A MULTIGRID-BASED SHIFTED-LAPLACIAN PRECONDITIONER FOR A FOURTH-ORDER HELMHOLTZ DISCRETIZATION.

N.UMETANI[*], S.P.MACLACHLAN[†], C.W. OOSTERLEE[‡§]

Abstract. In this paper, an iterative solution method for a fourth-order accurate discretization of the Helmholtz equation is presented. The method is a generalization of that presented in [10], where multigrid was employed as a preconditioner for a Krylov subspace iterative method. This multigrid preconditioner is based on the solution of a second Helmholtz operator with a complex-valued shift. In particular, we compare preconditioners based on a point-wise Jacobi smoother with those using an ILU(0) smoother, we compare using the prolongation operator developed by de Zeeuw in [37] with interpolation operators based on algebraic multigrid principles, and we compare the performance of the Krylov subspace method Bi-CGSTAB with the recently introduced induced dimension reduction method, IDR(s). These three improvements are combined to yield an efficient solver for heterogeneous high-wavenumber problems.

Key words. Helmholtz equation, non-constant high wavenumber, complex-valued multigrid preconditioner, algebraic multigrid, ILU smoother

AMS subject classifications. 65N55, 65F10, 78A45, 35J05

1. Introduction. Many authors, e.g. [5, 7, 13, 19], have contributed to the development of appropriate multigrid methods for the Helmholtz equation, but an efficient multigrid treatment of heterogeneous problems with high wavenumers arising in engineering settings has not yet been proposed in the literature. The multigrid method [4, 14] is known to be a highly efficient iterative method, for example, for discrete Poisson-type equations, even with fourth-order accurate discretizations [6, 33]. The Helmholtz equation, however, does not belong to the class of PDEs for which off-the-shelf multigrid methods perform efficiently. Convergence degradation and, consequently, loss of $O(N)$ complexity are caused by difficulties encountered in the smoothing and coarse-grid correction components; see [7, 33] for a discussion.

We present an efficient numerical solution technique for the heterogeneous high-wavenumber Helmholtz equation, discretized by fourth-order finite differences. Recently, in [10], a robust preconditioned Bi-CGSTAB method has been proposed for solving these problems, in which the preconditioner is based on a second Helmholtz equation with an imaginary shift. This preconditioner is a member of the family of shifted Laplacian operators, introduced in [20], and its inverse can be efficiently approximated by means of a multigrid iteration. Two-dimensional results, representative for geophysical applications, generated by second-order finite differences, have been presented in [26] and 3D results in [27].

In this paper, we generalize this solver and include, in particular, a fourth-order discretization of the Helmholtz operator in our discussion. The multigrid preconditioner is enhanced, in the sense that we replace the point-wise Jacobi smoother in the multigrid preconditioner by a variant of the incomplete lower-upper factorization smoother, ILU(0). Furthermore, we evaluate the performance of a prolongation

[*]Delft University of Technology, Delft, the Netherlands. Exchange student from Tokyo University

[†]Delft University of Technology, the Netherlands. Now at Tufts University, Medford, MA, US, The work of this author was supported by the European Community's Sixth Framework Programme, through a Marie Curie International Incoming Fellowship, MIF1-CT-2006-021927.

[‡]CWI, Center for Mathematics and Computer Science, Amsterdam, and Delft University of Technology, Delft, the Netherlands.

[§]Corresponding author: c.w.oosterlee@cwi.nl

scheme that originates from algebraic multigrid (AMG) [28]. Next to this, we also compare a recently introduced Krylov subspace method, called Induced Dimension Reduction [32] (IDR), with the commonly used Bi-CGSTAB [34] method. It is expected that these enhancements to the iterative solver can reduce both the number of iterations and the total CPU time needed for convergence. Moreover, we aim to reduce the size of the imaginary shift parameter in the shifted Laplacian preconditioner, so that an even faster solution method is obtained. Furthermore, the overall solution method with these algorithmic improvements is not strictly restricted to structured Cartesian grids, as it can be set up fully algebraically (a similar goal has been pursued in [1]). Although our method extends to solving problems on unstructured grids, we focus here on heterogeneous Helmholtz problems on Cartesian grids. We focus on the two-dimensional case; however, all of the method's ingredients can be directly generalized to three dimensions.

This paper is set up as follows. In Section 2, we briefly discuss the 2D Helmholtz equation, its field of application, and the discrete finite-difference formulations of second and fourth order. The iterative solution method, including the preconditioner and its components are presented in Section 3. Numerical results are presented in Section 4, where we evaluate the iterative solver.

2. The Helmholtz Equation and Its Applications. Accurately imaging the Earth is one of the major challenges in the hydrocarbon industry. Subsurface formations are mapped by measuring the time required for a seismic pulse to return to the surface after reflection from interfaces between formations with different physical properties. Variations in these reflection times, as recorded on the Earth's surface, usually indicate structural features in the strata below. Depths to reflecting interfaces can be determined from the times, using velocity information that can be obtained from the reflected signals themselves.

In geophysics, numerical methods can be employed in acoustic imaging techniques to gain insight into the geological structures deep within the Earth's subsurface [23,24]. Traditional mathematical imaging techniques based on the wave equation have been successfully developed based on a high-frequency approximation. When the medium is very complex (containing heterogeneities that result in strong lateral velocity variations), however, these so-called ray-based high-frequency migration techniques reach their limits. The industry is gradually moving to finite-difference-wave-equation migration [23], as the linearized acoustic wave equation can be used to accurately image steep dips. The numerical solution of the wave equation has greatly helped both forward modeling and migration of seismic wavefields in complex Earth structures, and it also serves as a starting point for solving the full inverse problem [25].

The wave equation can be solved in either the time or the frequency domain. When applying the Fourier transformation with respect to time to the acoustic wave equation, the Helmholtz equation is obtained. One advantage of this transformation to the frequency domain is that the problem becomes easier to solve, because the time variable, t, decouples. Instead of a single (coupled) equation in t, we get a series of decoupled equations for each wavenumber, k, that can be solved in parallel. Once a pressure solution in the frequency domain is obtained, the solution of the wave equation can be computed efficiently by using the forward Fourier transformation. To represent a solution of the wave equation for a given time interval accurately, however, a wide range of frequencies has to be resolved and the Helmholtz equation has to be solved independently for each frequency.

From the exploration-seismology point of view, the Earth is a heterogeneous semi-

infinite medium. The wavenumber can be large, which implies that the discretized Helmholtz operator gives rise to both positive and negative eigenvalues and, therefore, the discretization matrix, A_h, is indefinite. For 2D problems, however, the computation can be performed efficiently by using, for example, direct methods combined with nested-dissection reordering [12]. Only one LU decomposition is needed to calculate the solutions at multiple source locations. The result can be used for the computation of all of the wavefields, for all shots and, also, for the back-propagated receiver wavefields [24]. However, for 3D problems, the matrix sizes and bandwidths rapidly become too large and one has to fall back on iterative methods. In that case, one no longer has the advantages in the frequency domain related to the LU decomposition.

For the Helmholtz equation, unfortunately, many iterative methods suffer from slow convergence, especially if high frequencies need to be resolved, due to the indefiniteness. The development of fast iterative methods for high-frequency Helmholtz problems remains a subject of active research. One approach to iteratively solving this equation is presented below.

2.1. Mathematical Problem Definition. We start with the description of the 2D Helmholtz problem which we would like to solve,

$$-\nabla^2 u(\boldsymbol{x},\omega) - k(\boldsymbol{x})^2(1-\alpha i)u(\boldsymbol{x},\omega) = g(\boldsymbol{x},\omega),\ \boldsymbol{x}\in\Omega. \tag{2.1}$$

Unknown $u(\boldsymbol{x},\omega)$ represents the pressure field in the frequency domain, ∇^2 is the Laplacian operator, $k(\boldsymbol{x}) = \omega/c(\boldsymbol{x})$ is the wavenumber, with $c(\boldsymbol{x})$, the acoustic-wave velocity, which varies with position, and $\omega = 2\pi f$ denotes angular frequency, a scalar measure of rotation rate (f is the frequency in Hertz). Wavenumber k depends on $\boldsymbol{x}$ because of a spatially dependent speed of sound, $c(\boldsymbol{x})$. The source term is denoted by g. The medium is called barely attenuative if $0 \le \alpha \ll 1$, with α indicating the fraction of damping in the medium (and $i = \sqrt{-1}$, the imaginary unit). In geophysical applications, which are of our main interest, this damping can be up to 5% ($\alpha = 0.05$). While Equation (2.1) arises through the Fourier transform of a wave equation with a very simple model of damping, $(-\nabla^2 + (1-\alpha i)\partial_t^2)u = g$, it is closely related to the Fourier transform of the strongly damped wave equation, $(-\nabla^2 + \tau\partial_t\nabla^2 + \partial_t^2)u = g$ that yields, after scaling, $(-\nabla^2 - \frac{k^2}{1+\tau i k})u$. For small values of $\alpha = \tau k$, Equation (2.1) is an accurate approximation of the Fourier-domain strongly damped wave equation.

The semi-infinite physical domain needs to be truncated for a numerical treatment. A popular approach in geophysics in order to obtain a satisfactory near-boundary solution, without many artificial reflections, is to use the absorbing boundary layer (ABL) approach; see, for example, [16] or [21]. This unphysical boundary layer is used to gradually damp out the outgoing waves by adding dissipation in the equation outside the domain of interest. An efficient numerical technique should be robust with respect to this kind of feature. The absorption layers (denoted by Ω^e) are attached to the physical domain, Ω, (see Figure 2.1). In Ω^e, a damped Helmholtz equation (2.1) should be satisfied [30], with

$$\alpha = 0.25\frac{\parallel \boldsymbol{x} - \boldsymbol{x}_d \parallel^2}{\parallel \boldsymbol{x}_e - \boldsymbol{x}_d \parallel^2},\ \ \boldsymbol{x}\in\Omega^e, \tag{2.2}$$

where point $\boldsymbol{x}_d$ is a point at the boundary, Γ, and $\boldsymbol{x}_e$ a point at Γ^e (see Figure 2.1). At Γ_e, this equation is supplemented by second-order absorbing boundary conditions, as described in [2, 8].

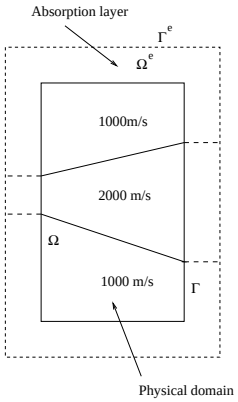


FIG. 2.1. *A 2D domain with ABL in the case of a regular heterogeneous wedge medium.*

2.2. Discretization. The equations are discretized either by a second- or a fourth-order finite-difference scheme, resulting in the linear system:

$$A_h \phi_h = b_h \,, \tag{2.3}$$

where ϕ_h and b_h represent the discrete frequency-domain pressure field and the source, respectively.

In a heterogeneous medium, the smallest velocity is usually selected based on the representative wavelength, λ_f. The number of wavelengths in a domain of size L equals L/λ_f. A dimensionless wavenumber, k, on a non-dimensional $[0,1]^2$ domain is defined by $k = 2\pi f L/c$, and a corresponding mesh size by $h = \lambda_f/(n_w L)$, with n_w the number of points per wavelength. With domain size $L = 1$, an accuracy requirement, for second-order discretizations, is that $kh \leq \pi/5 (\approx 0.63)$ for $n_w = 10$ points per wavelength, and $kh \leq 0.53$ with $n_w = 12$ points per wavelength. The number of grid points used assumes a linear connection between k and h. In order to avoid a reduction of accuracy due to *pollution of the solution*, however, k^2h^3 should be chosen constant, as stated in [3, 15]. For an iterative solution method, the requirement that kh should be constant is more severe and, so, this is the constraint that we consider here.

For second-order finite differences, the 5-point discretization stencil used to set up matrix A_h is well-known. For the absorbing boundary conditions at Γ_e, we also apply central differences. The discretization that is particularly of interest in this work is the $O(h^4)$ accurate discretization based on the Padé approximation. It is called the HO discretization in [29], with stencil,

$$A_h^{HO} \triangleq \frac{1}{h^2} \begin{bmatrix} -\frac{1}{6} & -\frac{2}{3} - \frac{(kh)^2(1-\alpha i)}{12} & -\frac{1}{6} \\ -\frac{2}{3} - \frac{(kh)^2(1-\alpha i)}{12} & \frac{10}{3} - \frac{2(kh)^2(1-\alpha i)}{3} & -\frac{2}{3} - \frac{(kh)^2(1-\alpha i)}{12} \\ -\frac{1}{6} & -\frac{2}{3} - \frac{(kh)^2(1-\alpha i)}{12} & -\frac{1}{6} \end{bmatrix}. \tag{2.4}$$

An important reason for choosing a higher-order discretization method is that the number of grid points per wavelength can be reduced compared to a second-order discretization. This results in smaller matrices for the same level of accuracy and,

thus, may lead to an algorithm that is more efficient overall, if the matrices associated with the higher-order discretization can be solved efficiently.

These matrices remain positive definite as long as k^2 is smaller than the first eigenvalue of the discrete Laplacian. The wavenumber in geophysical applications can, however, be large, which implies that the discretized Helmholtz equation gives rise to both positive and negative eigenvalues and, therefore, the discretization matrix, A_h, is indefinite. The size of the system of linear equations (2.3) gets very large for high frequencies. So, A_h in (2.3) is a large but sparse matrix, with complex-valued entries, because of the absorbing boundary conditions and the attenuative medium. It is symmetric but non-Hermitian.

2.2.1. Validation of the discretization. In order to validate the choice of boundary condition, ABL, and discretization, we compute the solution for a constant wavenumber problem in a homogeneous medium with the source function, representative for a seismic pulse, chosen as

$$g_h = \frac{1}{h^2}\delta(x_1 - \frac{1}{2}, x_2 - \frac{1}{32}).$$

Here, $\delta(\cdot,\cdot)$ represents the Dirac delta function, which is 1 when its argument is $(0,0)$, and 0 elsewhere. The scaling by h^2 guarantees that the solutions on fine and coarse grids are of the same amplitude, giving a discrete approximation of a δ-function distribution.

Two formulations of the boundary discretization are compared here. In the first, we prescribe the second-order absorbing boundary conditions directly at the physical boundaries, whereas, in the second formulation, the boundary discretization is based on an extra absorbing boundary layer, placed along all physical domain boundaries. An ABL of $n/4$ points is added to each side.

In this first numerical experiment, we fix the wavenumber, $k = 40$, and use a model domain, $(0,1)^2$, covered by a fine grid consisting of 256^2 points ($kh = 0.156$). Figure 2.2 presents the two corresponding solutions with the second-order discretization. An unphysical damping of the solution without the ABL can be observed near the domain boundaries.

Figure 2.3 presents solution profiles along the line $x = 0.125$, for the second- and fourth-order discretizations, with and without the ABL, on three meshes with

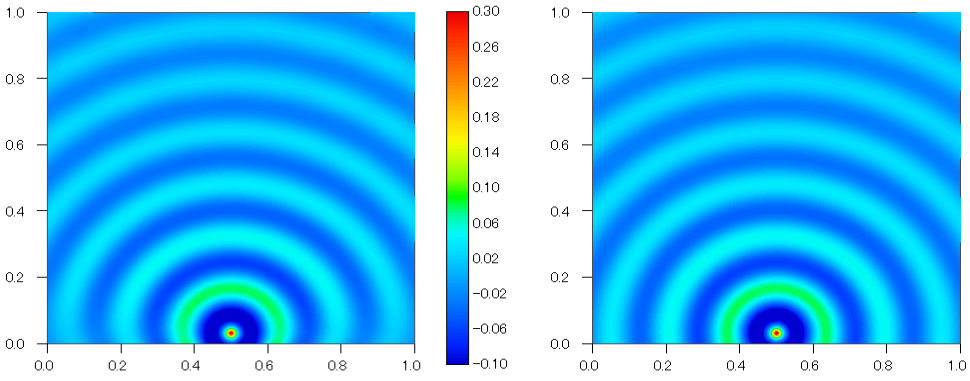
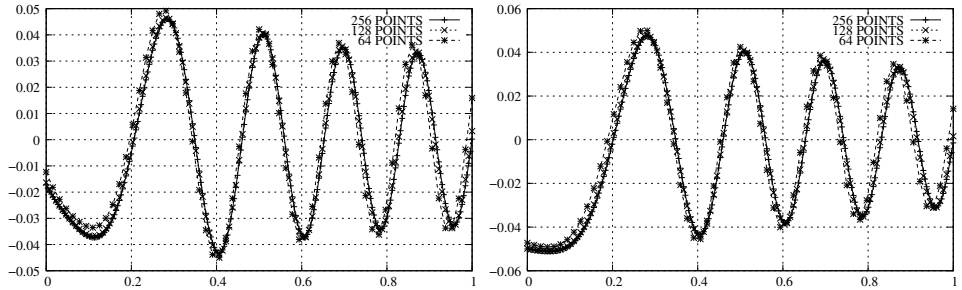
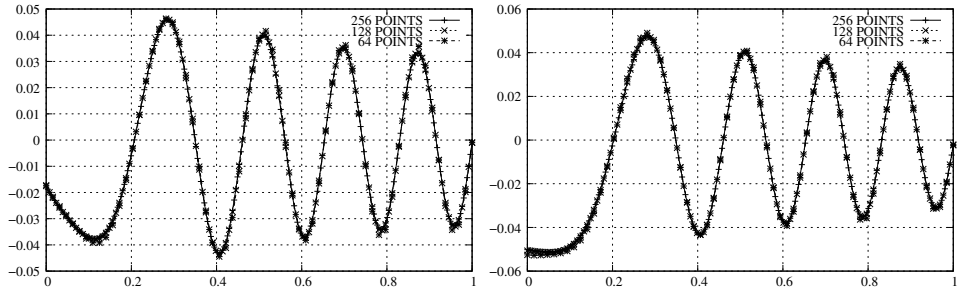


FIG. 2.2. *Numerical solutions for $k = 40$ and $h = 1/256$, without (left-side) and with (right-side) ABL.*



(a) No ABL, 2nd order

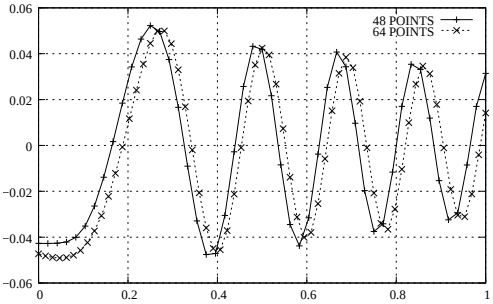
(b) With ABL, 2nd order



(c) No ABL, 4th order

(d) With ABL, 4th order

FIG. 2.3. *Comparison of the vertical line solutions, for three consecutive grid sizes, at* $x = 0.125$, $k = 40$*, with second- and fourth-order discretizations; Left side: No ABL, right side: with ABL.*



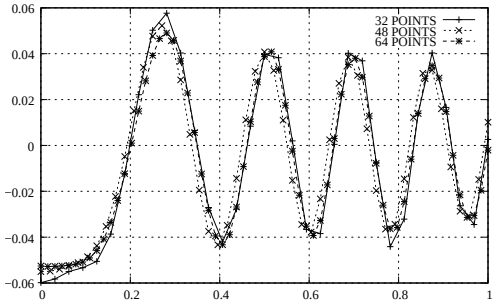


FIG. 2.4. *Comparison of the vertical line solutions on coarse grids, at* $x = 0.125$, $k = 40$*, left side: second-order discretization, right side: fourth-order discretization.*

$64^2, 128^2$ and 256^2 points. Grid convergence for the three mesh sizes is achieved in both cases, but the solution profiles, with and without ABL, differ significantly, especially near the domain boundaries. We also compare, for the same problem, the solution profiles on *coarse* grids, with the second- and fourth-order discretizations. The ABL is now attached to the domain. In Figure 2.4a, the profiles for the second-order discretization with 64^2 and 48^2 interior points are presented. A non-physical phase shift is observed on the grid that is too coarse ($kh > 0.8$) with this second-order discretization. In comparison, Figure 2.4b presents the solution profiles with the HO discretization on $32^2, 48^2$ and 64^2 grids, which converge nicely towards the physical solution. Thus, in this case, the fourth-order discretization with the ABL leads to an accurate numerical solution, already on relatively coarse grids.

3. Iterative Solution Method. Iterative solution methods for complex-valued indefinite systems based on Krylov subspace methods [31] are typically generalizations of the conjugate-gradient (CG) method. The Bi-conjugate gradient stabilized (Bi-CGSTAB) algorithm [34] is one of the better known Krylov subspace algorithms for non-Hermitian problems, which has been used for Helmholtz problems, for example, in [1, 10]. One of the advantages of Bi-CGSTAB, compared to full GMRES, is its limited memory requirements.

Without a preconditioner, however, the Krylov subspace methods converge very slowly, or not at all, for the problems of interest [9]. By preconditioning with a matrix, M_h^{-1}, we solve an equivalent linear system,

$$A_h M_h^{-1} \tilde{\phi}_h = b_h, \qquad \tilde{\phi}_h = M_h \phi_h. \tag{3.1}$$

The challenge, then, is to find a matrix, M_h, such that $A_h M_h^{-1}$ has a spectrum that is favorable for iterative solution with Krylov subspace methods, and whose inverse, M_h^{-1}, can be efficiently approximated.

In [10], a shifted-Laplacian operator was proposed as a preconditioner for the Helmholtz equation, with M_h defined as a discretization of

$$\mathcal{M} = -\nabla^2 - k^2(\boldsymbol{x})(\beta_1 - \beta_2 i). \tag{3.2}$$

Boundary conditions were set identically to those for the original Helmholtz equation. The influence of parameters β_1 and β_2 was evaluated in [10], and the optimal values for the solver proposed there were $(\beta_1, \beta_2) = (1, 0.5)$. Here, we will also consider $\beta_2 = 0.4$ (smaller values of β_2 do not lead to a converging algorithm with the components to be introduced below). The matrix after discretization of (3.2), M_h, is obtained from either the 5−point, $O(h^2)$, or the 9-point, $O(h^4)$, finite-difference discretization.

3.1. Fourier Analysis. The discrete Helmholtz matrix, A_h, as well as the preconditioner, M_h, allow us, assuming a constant wavenumber and Dirichlet boundary conditions, to apply Fourier analysis on the basis of discrete sine-eigenfunctions,

$$v_h^{p,q} = \sin(p\pi x)\sin(q\pi y), \tag{3.3}$$

to gain insight into the spectrum of $A_h M_h^{-1}$. With these discrete sine functions, $A_h M_h^{-1}$ is diagonalized, and the eigenvalues can easily be determined. As long as k^2 is not equal to any of the eigenvalues of the discrete Laplace operator, $A_h M_h^{-1}$ is nonsingular. Otherwise, the matrix is singular and its nullspace is spanned by the corresponding eigenfunctions (3.3).

We perform Fourier analysis here to visualize the effect of the choice of the parameter, β_2, as well as the choice of discretization on the clustering of the eigenvalues of the preconditioned system. This analysis gives a first indication of what we can expect from the solver. For both A_h and M_h, we use either the second-order discretizations or the fourth-order, HO stencils. We do not include damping in A_h in the analysis (we take $\alpha = 0$ in (2.1)), and concentrate here on the case $k = 100$ ($k^2 = 10^4$), $h = 1/160$. Preconditioners with complex Helmholtz terms give rise to a curved spectrum, see also [11]. Figure 3.1 presents the curved spectrum of $A_h M_h^{-1}$ for $(\beta_1, \beta_2) = (1, 0.4)$ in M_h, where both operators, A_h and M_h, are discretized by the fourth-order stencil. A very similar eigenvalue distribution is obtained with the second-order discretization. For $\beta_2 = 0.5$, we find essentially the same spectrum; however, more eigenvalues are in the vicinity of the origin.

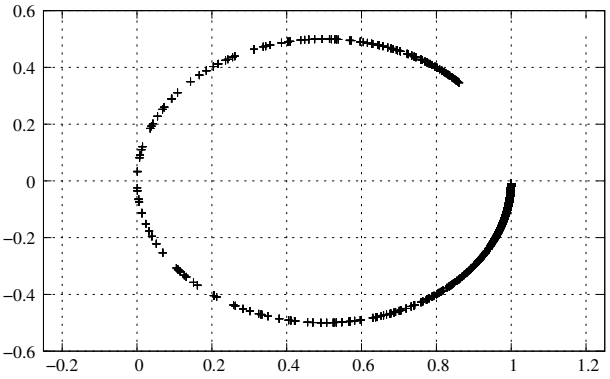
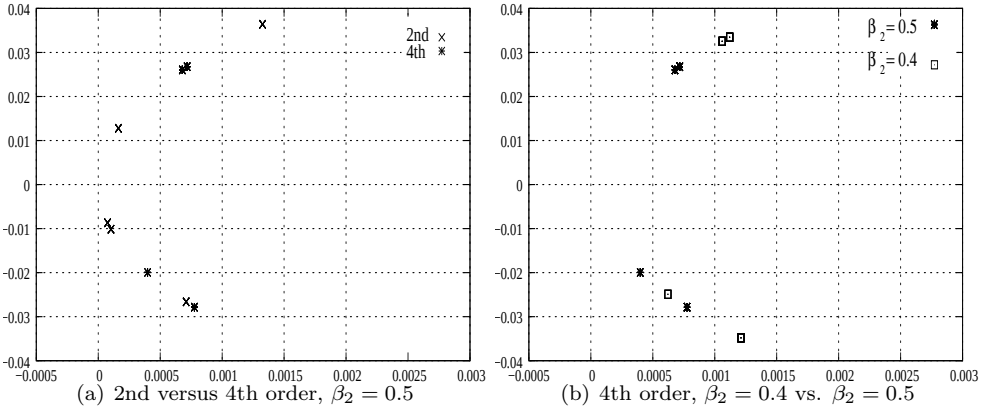


FIG. 3.1. *Spectral picture of $M_h^{-1}A_h$ with $\alpha = 0$, $(\beta_1, \beta_2) = (1, 0.4)$, $k = 100, h = 1/160$. Both operators are discretized by the fourth-order, HO discretization.*



(a) 2nd versus 4th order, $\beta_2 = 0.5$ (b) 4th order, $\beta_2 = 0.4$ vs. $\beta_2 = 0.5$

FIG. 3.2. *Comparison of zoomed spectral pictures of $M_h^{-1}A_h$ with $\alpha = 0$, $k = 100, h = 1/160$. (a): Comparison of eigenvalues near the origin discretized with second-order and with fourth-order discretizations; (b) Comparison of eigenvalues for $\beta_2 = 0.4$ and $\beta_2 = 0.5$, fourth-order discretizations.*

These near-zero eigenvalues are problematic for the convergence of the Krylov subspace method. Figure 3.2a shows a zoom of the spectrum near the origin, comparing (for $\beta_2 = 0.4$) the location of the eigenvalues near the origin with the second- and the fourth-order discretizations. In Figure 3.2b, we also compare the location of the eigenvalues near the origin for $\beta_2 = 0.4$ and $\beta_2 = 0.5$, with the fourth-order discretization, keeping $k = 100$, $h = 1/160$. With the fourth-order discretization, the eigenvalues stay further from the origin as compared to the second-order discretization. This has a positive effect on the convergence of the Krylov subspace method. The same is true when comparing the eigenvalues with $\beta = 0.4$ and $\beta_2 = 0.5$ where, as expected, the clustering with $\beta_2 = 0.4$ is more favorable for iterative solution.

When discretized with second-order finite differences, M_h, with $(\beta_1, \beta_2) = (1, 0.5)$, can be relatively easily handled by a multigrid method, which is confirmed by Local Fourier Analysis, a quantitative multigrid analysis tool [33]. This is due to the imaginary term, $\beta_2 i$ in the shifted Laplacian. LFA also indicates that M_h based on the fourth-order discretization can be dealt with in multigrid as efficiently as the second-order discretization. With the robust multigrid components presented below, includ-

ing the ILU(0) smoother used within the preconditioner, we aim to decrease the value of β_2, to $\beta_2 = 0.4$, and obtain an efficient preconditioned Krylov subspace method.

3.2. Multigrid Preconditioner. One multigrid cycle, based on standard grid coarsening and point-wise smoothing, can be used as an approximation to M_h^{-1}. In [10], an F(1,1)-cycle [33], with one pre- and one post-smoothing iteration, with a Jacobi smoother with under-relaxation parameter $\omega = 0.5$ was chosen for the high-wavenumber problems. The other multigrid components were:

(i) Restriction operators, I_h^H, based on 2D full weighting, whose stencil [33] reads:

$$I_h^H \triangleq \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H , \tag{3.4}$$

with h denoting the fine-, and H denoting the coarse-mesh size.

(ii) Prolongation operators, I_H^h, were 2D matrix-dependent interpolation, based on de Zeeuw's interpolation weights [37].

(iii) Coarse-grid matrices were based on Galerkin coarse-grid discretizations, defined as $M_H = I_h^H M_h I_H^h$.

In [10], it was shown that the full-weighting restriction combined with the matrix-dependent prolongation resulted in robust convergence for a variety of problems with irregular heterogeneities and strong contrasts. The inclusion of an ABL in the discretization does not lead to any multigrid convergence difficulties, as the components chosen are especially designed for problems with varying coefficients.

With a more powerful smoother, we expect that a robust multigrid method can be developed for approximately inverting matrices M_h that originate from a fourth-order discretization. As the smoother in the multigrid preconditioner, we replace the point-wise Jacobi smoother by an ILU smoother, well-known in the multigrid literature [17, 18, 35, 36, 38]. We choose here the ILU(0) variant, meaning that we do not allow any additional fill-in in the lower- and upper-triangular factors outside of the nonzero pattern of matrix M_h. An ILU(0) smoother is known to be more powerful than a point-wise Jacobi smoother for a number of test problems [35]. Strictly speaking, ILU methods do not only have a smoothing effect on the errors. A lexicographical version may also reduce low-frequency errors, especially when the entries of the remainder matrix, R_h, in

$$M_h = \hat{L}_h \hat{U}_h - R_h,$$

are relatively small.

3.3. AMG Type Interpolation. An efficient multigrid scheme relies on the effective complementarity of the chosen relaxation and interpolation procedures in reducing the error components in an approximate solution. The coarse-grid correction operator is designed to reduce errors that the chosen smoother is slow to attenuate. Such errors should lie in the range of interpolation, so that the coarse-grid correction may be effective. Here, we consider a fixed choice of coarse grid, using the same Cartesian coarse grids (doubling the mesh size in each direction) as in geometric multigrid and the multigrid method examined in [10], but evaluate an interpolation operator that is chosen based on algebraic multigrid (AMG) principles. The interpolation developed is largely based on the real-valued AMG interpolation from [28], and discussed for complex-valued equations in [22].

Consider, then, an error, e_h, that is not quickly reduced by relaxation. For many standard problems and smoothers, these errors coincide with those vectors that yield small residuals. For the purpose of interpolation, AMG assumes that the error, e_h, is much larger than its residual when measured point-wise, $(A_h e_h)_j \ll (e_h)_j$, for each fine-grid index j. Based on this property, we have

$$(A_h e_h)_j \approx 0 \Rightarrow a_{jj}(e_h)_j \approx -\sum_{k \neq j} a_{jk}(e_h)_k, \tag{3.5}$$

meaning that the value of the error at a fine-grid node, j, can be accurately approximated by the values from its neighboring nodes. If all neighboring nodes are also coarse-grid nodes, then (3.5) is easily turned into an interpolation formula.

With either the fixed coarsening considered here, or with any other sufficiently rapid coarsening procedure, fine-grid node j will have both fine-grid and coarse-grid nodes as neighbors. Designing an interpolation procedure can, then, be thought of as modifying the balance in (3.5) in such a way as to remove connections to other fine-grid neighbors of j while preserving the overall balance. This is typically done by applying a partition to the neighboring nodes of j that identifies some nodes as important, or strong, connections and other nodes as unimportant, or weak connections. That is, we write the set, $\{k \neq j\} = C_j \cup F_j^s \cup F_j^w$, where C_j is the set of strongly connected coarse-grid neighbors of j, and the disjoint sets, F_j^s and F_j^w, denote the strong fine-grid and weak connections, respectively.

The matrix arising from the Helmholtz equation is complex and, typically, the sum of the moduli of the off-diagonal elements is larger than that of the diagonal element in each row. In this case, a different criterion should be considered as a measure of the strong connections. Here, we give two common criteria for defining the set, S_j, of strong connections for node j, defining

$$S_j = \left\{ k : |a_{jk}| \geq \theta \max_{l \neq j} |a_{jl}| \right\},$$

or

$$S_j = \left\{ k : Re(a_{jk}) \geq \theta \max_{l \neq j} Re(a_{jl}) \right\}.$$

Parameter θ allows some adjustment of the number of connections chosen as strong (relative to the strongest connection); for many problems, $\theta = 0.25$ is considered to be a standard choice. Numerical experiments with the discrete complex-valued shifted Laplacian have revealed that sometimes divergence is observed for high wavenumber problems if we use the measure based on the norm. The measure based on the real part of the matrix elements gave a robust multigrid performance over a large range of wavenumbers and, thus, is used in the numerical results that follow.

It is expected that the weak connections of fine-grid node j can be discarded from the balance in (3.5). To remove these terms (in particular, the appearance of $(e_h)_k$ for $k \notin S_j$) without upsetting the balance, these terms are "lumped to the diagonal". In effect, this means that we make the approximation that $(e_h)_k \approx (e_h)_j$ for $k \in F_j^w$; while this approximation may not be very accurate, it is not harmful to make such a choice, since the connections involved are not important. Treating the strongly connected fine-grid neighbors of j is much more important, as these are connections that (by definition) cannot be easily dropped. In classical AMG methods, one assumes that these connections are well-represented on the coarse grid, by their

values at neighboring points. Then, an approximation may be made by considering the weighted average of the values at common coarse-grid neighbors of node j and its fine-grid neighbor, node k, resulting in the expression,

$$(e_h)_k \approx \frac{\sum_{l \in C_j} a_{kl}(e_h)_l}{\sum_{l \in C_j} a_{kl}}.$$

If there is no point in C_j such that $a_{kl} \neq 0$ (or if $\sum_{l \in C_j} a_{kl} = 0$), then node k is neglected in the interpolation formula. Making these substitutions in (3.5) and choosing for equality, we then have

$$a_{jj}(e_h)_j = -\sum_{k \in C_j} a_{jk}(e_h)_k - \sum_{k \in F_i^s} a_{jk} \frac{\sum_{l \in C_j} a_{kl}(e_h)_l}{\sum_{l \in C_j} a_{kl}} - \sum_{k \in F_i^w} a_{jk}(e_h)_j,$$

or $(e_h)_j = \sum_{k \in C_j} w_{jk}(e_h)_k$, for

$$w_{jk} = -\frac{a_{jk} + \sum_{m \in F_i^s} \frac{a_{jm} a_{mk}}{\sum_{l \in C_j} a_{ml}}}{a_{jj} + \sum_{m \in F_i^w} a_{jm}}.$$

With these weights, we can form the coarse-to-fine transfer matrix, W, from which we can express the overall prolongation matrix, I_H^h, as

$$I_H^h = \begin{bmatrix} W \\ I \end{bmatrix}.$$

We stress that while we only investigate the use of this interpolation for structured grids in this work, the use of these multigrid components enable the solution of unstructured-grid Helmholtz problems, which will be the subject of future work.

3.4. Induced Dimension Reduction, IDR(s). Finally, we discuss an alternative to the Bi-CGSTAB iterative method. Recently, an efficient alternative to Bi-CGSTAB has been proposed in [32], also with limited memory requirements, called Induced Dimension Reduction, IDR(s). The parameter, s, in IDR(s) determines the number of pre-defined vectors used to enhance the method's convergence. In [32], it has been shown that IDR(1) has similar cost in terms of the memory requirements and computational complexity as Bi-CGSTAB. With higher values of s, the algorithm's storage requirements increase, but, typically, its performance improves accordingly. With $s = 4$, for example, 17 vectors need to be stored, whereas Bi-CGSTAB requires only 7. Bi-CGSTAB is based on the idea of computing two mutually bi-orthogonal bases for the Krylov subspaces based on matrix, A_h, and its conjugate transpose, A_h^H; IDR(s), on the other hand, is based on the generation of residuals that are forced to be in subspaces of decreasing dimension. The IDR(s) algorithm is as easy to implement as Bi-CGSTAB and other Krylov methods. In pseudo-code, the algorithm reads:

while $||r_l|| >$ TOL or $l <$ MAXIT do
  for $k = 0$ to s do
    Solve c from $P^H \mathrm{dR}_l c = P^H r_l$
    $v = r_l - \mathrm{dR}_l c$; $t = A_h v$;
    if $k = 0$ then
      $\omega = (t^H v)/(t^H t)$;
    endif
    $\mathrm{dr}_l = -\mathrm{dR}_l c - \omega t$; $\mathrm{dx}_l = -\mathrm{dX}_l c + \omega v$;
    $r_{l+1} = r_l + \mathrm{dr}_l$; $x_{l+1} = x_l + \mathrm{dx}_l$;
    $l = l + 1$;
    $\mathrm{dR}_l = (\mathrm{dr}_{l-1}, \ldots, \mathrm{dr}_{l-s})$; $\mathrm{dX}_l = (\mathrm{dx}_{l-1}, \ldots, \mathrm{dx}_{l-s})$;
  end for
end while

Here, we make a comparison between Bi-CGSTAB and the IDR(s) method for the heterogeneous high-wavenumber problems with realistic absorbing boundary conditions.

4. Numerical Experiments. In this section, we perform several 2D numerical experiments of increasing complexity. We start with the constant wavenumber problem, which serves as a benchmark for the algorithmic choices, after which we evaluate the method's performance for a Helmholtz problem with a wedge heterogeneity ,and a model of the Sigsbee field.

4.1. Homogeneous problem. The first numerical experiments are based on the *homogeneous* Helmholtz problem on the square domain, $(0, 1)^2$, to gain insight into the overall performance of the solvers. The pulse source, g, is located near the surface, at $(\frac{1}{2}, \frac{1}{32})$ and is represented by the scaled delta function. We will evaluate the following three multigrid preconditioners:

1 A multigrid V(1,1)-cycle with de Zeeuw's prolongation operator, FW restriction and Jacobi smoothing with relaxation parameter $\omega = 0.5$. This is the solver from [10].
2 A multigrid V(0,1)-cycle with de Zeeuw's prolongation operator, FW restriction and ILU post-smoothing.
3 A multigrid V(0,1)-cycle with AMG's prolongation operator, FW restriction and ILU post-smoothing.

These preconditioners are combined with the Bi-CGSTAB and IDR(s) Krylov subspace solvers. The value of β_1 in the shifted Helmholtz preconditioner equals 1, β_2 is set to either 0.4 or 0.5.

4.1.1. Systematic comparison of the multigrid schemes. We start by evaluating the choices made for the multigrid components in the preconditioner. Thus, we compare ILU(0) relaxation with the point-wise Jacobi smoother, and the prolongation operator of de Zeeuw [37] with AMG interpolation (based on the real part of the operator elements). We first consider the fourth-order discrete preconditioner in (3.2) and evaluate its solution using multigrid. Here, we choose a 2D homogeneous model, without an ABL. Starting with a random initial guess, zero boundary conditions and right-hand side, we present the asymptotic multigrid convergence factor, ρ, measured after 300 iterations.

For a given h, we fix the wavenumber, k, by requiring that $kh = 0.625$, meaning there are approximately ten grid points per wavelength. Table 4.1 presents the multi-

multigrid preconditioner	β_2	h : 1/64	1/128	1/256	1/512
ω-Jacobi	0.4	0.66	0.87	0.90	0.91
Zeeuw-V(1,1)	0.5	0.65	0.65	0.65	0.65
ILU(0)	0.4	0.39	0.44	0.60	0.77
Zeeuw-V(0,1)	0.5	0.27	0.27	0.30	0.33
ILU(0)	0.4	0.38	0.46	0.56	0.68
AMG-V(0,1)	0.5	0.27	0.27	0.30	0.34

TABLE 4.1

Multigrid convergence factors for solving M_h, the discrete (fourth-order discretization) version of Equation (3.2).

grid convergence factors for two values of β_2 (with $\beta_1 = 1$) and varying mesh size h. It shows that for $\beta_2 = 0.5$, as in [10], a stable multigrid convergence for decreasing mesh sizes (and increasing wavenumbers) is obtained. For $\beta_2 = 0.4$, the multigrid convergence rates increase; however, the combination of the V(0,1)-cycle, the ILU(0) smoother and the AMG prolongation operator still performs well for $\beta_2 = 0.4$ and $h = 1/512$. The ILU(0) smoother shows a significant improvement in the multigrid convergence, compared with those using a V(1,1)-cycle with an ω-Jacobi smoother. Moreover, the multigrid convergence for the fourth-order discretization is similar to that of the second-order discretization (not shown here). In fact, for the second-order discretization and $\beta_2 = 0.4$, we observed a multigrid divergence for the combination of ILU(0) smoothing and AMG interpolation, whereas Table 4.1 presents a highly satisfactory convergence (even better than that with the weights based on de Zeeuw's interpolation).

4.1.2. Second- and fourth-order discretizations. We compare the convergence of the Krylov subspace solvers for the second- and fourth-order discretizations of both the original operator and the preconditioner. The ABL is also not included in this experiment. We test each setting with a random initial guess and a point source as a right-hand side. The iteration is terminated as soon as the relative residual is reduced to a prescribed tolerance of 10^{-6},

$$\frac{||r_i||}{||r_0||} \leq 10^{-6}. \tag{4.1}$$

Tables 4.2 and 4.3 present, for fixed $kh = 0.625$, the Bi-CGSTAB performance on four meshes, with the three multigrid preconditioners, for the second- and fourth-order discretizations, respectively.

For all of the solvers, we observe a linear increase in the number of iterations for increasing wavenumbers. The performance of the V(0,1) multigrid preconditioner with de Zeeuw's prolongation weights and ILU(0) smoothing appears to be the most robust among these choices. Its convergence for both values of β_2, and for both discretizations, is very satisfactory. An interesting observation, however, is that the performance of the AMG interpolation (as in Table 4.3) is, especially on the finest grid, significantly better for the fourth-order discretization. The CPU times per iteration reported for the fourth-order discretization are always somewhat higher than for the second-order problem, as we deal with 9-point discretization stencils on all grids. However, as stated earlier, we do not need the extremely fine grids for high-wavenumber problems in combination with the fourth-order schemes.

preconditioner	β_2	h : 1/64	1/128	1/256	1/512
ω-Jacobi	0.4	39 (0.26)	75 (1.7)	139 (12.6)	266 (99)
Zeeuw-V(1,1)	0.5	36 (0.22)	69 (1.5)	125 (11.3)	236 (88)
ILU(0)	0.4	17 (0.13)	28 (0.63)	48 (4.3)	94 (36)
Zeeuw-V(0,1)	0.5	19 (0.14)	32 (0.70)	52 (4.7)	98 (37)
ILU(0)	0.4	26 (0.17)	43 (0.94)	88 (7.9)	218 (83)
AMG-V(0,1)	0.5	27 (0.19)	42 (0.92)	83 (7.4)	162 (61)

TABLE 4.2

Bi-CGSTAB performance for the homogeneous model (second-order discretization) in terms of number of iterations and CPU time in seconds (in brackets).

multigrid preconditioner	β_2	h : 1/64	1/128	1/256	1/512
ω-Jacobi	0.4	35 (0.30)	70 (2.0)	122 (14.1)	215 (103)
Zeeuw-V(1,1)	0.5	32 (0.25)	62 (1.8)	110 (13.0)	201 (96)
ILU(0)	0.4	16 (0.17)	26 (0.78)	46 (5.9)	85 (46)
Zeeuw-V(0,1)	0.5	20 (0.19)	30 (0.89)	51 (6.5)	96 (51)
ILU(0)	0.4	16 (0.13)	26 (0.78)	45 (5.7)	84 (45)
AMG-V(0,1)	0.5	19 (0.16)	30 (0.91)	52 (6.6)	95 (51)

TABLE 4.3

Bi-CGSTAB performance for the homogeneous model (fourth-order discretization) in terms of number of iterations and CPU time in seconds (in brackets).

4.1.3. Comparison with IDR(s). In this subsection, we fix our discretization to be fourth order and compare the convergence with and without ABL. The preconditioner chosen in all tests is the multigrid V(0,1)-cycle with ILU(0) smoothing and AMG interpolation, that performed the best for the fourth-order problems tested above.

Table 4.4 presents the number of matrix-vector products for preconditioned Bi-CGSTAB, IDR(2) and IDR(4), with the CPU times given in brackets. The presentation of the number of matrix-vector products, rather than the number of iterations, enables us to compare fairly among the different Krylov subspace methods. We choose $h = 1/512$ and $k = 320$, so that $kh = 0.625$. For the computations with the ABL, we add $n/4$ points to all sides of the domain. Thus, there is an increase in CPU time per iteration, as seen in the table, comparing the results with and without ABL, whereas the number of iterations is reduced with the ABL. Adding $n/4$ points in the ABL increases the overall problem size by a factor of 2.25.

From the results in Table 4.4, we conclude that the performance of Bi-CGSTAB in combination with the multigrid preconditioner chosen is optimal in terms of performance relative to the cost in memory. This may be somewhat surprising as the use of the IDR(s) method is especially preferred when there is a significant gap in the number of iterations between Bi-CGSTAB and the full GMRES performance. IDR(s) is able to close this gap; i.e., with limited extra storage and work, IDR(s) may converge like GMRES even when Bi-CGSTAB performs poorly. For the Helmholtz problems evaluated here, however, the difference in the number of iterations between full GMRES and Bi-CGSTAB is less than 10 %, confirming the conclusion that Bi-CGSTAB is favorable in these cases. This may be a result of the powerful preconditioner.

discretization	multigrid prec.	β_2	Krylov subspace method Bi-CGSTAB	IDR(2)	IDR(4)
no ABL	ILU Zeeuw V(0,1)	0.4	170 (46)	145 (45)	149 (50)
		0.5	192 (51)	177 (54)	174 (59)
	ILU AMG V(0.1)	0.4	168(48)	145(46)	149(50)
		0.5	190(51)	176(56)	172(57)
with ABL	ILU Zeeuw V(0,1)	0.4	136 (84)	122 (83)	129 (93)
		0.5	168 (104)	149 (103)	154 (112)
	ILU AMG V(0,1)	0.4	134 (82)	125 (85)	127 (92)
		0.5	166 (103)	143 (98)	149 (108)

TABLE 4.4

Solver performance with Bi-CGSTAB and IDR(s) for the homogeneous problem, fourth-order discretizations, with and without the ABL, in terms of the number of matrix-vector products and CPU time in seconds (in brackets).

number of points n	β_2	discretization 2nd order	4th order
32	0.4	32 (0.16)	22 (0.11)
	0.5	31 (0.11)	27 (0.13)
64	0.4	20 (0.27)	17 (0.30)
	0.5	20 (0.27)	20 (0.34)
128	0.4	17 (0.84)	17 (1.20)
	0.5	21 (1.08)	19 (1.36)
256	0.4	18 (3.94)	17 (5.20)
	0.5	21 (4.66)	19 (5.88)

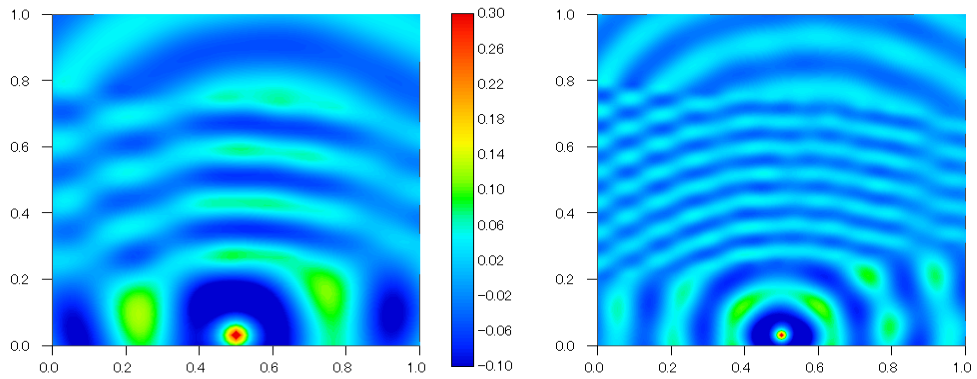
TABLE 4.5

Bi-CGSTAB performance for second- and fourth-order discretizations for the homogeneous model in terms of number of iterations and CPU time in seconds (in brackets).

4.1.4. Fixed wavenumber, increasing mesh sizes. Finally, we reconsider the 2D homogeneous model from Subsection 2.2.1, discretized on the unit square with the ABL. Wavenumber $k = 40$ is set in this experiment. The number of grid points increases in order to confirm grid independent convergence of the preconditioned Bi-CGSTAB solver for a fixed continuum problem. We terminate the iterations as soon as the relative residual is less than 10^{-6}.

The iterative solver in this experiment is based on the Bi-CGSTAB method with a V(0,1) multigrid preconditioner, with $\beta_1 = 1$, in which the ILU(0) smoother and the AMG prolongation are incorporated. Table 4.5 presents the number of Bi-CGSTAB iterations, plus the CPU time to reach the termination criterion, for two values of β_2, $\beta_2 = 0.4$ and $\beta_2 = 0.5$, with the second- and fourth-order discretizations. We observe the h-independent convergence rate for the iterative solver; with fixed wavenumber and h decreasing, approximately the same number of iterations is needed to satisfy the termination criterion, for both discretizations. The convergence for the two discretizations is very similar.

4.2. The Wedge Problem. In this section, we present numerical results for the so-called wedge problem. The domain, as in Figure 2.1, is a box, $(0, 1)^2$, in which a wedge-shaped heterogeneity is placed, and the location of the source is $(1/2, 1/32)$.



(a) $k = 40$ (b) $k = 80$

FIG. 4.1. *Numerical solutions for the wedge problem with* $k = 40$ *and* $k = 80$ *with ABL.*

The wave number inside the wedge region is k, and outside the wedge it is set to $k/2$. We employ the fourth-order discretization with the ABL here, with $n/4$ points on both sides in the ABL.

Figure 4.1 presents the solutions of the wedge problem for $k = 40$ and $k = 80$. Furthermore, for $k = 80$, we present in Figure 4.2 the vertical centerline solution profile obtained on three different meshes. These profiles converge as expected for increasing mesh sizes. With the fourth-order discretization, 64 points already seem sufficient for an accurate representation, whereas 128 points should be chosen according to the rule $kh = 0.625$.

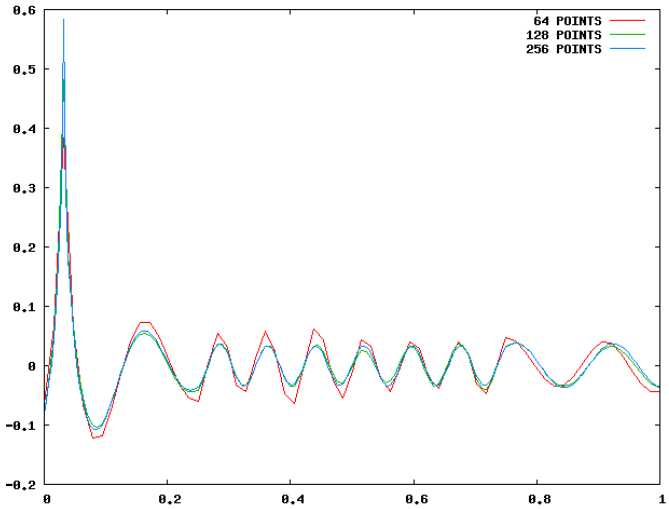


FIG. 4.2. *Centerline solution profile for the wedge problem with* $k = 80$ *with ABL, for different mesh sizes.*

Next, we examine the convergence of the preconditioned Bi-CGSTAB method, with the shifted Laplacian V(0,1)-multigrid preconditioner with the ILU(0) smoother and the AMG prolongation. Parameter kh is set to 0.625. Here, we also present

	damping	number of points (kh constant)			
β_2	α	64	128	256	512
0.4	0.0%	18 (0.31)	31 (2.16)	55 (16.4)	96 (117)
	1.0%	17 (0.34)	26 (1.84)	39 (11.8)	58 (72)
	2.5%	15 (0.28)	21 (1.52)	28 (8.5)	44 (54)
	5.0%	13 (0.28)	16 (1.22)	21 (6.5)	25 (32)
0.5	0.0%	23 (0.50)	38 (2.66)	68 (20.2)	121 (150)
	1.0%	21 (0.34)	31 (2.22)	46 (13.9)	71 (88)
	2.5%	18 (0.34)	27 (1.92)	37 (11.6)	49 (61)
	5.0%	15 (0.28)	19 (1.41)	26 (7.9)	28 (35)

TABLE 4.6

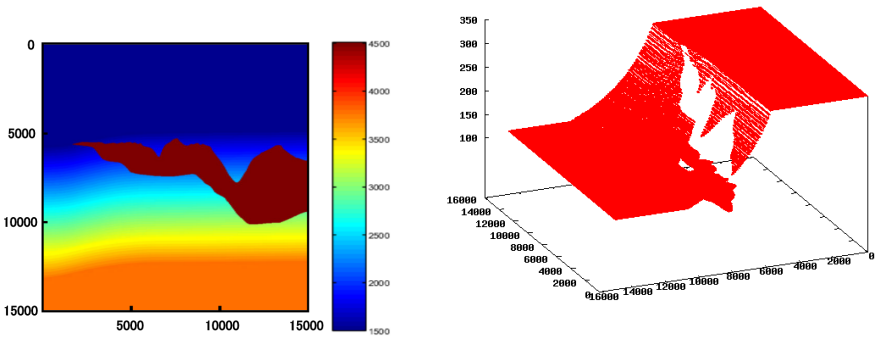
Bi-CGSTAB performance for the fourth-order discretization of the wedge model, with α % damping, in terms of number of iterations and CPU time in seconds (in brackets).

results with some damping included in the original Helmholtz equation. Parameter α in (2.1) varies between 0 and 0.05. The number of iterations and CPU time (in seconds) are presented in Table 4.6. We notice a significant improvement of the method's convergence, already when 1% damping is included in the original problem. With 5% damping, we even observe a *constant* number of iterations, for varying k. Compared to the performance of the solution method with the damped Jacobi smoother (not shown here), the results in Table 4.6 are significantly improved, both in terms of the number of iterations and in terms of the CPU time. Especially on the fine meshes, we see in Table 4.6 that it is beneficial to choose $\beta_2 = 0.4$.

4.3. The Sigsbee Problem. The Sigsbee2A synthetic dataset models the geologic setting found on the Sigsbee escarpment in the deep-water Gulf of Mexico. There is a substantial uniform layer of water at the top of the model. Here, we use a scaled version of the original Sigsbee model to test our iterative Helmholtz solver, see Figure 4.3. The size of the domain is $15000^2 m$ and a source is placed at $(7500, 117)$, near the top wall. The frequency chosen for this computation is $5Hz$.

The grid size consists of 512^2 points with an ABL of 128 points on each edge. The largest value of kh is 0.6135. Figure 4.4 presents the solution of this Helmholtz problem, where the fourth-order discretization is used. With the linear solver based on preconditioned Bi-CGSTAB with the V(0,1) multigrid preconditioner for the shifted Laplacian, using the ILU(0) smoother and AMG-based interpolation as essential components, we solve this problem in 61 iterations and 74.2 CPU seconds with $\beta_2 = 0.4$, and in 68 iterations and 85.5 CPU seconds for $\beta_2 = 0.5$. This convergence is highly satisfactory for this real-life setting. As a comparison, the original solver in [10] with $\beta_2 = 0.5$, the multigrid V(1,1)-cycle preconditioner, point-wise Jacobi smoothing and de Zeeuw's interpolation needed 216 iterations and 237 seconds CPU time.

5. Conclusion. In this paper, we have presented a fast iterative solver for heterogeneous high-wavenumber Helmholtz problems. The solver is an improvement of the work in [10]. A preconditioned Bi-CGSTAB solver has been developed where the preconditioner is based on a shifted Laplacian with a complex-valued shift. We have shown that it is possible to work with fourth-order finite differences, both in the discrete original problem, as well as in the preconditioner without any problem. An absorbing boundary layer improves the quality of the solution significantly and does not pose difficulties to the solution method proposed. We have focussed on discretizations obeying a linear relation between the wavenumber and the mesh size,



(a) Speed of sound, $c(\boldsymbol{x})$ for Sigsbee scaled domain

(b) Distribution of k

FIG. 4.3. *Domain for the scaled Sigsbee problem and the distribution of velocity, $c(\boldsymbol{x})$, and wavenumber, k.*

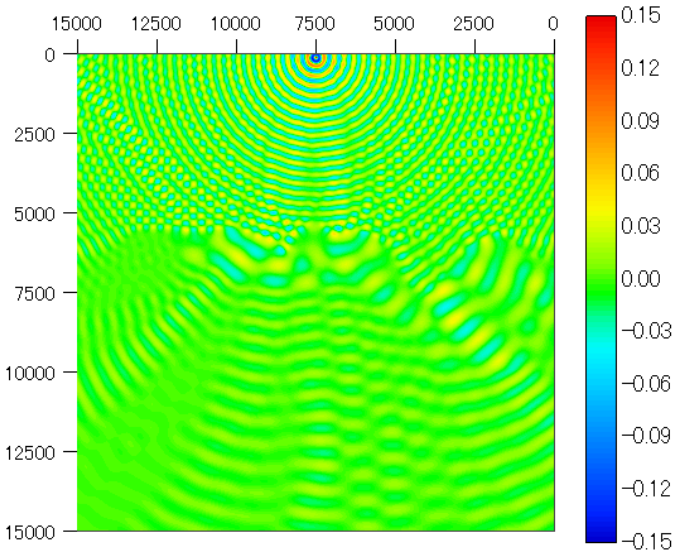


FIG. 4.4. *Solution of the scaled Sigsbee problem with ABL, frequency $5Hz$.*

$kh = 0.625$, here.

The fourth-order accurate shifted Laplacian preconditioner can easily be approximated by one V(0,1)-cycle of multigrid. In the multigrid preconditioner, we have included a powerful ILU(0) smoother and an AMG-based prolongation scheme. This enables us to choose a somewhat smaller imaginary shift parameter ($\beta_2 = 0.4$) in the preconditioner, which improves the solver's convergence (especially for high wavenumbers on fine meshes). The performance of the solver is significantly improved compared to the convergence results in [10]. With these additions, Helmholtz problems on unstructured grids can, in principle, be solved in the same manner.

REFERENCES

[1] T. Airaksinen, E. Heikkola, A. Pennanen, J. Toivanen, An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation. *J. Comp. Physics,* 226: 1196–1210, 2007.

[2] A. Bamberger, P. Joly, J.E. Roberts, Second-order absorbing boundary conditions for the wave equations: A solution for the corner problem. *SIAM J. Numer. Anal.,* 27: 323–352, 1990.

[3] A. Bayliss, C.I. Goldstein and E. Turkel, On accuracy conditions for the numerical computation of waves. *J. Comput. Phys.,* 59: 396–404, 1985.

[4] A. Brandt, Multi-level adaptive solutions to boundary-value problems. *Math. Comput.* 31: 333–390, 1977.

[5] A. Brandt, I. Livshits, Wave-ray multigrid methods for standing wave equations. *Elect. Trans. Numer. Anal.* 6: 162–181, 1997.

[6] W.L. Briggs, V.E. Henson, and S.F. McCormick, A multigrid tutorial: SIAM, Philadelphia, USA, 2000.

[7] H. R. Elman, O. G. Ernst and D. P. O'Leary, A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, *SIAM J. Sci. Comput.*, 23: 1291–1315, 2001.

[8] B. Engquist, A. Majda, Absorbing boundary conditions for the numerical simulation of waves, *Math. Comput.*, 31: 629–651, 1977.

[9] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Applied Num. Math.* 56: 648-666, 2006.

[10] Y.A. Erlangga, C.W. Oosterlee, C. Vuik, A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.* 27: 1471-1492, 2006.

[11] M. B. van Gijzen, Y. A. Erlangga, C. Vuik, Spectral Analysis of the Discrete Helmholtz Operator Preconditioned with a Shifted Laplacian *SIAM J. Sci. Comput.*, 29: 1942-1958, 2007.

[12] A. George, J.W. Liu, *Computer solution of large sparse positive definite systems,* Prentice-Hall, New Jersey, 1981.

[13] J. Gozani, A. Nachshon, E. Turkel, Conjugate gradient coupled with multigrid for an indefinite problem, in *Advances in Comput. Methods for PDEs V*, 425–427, 1984.

[14] W. Hackbusch, *Multi-grid methods and applications.* Springer, Berlin, 1985.

[15] F. Ihlenburg and I. Babuska, Finite element solution to the Helmholtz equation with high wave numbers. *Comput. Math. Appl.,* 30: 9-37, 1995.

[16] C.-H. Jo, C. Shin, J.H. Suh, An optimal 9-point, finite-difference, frequency space, 2-D scalar wave extrapolator, *Geophysics* 61(2): 529–537, 1996.

[17] R. Kettler, Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. *In:*W. Hackbusch and U. Trottenberg (eds.), Multigrid methods, *Lecture Notes in Mathematics* 960: 502–534, Springer, Berlin, 1982.

[18] M. Khalil, *Analysis of linear multigrid methods for elliptic differential equations with discontinuous and anisotropic coefficients.* Ph.D. Thesis, Delft University of Technology, Delft, Netherlands, 1989.

[19] S. Kim, S. Kim, Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media, *SIAM J. Sci. Comput.* 24: 684–701, 2002.

[20] A. L. Laird, M. B. Giles, *Preconditioned iterative solution of the 2D Helmholtz equation.* Report NA 02-12, Comp. Lab., Oxford Univ., 2002.

[21] Q. Liao, G.A. McMechan, Multifrequency viscoacoustic modeling and inversion. *Geophysics* 61(5): 1371–1378, 1996.

[22] S.P. MacLachlan, C.W.Oosterlee, *Algebraic multigrid solvers for complex-valued matrices.* To appear in SIAM J. Sci. Comput., 2008.

[23] W.A. Mulder, R.-E. Plessix, One-way and two-way wave-equation migration, *Geophysics*, 69: 1491–1504, 2004.

[24] W.A. Mulder, R.-R. Plessix, How to choose a subset of frequencies in frequency-domain finite-difference migration, *Geophys. J. Int.* 158: 801–812, 2004.

[25] R.G. Pratt, Seismic waveform inversion in frequency domain. Part I: theory and verification in a physical scale domain, *Geophysics*, 64: 888–901, 1999.

[26] C.D. Riyanti, Y.A. Erlangga, R.-E. Plessix, W.A. Mulder, C.W. Oosterlee, C. Vuik, A new iterative solver for the time-harmonic wave equation, *Geophysics* 71: 57-63, 2006.

[27] C.D. Riyanti, A. Kononov, Y.A. Erlangga, C. Vuik, C.W. Oosterlee, R-E Plessix and W.A. Mulder, A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *J. Comp. Physics* 224: 431-448, 2007.

[28] J.W. Ruge and K. Stüben, Algebraic Multigrid (AMG). *In:* S.F. McCormick (ed.), Multigrid

Methods, Frontiers in Appl. Math., SIAM Philadelphia, 5: 73–130, 1987.
[29] I. Singer and E. Turkel, High Order Finite Difference Methods for the Helmholtz Equation, *Comp. Meth. Appl. Mech. Eng.* 163:343-358, 1998.
[30] I. Singer, E. Turkel, A Perfectly Matched Layer for the Helmholtz Equation in a Semi-infinite Strip. *J. Comput. Phys.* 201 (2004) 439-465.
[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003
[32] P. Sonneveld, M.B. van Gijzen, *IDR(s): a new family of efficient algorithms for solving large nonsymmetric linear systems.* Delft Univ. Techn. Report 07-07, 2007.
[33] U. Trottenberg, C.W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, London, 2001.
[34] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Comput.*, 13: 631-645, 1992.
[35] P. Wesseling, *An introduction to multigrid methods.* John Wiley, Chichester, 1992.
[36] G. Wittum, On the robustness of ILU smoothing. *SIAM J. Sci. Comput.* 10: 699–717, 1989.
[37] P. M. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.*, 33: 1–27, 1990.
[38] P.M. de Zeeuw, Incomplete line LU as smoother and as preconditioner. *In:* W. Hackbusch, G. Wittum (eds.), Incompete decompositions (ILU) - algorithms, theory, and applications. Vieweg, Braunschweig, 215–224, 1993. *Int. J. Num. Methods in Fluids* 20: 59–74, 1995.