ and $_H$ always indicate the fine and coarse grids, with $h = 2H$.

In [10] and [62] the Laplacian is used as the preconditioner for CGNR, and is approximately inverted by SSOR and multigrid, respectively. As multigrid is very efficient for Poisson problems, a few multigrid iterations in the preconditioning steps of CGNR leads to a good iterative method. As analyzed in Sect. 5 and as well shown in the numerical examples in [10] and [62] the method however becomes inefficient for large k.

In [82], the definite Helmholtz equation, which is related to $(\beta_1, \beta_2) = (-1, 0)$ in our discussion in Sect. 5 is handled by multigrid. The convergence using this option is somewhat faster than that in [10] and [62].

In this section we discuss multigrid as the solver for the preconditioning system $M_h u_h = w_h$.

6.2 Multigrid Convergence for a Purely Imaginary Shift

Recall that our preconditioning operator reads, as in Sect. 5:

$$\mathcal{M} := -\nabla \cdot \nabla - \hat{\jmath} k^2. \tag{94}$$

A common way to analyze the convergence of multigrid is by using Rigorous Fourier Analysis (RFA). This approach is more quantitative than the approach discussed in [67]. For given multigrid components (type of smoother, relaxation factor, cycle, etc.) an estimate of the smoothing properties and two- (or three-) grid can be determined; See [20–22, 114, 122, 139] for the two-grid analysis and [141] for the three-grid analysis. The three-grid analysis gives some more

details about the coarse-grid correction. If a large difference occurs between the two-grid and the three-grid convergence factors, this is an indication for a problematic coarse-grid correction. For the complex Helmholtz preconditioner it is important to analyze the coarse-grid correction carefully (Sect. 3.3).

A software for two- and three-grid analysis is available nowadays and can be freely downloaded from: www.mgnet.org/mgnet-codes-wienands.html, with detailed instructions in [140].

In order to do Rigorous Fourier Analysis, we need some multigrid components. In this section, we detail the multigrid components that can be specified for approximately inverting a discrete version of the preconditioning operator $\mathcal{M}$ (49). We consider the 5-point discretization and denote the discrete preconditioner as M_h. Furthermore, we consider the *right* preconditioned system. Thus, we solve the equivalent system

$$A_h M_h^{-1} w_h = g_h, \qquad M_h u_h = w_h, \tag{95}$$

with $M_h u_h = w_h$ solved by one multigrid iteration. Standard multigrid coarsening, i.e., doubling the mesh size h in every direction is chosen.

Smoothers Classical iteration methods like Jacobi with underrelaxation and Gauss-Seidel iterations can be used as smoothers. We denote the two smoothers by ω_r-JAC and ω_r-GS, respectively. In principle one can choose the underrelaxation parameter $\omega_r \in \mathbb{C}$ (as in the two-stage complex Jacobi iteration). The Fourier analysis indicates that there is no real benefit from such a choice.

Prolongation and Restriction Operators For the intergrid transfer of the grid functions two (linear) transfer operators (39) and (40) are required. There are several ways to construct the transfer operators.

A frequently used prolongation operator is based on *bilinear interpolation* from $\mathcal{G}_H$ to $\mathcal{G}_h$. This interpolation is defined as (see Fig. 8)

$$I_H^h \widehat{v}_H(x,y) = \begin{cases} \widehat{v}_H(x,y) & \text{for } A, \\ \frac{1}{2}[\widehat{v}_H(x_1, x_2+h) + \widehat{v}_H(x_1, x_2-h)] & \text{for } q, \\ \frac{1}{2}[\widehat{v}_H(x_1+h, x_2) + \widehat{v}_H(x_1-h, x_2)] & \text{for } p, \\ \frac{1}{4}[\widehat{v}_H(x_1+h, x_2+h) + \widehat{v}_H(x_1+h, x_2-h) & \\ +\widehat{v}_H(x_1-h, x_2+h) + \widehat{v}_H(x_1-h, x_2-h)] & \text{for } r. \end{cases} \tag{96}$$

In order to better accommodate heterogeneities, it is natural to used an operator-dependent interpolation. What follows is the operator-dependent interpolation due to de Zeeuw's transfer operators [35], denoted by MD, which has been simplified to exploit symmetry of the linear system. See also [3] and [77] for other operator-dependent interpolations.

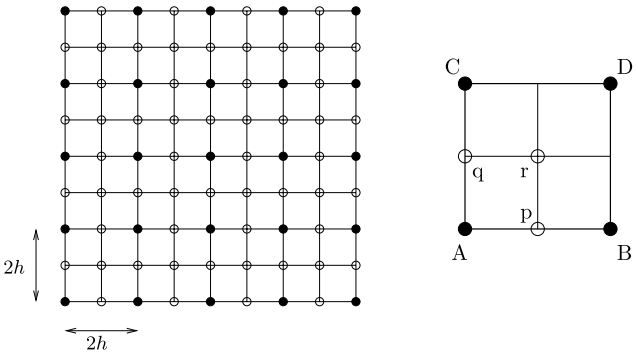

Fig. 8 Two grids in standard coarsening. The coarse grid is denoted by (•) while the fine grid by (○)

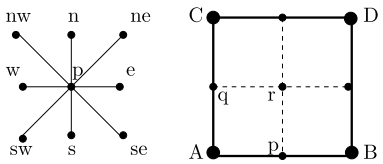

Fig. 9 *Left*: Nine point stencil with numbering (in this case, for point p), *Right*: Coarse grid cell and four fine cells (Coarse grid indices by capital and fine grid indices by lower case letters)

The notation in a stencil for the explanation of the prolongation is as in Fig. 9 (left side). The right picture in Fig. 9 shows one coarse and four fine grid cells with indices for the explanation of the interpolation weights. Capital letters denote coarse grid, lower case letters fine grid points. Operator element m_p^w, for example, denotes the west element of operator M_h at point p on the fine grid.

The corrections from the coarse to the fine grid are obtained by interpolation among nearest coarse grid neighbors. The operator-dependent interpolation weights, w, to determine the fine grid correction quantities e_h are derived with the following formulas:

- for fine grid points p in Fig. 9: $e_{h,p} = w_A e_{H,A} + w_B e_{H,B}$. $w_A = w_w$; $w_B = w_e$,

where

$$d_w = \max(|m_p^{sw} + m_p^w + m_p^{nw}|, |m_p^{sw}|, |m_p^{nw}|), \tag{97}$$

$$d_e = \max(|m_p^{se} + m_p^s + m_p^{ne}|, |m_p^{se}|, |m_p^{ne}|), \tag{98}$$

$$w_w = \frac{d_w}{d_w + d_e}, \qquad w_e = \frac{d_e}{d_w + d_e} \tag{99}$$

- for fine grid points q in Fig. 9: $e_{h,q} = w_A e_{H,A} + w_C e_{H,C}$. $w_A = w_s$; $w_C = w_n$,

with

$$d_n = \max(|m_q^{nw} + m_q^n + m_q^{ne}|, |m_q^{nw}|, |m_q^{ne}|), \tag{100}$$

$$d_s = \max(|m_q^{sw} + m_q^s + m_q^{se}|, |m_q^{sw}|, |m_q^{se}|), \tag{101}$$

$$w_s = \frac{d_s}{d_s + d_n}, \qquad w_n = \frac{d_n}{d_s + d_n}. \tag{102}$$

On the remaining points the prolongation is defined as follows:

- On fine grid points that are also coarse points: $e_h(A) = e_H(A)$,
- On points r: $e_h(r)$ is determined so that $M_h I_H^h e_H = 0$ at r.

$|\cdot|$ denotes the modulus, in this case. In [3], for d_w, for example, the lumped sum of three elements, $m_p^{sw} + m_p^w + m_p^{nw}$ is chosen. For satisfactory convergence it is, however, important to use the *modulus* of the operator elements, as in (97), (98), (100), (101) in the definition of the interpolation weights.

For the restriction, the simplest choice is injecting the value on a fine grid points to the corresponding coarse grid points. This is called *injection*. Injection, however, is not a good restriction for (94). A commonly used restriction operator is called the *full weighting* (FW) operator, which can be considered as a nine-point weighted average of a (fine) grid function on a coarse grid. For full weighting, the restriction operator reads

$$\begin{aligned} & d_H(x_1, x_2) \\ &= I_h^H d_h(x_1, x_2) \\ &= \frac{1}{16}[4d_h(x_1, x_2) + 2d_h(x_1 + h, x_2) + 2d_h(x_1 - h, x_2) \\ &\quad + 2d_h(x_1, x_2 + h) + 2d_h(x_1, x_2 - h) \\ &\quad + d_h(x_1 + h, x_2 + h) \\ &\quad + d_h(x_1 - h, x_2 + h) + d_h(x_1 + h, x_2 - h) \\ &\quad + d_h(x_1 - h, x_2 - h)], \end{aligned} \tag{103}$$

or, in stencil notation,

$$I_h^H = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H. \tag{104}$$

In general, we do not choose for the adjoint of the prolongation operator, which is commonly used but not absolutely necessary (see [3] and [33]), unless a combination of bilinear interpolation and full weighting is evaluated. In this case the adjoint of bilinear interpolation operator equals the full weighting operator. We observe in our numerical experiments especially with strongly varying coefficients that choosing the combination of full weighting and the operator-dependent interpolation brings to a robust method. For constant coefficients and mildly varying wavenumbers, however, bilinear interpolation (in combination with full weighting) also gives very satisfactory convergence results.

Table 10 Asymptotic convergence from Fourier analysis with numerical multigrid convergence, $M_{h,(0,1)}$. μ is the smoothing factor; ρ_{2g}, ρ_{3g} are the two- and three-grid convergence factor from Fourier analysis; ρ_h is the numerical multigrid convergence factor. The smoother is ω_r-RB-GS with $\omega_r = 1$

	μ	ρ_{2g}	$\|T_h^{2h}\|_S$	ρ_{3g}	$\|T_{2h}^{4h}\|_S$	ρ_h, V-cycle
V(1, 0)	0.25	0.25	0.56	0.24	0.56	0.235
V(1, 1)	0.06	0.06	0.14	0.06	0.14	0.055

Table 11 Number of multigrid V-cycles to solve the preconditioner $M_{h,(0,1)}$, with MD and FW as the transfer operators. The CPU time is presented for $k = 100$. The termination criterion is $\|r^j/r^0\| \le 10^{-6}$

Cycle	k					Time
	20	40	60	80	100	(sec)
V(1, 0)	9	9	9	9	9	1.01
V(1, 1)	7	8	6	8	8	1.07
V(2, 1)	4	6	8	5	6	1.16
V(1, 2)	4	4	7	4	5	0.97

In Table 10 we show results from RFA for the matrix related to the preconditioning operator (94), for $k = 100$. In this case, we have used multigrid with V-cycle and ω_r-RB-GS smoother, where RB stands for red-black ordering. The intergrid transfer operators are matrix-dependent (MD) interpolation based on [35] and the full weighting (FW) restriction operator. The use of matrix-dependent interpolation operator is somewhat natural since it is suitable for varying wavenumber (heterogeneous) problems.

In the table, the two- and three-grid convergence factor are denoted by ρ_{2g} and ρ_{3g} respectively, while ρ_h denotes the numerical multigrid convergence [122]. We compare V(1, 0)-cycle and V(1, 1)-cycle. (The first integer in the parentheses indicates the number of presmoothing. The second integer is the number of postsmoothing.)

From RFA, the asymptotic two-grid convergence factor for the V(1, 1)-cycle is about 0.06, which is in a good agreement with the numerical convergence. Furthermore, the norm of the two-grid operator is well below 0.2. Multigrid for (94) behaves very similarly to the definite real version of the Helmholtz operator (and of the Laplace operator). See Table 11 for the actual number of iterations to reduce the residual by 6 orders.

Another example from Fourier analysis applied to (94) is shown in Table 12 for ω_r-JAC, with JAC denoting Jacobi smoother. Here we have used F-cycle. With $\gamma = 2$,

Table 12 Comparison of asymptotic convergence from Fourier analysis with numerical multigrid convergence, $M_{h,(0,1)}$. μ is the smoothing factor; ρ_{2g}, ρ_{3g} are the two- and three-grid convergence factor from Fourier analysis; ρ_h is the numerical multigrid convergence factor. The smoother is ω_r-JAC with $\omega_r = 0.8$

	μ	ρ_{2g}	$\|T_h^{2h}\|_S$	ρ_{3g}, $\gamma = 2$	$\|T_{2h}^{4h}\|_S$	ρ_h, F-cycle
F(1, 0)	0.60	0.60	0.60	0.60	0.60	0.58
F(1, 1)	0.36	0.36	0.36	0.36	0.36	0.34

this means that the F-cycle is identical to the W-cycle. We set $\omega_r = 0.8$. Again, the agreement between the smoothing, two- and three-grid Fourier analysis results with one and two smoothing iterations and the numerical convergence is excellent, as presented in Table 12. The results obtained are very similar to the convergence factors for the Laplace operator with ω_r-JAC ([122]).

We note here that:

- For $M_{h,(0,1)}$ direct PDE coarse-grid discretization and RB-GS Gauss-Seidel relaxation can be adopted. With two smoothing iterations, this leads to a good multigrid method. The cost of RB-GS per iteration is almost twice as expensive as that of one Jacobi iteration.
- Bilinear interpolation in combination with half injection can be employed to obtain the 5-point stencil discretization on the coarse levels. This consequently reduces work on the coarse levels. This is however a divergent method for $M_{(0,1)}$ if V(1, 1)-cycle is used. W(1, 1)-cycle does better than V(1, 1)-cycle, but the improvement is not substantial. Furthermore, W-cycle is more expensive than V-cycle.
- One can also obtain a 5-point stencil on the coarse grids by applying a direct discretization to the operator (94). In this case, care should be taken in incorporating radiation boundary conditions in the coarse grid discretization. Our observation, however, leads to a conclusion that this is not also a good method.

It is important to note that multigrid is only used to approximately invert the preconditioner M in a Krylov subspace algorithm. Hence, it is not necessary to do multigrid iterations up to a very high accuracy, and in practice it is sufficient to only do one multigrid iteration. Then, we keep the preconditioning work as small as possible.

6.2.1 Numerical Experiment

Table 13 shows convergence results from the 2D homogeneous Helmholtz equation with radiation boundary conditions. For the linear solver, Bi-CGSTAB is used. We compare the case without preconditioner, with ILU(0) and ILU(1) preconditioners and with preconditioner $M_{h,(0,1)}$. For $M_{h,(0,1}$ we approximately invert it by using one V(1, 0) or V(1, 1) multigrid iteration.

Table 13 Number of Bi-CGSTAB iterations for a 2D constant wavenumber Helmholtz problem, shown for various k

k	10	20	40	50	100
grid	32^2	64^2	128^2	192^2	384^2
No-Prec	150	846	1857	3054	6283
ILU(0)	75	221	474	634	1340
ILU(1)	35	113	238	295	563
MG($V(1, 0)$)	18	36	63	71	133
MG($V(1, 1)$)	16	33	60	70	133

Observe that:

- with $M_{h,(0,1)}$ the number of iterations still increases but slowly with respect to k,
- V(1, 0) and V(1, 1) multigrid leads to almost the same number of Bi-CGSTAB iteration for all k's under consideration
- ILU(1) and multigrid require almost the same CPU time. While the number of iteration reduces dramatically, the use of multigrid increases the total amount of work per iteration.

6.3 Multigrid for a General Complex Shift

We can relax the restriction for β_1 and β_2 by only requiring that $\beta \neq 0$. Since multigrid is used to solve the preconditioner, the pair (β_1, β_2) can be determined from convergence properties of multigrid methods. It is not necessary that the preconditioner related to the optimal choice of (β_1, β_2) can be solved by multigrid with the typical *text book* multigrid convergence.

In [43] some possible choices for the pair (β_1, β_2) in (49) are considered and evaluated by using Fourier analysis. We present in Table 14 results of Fourier analysis for two preconditioners:

$$\mathcal{M}_{(1,1)} = -\nabla \cdot \nabla - (1 - \hat{j})k^2 \quad \text{and} \tag{105}$$

$$\mathcal{M}_{(1,\frac{1}{2})} = -\nabla \cdot \nabla - \left(1 - \frac{1}{2}\hat{j}\right)k^2. \tag{106}$$

In both cases, F(1, 1) multigrid cycle is used; this cycle is more effective than V-cycle but requires less work than W-cycle. With only three grid levels, there is no different between F-cycle and W-cycle. The smoother is point Jacobi with relaxation $\omega_r = 0.7$ for (105) and $\omega_r = 0.5$ for (106).

Even though multigrid is more efficient for solving the preconditioner with $(\beta_1, \beta_2) = (1, 1)$ than the preconditioner with $(\beta_1, \beta_2) = (1, \frac{1}{2})$, the latter is more effective if it is used for Krylov subspace acceleration. This is observed in Table 15, where the 2D Helmholtz equation with constant k is solved by Bi-CGSTAB with preconditioners

Table 14 Multigrid convergence analysis for $(\beta_1, \beta_2) = (1, 1)$ and $(\beta_1, \beta_2) = (1, 0.5)$, with F(1,1)-cycle. See the text for details

(β_1, β_2)	μ	ρ_{2g}	ρ_{3g}, $\gamma = 2$	ρ_h, F(1, 1)
$(1, 1)$	0.47	0.47	0.47	0.45
$(1, \frac{1}{2})$	0.60	0.60	0.60	0.61

Table 15 Bi-CGSTAB iterations to reduce the initial residual by 7 orders of magnitude. For every (β_1, β_2): number of iterations (first row) and CPU time in second (second row). "–" means "not computed"

(β_1, β_2)	k						
	40	50	80	100	150	200	500
$(0, 1)$	57	73	112	126	188	–	–
	0.44	0.92	4.3	7.7	28.5	–	–
$(1, 1)$	36	39	54	74	90	114	291
	0.30	0.51	2.2	4.5	13.9	30.8	515
$(1, 0.5)$	26	31	44	52	73	92	250
	0.21	0.40	1.8	3.3	10.8	25.4	425

(105) and (106). Moreover, compared to the purely imaginary shift preconditioner $\mathcal{M}_{(0,1)}$, the preconditioner with $(\beta_1, \beta_2) = (1, 0.5)$ requires more than 2.5 times less iterations and CPU times. Hence, it is even sufficient to employ a multigrid iteration with a convergence factor $\rho_h \approx 0.6$.

We note that within the context of Krylov subspace acceleration, it is difficult to have a quantitative estimate of a Krylov subspace method with multigrid-based preconditioner. Qualitative insights can be obtained from the spectrum of the preconditioned system. This is can be done via Fourier analysis by considering the regular splitting of M_h, which is discussed in detail in [43]. This allows us to visualize the corresponding eigenvalues for various multigrid cycles.

6.3.1 Numerical Experiment

We consider a simple 2D heterogeneous problem, called the wedge problem, to illustrate the effectiveness of this new shift $(\beta_1, \beta_2) = (1, 0.5)$. Furthermore, three boundary conditions are implemented: the 2nd order absorption condition, the damping layer and PML. Figure 10 shows the convergence results. Compared to the purely imaginary shift $(\beta_1, \beta_2) = (0, 1)$, a convergence speed up of more than three times in terms of number of iteration is gained. Since the work per iteration is generally the same, CPU time is also reduced by the same order. Interestingly, the convergence also hardly depends on the type of boundary conditions used.

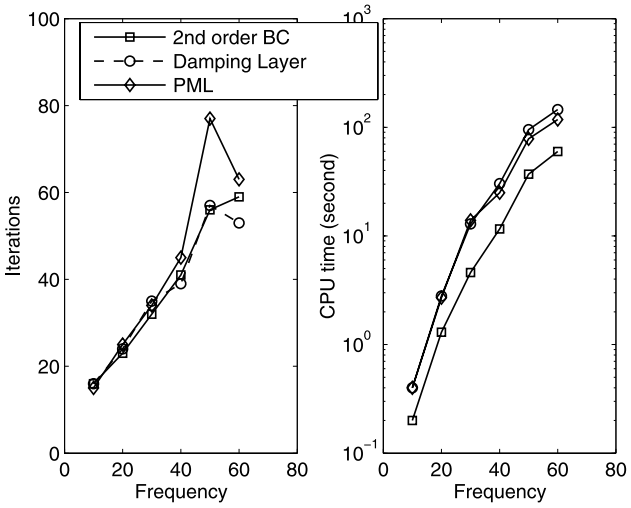

Fig. 10 Convergence results for the wedge problem

6.3.2 Summary

Based on our analysis and observations, in order to obtain a robust and efficient iterative method for the Helmholtz equation we require to use the following:

- Bi-CGSTAB as the outer iteration to solve the Helmholtz matrix. This algorithm requires minimal memory than GMRES (and QMR, slightly) and is faster than GMRES and QMR. We note that the right preconditioned Bi-CGSTAB must be employed because this requires only two preconditioner solves (instead of four for the left-preconditioned version)
- The preconditioner is obtained from a discretization of the operators: For the Helmholtz equation:

$$\mathcal{M} := -\nabla \cdot \nabla - (1 - 0.5\hat{j})k^2. \tag{107}$$

For the damping layers:

$$\mathcal{M}_{\text{Damp}} := -\nabla \cdot \nabla - (1 - 0.5\hat{j})(1 - \alpha\hat{j})k^2. \tag{108}$$

For PML:

$$\mathcal{M}_{\text{PML}} := -\frac{\partial}{\partial x_1}\left(\frac{s_2}{s_1}\frac{\partial}{\partial x_1}\right) - \frac{\partial}{\partial x_2}\left(\frac{s_1}{s_2}\frac{\partial}{\partial x_2}\right) - (1 - 0.5\hat{j})\frac{\omega^2}{c^2}s_1 s_2. \tag{109}$$

As will be shown in Sect. 7, $\mathcal{M}$ can be discretized differently from the Helmholtz operator. But it is important that the same boundary conditions are used in the Helmholtz equation and the preconditioner operator.
- One multigrid iteration is performed on the preconditioner with the following components: matrix-dependent interpolation, the full weighting, F-cycle with one pre- and postsmoothing with the Jacobi iteration and relaxation factor $\omega_r = 0.5$.

6.4 Inner-Outer Krylov Iterations

Beside multigrid, the preconditioner $\mathcal{M}$ can also be approximately inverted by using Krylov subspace iterations, yielding to an inner-outer iteration process [125]. Note that in this case, the preconditioner is no longer constant. Since Krylov subspace methods are not a linear method, the subspace generated by this process does not lead to the standard Krylov subspace similar to that in the case of constant preconditioner. One can do two approaches in this case. The first approach is to solve the preconditioning problems very accurately (usually up to the machine precision). In this case, another preconditioner may be needed in order to reduce work in the inner iteration. The other approach is by using flexible Krylov iterations, by using, e.g., FGMRES [103], GMRESR [130] or FQMR [116].

6.5 Extension to the 3D Helmholtz Equation

In 3D, the linear system arising from discretization of the Helmholtz equation becomes extremely large with large bandwidth. In this case a black box (or algebraic) incomplete LU factorization becomes impractical. Fill-in can be very excessive, which require more work for the preconditioner solve. However, the shifted Laplace preconditioner can be easily extended to 3D [45, 102]. One that is needed is an efficient 3D multigrid solver.

In principle a multigrid method with standard coarsening can be used for the preconditioner solve. However, if a 3D multigrid method with a 2D *semi-coarsening* strategy [36, 120, 138] combined with line-wise ω_r-Jacobi smoothing in the *third* direction is used, much of results from 2D multigrid with standard coarsening can be employed. In this 3D multigrid method, the coarsening is only performed simultaneously in two directions; the third direction is kept uncoarsened. This strategy is illustrated in Fig. 11.

The transfer operator is adapted as follows. We assume that on the finest level the 27-point stencil discretization is used.

To determine the 3D interpolation weights, we consider the 27-point stencil matrix (see Fig. 12), written as follows [138]:

$$
\begin{aligned}
(Mu)_{i_1,i_2,i_3} = \sum_{iz=-1,0,1} \big(& m(iz)^1_{i_1,i_2,i_3} u_{i_1-1,i_2-1,i_3+iz} \\
& + m(iz)^2_{i_1,i_2,i_3} u_{i_1,i_2-1,i_3+iz} \\
& + m(iz)^3_{i_1,i_2,i_3} u_{i_1+1,i_2-1,i_3+iz} \\
& + m(iz)^4_{i_1,i_2,i_3} u_{i_1-1,i_2,i_3+iz} \\
& + m(iz)^5_{i_1,i_2,i_3} u_{i_1,i_2,i_3+iz} \\
& + m(iz)^6_{i_1,i_2,i_3} u_{i_1+1,i_2,i_3+iz} \\
& + m(iz)^7_{i_1,i_2,i_3} u_{i_1-1,i_2+1,i_3+iz} \\
& + m(iz)^8_{i_1,i_2,i_3} u_{i_1,i_2+1,i_3+iz} \\
& + m(iz)^9_{i_1,i_2,i_3} u_{i_1+1,i_2+1,i_3+iz} \big).
\end{aligned} \tag{110}
$$

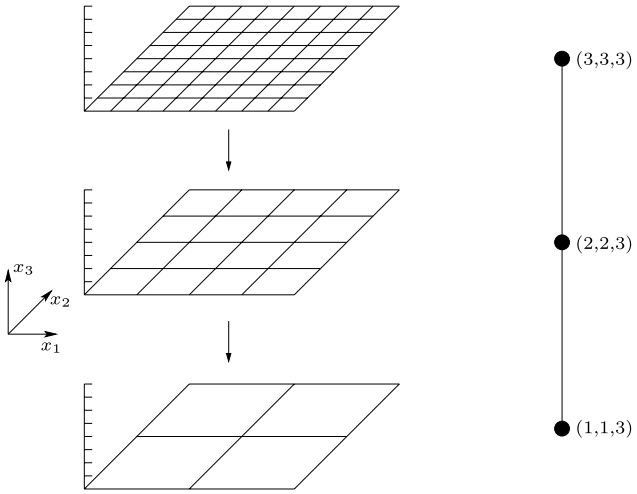

Fig. 11 Semicoarsening of three grid levels: standard coarsening in two directions (x_1 and x_2), the third (x_3) direction is kept uncoarsened

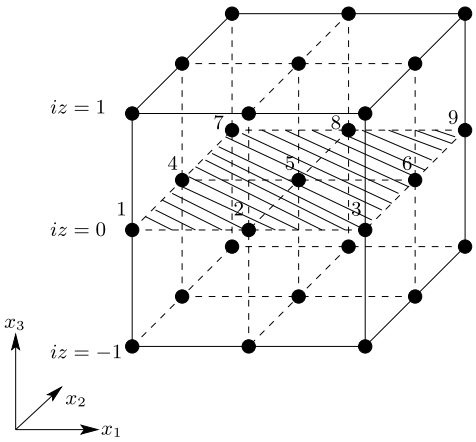

Fig. 12 The 27-point stencil

In case the coarsening is only done in (x_1, x_2) direction, a lumped 9-point stencil matrix $\widetilde{M}_h$ in an (x_1, x_2)-plane is defined as:

$$
\begin{aligned}
(\widetilde{M}\phi)_{i_1,i_2,i_3} = {} & \widetilde{m}^1_{i_1,i_2,i_3} u_{i_1-1,i_2-1,i_3} + \widetilde{m}^2_{i_1,i_2,i_3} u_{i_1,i_2-1,i_3} \\
& + \widetilde{m}^3_{i_1,i_2,i_3} u_{i_1+1,i_2-1,i_3} + \widetilde{m}^4_{i_1,i_2,i_3} u_{i_1-1,i_2,i_3} \\
& + \widetilde{m}^5_{i_1,i_2,i_3} u_{i_1,i_2,i_3} + \widetilde{m}^6_{i_1,i_2,i_3} u_{i_1+1,i_2,i_3} \\
& + \widetilde{m}^7_{i_1,i_2,i_3} u_{i_1-1,i_2+1,i_3} + \widetilde{m}^8_{i_1,i_2,i_3} u_{i_1,i_2+1,i_3} \\
& + \widetilde{m}^9_{i_1,i_2,i_3} u_{i_1+1,i_2+1,i_3},
\end{aligned}
$$

Table 16 Performance of preconditioned Bi-CGSTAB in terms of the number of iterations and CPU time (in sec.) to reach convergence for the Helmholtz equation with constant wavenumber k. $kh = 0.625$ and the preconditioner is $M_{(1,0.5)}$

k	BI		MD	
	Iter	Time	Iter	Time
10	9	0.65	9	0.71
20	13	6.89	13	6.64
30	17	25.52	18	27.63
40	21	75.99	21	71.35
50	24	136.31	24	136.33
60	26	251.62	27	276.88

with

$$\widetilde{m}^p_{i_1,i_2,i_3} = m(-1)^p_{i_1,i_2,i_3} + m(0)^p_{i_1,i_2,i_3} + m(1)^p_{i_1,i_2,i_3},$$
$$p = 1, 2, \ldots, 9.$$

Based on the lumped 9-point stencil the coarse-to-fine grid interpolation can be determined in the same way as described in the previous section for 2D for bilinear interpolation (BI) or in [35] for operator-dependent interpolation (MD).

Numerical Experiment Numerical results are shown in Table 16 for various wavenumbers k. The convergence results shown in the table are similar to the 2D counterparts, or even faster convergence is observed. In the constant wavenumber case, the use of bilinear or operator-dependent interpolation in multigrid does not lead to a different computational performance.

7 Numerical Examples with Realistic Problems

In this section we consider two difficult problems arising from seismic inversion in geophysics. In seismic inversion one tries to generate an image of the earth's substructure from a set of recorded seismic data. In the inversion process, to generate such an image one needs to do forward modeling. Forward modeling means solving a wave equation, which in our case is the 2D or 3D Helmholtz equation. The initial velocity background is usually taken from a synthetic model. During the course of inversion process, the synthetic model is continuously updated until a convergence is reached. In the case of convergence, the computed data in the recording positions for the updated synthetic model is the same as the recorded data (see, e.g., [99, 118]). To obtain "good" images, the data are sampled for various frequencies ranging from usually 5 to 70 Hz. Since, for every sampled frequency forward modeling must be performed successfully, it is crucial to have a convergence iterative method for all frequencies.

Table 17 Bi-CGSTAB convergence for the Marmousi problem. CPU time (in parentheses) is measured in seconds

f (Hz)	Grid	(β_1, β_2)	
		(0,1)	(1,0.5)
1	751 × 201	91 (36)	39 (16)
10	751 × 201	187 (72)	54 (22)
20	1501 × 401	388 (642)	98 (167)
30	2001 × 534	484 (1472)	144 (444)

We consider two synthetic models: 2D Marmousi model [19] and 3D saltdome model of the North sea [102]. These are typically hard problem and, to author's knowledge, there exists no iterative method which gives convergent solution for all frequencies. See, for example, [98] for convergence problems of an iterative solver applied to the 2D Marmousi case.

7.1 2D Marmousi Problem

A part of 2D Marmousi problem is depicted in Fig. 13. This model represents a 2D earth's substructure with a strong velocity contrast: from $c = 1500$ m/s to 4000 m/s, and covers an area of 1.6×6 km^2. A source of different frequencies, ranging from 10 Hz to 60 Hz, is located at the center just below the upper surface.

7.1.1 Convergence on a Sequential Machine

To solve the problem, we use Bi-CGSTAB preconditioned by the shifted Laplacian preconditioner of the form (106), i.e. with $(\beta_1, \beta_2) = (1, 1/2)$, which results in the best choice for constant wavenumber cases. The extension to heterogeneous case is very easy. In this case, we only include the variation of $k = 2\pi f/c$ in the discretization of (106).

For multigrid, we have used the multigrid components summarized in Sect. 6.3.2.

Table 17 presents convergence results based on a single processor machine. The figures shown there are related to the reduction of the initial residual by 7 orders of magnitude. We show also convergence results for $(\beta_1, \beta_2) = (0, 1)$. Compared to the latter, the use of $(\beta_1, \beta_2) = (1, 0.5)$ in the preconditioner leads to a convergence speed up of a factor of more than 3. This gain is slightly more than the gain we can get in the constant wavenumber case. Furthermore, the number of iterations depends linearly on the frequency, with only a small proportionality constant.

In [44] ILU(0) and ILU(1) have been used as the preconditioners for Bi-CGSTAB. For the 2D Marmousi problem with frequency $f = 20$ Hz, the shifted Laplace preconditioner with $(\beta_1, \beta_2) = (1, 0.5)$ leads to a convergence which is about 40 and 18 times faster than ILU(0) and ILU(1), respectively, in terms of the number of iterations. The iteration

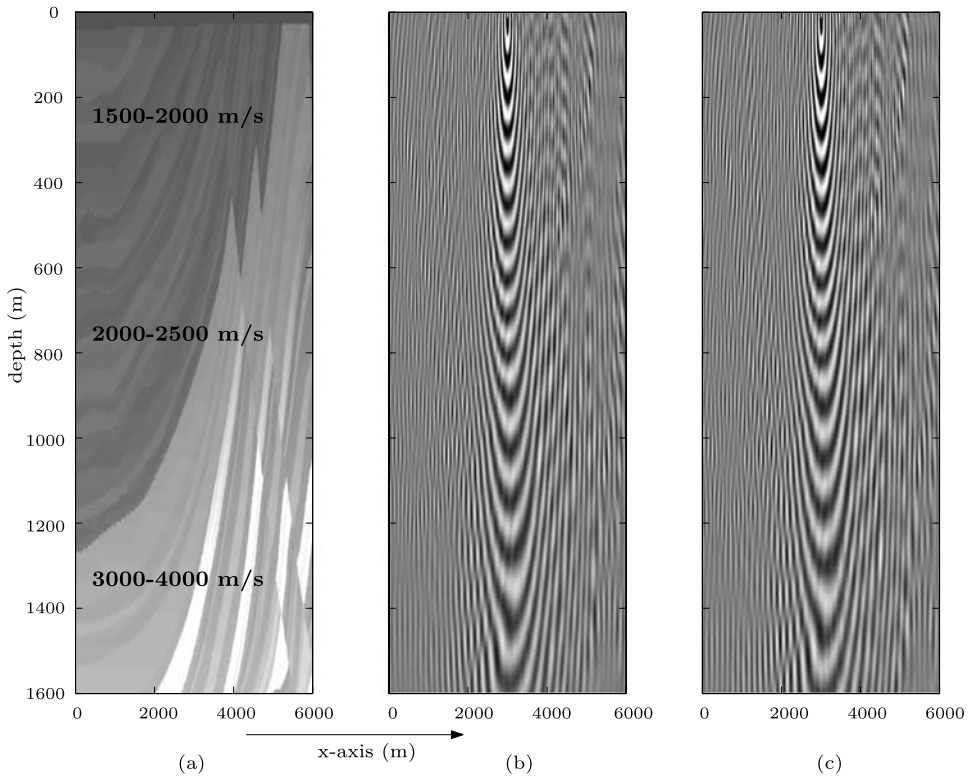

Fig. 13 Marmousi problem (not to scale). **a** Velocity distribution in meter/s, **b** real part of the solution for $f = 20$ Hz, 5-point stencil, **c** real part of the solution for $f = 20$ Hz, 9-point stencil ($\gamma = 2/5$) with absorption layers

Table 18 Bi-CGSTAB convergence for the Marmousi problem. The Helmholtz equation is discretized by using a 9-point stencil ($\gamma = 2/5$). The preconditioner is discretized with the 5-point stencil. 50 gridpoints are added for absorption layers

f (Hz)	$(\beta_1, \beta_2) = (1, 0.5)$	
	#iter	CPU (s)
1	9	9
10	47	37
20	87	193
30	138	513

process requires CPU time which is about 9 and 5 times less than ILU(0) and ILU(1), respectively.

This Marmousi problem has also been solved by using the 9-point stencil with absorption layers added surrounding the original physical domain. Fifty grid points are used in the absorption layers. In these layers, α in (11) increases quadratically outwardly, starting from zero at the interface between the physical region and absorption layers.

Here, we have used only the five-point stencil to discretize the preconditioning operator; Thus the Helmholtz equation and the preconditioning operator are discretized differently. Convergence results are shown in Table 18. The real part of the solutions at 20 Hz is as in Fig. 13c.

Compared to the results in Table 17, the use of the 9-point stencil does not influence the convergence negatively, as the convergence is similar to that with the 5-point stencil. The CPU time increases somewhat due to a larger size, which is due to the inclusion of the absorption layers. Furthermore, this result suggests that the preconditioning operator needs not be discretized by the same discretization as the Helmholtz equation.

7.1.2 Performance on Parallel Machines

To solve high frequency problems, the computational grid has to be refined to be able to capture the smallest wavelength in the solution. As shown in Table 17 of the previous subsection, we only compute solutions up to $f = 30$ Hz. Beyond this frequency, the computing power becomes very demanding, and parallel computing and algorithms become necessary.

Implementations of the shifted Laplacian preconditioner on a parallel machine are discussed in [81, 101, 102]. As the method mainly consists of a Krylov subspace and a multigrid method, parallelization of the method depends only on the parallelism of the Krylov and multigrid method. Note that both methods consist mainly of matrix-vector multiplications, inner products of two vectors and vector updates. Multiplication of a matrix by a vector, which is the costliest part in both methods, can be done in parallel. This is also

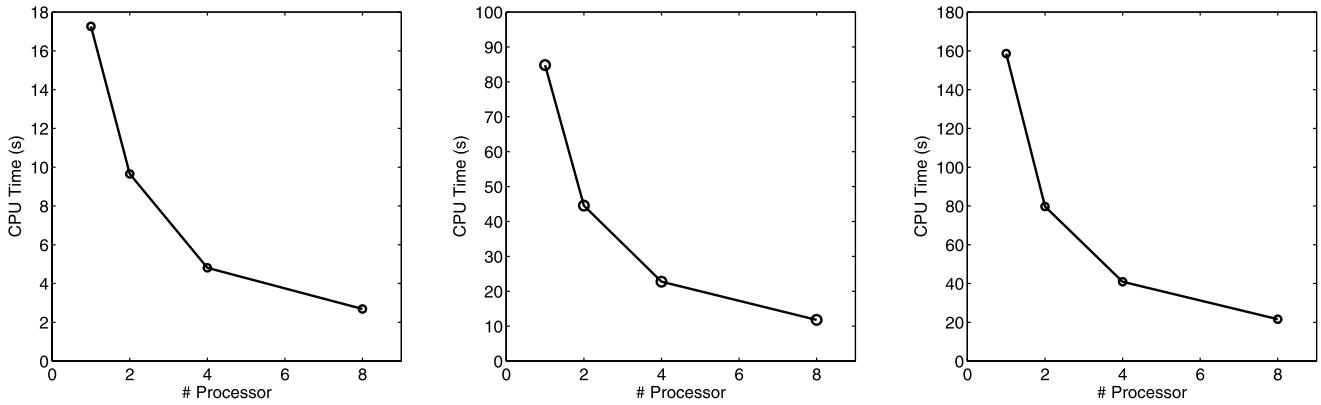

Fig. 14 CPU time vs. # processors

Table 19 Bi-CGSTAB convergence for the 3D saltdome model

f (Hz)	# unknown	# proc.	Iteration	CPU time (hour)
5	11E+6	9	111	0.55
10	55E+6	18	216	3.30

the case for the inner products and vector updates. In particular, Bi-CGSTAB is a parallelizable algorithm. For parallel multigrid methods, see, e.g., [95, 96].

Figure 14 shows CPU time measured for different numbers of processors used for solving the Marmousi problem with some physical damping, which suggests that the method is scalable.

7.2 3D Saltdome Model

The 3D saltdome velocity model mimics a complex earth's structure in the North sea. This model covers a volume of $8920 \times 4440 \times 5100$ m^3, with a velocity contrast ranging from 1700 m/s to 4900 m/s. A source is located on the upper surface of the computational box at position $(4460, 2220, 20)$.

Table 19 shows convergence results for two frequencies [102]: 5 Hz and 10 Hz, solved on 11 million and 50 million gridpoints, making them impossible to fit in a single processor machine. In this case we have used a machine with 9 and 18 processors, respectively. Here we do not study scalability of the parallel implementation for the 3D saltdome. One observes that the convergence depends linearly on the frequency.

A scalability study for 3D cases is done in [102] for a 3D wedge model in a unit cube with a reference wavenumber $k_{\text{ref}} = 150$, solved on 55 million gridpoints. For a various number of processors, the method reduces the initial residual by 6 orders of magnitude within 91 iterations. The parallel scalability is satisfactory. The scalability is however less if many processors (in this case, more than 16 processors) are used, due to communication time between each processor.

8 Conclusion

Iterative methods for large Helmholtz problems are still an active research. In the last decade, progress has been made within different iterative method framework.

In this paper we summarized development of the shifted Laplacian preconditioner for the Helmholtz equation, and gave some numerical examples arising from realistic applications, in which fast convergence is necessarily demanded. This preconditioner is very flexible in the sense that it is readily generalizable to higher dimensions, different discretization methods, different boundary conditions and problems with heterogeneities. It can also be implemented within sequential and parallel algorithms.

The first application of this method on scattering of an obstacle using finite element methods has been reported in [125]. The preconditioner is solved by Bi-CGSTAB preconditioned by ILU. Ideally, following the same outline discussed in this paper, the preconditioner is handled by multigrid. For general finite element methods, this will require algebraic multigrid methods.

Beside multigrid, domain decomposition methods and ILU factorizations can as well be used to perform the preconditioning steps. The use of ILU to approximate the shifted Laplacian preconditioners are reported in [44]. The convergence of Bi-CGSTAB combined by ILU applied to the shifted Laplacian is very similar to ILU applied to the Helmholtz matrix.

Considering as well spectral pictures shown in this paper, we notice that too many small eigenvalues close to the origin contribute to slow convergence, usually observed in the initial stage of the iterations. In Krylov subspace methods, such slow convergence can be overcome by incorporating deflation techniques [51, 93, 94]. Deflation, however, should be done on the product $M^{-1}A$, and *not* on A. If one can find a way to include deflation, it can be expected that the method will converge even faster.

Acknowledgement The author would like to thank the Dutch Ministry of Economic Affairs for the financial support they provided under the project BTS01044 during the research which led to the method presented in this paper.

References

1. Abarbanel S, Gottlieb D (1997) A mathematical analysis of the PML method. J Comput Phys 134:357–363
2. Abarbanel S, Gottlieb D (1998) On the construction and analysis of absorbing layers in CEM. Appl Numer Math 27:331–340
3. Alcouffe RE, Brandt A, Dendy JE Jr, Painter JW (1981) The multi-grid method for the diffusion equation with strongly discontinuous coefficients. SIAM J Sci Comput 2:430–454
4. Arnoldi WE (1951) The principle of minimized iterations in the solution of the matrix eigenvalue problem. Q Appl Math 9:17–29
5. Babuska I, Sauter S (1997) Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?. SIAM J Numer Anal 27:323–352
6. Babuska I, Ihlenburg F, Strouboulis T, Gangaraj SK (1997) Posteriori error estimation for finite element solutions of Helmholtz's equation. Part I: the quality of local indicators and estimators. Int J Numer Methods Eng 40:3443–3462
7. Babuska I, Ihlenburg F, Strouboulis T, Gangaraj SK (1997) Posteriori error estimation for finite element solutions of Helmholtz's equation. Part II: estimation of the pollution error. Int J Numer Methods Eng 40:3883–3900
8. Bamberger A, Joly P, Roberts JE (1990) Second-order absorbing boundary conditions for the wave equation: a solution for the corner problem. SIAM J Numer Anal 27:323–352
9. Bayliss A, Gunzburger M, Turkel E (1982) Boundary conditions for the numerical solution of elliptic equations in exterior regions. SIAM J Appl Math 42:430–451
10. Bayliss A, Goldstein CI, Turkel E (1983) An iterative method for Helmholtz equation. J Comput Phys 49:443–457
11. Bayliss A, Goldstein CI, Turkel E (1985) The numerical solution of the Helmholtz equation for wave propagation problems in underwater acoustics. Comput Math Appl 11:655–665
12. Bayliss A, Goldstein CI, Turkel E (1985) On accuracy conditions for the numerical computation of waves. J Comput Phys 59:396–404
13. Benamou JD, Despres B (1997) Domain decomposition method for the Helmholtz equation and related optimal control problems. J Comput Phys 136:62–88
14. Benzi M, Haws JC, Tuma M (2000) Preconditioning highly indefinite and nonsymmetric matrices. SIAM J Sci Comput 22:1333–1353
15. Berenger JP (1994) A perfectly matched layer for the absorption of electromagnetic waves. J Comput Phys 114:185–200
16. Berenger JP (1996) Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. J Comput Phys 127:363–379
17. Berkhout AJ (1982) Seismic migration: imaging of acoustic energy by wave field extrapolation. Elsevier, Amsterdam
18. Bollöffer M (2004) A robust and efficient ILU that incorporates the growth of the inverse triangular factors. SIAM J Sci Comput 25:86–103
19. Bourgeois A, Bourget M, Lailly P, Poulet M, Ricarte P, Versteeg R (1991) Marmousi, model and data. In: Marmousi experience, pp 5–16
20. Brackenridge K (1993) Multigrid and cyclic reduction applied to the Helmholtz equation. In: Melson ND, Manteuffel TA, McCormick SF (eds) Proc 6th Copper Mountain conf on multigrid methods, pp 31–41
21. Brandt A (1977) Multi–level adaptive solutions to boundary–value problems. Math Comput 31:333–390
22. Brandt A (2002) Multigrid techniques: 1984 guide with applications to fluid dynamics. Technical Report GMD-Studie 85, GMD Sankt Augustine, Germany
23. Brandt A, Livshits I (1997) Wave-ray multigrid methods for standing wave equations. Electr Trans Numer Anal 6:162–181
24. Brandt A, Ta'asan S (1986) Multigrid method for nearly singular and slightly indefinite problems. In: Proc EMG'85 Cologne, 1986, pp 99–121
25. Brezinzky C, Zaglia MR (1995) Look-ahead in bi-cgstab and other product methods for linear systems. BIT 35:169–201
26. Briggs WL (1988) A multigrid tutorial. SIAM, Philadelphia
27. Chow E, Saad Y (1997) ILUS: an incomplete LU factorization for matrices in sparse skyline format. Int J Numer Methods Fluids 25:739–749
28. Clayton R, Engquist B (1977) Absorbing boundary conditions for acoustic and elastic wave equations. Bull Seis Soc Am 67(6):1529–1540
29. Colloni F, Ghanemi S, Joly P (1998) Domain decomposition methods for harmonic wave propagation: a general presentation. Technical Report, INRIA RR-3473
30. Colton D, Kress R (1983) Integral equation methods in scattering theory. Willey, New York
31. Colton D, Kress R (1998) Inverse matrix and electromagnetic scattering theory. Springer, Berlin
32. D'Azevedo EF, Forsyth FA, Tang WP (1992) Towards a cost effective ILU preconditioner with high level fill. BIT 31:442–463
33. Dendy J Jr (1983) Blackbox multigrid for nonsymmetric problems. Appl Math Comput 13:261–283
34. Deraemaeker A, Babuska I, Bouillard P (1999) Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two, and three dimensions. Int J Numer Methods Eng 46:471–499
35. de Zeeuw PM (1990) Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. J Comput Appl Math 33:1–27
36. de Zeeuw PM (1996) Development of semi-coarsening techniques. Appl Numer Math 19:433–465
37. Drespes B (1990) Domain decomposition method and Helmholtz problems. In: Cohen G, Halpern L, Joly P (eds) Mathematical and numerical aspects of wave propagation phenomena. SIAM, Philadelphia, pp 42–51
38. Elman HC (1986) A stability analysis of incomplete LU factorizations. Math Comput 47:191–217
39. Elman HR, Ernst OG, O'Leary DP (2001) A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. SIAM J Sci Comput 22:1291–1315
40. Engquist B, Majda A (1977) Absorbing boundary conditions for the numerical simulation of waves. Math Comput 31:629–651
41. Erlangga YA, Vuik C, Oosterlee CW (2004) On a class of preconditioners for solving the Helmholtz equation. Appl Numer Math 50:409–425
42. Erlangga YA, Vuik C, Oosterlee CW (2005) On a robust iterative method for heterogeneous Helmholtz problems for geophysical applications. Int J Numer Anal Model 2:197–208
43. Erlangga YA, Oosterlee CW, Vuik C (2006) A novel multigrid-based preconditioner for the heterogeneous Helmholtz equation. SIAM J Sci Comput 27:1471–1492
44. Erlangga YA, Vuik C, Oosterlee CW (2006) Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. Appl Numer Math 56:648–666
45. Erlangga YA, Vuik C, Oosterlee CW (2006) A semicoarsening-based multigrid preconditioner for the 3D inhomogeneous Helmholtz equation. In: Wesseling P, Oosterlee CW, Hemker P

(eds) Proceedings of the 8th European multigrid conference, September 27–30, 2005, Scheveningen, TU Delft, The Netherlands
46. Fan K (1960) Note in M-matrices. Q J Math Oxford Ser 2 11:43–49
47. Farhat C, Macedo A, Lesoinne M (2000) A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. Numer Math 85:283–308
48. Fish J, Qu Y (2000) Global-basis two-level method for indefinite systems. Int J Numer Methods Eng 49:439–460
49. Fish J, Qu Y (2000) Global-basis two-level method for indefinite systems. Part I: convergence studies. Int J Numer Methods Eng 49:461–478
50. Fletcher R (1975) Conjugate gradient methods for indefinite systems. In: Watson GA (ed) Proc the 1974 Dundee biennial conf on numerical analysis, pp 73–89
51. Frank J, Vuik C (2001) On the construction of deflation-based preconditioners. SIAM J Sci Comput 23:442–462
52. Freund RW (1992) Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. SIAM J Sci Stat Comput 13(1):425–448
53. Freund RW (1997) Preconditioning of symmetric but highly indefinite linear systems. In: Sydow A (ed) 15th IMACS world congress on scientific computation modelling and applied mathematics, vol 2. Numerical mathematics, pp 551–556
54. Freund RW, Nachtigal NM (1991) QMR: A quasi minimum residual method for non-Hermitian linear systems. Numer Math 60:315–339
55. Gander MJ, Nataf F (2000) AILU: a preconditioner based on the analytical factorization of the elliptical operator. Numer Linear Algebra Appl 7:543–567
56. Gander MJ, Nataf F (2001) AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization. J Comput Acoust 9:1499–1509
57. Gander MJ, Nataf F (2005) An incomplete LU preconditioner for problems in acoustics. J Comput Acoust 13:455–476
58. George A, Liu JW (1981) Computer solution of large sparse positive definite systems. Prentice-Hall, Englewood Cliffs
59. Ghanemi S (1998) A domain decomposition method for Helmholtz scattering problems. In: Bjørstad, Espedal, Keyes, (eds) The ninth intl conf on domain decomposition methods, pp 105–112
60. Ghosh-Roy DN, Couchman LS (2002) Inverse problems and inverse scattering of plane waves. Academic, London
61. Goldstein CI (1986) Multigrid preconditioners applied to the iterative methods of singularly perturbed elliptic boundary value and scattering problems. In: Innovative numerical methods in engineering. Springer, Berlin, pp 97–102
62. Gozani J, Nachshon A, Turkel E (1984) Conjugate gradient coupled with multigrid for an indefinite problem. In: Advances in comput methods for PDEs V, pp 425–427
63. Greenbaum A (1997) Iterative methods for solving linear systems. SIAM, Philadelphia
64. Grote MJ, Huckel T (1997) Parallel preconditioning with sparse approximate inverses. SIAM J Sci Comput 18:838–853
65. Gutknecht MH, Ressel KJ (2000) Look-ahead procedures for Lanczos-type product methods based on three-term recurrences. SIAM J Matrix Anal Appl 21:1051–1078
66. Hackbusch W (1978) A fast iterative method for solving Helmholtz's equation in a general region. In: Schumman U (ed) Fast elliptic solvers. Advance Publications, London, pp 112–124
67. Hackbusch W (2003) Multi-grid methods and applications. Springer, Berlin
68. Hadley GR (2006) A complex Jacobi iterative method for the indefinite Helmholtz equation. J Comput Phys 203:358–370
69. Harari I (2006) A survey of finite element methods for time-harmonic acoustics. Comput Methods Appl Mech Eng 195:1594–1607
70. Harari I, Turkel E (1995) Accurate finite difference methods for time-harmonic wave propagation. J Comput Phys 119:252–270
71. Heikkola E, Rossi T, Toivanen J (2000) A parallel fictitious domain decomposition method for the three-dimensional Helmholtz equation. Technical Report No B 9/2000, Dept Math Info Tech, Univ Jÿvaskÿla
72. Hestenes MR, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. J Res Nat Bur Stand 49:409–435
73. Ihlenburg F, Babuska I (1995) Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. Int J Numer Methods Eng 38:3745–3774
74. Ihlenburg F, Babuska I (1995) Finite element solution of the Helmholtz equation with high wave number. Part I: the h-version of the FEM. Comput Math Appl 30(9):9–37
75. Ihlenburg F, Babuska I (1997) Finite element solution of the Helmholtz equation with high wave number. Part II: the hp-version of the FEM. SIAM J Numer Anal 34:315–358
76. Jo C-H, Shin C, Suh JH (1996) An optimal 9-point, finite difference, frequency space, 2-D scalar wave extrapolator. Geophysics 61(2):529–537
77. Kettler R (1982) Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In: Hackbusch W, Trottenberg U (eds) Multigrid methods. Lecture notes in mathematics, vol 960, pp 502–534
78. Kim S (1994) A parallezable iterative procedure for the Helmholtz equation. Appl Numer Math 14:435–449
79. Kim S (1995) Parallel multidomain iterative algorithms for the Helmholtz wave equation. Appl Numer Math 17:411–429
80. Kim S (1998) Domain decomposition iterative procedures for solving scalar waves in the frequency domain. Numer Math 79:231–259
81. Kononov AV, Riyanti CD, de Leeuw SW, Vuik C, Oosterlee CW (2006) Numerical performance of parallel solution of heterogeneous 2d Helmholtz equation. In: Wesseling P, Oosterlee CW, Hemker P (eds) Proceedings of the 8th European multigrid conference, TU Delft
82. Laird AL, Giles MB (2002) Preconditioned iterative solution of the 2D Helmholtz equation. Technical Report NA 02-12, Comp Lab, Oxford Univ
83. Lanczos C (1950) An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J Res Nat Bur Stand 45:255–282
84. Lanczos C (1952) Solution of systems of linear equations by minimized iterations. J Res Nat Bur Stand 49:33–53
85. Larsson E (1999) Domain decomposition method for the Helmholtz equation in a multilayer domain. SIAM J Sci Comput 20:1713–1731
86. Lee B, Manteuffel TA, McCormick SF, Ruge J (2000) First-order system least-squares for the Helmholtz equation. SIAM J Sci Comput 21:1927–1949
87. Lele SK (1992) Compact finite difference schemes with spectral-like resolution. J Comput Phys 103(1):16–42
88. Lynch RE, Rice JR (1980) A high-order difference method for differential equations. Math Comput 34(150):333–372
89. Made MMM (2001) Incomplete factorization-based preconditionings for solving the Helmholtz equation. Int J Numer Methods Eng 50:1077–1101
90. Manteuffel TA, Parter SV (1990) Preconditioning and boundary conditions. SIAM J Numer Anal 27(3):656–694
91. Meijerink JA, van der Vorst HA (1977) An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math Comput 31(137):148–162
92. Meijerink JA, van der Vorst HA (1981) Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems. J Comput Phys 44:134–155
93. Morgan RB (1995) A restarted GMRES method augmented with eigenvectors. SIAM J Matrix Anal Appl 16:1154–1171

94. Nicolaides RA (1987) Deflation of conjugate gradients with applications to boundary value problems. SIAM J Numer Anal 24:355–365
95. Oosterlee CW (1995) The convergence of parallel multiblock multigrid methods. Appl Numer Math 19:115–128
96. Oosterlee CW, Washio T (1998) An evaluation of parallel multigrid as a solver and as a preconditioner for singularly perturbed problems. SIAM J Sci Comput 19:87–110
97. Otto K, Larsson E (1999) Iterative solution of the Helmholtz equation by a second order method. SIAM J Matrix Anal Appl 21:209–229
98. Plessix RE, Mulder WA (2004) Separation-of-variables as a preconditioner for an iterative Helmholtz solver. Appl Numer Math 44:385–400
99. Pratt RG, Worthington MH (1990) Inverse theory applied to multi-source cross-hole tomography. Part 1: acoustic wave-equation method. Geophys Prosp 38:287–310
100. Quarteroni A, Valli A (1999) Domain decomposition methods for partial differential equations. Oxford Science Publications, Oxford
101. Riyanti CD, Kononov AV, Vuik C, Oosterlee CW (2006) Parallel performance of an iterative solver for heterogeneous Helmholtz problems. In: SIAM conference on parallel processing for scientific computing, San Fransisco, CA
102. Riyanti CD, Kononov A, Erlangga YA, Vuik C, Oosterlee CW, Plessix R-E, Mulder WA (2007) A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. J Comput Phys 224(1):431–448
103. Saad Y (1993) A flexible inner-outer preconditioned GMRES algorithm. SIAM J Sci Comput 14:461–469
104. Saad Y (1994) ILUT: a dual threshold incomplete LU factorization. Numer Linear Algebra Appl 1:387–402
105. Saad Y (2003) Iterative methods for sparse linear systems. SIAM, Philadelphia
106. Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7(12):856–869
107. Schenk O, Gärtner K (2004) Solving unsymmetric sparse systems of linear equations with PARDISO. J Future Gen Comput Syst 20:475–487
108. Schenk O, Gärtner K (2006) On fast factorization pivoting methods for symmetric indefinite systems. Electron Trans Numer Anal 23:158–179
109. Singer I, Turkel E (1998) High-order finite difference methods for the Helmholtz equation. Comput Methods Appl Mech Eng 163:343–358
110. Singer I, Turkel E (2006) Sixth order accurate finite difference scheme for the Helmholtz equations. J Comput Acoust 14(3):339–351
111. Smith B, Bjorstad P, Gropp W (1996) Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press, Cambridge
112. Sonneveld P (1989) CGS: a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J Sci Stat Comput 10:36–52
113. Strikwerda JC (1989) Finite difference schemes and partial differential equations. Wadsworth & Brooks/Cole, Pacific Groove
114. Stüben K, Trottenberg U (1982) Multigrid methods: fundamental algorithms, model problem analysis and applications. In: Hackbusch W, Trottenberg U (eds) Lecture notes in math, vol 960, pp 1–176
115. Susan-Resiga RF, Atassi HM (1998) A domain decomposition method for the exterior Helmholtz problem. J Comput Phys 147:388–401
116. Szyld DB, Vogel JA (2001) A flexible quasi-minimal residual method with inexact preconditioning. SIAM J Sci Comput 23:363–380
117. Tam CKW, Webb JC (1993) Dispersion-relation-preserving finite difference schemes for computational acoustics. J Comput Phys 107(2):262–281
118. Tarantola A (1984) Inversion of seismic reflection data in the acoustic approximation. Geophysics 49:1259–1266
119. Tezaur R, Macedo A, Farhat C (2001) Iterative solution of large-scale acoustic scattering problems with multiple right hand-sides by a domain decomposition method with Lagrange multipliers. Int J Numer Methods Eng 51:1175–1193
120. Thole CA, Trottenberg U (1986) Basic smoothing procedures for the multigrid treatment of elliptic 3-d operators. Appl Math Comput 19:333–345
121. Tosseli A, Widlund O (2005) Domain decomposition methods. Springer, Berlin
122. Trottenberg U, Oosterlee C, Schüller A (2001) Multigrid. Academic, New York
123. Tsynkov S, Turkel E (2001) A Cartesian perfectly matched layer for the Helmholtz equation. In: Tourette L, Harpern L (eds) Absrobing boundaries and layers, domain decomposition methods applications to large scale computation. Springer, Berlin, pp 279–309
124. Turkel E (2001) Numerical difficulties solving time harmonic equations. In: Multiscale computational methods in chemistry and physics. IOS, Ohmsha, pp 319–337
125. Turkel E, Erlangga YA (2006) Preconditioning a finite element solver of the Helmholtz equation. In: Wesseling P, Oñate EO, Périaux J (eds), Proceedings ECCOMAS CFD 2006, TU Delft
126. van der Vorst HA (1992) Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput 13(2):631–644
127. van der Vorst HA (2003) Iterative Krylov methods for large linear systems. Cambridge University Press, New York
128. van der Vorst HA, Melissen JBM (1990) A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex systems. IEEE Trans Magn 26(2):706–708
129. van der Vorst HA, Vuik C (1993) The superlinear convergence behaviour of GMRES. J Comput Appl Math 48:327–341
130. van der Vorst HA, Vuik C (1994) GMRESR: a family for nested GMRES methods. Numer Linear Algebra Appl 1(4):369–386
131. van Gijzen M, Erlangga YA, Vuik C (2007) Spectral analysis of the shifted Laplace precondtioner. SIAM J Sci Comput 29(5):1942–1958
132. Vandersteegen P, Bienstman P, Baets R (2006) Extensions of the complex Jacobi iteration to simulate photonic wavelength scale components. In: Wesseling P, Oñate E, Périaux J (eds) Proceedings ECCOMAS CFD 2006, TU Delft
133. Vandersteegen P, Maes B, Bienstman P, Baets R (2006) Using the complex Jacobi method to simulate Kerr non-linear photonic components. Opt Quantum Electron 38:35–44
134. Vanek P, Mandel J, Brezina M (1996) Algebraic multigrid based on smoothed aggregation for second and fourth order problems. Computing 56:179–196
135. Vanek PV, Mandel J, Brezina M (1998) Two-level algebraic multigrid for the Helmholtz problem. Contemp Math 218:349–356
136. Vuik C, Erlangga YA, Oosterlee CW (2003) Shifted Laplace preconditioner for the Helmholtz equations. Technical Report 03-18, Dept Appl Math Anal, Delft Univ Tech, The Netherlands
137. Waisman H, Fish J, Tuminaro RS, Shadid J (2004) The generalized global basis (GGB) methods. Int J Numer Methods Eng 61:1243–1269
138. Washio T, Oosterlee CW (1998) Flexible multiple semicoarsening for three dimensional singularly perturbed problems. SIAM J Sci Comput 19:1646–1666
139. Wesseling P (1992) An introduction to multigrid methods. Willey, London

140. Wienands R, Joppich W (2004) Practical Fourier analysis for multigrid methods. Chapman & Hall/CRC, London
141. Wienands R, Oosterlee CW (2001) On three-grid Fourier analysis of multigrid. SIAM J Sci Comput 23:651–671
142. Zhou L, Walker HF (1994) Residual smoothing techniques for iterative methods. SIAM J Sci Comput 15(2):297–312