A Semicoarsening-based Multigrid Preconditioner for 3D Inhomogeneous Helmholtz Equations

Y.A. Erlangga, C. Vuik, C.W. Oosterlee

January 3, 2006

Abstract

In this paper an iterative solution method for the 3D Helmholtz equation is presented. The method is a generalization of the method presented in [*Erlangga, Oosterlee, Vuik, SIAM J. Sci. Comput., to appear*] for the 2D heterogeneous Helmholtz equation. The method employs 3D multigrid with 2D semicoarsening as a preconditioner for a Krylov subspace iteration method. This multigrid method is, however, not applied to the Helmholtz operator directly, but to a Helmholtz operator with complex shift, a so-called shifted Laplacian preconditioner. Numerical results obtained on a sequential machine indicate the efficiency and the robustness of the method.

Keywords. 3D Helmholtz equation, nonconstant high wavenumber, Krylov subspace methods, complex multigrid preconditioner, semicoarsening
AMS subject classifications. 65N55, 65F10, 65N06, 65N22, 78A45, 76Q05

1 Introduction

Iterative methods have become very popular for solving sparse, large linear systems. Direct solution methods typically suffer from sparsity fill-in, especially for systems with a wide bandwidth. This increases the amount of storage needed and the computational work required for solving matrices drastically. Matrices with a wide bandwidth occur particularly in 3D applications. Iterative methods, on the contrary, are usually constructed based on matrix-vector products, that exploit the sparsity of a matrix. Highly efficient iterative methods, that are commonly used are multigrid [16] and preconditioned Krylov subspace methods [14].

The multigrid method is known to be an efficient iterative method for solving linear systems derived from nicely elliptic partial differential equations (PDEs). A tuned multigrid method can result in $O(N)$ complexity for solving Poisson type equations.

The Helmholtz equation, however, does not belong to the class of PDEs which multigrid can solve efficiently. Convergence degradation, and consequently loss of $O(N)$ complexity are caused by difficulties encountered in the smoothing and coarse grid correction components, the two basic principles of multigrid; see [4, 8] for discussion. Since the Helmholtz equation appears in many fields, e.g. in under-water acoustics, seismics and electromagnetics, the need for an efficient 3D iterative solver is high. Many authors, e.g. [9, 3, 4, 10], have contributed to the development of appropriate multigrid methods for the Helmholtz equation. Multigrid is then used either as a solver or as a preconditioner.

In [8] we have presented a 2D preconditioner based on the Helmholtz equation with an imaginary shift. The iterations are performed in the context of Krylov subspace methods. This preconditioner is approximately inverted by means of one iteration of multigrid, which is highly efficient. It appears as a member of the family of shifted Laplacian operators introduced in [6, 7]. In [7, 8], we applied the iterative method to various 2D Helmholtz problems with inhomogeneity in the medium, like to a Marmousi [2] problem arising from geophysics and mimicking the Earth's subsurface layer with strong contrasts.

Similar approaches have also been proposed in [9], where a Laplacian is used as a preconditioner for Conjugate Gradients; In [11] the *definite* version of the Helmholtz equation resulted in a faster convergence. The preconditioner in [8] is an efficient generalization of these.

In this paper we generalize the 2D method to three spatial dimensions. We solve 3D Helmholtz problems with heterogeneous media. Following the basics from [6, 7, 8], the method presented here consists of a Krylov subspace method (in our case Bi-CGSTAB), a preconditioning operator that is handled by multigrid. A 3D multigrid algorithm based on 2D semicoarsening, plus line-wise smoothing in the third direction is used. This may seem not natural for problems without anisotropy, like the Helmholtz equation. However, anisotropies may occur also here in the form of different numbers of grid points in different spatial directions.

In Section 2 we briefly introduce the 3D Helmholtz equation, the preconditioning operator, and their discrete finite difference formulations. Section 3 discusses the 3D multigrid method with semicoarsening. Numerical results are presented in Section 4.

2 3D Helmholtz equation and multigrid

We consider the three-dimensional Helmholtz equation in a cubic domain $\Omega \subset \mathbb{R}^3$, with $\boldsymbol{x} = (x_1, x_2, x_3)$:

$$\mathcal{A}u(\boldsymbol{x}) := -\left(\partial_{x_1x_1} + \partial_{x_2x_2} + \partial_{x_3x_3} + k^2(\boldsymbol{x})\right) u(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad (1)$$

where k is the wavenumber. The boundary conditions on $\Gamma = \partial\Omega$ are set on the faces, edges and corners of the domain. On each part of Γ we set the following absorbing boundary layer conditions (see [1, 5]):

$$\text{Faces:} \qquad \mathcal{B}u|_{face} := \pm\frac{\partial u}{\partial x_i} - \hat{j}ku - \frac{\hat{j}}{2k}\sum_{1\leq j\neq i\leq 3}\frac{\partial^2 u}{\partial x_j^2} = 0, \qquad i = 1, \dots, 3, \quad (2)$$

where x_i indicates the coordinate perpendicular to the face;

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

with x_i the coordinate parallel to the edge.

$$\text{Corners:} \qquad \mathcal{B}u|_{corner} := -2\hat{j}ku + \sum_{i=1}^{3} \pm \frac{\partial u}{\partial x_i} = 0. \tag{4}$$

In (2)–(4) $\hat{j}$ is the complex identity, i.e. $\hat{j} = \sqrt{-1}$.

If (1) is discretized using the 7-point central finite difference scheme, a linear system

$$A\phi = g, \qquad A \in \mathbb{C}^{N\times N}, \qquad \phi, g \in \mathbb{C}^N, \tag{5}$$

is obtained. Stencil A_h corresponding to matrix A then reads:

$$A_h \triangleq \frac{1}{h^2} \left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h , \quad \begin{bmatrix} 0 & 0-1 & 0 \\ -1 & 6-(kh)^2 & -1 \\ 0 & 0-1 & 0 \end{bmatrix}_h , \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right]_h . \tag{6}$$

One can also choose another discretization, e.g. a 19- or 27-point stencil.

Matrix A in (5) is sparse, large and symmetric, but indefinite and ill-conditioned for sufficiently large k. A Krylov subspace method applied to (5) without a preconditioner converges extremely slow. To improve the convergence a preconditioner is used. Preconditioning based on an incomplete factorization of A does, however, not lead to a robust iterative method in the case of high wavenumbers and heterogeneities; see the numerical results in [7]. Therefore, we use the shifted Laplacian operator for preconditioning, which in 3D takes the form:

$$\mathcal{M} := -\left(\partial_{x_1x_1} + \partial_{x_2x_2} + \partial_{x_3x_3} + (1 - 0.5\hat{j})k^2 \right). \tag{7}$$

The preconditioning matrix M is constructed by discretization of (7) using central differences. In the construction, the boundary conditions (2)–(4) are also incorporated. With M as the preconditioner, the (right) preconditioned linear system to be solved is given by:

$$AM^{-1}(M\phi) = g. \tag{8}$$

In practice, M is never inverted explicitly. We, for example, will use one multigrid iteration to approximately invert M. System (8) is then solved using a Krylov subspace method. We have chosen Bi-CGSTAB [18], which performed best in 2D. Multigrid's cycle type(V-, F- or W-cycle) and smoother (Jacobi or Gauß-Seidel) have to be prescribed, as the number of pre- and post-smoothing steps, ν_1 and ν_2, respectively.

For three-dimensional problems, the choice of multigrid components is larger than in 2D. In general, a multigrid method with standard coarsening, i.e. with doubling the mesh size in all directions, results in highly efficient 3D Poisson solvers. For anisotropic Poisson problems, standard coarsening combined with pointwise smoothing does not converge well. In this case, the use of line (or plane) smoothers is mandatory with standard coarsening. Point smoothers can lead to a fast converging method if a proper semicoarsening strategy is incorporated (see [15]).

In more complicated situations, with strong anisotropies in arbitrary directions, the use of *algebraic* multigrid is natural. In this paper we use *geometric* multigrid. Operator (7) with boundary conditions (2), (3) and (4) does not need a non-standard multigrid treatment; standard coarsening multigrid method with point smoother may be sufficient. This is confirmed in 2D. However, as nonsquare grid cells may occur in one pre-defined direction, we employ semicoarsening in two directions and linewise smoothing in the third direction. Grid anisotropies occur, for example, if the domain of interest is of a different height, compared to the domain widths. Further, all 2D multigrid components are immediately usable in 3D setting. We only change the pointwise smoother to a linewise smoother in the uncoarsened third dimension. The interpolation operator can be built based on bilinear interpolation or 2D interpolation operator by de Zeeuw [20]. The restriction operator can also be determined as in 2D. The generalization of the pointwise damped Jacobi smoother in [8] is the linewise damped Jacobi smoother. Notice that the number of grid points on coarse grids now reduces only with a factor of four, as compared to a factor of eight in the case of standard grid coarsening. The 3D multigrid method with (x_1, x_2)-semicoarsening is illustrated in Fig. 1 for three grid levels. As can be

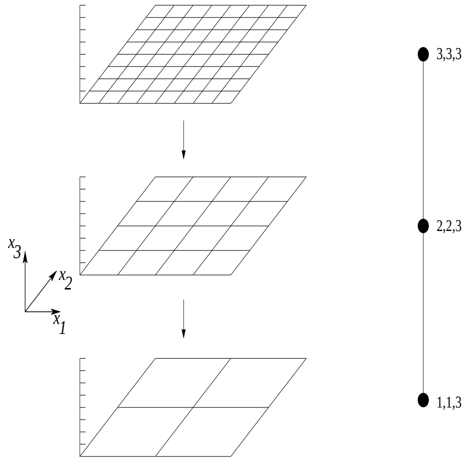

Figure 1: Semicoarsening of three grid levels: standard coarsening in two directions (x_1 and x_2), the third (x_3) direction is kept uncoarsened

seen, doubling the mesh size only takes place in the x_1- and x_2-directions, while the x_3-direction is kept uncoarsened. We will show in the numerical results that the choice of the coarsening directions is unimportant for Helmholtz problems on square grids.

Semicoarsening strategies other than the one shown in Fig 1 also exist; see e.g. [12] for a method called 'Multiple Semicoarsened Grids (MSG) and [13] for 'Multigrid also used as a Smoother' (MG-S). These robust strategies are mostly designed for problems with general anisotropies and for the Euler equations. The strategy in [13] (later extended to 3D in [19]) requires smoothing in the x_1- and x_2-directions before the grid is coarsened simultaneously in the x_1- and x_2-directions. This process will give also for isotropic 3D Helmholtz problems a slightly better convergence, but it requires substantially extra work on coarser levels and extra information on coarse grids (thus, storage).

3 Multigrid to approximate M

The 3D multigrid method with (x_1, x_2)-semicoarsening and linewise smoothing is designed as follows: Let us consider the preconditioner operator (7), discretized by the seven-point stencil on a grid $\mathcal{G}_h = \{(i_1 h, i_2 h, i_3 h)|$
$i_1, i_2, i_3 = 1, \ldots, n\}$, with $h = 1/n$.

For a damped x_3-line Jacobi relaxation with relaxation factor $0 < \omega_{jac} \leq 1$ we have the following iteration:

$$(M_{x_3} + D)\widetilde{\phi}^{j+1} + (M_{x_1} + M_{x_2})\phi^j = g, \tag{9}$$

$$\phi^{j+1} = \omega_{jac}\widetilde{\phi}^{j+1} + (1 - \omega_{jac})\phi^j, \tag{10}$$

or

$$\begin{aligned} \frac{1}{\omega_{jac}}(M_{x_3} + D)\phi^{j+1} &= g - (M_{x_1} + M_{x_2})\phi^j + \frac{1 - \omega_{jac}}{\omega_{jac}}(M_{x_3} + D)\phi^j, \\ &= g - M\phi^j + \frac{1}{\omega_{jac}}(M_{x_3} + D)\phi^j, \qquad (11) \end{aligned}$$

with

$$(M_{x_1}\phi)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1+1,i_2,i_3} + u_{i_1-1,i_2,i_3}\right), \tag{12}$$

$$(M_{x_2}\phi)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1,i_2+1,i_3} + u_{i_1,i_2-1,i_3}\right), \tag{13}$$

$$(M_{x_3}\phi)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1,i_2,i_3+1} + u_{i_1,i_2,i_3-1}\right), \tag{14}$$

$$(D\phi)_{i_1,i_2,i_3} := \frac{1}{h^2}\left(6 - (1 - 0.5\hat{\jmath})k^2h^2\right)u_{i_1,i_2,i_3}. \tag{15}$$

It is then possible to determine the smoothing factor of this method in the case of semicoarsening.

In [8] we analyze and show that 2D standard multigrid with point Jacobi smoother results in a robust method for the 2D version of the preconditioner (7) for various kh. For $\omega_{jac} = 0.5$ a 2D F(1,1)-multigrid algorithm with an acceptable convergence factor (usually 0.5-0.7) is obtained. This convergence is sufficient to approximately invert the preconditioner. As compared to exact inversion, one multigrid iteration of this kind does not influence convergence of a Krylov subspace method applied to the Helmholtz equation negatively. Furthermore we use the following multigrid components:

(i) restriction, $I_h^H : \mathcal{G}_h \rightarrow \mathcal{G}_H$, is 2D full weighting,

(ii) prolongation, $I_H^h : \mathcal{G}_H \rightarrow \mathcal{G}_h$, is either 2D bilinear interpolation or 2D matrix-dependent interpolation,

(iii) coarse grid matrix is based on the Galerkin coarse grid discretization, defined as $M_H = I_h^H M_h I_H^h$.

Prolongation operator. Next to the bilinear interpolation, we consider a matrix-dependent interpolation based on the operator proposed by de Zeeuw in [20]. This type of prolongation is especially useful for problems with strong heterogeneities. In the original formulation, this matrix dependent interpolation is developed based on the splitting of symmetric and non-symmetric part of the linear system. In our case, symmetry of the linear system reduces the representation of the interpolation operator. Furthermore, we modify the operator such that the weights are now based on the modulus of some complex-valued constant.

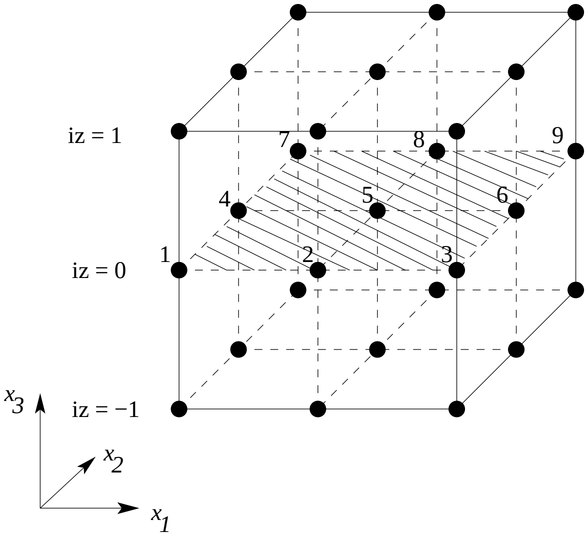

Figure 2: The 27-point stencil

To determine the 3D interpolation weights, we consider the 27-point stencil matrix (see Fig. 2)

$$\begin{aligned}(M\phi)_{i_1,i_2,i_3} &= \sum_{iz=-1,0,1}[m(iz)^1_{i_1,i_2,i_3}\phi_{i_1-1,i_2-1,i_3+iz} \\ &+ m(iz)^2_{i_1,i_2,i_3}\phi_{i_1,i_2-1,i_3+iz} + m(iz)^3_{i_1,i_2,i_3}\phi_{i_1+1,i_2-1,i_3+iz} \\ &+ m(iz)^4_{i_1,i_2,i_3}\phi_{i_1-1,i_2,i_3+iz} + m(iz)^5_{i_1,i_2,i_3}\phi_{i_1,i_2,i_3+iz} \\ &+ m(iz)^6_{i_1,i_2,i_3}\phi_{i_1+1,i_2,i_3+iz} + m(iz)^7_{i_1,i_2,i_3}\phi_{i_1-1,i_2+1,i_3+iz} \\ &+ m(iz)^8_{i_1,i_2,i_3}\phi_{i_1,i_2+1,i_3+iz} + m(iz)^9_{i_1,i_2,i_3}\phi_{i_1+1,i_2+1,i_3+iz}] \quad (16)\end{aligned}$$

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

$$\begin{aligned}(\widetilde{M}\phi)_{i_1,i_2,i_3} &= \widetilde{m}^1_{i_1,i_2,i_3}\phi_{i_1-1,i_2-1,i_3} \\ &+\widetilde{m}^2_{i_1,i_2,i_3}\phi_{i_1,i_2-1,i_3} + \widetilde{m}^3_{i_1,i_2,i_3}\phi_{i_1+1,i_2-1,i_3} \\ &+\widetilde{m}^4_{i_1,i_2,i_3}\phi_{i_1-1,i_2,i_3} + \widetilde{m}^5_{i_1,i_2,i_3}\phi_{i_1,i_2,i_3} \\ &+\widetilde{m}^6_{i_1,i_2,i_3}\phi_{i_1+1,i_2,i_3} + \widetilde{m}^7_{i_1,i_2,i_3}\phi_{i_1-1,i_2+1,i_3} \\ &+\widetilde{m}^8_{i_1,i_2,i_3}\phi_{i_1,i_2+1,i_3} + \widetilde{m}^9_{i_1,i_2,i_3}\phi_{i_1+1,i_2+1,i_3}, \qquad (17)\end{aligned}$$

with

$$\widetilde{m}^p_{i_1,i_2,i_3} = m(-1)^p_{i_1,i_2,i_3} + m(0)^p_{i_1,i_2,i_3} + m(1)^p_{i_1,i_2,i_3},\ p = 1, 2, \ldots, 9. \qquad (18)$$

Based on the lumped 9-point stencil the coarse-to-fine grid operator can be determined from the following:

$$\begin{aligned}v_{2i_1-1,2i_2-1,i_3} &= u_{i_1,i_2,i_3} \\ v_{2i_1,2i_2-1,i_3} &= w1_{2i_1,2i_2-1,i_3}u_{i_1,i_2,i_3} + w2_{2i_1,2i_2-1,i_3}u_{i_1+1,i_2,i_3} \\ v_{2i_1-1,2i_2,i_3} &= w1_{2i_1-1,2i_2,i_3}u_{i_1,i_2,i_3} + w3_{2i_1-1,2i_2,i_3}u_{i_1,i_2+1,i_3} \\ v_{2i_1,2i_2,i_3} &= w1_{2i_1,2i_2,i_3}u_{i_1,i_2,i_3} + w2_{2i_1,2i_2,i_3}u_{i_1+1,i_2,i_3} \\ &+ w3_{2i_1,2i_2,i_3}u_{i_1,i_2+1,i_3} + w4_{2i_1,2i_2,i_3}u_{i_1+1,i_2+1,i_3},\end{aligned}$$

with the weights $w1, \ldots, w4$ determined from the 2D interpolation weights [20], see also [19]. Here we explain the matrix-dependent interpolation weights in some more detail (for bilinear interpolation in multigrid we refer to, for example, [16]).

The derivation of the matrix-dependent prolongation weights starts by splitting $\widetilde{M}$ into a symmetric and an antisymmetric part:

$$\widetilde{M_S} = \frac{1}{2}\left(\widetilde{M} + \widetilde{M}^T\right),\ \widetilde{M_T} = \frac{1}{2}\left(\widetilde{M} - \widetilde{M}^T\right). \qquad (19)$$

The elements of the symmetric and the antisymmetric parts are denoted by $\widetilde{m}_s$ and $\widetilde{m}_t$, respectively. In our case, $\widetilde{M_T} = 0$ *on the finest grid only.* On

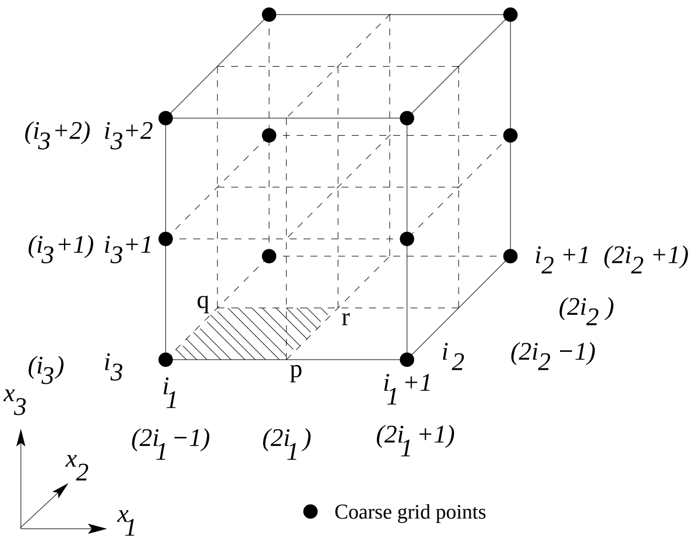

Figure 3: Coarse and fine grid cells for (x_1, x_2)-semicoarsening

the coarse grids, due to the lumping, unsymmetry enters the the definition of the matrices and thus the computations. Therefore, we adopt the unsymmetric strategy as presented in [19] in the definition of the interpolation weights. For the symmetric part, we have

$$
\begin{aligned}
d_w &= \max(|\widetilde{m}^1_{s,2i_1,2i_2-1,i_3} + \widetilde{m}^4_{s,2i_1,2i_2-1,i_3} + \widetilde{m}^7_{s,2i_1,2i_2-1,i_3}|, \\
&\quad |\widetilde{m}^1_{s,2i_1,2i_2-1,i_3}|, |\widetilde{m}^7_{s,2i_1,2i_2-1,i_3}|), \\
d_e &= \max(|\widetilde{m}^3_{s,2i_i,2i_2-1,i_3} + \widetilde{m}^6_{s,2i_1,2i_2-1,i_3} + \widetilde{m}^9_{s,2i_1,2i_2-1,i_3}|, \\
&\quad |\widetilde{m}^3_{s,2i_1,2i_2-1,i_3}|, |\widetilde{m}^9_{s,2i_1,2i_2-1,i_3}|), \\
d_n &= \max(|\widetilde{m}^7_{s,2i_1-1,2i_2,i_3} + \widetilde{m}^8_{s,2i_1-1,2i_2,i_3} + \widetilde{m}^9_{s,2i_1-1,2i_2,i_3}|, \\
&\quad |\widetilde{m}^7_{s,2i_1-1,2i_2,i_3}|, |\widetilde{m}^9_{s,2i_1-1,2i_2,i_3}|), \\
d_s &= \max(|\widetilde{m}^1_{s,2i_1-1,2i_2,i_3} + \widetilde{m}^2_{s,2i_1-1,2i_2,i_3} + \widetilde{m}^3_{s,2i_1-1,2i_2,i_3}|, \\
&\quad |\widetilde{m}^1_{s,2i_1-1,2i_2,i_3}|, |\widetilde{m}^3_{s,2i_1-1,2i_2,i_3}|), \\
\sigma_1 &= \frac{1}{2}\min\left(1, \left|1 - \frac{\sum_{p=1}^{9} \left|\widetilde{m}^p_{s,2i_1,2i_2-1,i_3}\right|}{\widetilde{m}^5_{s,2i_1,2i_2-1,i_3}}\right|\right), \\
\sigma_2 &= \frac{1}{2}\min\left(1, \left|1 - \frac{\sum_{p=1}^{9} \left|\widetilde{m}^p_{s,2i_1-1,2i_2,i_3}\right|}{\widetilde{m}^5_{s,2i_1-1,2i_2,i_3}}\right|\right).
\end{aligned}
$$

For the unsymmetric part, two parameters c_1 and c_2 are defined as:

$$\begin{aligned}
c_1 &= \widetilde{m}^3_{t,2i,2j-1,k} + \widetilde{m}^6_{t,2i,2j-1,k} + \widetilde{m}^9_{t,2i,2j-1,k} - (\widetilde{m}^1_{t,2i,2j-1,k} + \widetilde{m}^4_{t,2i,2j-1,k} + \widetilde{m}^7_{t,2i,2j-1,k}) \\
c_2 &= \widetilde{m}^7_{t,2i-1,2j,k} + \widetilde{m}^8_{t,2i-1,2j,k} + \widetilde{m}^9_{t,2i-1,2j,k} - (\widetilde{m}^1_{t,2i-1,2j,k} + \widetilde{m}^2_{t,2i-1,2j,k} + \widetilde{m}^3_{t,2i-1,2j,k}).
\end{aligned}$$

As mentioned, the elements of the symmetric and the antisymmetric parts are denoted by subscript s and t, respectively. Using these quantities the matrix-dependent weights on the west, east, north and south are determined as follows:

$$\begin{aligned}
w_w &= \sigma_1 \left(1 + \frac{d_w - d_e}{d_w + d_e} + \frac{c_1}{d_w + d_e + d_n + d_s}\right), \quad w_e = 2\sigma_1 - w_w, \\
w_n &= \sigma_2 \left(1 + \frac{d_s - d_n}{d_s + d_n} + \frac{c_2}{d_w + d_e + d_n + d_s}\right), \quad w_s = 2\sigma_2 - w_n.
\end{aligned}$$

The weights $w1, \ldots, w4$ can now be computed, i.e.,

- for $(2i_1, 2i_2 - 1, i_3)$

  $$w1_{2i_1,2i_2-1,i_3} = \min(2\sigma_1, \max(0, w_w)), \; w2_{2i_1,2i_2-1,i_3} = \min(2\sigma_1, \max(0, w_e)).$$

- for $(2i_1 - 1, 2i_2, i_3)$

  $$w1_{2i_1-1,2i_2,i_3} = \min(2\sigma_1, \max(0, w_s)), \; w3_{2i_1-1,2i_2,i_3} = \min(2\sigma_1, \max(0, w_n)).$$

- for $(2i_1, 2i_2, i_3)$

$$\begin{aligned}
w1_{2i_1,2i_2,i_3} &= \frac{\widetilde{m}^1_{2i_1,2i_2,i_3} + \widetilde{m}^2_{2i_1,2i_2,i_3} \cdot w1_{2i_1,2i_2-1,i_3} + \widetilde{m}^4_{2i_1,2i_2,i_3} \cdot w1_{2i_1-1,2i_2,i_3}}{\widetilde{m}^5_{2i_1,2i_2,i_3}}, \\
w2_{2i_1,2i_2,i_3} &= \frac{\widetilde{m}^3_{2i_1,2i_2,i_3} + \widetilde{m}^2_{2i_1,2i_2,i_3} \cdot w2_{2i_1,2i_2-1,i_3} + \widetilde{m}^6_{2i_1,2i_2,i_3} \cdot w1_{2i_1+1,2i_2,i_3}}{\widetilde{m}^5_{2i_1,2i_2,i_3}}, \\
w3_{2i_1,2i_2,i_3} &= \frac{\widetilde{m}^7_{2i_1,2i_2,i_3} + \widetilde{m}^4_{2i_1,2i_2,i_3} \cdot w3_{2i_1-1,2i_2,i_3} + \widetilde{m}^8_{2i_1,2i_2,i_3} \cdot w1_{2i_1,2i_2+1,i_3}}{\widetilde{m}^5_{2i_1,2i_2,i_3}}, \\
w4_{2i_1,2i_2,i_3} &= \frac{\widetilde{m}^9_{2i_1,2i_2,i_3} + \widetilde{m}^6_{2i_1,2i_2,i_3} \cdot w3_{2i_1+1,2i_2,i_3} + \widetilde{m}^8_{2i_1,2i_2,i_3} \cdot w2_{2i_1,2i_2+1,i_3}}{\widetilde{m}^5_{2i_1,2i_2,i_3}}.
\end{aligned}$$

Restriction operator. In the definition of Galerkin coarse grid matrices, the restriction operator is most naturally defined as the transpose conjugate of the interpolation operator. It is, however, not always necessary to chose this option. In our numerical tests in [8], we found that the full weighting restriction in combination with either bilinear or matrix-dependent prolongation operator, resulted in a robust overall solution method for Helmholtz problems with irregular heterogenieties and strong contrast. With the choice, $I_h^H \neq (I_H^h)^*$, however, we cannot guarantee the symmetry of coarse grid matrices after Galerkin coarse grid process, eventhough M_h is symmetric.

The full weighting restriction operator is defined, in stencil notation, by:

$$I_h^H \triangleq \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^H . \tag{20}$$

4 Numerical results

In this section we present numerical results for several 3D Helmholtz model problems, as schematically shown in Figures 4 and 5. Since the numerical experiments are performed on a single processor machine, the problem sizes that can be resolved when setting the accuracy law "kh constant" are limited. A parallel implementation in 3D becomes inevitable and is on its way. The examples here are, however, representative to show the robustness of the shifted Laplacian preconditioner and it gives a first glance of its effectiveness in combination with Bi-CGSTAB to accelerate the numerical solution of the 3D Helmholtz equation with regular heterogeneities. There is no reason to believe that the solver will not perform well for problems with irregular heterogeneities.

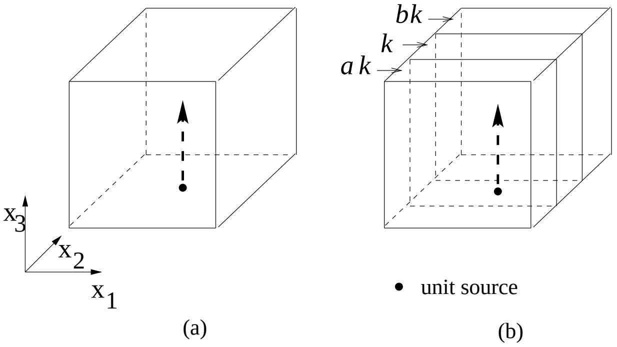

Figure 4: Three-dimensional problems: (a) constant k (b) three layers, k is varied in the x_2-direction.

4.1 3D constant wavenumber

We first consider a problem at constant wavenumber in $\Omega = (0,1)^3$. At the boundaries $\Gamma = \partial\Omega$ the first-order radiation conditions are prescribed. They are discretized by a one-sided finite difference scheme. A unit source is situated at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Starting with a zero initial guess, Bi-CGSTAB iterates until the residual at the j-th iteration is reduced by 7 orders of magnitude:

$$\|r^j\|_2 \leq 10^{-7}\|r^0\|_2. \tag{21}$$

The numerical performance for this first test case is presented in Table 1 for various wavenumbers k, obtained on grids with mesh size resolution h, satisfying $kh = 0.625$ ($\sim$ 10 grid points per wavelength). Compared to solving an equivalent problem in 2D, the 3D convergence results show a very similar trend as their 2D counterparts (or even faster 3D convergence is observed). In the constant wavenumber case, the use of bilinear or operator-dependent interpolation in multigrid does, in this example, not lead to a substantially different computational performance. The effect of different grid resolutions on the numerical performance is shown in Table 2 for $k = 10, 20$ and 30. Except for $k = 10$, for which the number of iterations increases slightly on the finer grids, the results indicate an almost h-independent convergence of preconditioned Bi-CGSTAB in combination with multigrid.

However, with the operator-dependent interpolation an increasing number of iterations is observed for fine grids. For example, for $k = 10$ on a 96^3 mesh (~ 60 grid points per wavelength), Bi-CGSTAB does not converge in 30 iterations. In this case, we find that the multigrid method defined with (x_1, x_2)-semicoarsening and linewise Jacobi, does not converge as a stand-alone solver for the preconditioner. The h-independent convergence is, however, again recovered for high wavenumbers. Further investigation of this phenomenon is needed in the near future.

Table 1: 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, $kh = 0.625$

	BI		MD	
k	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

Table 2: Number of iterations of preconditioned Bi-CGSTAB to reach convergence for the constant wavenumber Helmholtz equation, solved for different grid resolutions. The "–" means that the computation is not performed because $kh > 0.625$

	BI					MD				
	Grid:					Grid:				
k	16^3	32^3	48^3	64^3	96^3	16^3	32^3	48^3	64^3	96^3
10	9	9	10	11	18	9	10	15	16	>30
20	–	13	13	12	14	–	13	13	13	19
30	–	–	17	16	17	–	–	17	17	17

4.2 3D three-layer problems

A second problem considered is a so-called three-layer problem. The medium is divided into three parts, where the wavenumber varies in Ω according to

$$k = \begin{cases} ak_{\text{ref}} & 0 \leq x_2 < 1/3, \\ k_{\text{ref}} & 1/3 \leq x_2 < 2/3, \\ bk_{\text{ref}} & 2/3 \leq x_2 \leq 1. \end{cases} \tag{22}$$

Constants a and b in (22) are prescribed in the tables below.

The problem is presented in Figure 4b. A source is located at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, 0)$. With this model problem we investigate the influence of different semicoarsening strategies on the convergence of the preconditioned Bi-CGSTAB solver in the presence of a simple heterogeneity. We first consider the case where the coarsening is done in the direction of the strong variation in k, i.e. we apply (x_1, x_2)-semicoarsening. We prescribe again a zero initial guess and terminate the iterations when criterion (21) is satisfied.

Table 3 presents the convergence results for two pairs of numbers (a, b), that determine the physical contrast in the media (22). For the (a, b) values prescribed in Table 3 we observe a significant effect of an increasing contrast on the numerical performance. The number of iterations increases almost linearly with respect to k_{ref}. Here, the use of the operator-dependent interpolation is found to be slightly more effective than bilinear interpolation. For small wavenumbers the bilinear interpolation outperforms the operator-dependent interpolation.

Table 3: Bi-CGSTAB iteration to reach convergence for three layers problems with (x_1, x_2)-semicoarsening, $k_{\text{ref}}h = 0.625$, BI is bilinear, MD matrix-dependent interpolation

k_{ref}	$(a,b) = (1.2, 1.5)$				$(a,b) = (1.2, 2.0)$			
	Iter		Time(s)		Iter		Time(s)	
	BI	MD	BI	MD	BI	MD	BI	MD
10	9	12	0.71	0.81	14	16	0.98	0.99
20	18	19	8.47	9.02	24	29	12.07	12.89
30	30	29	42.50	41.91	36	43	57.07	53.33
40	36	33	114.44	107.68	49	56	173.50	172.07
50	49	40	261.18	218.40	65	68	382.73	381.49
60	51	48	470.62	449.48	78	75	736.10	713.43

We also apply (x_1, x_3)-semicoarsening, in order to see the effect of different coarsening strategies on the numerical convergence. This means that the direction with variation in k (i.e. x_2) is now uncoarsened. The convergence results are shown in Table 4. Compared with the convergence results in Table 3, we see that the results from the two semicoarsening strategies are more or less identical. This implies that the semicoarsening directions can be chosen independently of the direction of contrast. Similar convergence results as the (x_1, x_2)-semicoarsening case are also obtained for variation in the interpolation operator.

Table 4: Bi-CGSTAB iteration to reach convergence for three layers problems with (x_1, x_3)-semicoarsening, $k_{\text{ref}}h = 0.625$

$(a, b) =$	$(1.2, 1.5)$		$(1.5, 2.0)$	
k_{ref}	BI	MD	BI	MD
10	11	12	13	14
20	18	18	23	25
30	30	28	36	35
40	36	34	50	49
50	49	40	65	62
60	51	49	79	68

The numerical tests in Tables 3 and 4 are obtained on grids whose resolution is based on criterion $k_{\text{ref}}h \leq 0.625$. This is related to the minimal number of grid points per wavelength for a 7-point $O(h^2)$-stencil. However, on the top and bottom layers of the three layer problem the grid resolution drops below this criterion. It is therefore safest to prescribe the grid resolution based on the largest wavenumber in the medium. For example, in the case of $(a, b) = (1.2, 1.5)$ and $k = 20$, $k_{\max} = 30$. Hence, instead of using $h^{-1} = 32$ we now require $h^{-1} = 48$. Table 5 then displays results with these finer meshes, for $k_{\text{ref}} = 10, 20$ and 30. We use the operator-dependent interpolation and observe a convergence dependence on h only for $k_{\text{ref}} = 10$. This may be due to the fact that the preconditioner is not accurately inverted by multigrid for the lower wavenumbers. The h-dependence becomes much less significant as the wavenumber increases. The convergence is then asymptotically independent of h.

Table 5: Number of Bi-CGSTAB iterations to reach convergence on finer grids for a three layers problem with (x_1, x_2)-semicoarsening. The "–" means that the computation is not performed because $k_{\text{ref}}h > 0.625$

	$(a, b) = (1.2, 1.5)$					
k_{ref}	16^3	32^3	48^3	64^3	80^3	96^3
10	12	12	11	14	19	35
20	–	19	16	23	14	16
30	–	–	29	23	20	20

4.3 The 3D wedge problem

A third and final 3D example is the so-called 3D wedge problem, depicted in Figure 5. A unit source is located at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, 0)$. For multigrid iteration, the bilinear (BI) and operator-dependent interpolation (MD) are compared. The restriction operator is the full weighting (FW) operator. The convergence results are presented in Table 6 for various values of k_{ref}.

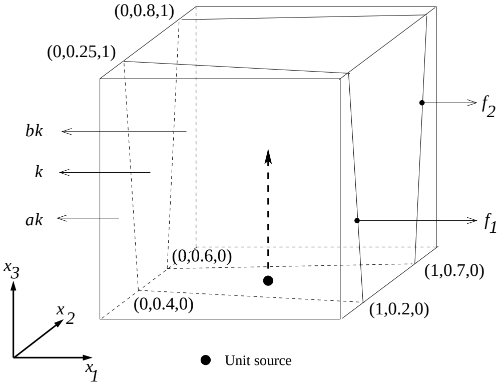

Figure 5: Wedge problem: $f_a(x_1, x_2, x_3) = 0.5x_1 + 2.5x_2 + 0.375x_3 - 1 = 0$, $f_b(x_1, x_2, x_3) = -\frac{1}{6}x_1 + \frac{5}{3}x_2 - \frac{1}{3}x_3 - 1 = 0$

For this problem, similar convergence results as for the previous problems are observed. For high wavenumbers the operator-dependent interpolation is superior to the bilinear interpolation.

Table 6: Bi-CGSTAB iteration to reach convergence for a three-dimensional wedge problem with (x_1, x_2)-semicoarsening, $k_{\text{ref}}h = 0.625$

	$(a,b) = (1.2, 1.5)$				$(a,b) = (1.2, 2.0)$			
	Iter		Time(s)		Iter		Time(s)	
k_{ref}	BI	MD	BI	MD	BI	MD	BI	MD
10	11	12	0.80	0.86	14	14	0.96	0.97
20	17	18	8.83	9.06	27	28	13.09	13.76
30	25	23	40.03	37.89	37	36	56.68	55.94
40	31	29	111.27	106.27	53	50	181.05	175.53
50	40	38	258.67	249.64	66	62	413.04	392.29
60	47	43	510.92	474.95	79	76	831.94	808.33

These results compare very well with the results obtained for the three layer problem. For reasonably high wavenumbers we obtain a robust and efficient convergence with the 3D shifted Laplacian preconditioner.

5 Conclusion

In this paper we have described an iterative method for solving 3D Helmholtz problems with heterogeneities at possibly at high wavenumbers. The method is based on Bi-CGSTAB, preconditioned by a complex-shifted Laplacian operator. The latter operator is approximately inverted by a special multigrid variant with (x_1, x_2)-semicoarsening. Numerical results have been presented for several 3D problems, which indicate the robustness and effectiveness of the method.

In this paper, convergence results of a sequential algorithm on one processor have been shown. Therefore, the 3D problems presented are not yet realistic real-life problems. The method proposed is, however, easily parallelizable, as all components in the current implementation are well parallelizable: e.g., the matrix-vector products, inner products, and line Jacobi for smoothing in the multigrid algorithm. In 2D, the parallel implementation is finalized. Following that parallel implementation, we expect that a similar numerical and parallel performance also in three dimensions.

References

[1] 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 (1990), pp. 323–352

[2] A. Bourgeois, M. Bourget, P. Lailly, M. Poulet, P. Ricarte, R. Versteeg, Marmousi, model and data, In *Marmousi Experience*, EAEG, 1991, pp. 5–16.

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

[4] 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 (2001), pp. 1291-1315.

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

[6] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, *Appl. Numer. Math.*, 50 (2004), pp. 409–425.

[7] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, *Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation*, Appl. Numer. Math. to appear.

[8] Y.A. Erlangga, C.W. Oosterlee, C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput. to appear.

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

[10] S. Kim, S. Kim, Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media, *SIAM J. Sci. Computa.* 24 (2002), pp. 684–701.

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

[12] W.A. Mulder, A new multigrid approach to convection problems, *J. Comput. Phys.* 83 (1989), pp. 303–323.

[13] C.W. Oosterlee, The convergence of parallel multiblock multigrid methods, *App. Numer. Math.* 19 (1995), pp. 115–128.

[14] Y. Saad, *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, 2003

[15] C.-A. Thole, U. Trottenberg, Basic smoothing procedures for the multigrid treatment of elliptic 3D operators, *Appl. Math. Comput.*, 19 (1986), pp. 333–345.

[16] U. Trottenberg, C.W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[17] R.S. Varga, *Matrix Iterative Analysis*, Prentice-Hall Inc., New Jersey, 1962

[18] 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 (1992), pp. 631-645.

[19] T. Washio, C.W. Oosterlee, Flexible multiple semicoarsening for three-dimensional singularly perturbed problems, *SIAM J. Sci. Comput.* 19 (1998), pp. 1646–1666.

[20] P. M. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.*, 33 (1990), pp. 1–27.