ACCURACY PROPERTIES OF THE MULTIGRID ALGORITHM FOR HELMHOLTZ EQUATIONS

IRENE LIVSHITS* AND ACHI BRANDT†

Abstract. Helmholtz equations with their highly oscillatory solutions play an important role in physics and engineering. These equations present the main computational difficulties typical to acoustics, electromagnetic and other wave problems. They are often accompanied by radiation boundary conditions and are considered on infinite domains. Solving them numerically using standard procedures, including multigrid, is too expensive. The wave-ray multigrid algorithm efficiently solves the Helmholtz equations and naturally incorporates the radiation boundary conditions. Important accuracy properties of the wave-ray solver are discussed in this paper.

Using various mode analyses, we show that, with the right choice of parameters, this algorithm can obtain an approximation to the differential solution with accuracy that equals the accuracy of the target grid discretization. Moreover, the boundary conditions can be introduced with any desired accuracy. Our theoretical conclusions are confirmed by numerical experiments.

Key words. Helmholtz equations, accuracy, multigrid, wave-ray

AMS subject classifications. 65N15, 65N55, 35C99

1. Introduction.

These studies are aimed at constructing fast and accurate numerical algorithms for the Helmholtz equation

$$\Delta u(x,y) + k^2 u(x,y) = f(x,y), \qquad (x,y) \in \mathbf{R}^2, \tag{1.1}$$

with the radiation boundary conditions. Our goal is to approximate the solution of (1.1) on a finite domain Ω of diameter d. We primarily consider domains with d that satisfies $kd \gg 1$. This makes the problem highly indefinite, and hard to solve.

Although the solution process we describe can solve more general problems, our analysis here is focused on the model problem (1.1) when the wave number k is constant, $f(x,y)$ has compact support $\Omega_f \subset \mathbf{R}^2$ and employed is the second-order discretization of (1.1)

$$(L^h u^h)_{i,j} = \frac{u^h_{i-1,j} + u^h_{i+1,j} + u^h_{i,j-1} + u^h_{i,j+1} - 4u^h_{i,j}}{h^2} + k^2 u^h_{i,j} = f_{i,j}, \tag{1.2}$$

where $u^h_{i,j}$ approximates $u(x_i,y_j)$ and $f_{i,j} \approx f(x_i,y_j)$, $i = 0,\dots,N_x$, $j = 0,\dots,N_y$, $\Omega_f \subset \Omega$, and Ω is much wider than Ω_f. For simplicity, Ω is chosen to be the square:

$$\Omega = \{(x,y), \qquad -d/2 \le x \le d/2, \quad -d/2 \le y \le d/2\}, \tag{1.3}$$

and hence $N_x = N_y = N$, $x_i = -d/2 + ih$, and $y_j = -d/2 + jh$.

For $kd \gg 1$, the cost of solving (1.2) can be overwhelming. To provide an accurate approximation to the differential solutions, discretization of (1.1) must employ many, actually more than $O(k^3 d^3)$, grid points. Moreover, to converge to the numerical solution, a conventional solver should employ many, actually more than $O(k^6 d^6)$, sweeps of a slow (e.g., Kaczmarz) relaxation scheme, since faster schemes diverge for some components. So the total work grows at least proportionally to $O(k^9 d^9)$.

[*] Department of Mathematical Sciences, Ball State University, Muncie, Indiana, 47306, USA (ilivshits@bsu.edu).

[†] Department of Mathematics & Computer Sceince, Weizmann Institute of Science, Rehovot 76100, Israel. The work of this author was supported in part by Israel Science Foundation grant 295/01.

The above provides motivation to use multigrid methods ([8], [10], [14], [15]). However, standard multigrid procedures are not efficient for the highly indefinite Helmholtz equations. This is because there is a range of Fourier error components $e^{i(\omega_1 x+\omega_2 y)}$ that the standard multigrid cycle cannot reduce efficiently. Those are the components with $\omega_1^2+\omega_2^2 \approx k^2$; we call them *characteristic*. On fine grids their amplitudes are much larger than the amplitudes of the corresponding normalized residuals, making them almost invisible to any local processing. Hence, they are only very slowly reduced by relaxation. On the other hand, unlike regular smooth components, these have poor approximations on coarse grids, which, therefore, cannot serve to correct the finer-grid solutions.

As a result, the characteristic components need a special multigrid treatment, and this is the basis of the wave-ray algorithm. Its fast convergence was the subject of [5] . There it was shown that the algorithm has high multigrid efficiency at a cost comparable to the cost of solving Poisson equation with the standard cycle.

In the present paper we study the accuracy aspects of the wave-ray algorithm. The goal of this algorithm, employing the second-order discretization, is to provide accurate $O(h^2)$ approximation to the differential solution while costing only $O(h^{-2})$ computer operations. This is not as straightforward as for definite elliptic systems, both due to the phase errors and because the radiation boundary conditions need to be brought accurately from the coarsest level of the algorithm to the target level.

Obviously, by choosing a smaller finest grid mesh-size h, we should expect a more accurate solution. Moreover, since the second-order scheme is employed in (1.2), the accuracy should improve as $O(h^2)$. In the wave-ray algorithm, however, the following scenario is possible: after choosing a smaller h, the approximation error remains almost unchanged. In this paper we discuss different factors and the error they cause that lead to such *accuracy inefficiency*. We then suggest the ways to deal with them. The ultimate goal is to minimize the effect of these errors on the total error to the level of becoming negligible compared to the finest-grid discretization error, which then regains its principal error term status (as in traditional multigrid solvers).

This paper will not address a difficult issue of pollution that appears when the Helmholtz equation with the wave number k must be discretized on a grid with a given mesh-size h such that kh is large. Then even the finest discrete Helmholtz equation is not accurate enough: its solution will be a poor approximation to the differential solution. Pollution and how to deal with it (when possible) is in detail discussed in [3], [9], [13], et al.

The outline of the paper is as follows: We start with a description of the algorithm. Its efficiency is achieved by using two types of solution representation: wave and ray. (This gave the algorithm its name `wave-ray`. The grids on which the wave and the ray equations are discretized are called the *wave grids* and the *ray grids*, respectively.) This technique can cause inaccuracies, unless the algorithmic parameters are properly chosen.

In section 2 we discuss discretization errors and their influence on the choice of the finest mesh-sizes in the wave and in the ray discretization. Unlike in standard multigrid procedures, coarse ray grids often (away from Ω_f) provide more accurate approximation to the differential solution than the finer ray or wave representations.

In section 3 our subject is the approximation of the differential problem on an infinite domain by a solver which employs finite domains on many levels. In particular, we deal with the question of how to impose the radiation boundary conditions.

We conclude the paper with numerical results and discuss their agreement with our theoretical conclusions.

1.1. Wave – Ray Algorithm for Helmholtz Equations. First, we define the circle of *principal* Fourier components to be the subset of characteristics components with frequencies $\omega_1^2 + \omega_2^2 = k^2$. They satisfy (1.1) with $f \equiv 0$ and have a significant role in the algorithm. Among all principal components we consider a finite set of *lattice* principal components uniformly spaced on the principal circle. We denote them (k_1^ℓ, k_2^ℓ), $\ell = 1, \ldots, L$; L is the number of lattice components. We use the word `lattice` to describe the discretization of the principal circle and to distinguish it from `grids` in the physical space. For the principal lattice component $e^{i(k_1^\ell x + k_2^\ell y)}$, we define a rotated Cartesian coordinate system (ξ^ℓ, η^ℓ) with ξ^ℓ being parallel to (k_1^ℓ, k_2^ℓ), the *propagation* direction.

Each characteristic component can be represented as a product of a lattice principal component and a smooth envelope function. Hence, a general representation of the error that cannot be efficiently reduced by a regular multigrid cycle can be given by

$$v(x,y) = \sum_{\ell=1}^{L} \hat{v}^\ell(x,y) e^{i(k_1^\ell x + k_2^\ell y)}, \tag{1.4}$$

where $\hat{v}^\ell(x,y)$ are smooth functions. Note that the functions $\hat{v}^\ell(x,y)$ are not uniquely defined: Fourier components of v with frequencies between two lattice points can be assigned to these points with arbitrary weights, yet still yielding the desired smoothness. Considered in the corresponding rotated coordinates, the scale of smoothness of $\hat{v}^\ell(x,y)$ is $O(L^2/k)$ in the propagation direction ξ^ℓ, and $O(L/k)$ in the η^ℓ direction. (In other words, the function $v(x,y)$ essentially consists of Fourier components with frequencies in a ring of width $O(k/L^2)$ around the principal circle.)

We call the functions $\hat{v}^\ell(x,y)$ the *ray functions*; the equations we derive for them – the *ray equations*. The residual ray equation for $\hat{v}^\ell(x,y)$ is

$$\Delta \hat{v}^\ell(x,y) + 2ik_1^\ell \frac{\partial}{\partial x}\hat{v}^\ell(x,y) + 2ik_2^\ell \frac{\partial}{\partial y}\hat{v}^\ell(x,y) = \hat{r}^\ell(x,y). \tag{1.5}$$

This results directly from the Helmholtz equation, with $\hat{r}^\ell(x,y)$ being the smooth weights in the ray representation of the wave residual function:

$$Lv(x,y) = r(x,y) = \sum_{\ell=1}^{L} \hat{r}^\ell(x,y) e^{i(k_1^\ell x + k_2^\ell y)}. \tag{1.6}$$

Each principal component $e^{i(k_1^\ell x + k_2^\ell y)}$ in its rotated coordinates has the form $e^{ik\xi^\ell}$, so Eqs. (1.5) can be simplified there to

$$\Delta \hat{v}^\ell(\xi^\ell,\eta^\ell) + 2ik\frac{\partial}{\partial_{\xi^\ell}}\hat{v}^\ell(\xi^\ell,\eta^\ell) = \hat{r}^\ell(\xi^\ell,\eta^\ell), \quad (\xi^\ell,\eta^\ell) \in \hat{\Omega}^\ell. \tag{1.7}$$

Since the problem is defined over the entire space $(x,y) \in \mathbf{R^2}$, we have to define a large enough finite computational domain $\hat{\Omega}^\ell$ over which $\hat{v}^\ell$ will be computed. For convenience we choose $\hat{\Omega}^\ell$ to be rectangular

$$\hat{\Omega}^\ell = \{(\xi^\ell,\eta^\ell) : \quad \xi_0^\ell \le \xi^\ell \le \xi_1^\ell, \quad -\eta_0^\ell \le \eta^\ell \le \eta_0^\ell\}, \tag{1.8}$$

where appropriate values for ξ_0^ℓ, ξ_1^ℓ and η_0^ℓ will be a subject of the analysis below. We call the set of points (ξ^ℓ, η^ℓ) on the lines $\xi^\ell = \xi_0^\ell$, $\xi^\ell = \xi_1^\ell$ and $\eta^\ell = \pm\eta_0^\ell$ the `entrance`, the `exit` and the `side` boundaries, respectively.

(Each equation (1.7) in the rotated coordinates allows the most aggressive discretization ratio: by the factor of four (instead of two) in the propagation direction. The use of the standard Cartesian (x, y) coordinates is also possible. Moreover, it too allows a special coarsening at least for some components: for instance, for $L = 8$, a half of the components propagates in either x or y direction, and, therefore, for them the same coarsening can be applied. For a higher value of L, for most of the components their propagation direction is closer to either x or y, and therefore some stronger coarsening can be applied there, with the factors of three or four, depending on the component in question and the accuracy requirements.)

The error for which the usual multigrid cycle is not effective can efficiently be reduced by a multigrid procedure which on coarse grids represents the smooth ray functions $\hat{v}^\ell(x, y)$ rather than the non-smooth function $v(x, y)$ itself. The coarser the grid the larger the scale of smoothness of $\hat{v}^\ell(x, y)$ represented on it. This requires a larger L, i.e., a finer lattice of principal components.

Let the ray levels be enumerated by $n = 1, \dots, N$ so that $n = N$ is the level with the coarsest grid and the finest lattice. For the *n-th* level, the principal lattice is then composed of the frequencies

$$(k_1^{n,\ell}, k_2^{n,\ell}) = (k \cos\theta^{n,\ell}, k \sin\theta^{n,\ell}), \qquad \ell = 1, \dots, L^n, \tag{1.9}$$

where $\theta^{n,\ell} = 2\pi\ell/L^n$; the corresponding ray functions are denoted $\hat{v}^{n,\ell}$. At each level n, each ray function $\hat{v}^{n,\ell}(x, y)$ in its rotated coordinates $(\xi^{n,\ell}, \eta^{n,\ell})$ is discretized on a uniform grid with mesh-sizes $h_\xi^n = O((L^n)^2/k)$ and $h_\eta^n = O(L^n/k)$. Our actual choice of parameters, based on a careful mode analysis, has been:

$$L^n = 2^{n+2}, \qquad h_\xi^n = C(L^n)^2/(32k), \qquad h_\eta^n = C(L^n)/(8k), \tag{1.10}$$

where typically $1.25 \le C \le 2.5$.

The wave-ray cycle is composed of two parts. The first one, the wave sub-cycle, is a standard multigrid cycle applied to (1.2). It proceeds from the finest wave grid to the coarsest; there is actually no need to go to grids with mesh-size coarser than $4/k$ (this is because for such h the system (1.2) becomes so strongly diagonal dominant that the Gauss-Seidel relaxation converges very fast). Few Kaczmarz or Gauss-Seidel relaxation sweeps are performed on each wave level except for the ones with mesh-sizes $h \approx 2/k$. Relaxation on the coarsest grid reduces the smoothest part of the error, while relaxation on the fine levels reduce its high-frequency components. The wave subcycle does not change the characteristics components effectively, and, consequently, it leaves an error of the type (1.4). Also it does not treat the radiation boundary conditions, which are associated only with principal components and, therefore, ray representation. In this part of the algorithm the sole role of the coarse grids is to produce smooth non-characteristic corrections to the finer grid approximations in the interior of the finer grid computational domains.

The second part of the algorithm starts with a ray sub-cycle, which is needed to reduce characteristic errors and to impose the radiation boundary conditions. First, the wave residual $r(x, y)$ is calculated on some sufficiently fine wave level with mesh-size $h = O(d^{-1/2}k^{-3/2})$ – still fine enough to produce small phase errors throughout the computational domain. From this grid, the wave residual is transferred directly to the wave grid with $h \approx 1/k$. There, for each $1 \leq \ell \leq L^1$ this wave residual is multiplied by $e^{-i(k_1^\ell x + k_2^\ell y)}$ resulting in

$$r(x,y)e^{-i(k_1^\ell x + k_2^\ell y)} = \hat{r}^\ell(x,y) + \sum_{m \neq \ell} \hat{r}^m(x,y) e^{i(\Delta k_1^m x + \Delta k_2^m y)}, \tag{1.11}$$

where $k_j^m = k_j^{1,m}$, $\Delta k_j^m = k_j^m - k_j^\ell$, $j = 1, 2$. Notice that the first term on the right-hand-side of (1.11) is very smooth compared to each term of the sum. The next step is to approximate each $\hat{r}^\ell$ in the rotated coordinates $(\xi^\ell, \eta^\ell) = (\xi^{1,\ell}, \eta^{1,\ell})$. For this purpose a procedure called *separation* is applied, denoted

$$\hat{r}^\ell = \hat{r}^{1,\ell} = W_{1,\ell}(r(x,y)e^{-i(k_1^\ell x + k_2^\ell y)}),\ \ell = 1, \ldots, L^1. \tag{1.12}$$

Here $W_{1,\ell}$ is a product of two weighting operators that accurately transfer smooth components from the wave grid with mesh-size $h \approx 1/k$ first to an intermediate ray grid with mesh-size $(h_\xi, h_\eta) \approx (2/k, 2/k)$, and then to the ray grid $(\xi^{1,\ell}, \eta^{1,\ell})$ with mesh-size $(h_\xi^1, h_\eta^1) \approx (4/k, 2/k)$. The operators are designed to almost annihilate components that are highly oscillatory on the current scale.

It is important to note here that the entire solver employs the Full Approximation Scheme (FAS, see e.g. [4]). This means that, although the equations at each level are basically equations for the correction $\hat{v}^{n,\ell}$, driven by the next-finer-level residuals, as in (1.7), they are modified (by adding an appropriate term to the right-hand-side) so that their solution always represents the full approximation $\hat{u}^{n,\ell} = \hat{v}^{n,\ell} + \tilde{u}^{n,\ell}$, where $\tilde{u}^{n,\ell}$ is an approximation to $\hat{u}^{n,\ell}$. The algorithm employs the value of $\hat{u}^{n,\ell}$ at the end of the previous cycle, as the first approximation to $\tilde{u}^{n,\ell}$. Such approximation satisfies the radiation boundary condition, and it does not require an update from the wave part of the cycle since characteristic components are not changed there. At the first cycle $\tilde{u}^{n,\ell}$ satisfies the radiation boundary conditions at the entrance and side boundaries, and it is zero elsewhere.

The FAS right-hand-side is calculated by

$$\hat{f}^{n,\ell} = \hat{r}^{n,\ell} + \hat{L}^n \tilde{u}^{n,\ell}. \tag{1.13}$$

The FAS is chosen to enable the imposition of the radiation boundary conditions at the coarsest ray level.

As the model boundary conditions, we have chosen the Dirichlet radiation boundary conditions that can be formulated as follows: for each direction, the amplitude of the ray propagating in this direction from infinity into Ω is given.

After the discrete ray equations are relaxed on the finest ray level ($n = 1$), and the new solution approximations are calculated, the ray sub-cycle proceeds (if necessary) to increasingly coarser ray levels ($n = 2, \ldots, N$). This includes recursive derivation of ray equations on increasingly coarser ray grids, having increasingly finer lattices (larger $L = L^n$). The equations on each grid have the ray form (1.7), discretized; they are the result of a separation process applied to the residuals of the next finer neighboring grids. Namely, to calculate $\hat{r}^{n,\ell}$ for odd ℓ, the corresponding ray residual function from the finer grid is used:

$$\hat{r}^{n,\ell} = W_{n-1,(\ell+1)/2}^{n,\ell} \hat{r}^{n-1,(\ell+1)/2}, \tag{1.14}$$

where here and below $W^{n,\ell}_{n-1,j}$ consists again of a product of two weighting operators, acting from the grid $(\xi^{n-1,j},\eta^{n-1,j})$ to the target $(\xi^{n,\ell},\eta^{n,\ell})$ designed to approximate the smooth components and annihilate high-frequencies on the considered scale.

For even ℓ, $\hat{r}^{n,\ell}$ is calculated from two finer grid residuals:

$$\begin{aligned}\hat{r}^{n,\ell} = W^{n,\ell}_{n-1,\ell/2}&\left(\hat{r}^{n-1,\ell/2}e^{i((k_1^{n-1,\ell/2}-k_1^{n,\ell})x+(k_2^{n-1,\ell/2}-k_2^{n,\ell})y)}\right)+\\ &W^{n,\ell}_{n-1,\ell/2+1}\left(\hat{r}^{n-1,\ell/2+1}e^{i((k_1^{n-1,\ell/2+1}-k_1^{n,\ell})x+(k_2^{n-1,\ell/2+1}-k_2^{n,\ell})y)}\right).\end{aligned}\tag{1.15}$$

The lattice count should be taken modulo L, i.e., for $\ell = L^n$ take $\ell/2+1 = L^{n-1}+1 \equiv 1(\mathrm{mod}\, L^{n-1})$.

On the coarsest grid the radiation boundary conditions are imposed, facilitated by the nearly pure ray representation at that grid. The ray equations are solved there fast by a line Gauss-Seidel relaxation. Then the boundary conditions along with corrections are interpolated through the intermediate ray grids (without any relaxation) to the finest ray grid.

Interpolation from a coarser ray grid (with a finer lattice) to the next finer ray grid (with coarser lattice) is performed differently on the boundary of the finer grid domain and in its interior. On the boundary $\hat{\Gamma}^{n,\ell}$ of the computational domain $\hat{\Omega}^{n,\ell}$ the solution values are interpolated directly from the three neighboring coarser grids:

$$\begin{aligned}\hat{u}^{n,\ell} = \frac{1}{2}I^{n,\ell}_{n+1,2\ell-2}&\left(\hat{u}^{n+1,2\ell-2}e^{i((k_1^{n+1,2\ell-2}-k_1^{n+1,2\ell-1})x+(k_2^{n+1,2\ell-2}-k_2^{n+1,2\ell-1})y)}\right)+\\ I^{n,\ell}_{n+1,2\ell-1}\hat{u}^{n+1,2\ell-1}+\frac{1}{2}I^{n,\ell}_{n+1,2\ell}&\left(\hat{u}^{n+1,2\ell}e^{i((k_1^{n+1,2\ell+1}-k_1^{n+1,2\ell-1})x+(k_2^{n+1,2\ell}-k_2^{n+1,2\ell-1})y)}\right),\end{aligned}$$

where the interpolation operator $I^{n,\ell}_{n+1,j}$ interpolates from the coarser grid $\hat{\Omega}^{n+1,j}$ to the finer grid $\hat{\Omega}^{n,l}$. This interpolation of the boundary conditions is the reason for using FAS, since it requires to approximate solutions rather than corrections on all grids.

In the interior, the solution is only *corrected* by the coarse-grid approximations:

$$\begin{aligned}\hat{u}^{n,\ell} = \hat{u}^{n,\ell}+\frac{1}{2}I^{n,\ell}_{n+1,2\ell-2}&\left(\hat{v}^{n+1,2\ell-2}e^{i((k_1^{n+1,2\ell-2}-k_1^{n+1,2\ell-1})x+(k_2^{n+1,2\ell-2}-k_2^{n+1,2\ell-1})y)}\right)+\\ I^{n,\ell}_{n+1,2\ell}\hat{v}^{n+1,2\ell}+\frac{1}{2}I^{n,\ell}_{n+1,2\ell}&\left(\hat{v}^{n+1,2\ell}e^{i((k_1^{n+1,2\ell}-k_1^{n+1,2\ell-1})x+(k_2^{n+1,2\ell}-k_2^{n+1,2\ell-1})y)}\right).\end{aligned}$$

Here corrections $\hat{v}^{n+1,j}$ are given by $\hat{v}^{n+1,j} = \hat{u}^{n+1,j} - \tilde{u}^{n+1,j}$, where $\tilde{u}^{n+1,j}$ again is the value of $\hat{u}^{n+1,j}$ at the end of the previous cycle.

No relaxation is performed on the way to the finest ray grid. From there, the ray solutions (for the boundary) and the ray corrections (for the interior) are interpolated to the wave grid with $h \approx 1/k$, where they are summed as in (1.4) and used to correct the previous wave solution. Similar to the ray-to-ray interpolation, the wave correction (at the interior) and the solution values (at the boundary) are interpolated to the finer wave grids, with a small number of relaxation sweeps performed on each grid.

See the Appendix below for a formal description of the entire algorithm.

2. Discretization Errors. In this section we discuss how to achieve an accurate approximation to the differential solutions in the interior of the finest computational domain. We examine the point-wise approximation to the differential solutions, analyze the influence of discretization errors, and conclude that the main source of inaccuracy are the phase errors, coming from both the wave and the ray representations. In our studies we measure only the errors eventually resulting in the interior of the target finest wave grid.

2.1. Phase Errors. The *phase* error is the difference in phases of the discrete and differential waves, accumulated along their propagation path due to their different wavelengths. This error can become very significant when the solution propagates along many wavelengths.

A peculiarity of our algorithm is that, unlike common discretization schemes, in the regions where the equation is homogeneous, the coarse ray grids provide more accurate approximations to the differential solution than the finer ray, and often are better than the wave grids. The ultimate role of the fine wave grids is to resolve the right-hand-side f at scales smaller than or just comparable to the wavelength $2\pi/k$. The details are discussed in this section.

2.1.1. Wave Phase Error. We define a *wave discrete principal component* $e^{i(k_1^h x+k_2^h y)}$ as any one that satisfies the discrete homogeneous equations (1.2),

$$L^h e^{i(k_1^h x+k_2^h y)} = 0. \tag{2.1}$$

By Taylor expansion in (1.2) it can be shown that, for $kh \ll 1$, the value of the corresponding discrete wave number $|k^h| = \sqrt{(k_1^h)^2+(k_2^h)^2}$ can be approximated by

$$|k^h| \approx k\left(1+\frac{k^2h^2}{\gamma}\right), \qquad 24 \le \gamma \le 48, \tag{2.2}$$

where the exact value of γ depends on the ratio $k_1^h : k_2^h$. The accumulated value of the **relative** phase error (**the phase error over the period** $2\pi/k$) propagating through the $kd/2\pi$ wavelengths of Ω can therefore be estimated as

$$E(kd,kh) \approx kd\frac{k^2h^2}{2\pi\gamma}. \tag{2.3}$$

To provide an accurate approximation to the differential solution on the target wave grid, the finest mesh-size h should be chosen so that $E(kd,kh)$ is small:

$$E(kd,kh) \ll 1. \tag{2.4}$$

2.1.2. Ray Phase Error. In standard multigrid procedures the accuracy of discretization on coarse grids influences the speed of convergence of the solver rather than the accuracy of the solution. In our algorithm the ray grids serve not only to correct the characteristic error components, but also to introduce the radiation boundary conditions, and to approximate the principal components invisible on the finest wave grids. Hence, the quality of the solution approximation on the ray grids needs to be addressed. (Since the discussion below is relevant for any ℓ^{th} ray function and ray operator on any n^{th} level, we omit the superscripts ℓ and n.)

We define a *ray discrete principal component* $e^{i(\theta_\xi\xi+\theta_\eta\eta)}$ as any one which satisfies

$$\hat{L}^h e^{i(\theta_\xi\xi+\theta_\eta\eta)} = 0,$$

where $\hat{L}^h$ is a second-order discretization of (1.7), given in the (ξ, η) coordinates by

$$\hat{L}^h \hat{v}_{i,j} = \frac{\hat{v}_{i-1/2,j-1} + \hat{v}_{i-1/2,j+1} - 2\hat{v}_{i-1/2,j}}{2h_\eta^2} + \frac{\hat{v}_{i+1/2,j-1} + \hat{v}_{i+1/2,j+1} - 2\hat{v}_{i+1/2,j}}{2h_\eta^2} +$$

$$\frac{\hat{v}_{i-3/2,j} - \hat{v}_{i-1/2,j} - \hat{v}_{i+1/2,j} + \hat{v}_{i+3/2,j}}{2h_\xi^2} + 2ik\frac{\hat{v}_{i+1/2,j} - \hat{v}_{i-1/2,j}}{h_\xi}. \tag{2.5}$$

On the scale of the employed grid, the term $2ikv_\xi$ is the principal term, and its most accurate (second-order) discretization is the main factor in our choice of the discrete scheme. Our operator provides an accurate short central discretization for $2ikv_\xi$, and also a second order accuracy for the remaining terms (it employs the Crank-Nicholson scheme for $v_{\eta\eta}$). (The standard 5-point stencil for the Laplace operator would come with the long central differences for the principal term leading to a weak diagonal term.) Discreization (2.5) also allows an efficient η-line relaxation (marching from entrance to exit, solving (2.5) simultaneously for all $\hat{v}_{i,j}$ with the same j, cf. Sec 3.2) on the ray grids. (Substitution of $e^{i(\theta_\xi\xi+\theta_\eta\eta)}$ into (2.5) gives

$$\frac{2}{h_\eta^2}\cos\left(\frac{\phi_\xi}{2}\right)\Big[\cos(\phi_\eta) - 1\Big] - \frac{4k}{h_\xi}\sin\left(\frac{\phi_\xi}{2}\right) + \frac{1}{h_\xi^2}\left[\cos\left(\frac{3\phi_\xi}{2}\right) - \cos\left(\frac{\phi_\xi}{2}\right)\right] = 0,$$

where $\phi_\xi = \theta_\xi h_\xi$ and $\phi_\eta = \theta_\eta h_\eta$. Note that the highest frequency $(\theta_\xi, \theta_\eta)$, which must be represented on each ray grid (so that any principal and characteristic component has an accurate representation on at least one of the L ray grids) is given by

$$\theta_\xi \approx k\frac{\pi^2}{2L^2}, \quad \theta_\eta \approx k\frac{\pi}{L}. \tag{2.6}$$

This is the frequency associated with the principal component at the mid-point between two neighboring lattice points. With (1.10), the arguments of sines and cosines are sufficiently small for (2.6) to be replaced by its approximation:

$$-\theta_\eta^2 + \frac{1}{8}\theta_\eta^2\phi_\xi^2 - 2k\theta_\xi + \frac{k}{12}\theta_\xi\phi_\xi^2 - \theta_\xi^2 \approx 0. \tag{2.7}$$

Using (2.7) we can approximate the size of a discrete ray wave number $\hat{k}_L = (\theta_\xi + k, \theta_\eta)$:

$$|\hat{k_L}|^2 = k^2 + 2k\theta_\xi + \theta_\xi^2 + \theta_\eta^2 \approx k^2\left(1 + \frac{1}{8k^2}\theta_\eta^2\phi_\xi^2 + \frac{1}{12k}\theta_\xi\phi_\xi^2\right), \tag{2.8}$$

or, by (1.10) and (2.6),

$$|\hat{k}_L| \approx k\left(1 + \frac{\beta}{L^2}\right), \tag{2.9}$$

with $\beta \approx 0.039C^2, 1.25 \leq C \leq 2.5$. The phase error arises from the propagation of principal components emanating from the radiation boundary conditions or from the right-hand-side source. For instance, at a distance $\hat{d}$ from the entering boundary, the phase error can be estimated as

$$E(k\hat{d}, L) \approx k\hat{d}\frac{\beta}{2\pi L^2}. \tag{2.10}$$

2.2. RHS Truncation Errors. The phase error is the main discretization error in approximating the homogeneous equations. To estimate those truncation errors which arise in approximating the effect of a non-homogeneous right-hand-side, consider a right-hand-side $f(x,y)$, a differential solution $u(x,y)$ and an approximate difference solution $u^h(x,y)$ in Ω of the form

$$f(x,y) = \int\limits_{\mathbf{t}} \hat{f}(\mathbf{t}) e^{i(t_1 x + t_2 y)} d\mathbf{t}, \quad u(x,y) = \int\limits_{\mathbf{t}} \hat{u}(\mathbf{t}) e^{i(t_1 x + t_2 y)} d\mathbf{t},$$

and

$$u^h(x,y) = \int\limits_{\mathbf{t}} \hat{u}^h(\mathbf{t}) e^{i(t_1 x + t_2 y)} d\mathbf{t}.$$

For any frequency $\mathbf{t} = (t_1, t_2), \ t = |\mathbf{t}|$, the following holds

$$\hat{u}(\mathbf{t}) = \frac{\hat{f}(\mathbf{t})}{k^2 - t^2} \tag{2.11}$$

and

$$\hat{u}^h(\mathbf{t}) \approx \frac{\hat{f}(\mathbf{t})}{k^2 - t^2(1 - 2t^2h^2/\gamma)}, \tag{2.12}$$

where γ is defined in (2.2). The difference between (2.11) and (2.12) is $O(t^2h^2)$, meaning that the finest wave mesh-size should be fine enough to resolve the highest-frequency components of f which one cares to approximate. More precisely, for $|t - k|/k \geq O(1)$, the relative error

$$E(\mathbf{t}) = \frac{|\hat{u}(\mathbf{t}) - \hat{u}^h(\mathbf{t})|}{|\hat{u}(\mathbf{t})|} \tag{2.13}$$

is small when t^2h^2 is small.

The more severe requirements on the mesh-size arise however from the characteristic components, where t is close to k. For $|t-k|/k = \delta \ll 1$, using (2.11) – (2.12), one can show that

$$E(\mathbf{t}) = \frac{k^2h^2(1 + O(\delta)}{\gamma\delta + k^2h^2(1 + O(\delta))} \approx \frac{k^2h^2}{\gamma\delta}. \tag{2.14}$$

For this truncation error to be small, the target mesh-size h should satisfy

$$\frac{k^2h^2}{\gamma\delta_{\min}} \ll 1, \tag{2.15}$$

where $\delta_{\min} = \min_{t=|\mathbf{t}|, \hat{u}(\mathbf{t}) \neq 0} \frac{|t-k|}{k}$. The problem (1.2) is considered on a computational domain of size d and a grid with mesh-size h. Only a finite number of Fourier components is distinguished on this grid. The minimal distance between such components, in the phase space, is $2\pi/d$. Therefore, $\delta_{min} = \frac{2\pi}{kd}$, and (2.15) can be replaced by

$$kd\frac{k^2h^2}{2\pi\gamma} \ll 1. \tag{2.16}$$

Note that requirement (2.16) is identical with the condition resulting from the wave phase error. Indeed,

$$E(t) = E(kd, kh)(1 + O(\delta)). \tag{2.17}$$

We have excluded from consideration the special case of resonance, when $\hat{f}/(k^2 - t^2)$ is not integrable, which would require a different formulation in the time domain.

3. Boundary Conditions. One of the possible reasons for slowness of multigrid solvers is inefficient introduction of boundary conditions. If done on a sufficiently fine grid, it takes many (expensive) iterations until the boundary information propagates from the boundaries into the domain. The remedy from such slowness is to use coarse grids. This, however, is not a good solution for the standard multigrid solvers applied to the Helmholtz equation since coarse wave grids are not suitable for representing accurate solutions. Instead, the wave-ray algorithm employs its ray grids to introduce the boundary conditions. These grids are very coarse and also very accurately resolve the ray functions.

In this paper we consider the Dirichlet radiation boundary conditions given by the amplitudes of the waves incoming to the computational domain from infinity. These boundary conditions are actually formulated as Dirichlet boundary conditions for the ray functions: at the coarsest ray level each entering ray is represented on the grid (or divided between two grids) with the closest propagation direction, on which its boundary values are indeed very smooth. Let us note that all exiting-only rays (rays which do not enter the computational domain but are originated in Ω_f) can actually exit only through the exit boundary. Hence on all other boundaries of each of the ray domains we can impose as boundary conditions the incoming rays (or zero, if no incoming rays are assumed). No boundary conditions are needed at the exit boundaries since the discrete equations (2.5) and the order in which we relax them (from entrance to exit) ensure that information propagation in the negative ξ direction is effectively prohibited.

The wave-ray approach can be also used for an efficient introduction of other RBCs: for instance, Sommerfeld-like boundary conditions (discussed for instance in [1], [2]). This can be done since near the boundaries (remote from Ω_f), any solution is purely of a ray nature and can be represented in the ray form of type (1.4). Then from the RBC conditions formulated as differential equations for a wave function, it is easy to find a set of differential equations that describe boundary conditions for its ray functions. This set can be separated into conditions for individual components or, at most, for a pair of components with opposite propagation direction. In this case, these two components should be approximated together, still very efficiently since considered on coarse grids. The transition to these *other* **boundary conditions has not yet been done by the authors.**

The radiation boundary conditions (RBC) can be introduced by imposing the values of the ray functions $\hat{v}^\ell$ in (1.7) at the entrance and the side boundaries on the coarsest ray level as Dirichlet boundary conditions. Since $kh_\xi \gg 1$ on all ray levels, the second derivative term $\hat{v}_{\xi\xi}$ in (1.7) is negligible on the scale (h_ξ, h_η), making (1.7) almost parabolic. Therefore, no condition is needed at the exit. There, instead of

(2.5), the algorithm employs strictly upstream discretization, and the solution values are treated as the interior rather than the boundary.

To guarantee an accurate introduction of the RBC, we have to avoid a possible influence of the interior solution on the boundary values, that can in its turn affect the solution in the domain of interest. First we discuss how an implementation of the wave-ray algorithm can pollute the boundary conditions, and how to introduce the RBC accurately – on finite domains and with a finite number of ray functions. The second question is how to transfer these conditions to the finer ray, and then wave, grids. In our algorithm the wave solution on the boundary is evaluated by direct interpolation from the coarser (eventually, ray) grids. It means that some interpolation error is necessarily introduced on the boundaries and affects the solution accuracy there and in the interior. However, we will show that this influence decreases fast as the error propagates inside the computational domain. We analyze the interpolation from both wave and ray domains.

3.1. Side Boundaries: Reflected Rays. When imposing the radiation boundary conditions at the coarsest ray level, at the entrance these conditions are naturally defined as Dirichlet boundary conditions without having any negative effects. However, this is not so for the side boundaries, because on each ray grid there are principal components that actually exit through one or the other side boundary.

For the sake of explanation, we define the *inclination angle* as follows: if a principal components in the (ξ,η) coordinates has a frequency $k(\cos\alpha,\sin\alpha)$ then α is its inclination angle on the corresponding grid.

In the assignment of the RBC at the coarsest ray level, each principal component with inhomogeneous boundary conditions is distributed only between the two neighboring lattice components, hence on each of the two corresponding ray grids its inclination is less than $2\pi/L$. The distribution weights are proportional to the distances between the corresponding principal frequency and the lattice points in the phase space. Hence, on each ray grid only principal components with inclinations in the range $|\alpha| \le 2\pi/L$ are represented.

Let us consider, for example, the side boundary $\eta = -\eta_0$. This is an entrance boundary for all components with $\alpha \ge 0$. Therefore, we have to impose Dirichlet boundary conditions for the ray function there as we do at the entrance. For the principal components with $\alpha < 0$, this boundary is obviously an exit. Since the boundary conditions are already imposed, upon reaching this boundary these components do not leave the ray domain. Instead, they turn into erroneous reflected rays. To avoid their influence, we choose the parameters of the algorithm so that these rays will not enter Ω as defined in (1.3). This can be achieved by moving ray side boundaries away from Ω. We thus define the ray domain $\hat{\Omega}^\ell$ by

$$\hat{\Omega}^\ell = \{(\xi^\ell,\eta^\ell) : -d^\ell/2 - \Delta d_\xi \le \xi^\ell \le d^\ell/2,\ \ -d^\ell/2 - \Delta d_\eta \le \eta^\ell \le d^\ell/2 + \Delta d_\eta\},$$

where $d^\ell = c^\ell d$ and c^ℓ is chosen so that the ray domain, even for $\Delta d_\eta = 0$, completely covers the wave domain ($1 \le c^\ell \le \sqrt{2}$ depending on the lattice point). Here Δd_η is how far the ray domains should be extended in the η direction, and Δd_ξ is how far they should be extended in the negative ξ direction (as discussed in the next section). On the finest ray level, with $L = 8$ and the maximum inclination angle $\alpha = \pi/4$, if Δd_η satisfies

$$\Delta d_\eta = \tan(\pi/4)\frac{d}{2} = \frac{d}{2}, \tag{3.1}$$

then no reflected rays enter the target domain Ω.

If the algorithm employs more than one ray grid, the coarser ray domains must meet two requirements. First, similar to the finest ray grids, they should be large enough not to allow the erroneous reflected rays into the target domain. Since the maximum inclination angle decreases as the algorithm proceeds to the coarser grids and the finer lattices, extension (3.1) is sufficient for all grids. More importantly, the coarser ray domains should cover neighboring finer ray domains, i.e., they should be of the size of the finer ray domains or larger (if the propagation directions are not aligned). The latter effect, however, diminishes as the difference between the propagation directions of neighboring principal components becomes smaller on the finer lattices. Moreover, since the involved grids are coarser, this effect is negligible in terms of the overall computational work.

Let us notice that since the algorithm allows to avoid the effect of reflected rays in the computational domain, there is *no need* **to impose artificial absorbing boundary conditions, such as Perfect Matching Layers (PML), e.g., [11], [12], which are actually used for the purpose of avoiding reflection.**

3.2. Entrance: Backward Residual Influence. In the nearest neighborhood of the entrance boundary, on each ray grid we assume the solution to be purely of the ray form, i.e., it is governed only by the RBC. Ideally, there should be no impact of the interior solution on the upstream values. This would be true if we used a strictly upstream discretization for the ray equations. To provide an accurate ray discretization, we have however chosen the discretization (2.5) which employs one downstream grid point. This implies upstream spread of changes in the solution from the region with non-zero residuals in the right-hand-side of (1.7). The question then arises: how far back to place the entrance boundary from the region where residuals from the finer grids are defined so that the entrance RBC is not polluted by the solution behavior in the interior.

An easy way to see the upstream impact of residuals is to analyze the changes in neighboring-line residuals introduced upon relaxing equations on one line, e.g., the line $\xi = \tilde{\xi}$. Such equations are relaxed using line Gauss-Seidel relaxation to change the solution values on the line $\xi = \tilde{\xi} + h_\xi/2$. Assuming that a Fourier component $r(\eta) = R_0 e^{i\omega\eta}$ of residuals exists on the line $\tilde{\xi}$, one line Gauss-Seidel relaxation sweep will change the solution on the line $\tilde{\xi} + h_\xi/2$ by $V_0 e^{i\omega\eta}$, where by (2.5)

$$\left(\frac{2ik}{h_\xi} - \frac{1}{2h_\xi^2} + \frac{\cos(\omega h_\eta) - 1}{h_\eta^2}\right) V_0 = -R_0. \tag{3.2}$$

Such a change will add $R_{-1}e^{i\omega\eta}, R_1 e^{i\omega\eta}$ and $R_2 e^{i\omega\eta}$ to the residuals on the lines $\xi = \tilde{\xi} - h_\xi$, $\xi = \tilde{\xi} + h_\xi$, and $\xi = \tilde{\xi} + 2h_\xi$, respectively, where by (2.5)

$$R_{-1} = R_2 = \frac{1}{2h_\xi^2} V_0 \tag{3.3}$$

and

$$R_1 = \left(-\frac{2ik}{h_\xi} - \frac{1}{2h_\xi^2} + \frac{\cos \omega h_\eta - 1}{h_\eta^2}\right) V_0. \tag{3.4}$$

Hence, and by (1.10),

$$\frac{|R_{-1}|}{|R_0|} < \frac{1}{4kh_\xi} = \frac{4}{L^2} \tag{3.5}$$

Since the smallest value of L in our algorithm is 8, the ratio always satisfies

$$\frac{|R_{-1}|}{|R_0|} \leq \frac{1}{16}. \tag{3.6}$$

If m_ξ relaxation sweeps are performed, the backward influence of the residual will continue to propagate and will reach the line $\xi = \tilde{\xi} - m_\xi h_\xi$. The size of the residual there, however, will not exceed $R_0 L^{-2m_\xi}$. Thus, at the distance of only few mesh-sizes the backward influence of the interior residuals becomes really negligible. It also follows from (3.5) that on coarser ray grids the backward influence per unit length remains the same.

In our solver, for technical reasons, near the boundaries the solution is always defined by the ray formulation with the homogeneous right-hand-side. Therefore, there is no need to extend the computational domain in the negative propagation direction.

3.3. Interpolation of Boundary Conditions. The subject of this section is how to diminish the influence of errors in boundary interpolation. Unlike many other boundary conditions, the RBC are defined on the coarsest (ray) grid. From there the boundary values are gradually interpolated to the finer ray grid, then to the intermediate wave grids, and, eventually, to the target wave grid.

Unlike the interior values, the boundary values are never changed (improved) on the finer grids, and, therefore, the errors that arise in the boundary interpolations are never corrected there. Therefore, to obtain accurate solutions, it is necessary to be possible to obtain for all *principal components* (those participating in the RBC) as small interpolation errors as one wishes. We show below that this is always possible by employing high enough interpolation orders along the boundaries. (No such high orders are needed in the interior of the fine grid.) Furthermore, we show that the effect of these boundary errors on the solution *inside* the domain remains equally small.

3.3.1. Interpolation from a Ray Grid. In the coming discussion we need to distinguish between entrance and side boundaries.

Let us first consider a Fourier component of the error originated from interpolating a principal component on the *entrance boundary*, $\xi = \xi_0$, and thus having the form

$$v(\xi, \eta) = V(\xi) e^{i\omega_\eta \eta}. \tag{3.7}$$

Substitution of (3.7) into (1.7) leads to an ODE for $V(\xi)$:

$$V''(\xi) + 2ikV'(\xi) - \omega_\eta^2 V(\xi) = 0. \tag{3.8}$$

All principal components that should be approximated on a given ray grid have the frequencies that satisfy

$$(w_\xi, w_\eta) = k(\cos\theta - 1, \sin\theta), \quad -\pi/L \leq \theta \leq \pi/L. \tag{3.9}$$

This, in particular, means that $|\omega_\eta|$ is always less or equal to k, and therefore the error amplitude

$$V(\xi) = V(\xi_0) e^{-i(k \pm \sqrt{k^2 - \omega_\eta^2})\xi} \tag{3.10}$$

is oscillatory and neither decreases nor increases as the error propagates from the boundary into the domain. The initial amplitude $V(\xi_0)$ can be made arbitrarily small, because by (3.8) and (1.10), the ray functions represented on each grid are relatively smooth in the η direction:

$$|w_\eta h_\eta| \leq 1, \tag{3.11}$$

where h_η is the mesh-size in the η direction of the coarse grid. Such functions can be interpolated from the coarse ray grids to the finer grids very accurately just by using an interpolation of a high enough order.

Let us next consider a *side boundary*, for definiteness the boundary $\eta = -\eta_0$. Substituting an interpolation error component of the form

$$\hat{v}(\xi,\eta) = \hat{V}(\eta)e^{i\omega_\xi \xi} \tag{3.12}$$

into the ray equation (1.7), we obtain an ODE for $V(\eta)$:

$$\hat{V}'' - (\omega_\xi^2 + 2k\omega_\xi)\hat{V} = 0. \tag{3.13}$$

By (3.9), $-2k < \omega_\xi < 0$, and therefore $\omega_\xi^2 + 2k\omega_\xi < 0$, any solution of (3.13) is again oscillatory, with an amplitude that neither decreases nor increases as it propagates into the computational domain. Fortunately, it too can be reduced as much as one wishes by increasing the interpolation order. This is because, by (1.10) and (3.9), for the highest principal frequency that still needs to be accurately represented on the fine ray grid there holds

$$|\omega_\xi| h_\xi < 0.4. \tag{3.14}$$

Comparison between (3.11) and (3.14) shows that the order of interpolation operators used along the side boundaries can be lower than that needed along the entrance.

3.3.2. Interpolation from a Wave Grid. We consider here two wave grids: a fine one, with a mesh-size h in both directions, and a coarse, with the mesh-size $2h$. We examine the propagation of a component of an interpolation error, resulted from interpolating a principal component $e^{i(k_1x+k_2y)}, k_1^2 + k_2^2 = k^2$, in the form

$$v(x,y) = V(y)e^{ik_1x}, \tag{3.15}$$

introduced, for example, on the boundary $y = y_0 = -d/2$. Its amplitude, $V(y)$ satisfies the equation

$$V''(y) + (k^2 - k_1^2)V(y) = 0, \tag{3.16}$$

hence

$$V(y) = V(y_0)e^{\pm ik_2^2 y}, \tag{3.17}$$

so $|V(y| \leq |V(y_0)|$. Using interpolation operators of a sufficiently high order, the initial amplitude $V(y_0)$ for such values of k_2 can be made as small as desired, since $|k_2|2h \leq k2h \leq 1$; this is because for any wave grid to which the boundary conditions are interpolated, its mesh-size satisfies $kh \leq 0.5$. These are grids finer than the reconstruction level scale h_b, where $kh_b \approx 1$; no boundary conditions are interpolated to the coarser levels.

All these results show that the boundary interpolation errors do not decay as they propagate into the computational domains; they can be diminished only by achieving small initial (on the boundaries) error amplitudes. To guarantee that, for the components of interest, the algorithm should employ interpolations of high orders (possibly higher than the typical, for standard multigrid solvers, second or fourth orders). Fortunately such high-order, and therefore expensive, interpolation procedures are only needed for interpolation to the ray entrance boundaries and to the boundaries of the wave grids with $kh \approx 1$ and $kh \approx 0.5$. (For the finer wave grids the principal components are smooth and can be accurately approximated by regular interpolation procedures). Since this interpolation work is needed only on the boundaries and only on the coarse grid domains, its total cost is insignificant.

The results in Table 3.1 show how the increase of the interpolation order decreases the interpolation error.

Table 3.1

Error amplitudes obtained in interpolation of the Fourier component $exp(i\omega x)$ from the grid with $\omega h \approx 1$ (the most oscillatory component that appear in the boundary interpolation) in the x direction . The parameter of the Table is the interpolation order.

order	2	4	6	8
error	1.2e-01	2.1e-02	4.1e-03	8.4e-04

4. Numerical Experiments. In all numerical experiments the wave-ray algorithm was applied to the model problem (1.1) considered on the computational domain

$$\Omega = \{(x,y): \quad -d/2 \le x,y \le d/2\} \tag{4.1}$$

and with the right-hand-side support

$$\Omega_f = \{(x,y)| \quad x^2+y^2 \le 5^2\}. \tag{4.2}$$

In all experiments with the non-homogeneous right-hand-side, $f(x,y)$ is chosen to satisfy

$$f(x,y) = L\underline{u}(x,y), \tag{4.3}$$

where $\underline{u}(x,y)$ is a polynomial of $r=\sqrt{x^2+y^2}$ for $|r| \le 5$, a Hankel function of r for $|r| > 5$, and it is in $\mathbf{C}^4(\mathbf{R^2})$. The right-hand-side $f(x,y)$ is therefore a polynomial for $|r| \le 5$ and zero elsewhere. In most experiments we assume $k=1$.

The size d of the target domain Ω varies for different tables. The algorithm employs only one ray level, with eight lattice components ($n=1, L=L^1=8$). The RBC are either homogeneous, or inhomogeneous with just one principal component. This component is imposed on one ray grid that corresponds to the lattice principal component $(\cos(0),\sin(0))$. The algorithm employs (except Tables 4.10-4.12) cubic interpolation. The number of extended mesh-sizes in the ξ direction for ray domains is usually $m_\xi = \Delta d_\xi/h_\xi = 5$. In the η direction, the extension of the ray domains depends on the size of the target computational domain, and is given by (3.1). The values of Δd_ξ and Δd_η become experimental parameters in Tables 4.5–4.7.

The errors, calculated in the L_2 norm, measure the difference between the discrete solution $u^h_{i,j}$ and the differential one $\underline{u}(x_i,y_j)$. In most experiments the errors are calculated in the domain

$$\Omega_e = \{(x,y): -15 \le x,y \le 15\}, \tag{4.4}$$

unless stated otherwise. Note that Ω_e is located at the center of Ω. Therefore, the wave and the ray phase errors arising from the right-hand-side do not depend on d. On the other hand, the phase errors caused by the non-homogeneous radiation boundary conditions depend on d, since d defines the distance that the principal component travels before reaching Ω_e.

The goal of our numerical experiments is to show the influence of different error terms and to confirm our theoretical conclusions.

Table 4.1

Errors for different finest mesh-sizes h and computational domain sizes d; the results shown are for the model problems with the right-hand-side defined by (4.3). The RBC are homogeneous. The errors are calculated in Ω_e, using the L_2 norm.

	$d = 32$	$d = 64$	$d = 128$
$h = 0.5$	2.01e-02	2.24e-02	2.25e-02
$h = 0.25$	4.27e-03	5.11e-03	5.20e-03
$h = 0.125$	3.10e-03	3.07e-03	3.12e-03

4.1. Wave Truncation Errors. Since the RHS in Table 4.1 is not homogeneous, the error includes the wave truncation (non-phase) and the wave phase errors. For this model problem (with homogeneous RBC) and for the error measured in Ω_e, none of these errors depend on d and both are proportional to h^2. The errors for $h = 0.25$ and $h = 0.5$ indeed exhibit these properties, and therefore we conclude that there the wave errors are dominant. This, however, changes for $h = 0.125$, where the error decrease with h weakens, meaning that the resulting error is a combination of other errors, such as ray phase errors (the non-polynomial part of the solution is of the ray nature, and consists of a variety of ray components) and interpolation errors. In the next experiments we will investigate the behavior of such errors and the ways to diminish them.

4.2. Phase Errors. In Table 4.2, there is neither RHS related errors (the right-hand-side is homogeneous), nor RBC phase ray errors , since the solution ray functions are constant (because the RBC are defined at a lattice component). The errors depend linearly on the domain size d and quadratically on the target mesh-size h. These are the properties of the *wave phase error* (2.3), which is the main error source in these experiments.

Table 4.2

Errors for different finest mesh-sizes h and computational domain sizes d; the results shown are for the model problems with the homogeneous right-hand-side. The RBC are non-homogeneous for the lattice component $(\cos(0), \sin(0))$. *The errors are calculated in Ω_e, using the L_2 norm.*

	$d = 32$	$d = 64$	$d = 128$
$h = 0.5$	1.64e-01	3.64e-01	6.87e-01
$h = 0.25$	4.03e-02	8.74e-02	1.66e-01
$h = 0.125$	1.05e-02	2.09e-02	4.12e-02

In the experiments presented in Table 4.3, the RBC conditions are defined for a non-lattice principal component, and this introduces the RBC ray phase errors. Unlike the wave phase errors, they should not depend on the target mesh-size h but still depend on d.

The results show that the errors depend almost linearly on d confirming their phase nature. Their dependence on h, however, is mixed: the error is a combination of the $O(h^2)$ wave and the $O(1)$ ray phase errors (by (2.3) and (2.10), respectively).

Table 4.3

Errors for different finest mesh-sizes h and computational domain sizes d. The right-hand-side is homogeneous. The RBC are non-homogeneous for the non-lattice principal component $(\cos(\pi/16), \sin(\pi/16))$*. The errors are calculated in* Ω_e*, using the* L_2 *norm.*

	$d = 32$	$d = 64$	$d = 128$
$h = 0.5$	1.72e-01	3.32e-01	6.42e-01
$h = 0.25$	5.09e-02	9.31e-02	1.74e-01
$h = 0.125$	2.24e-02	3.43e-02	5.83e-02

In the next Table we compare the errors obtained for different values of the wave number, k. The range of k is chosen to allow the use of the same *same* wave and ray grids for all values of k. (For any pair of values $k = k_0$ and $k = 2k_0$, different grids should be chosen for separation ($kh \approx 1$) and for the ray representation (1.10). Therefore, in our experiments we consider k such that $k_0 = 0.7 \leq k < 1.4 = 2k_0$.)

Table 4.4

Errors for different values of k; the RHS is homogeneous, the RBC are non-homogeneous for the principal component $(\cos(\alpha), \sin(\alpha))$*,* $h = 0.125$*,* $d = 64$*.*

k	0.7	0.9	1.0	1.1	1.2	1.3
$\alpha = 0$	2.82e-02	6.05e-02	8.05e-02	1.06e-01	1.38e-01	1.76e-01
$\alpha = \pi/16$	5.50e-02	6.69e-02	8.97e-02	1.16e-01	1.48e-01	1.83e-01

For $\alpha = 0$, the error ratios are proportional to the cubic power of the wave number ratios, which is a characteristic of the wave phase error. For $\alpha = \pi/16$, instead of cubic dependence, the power is $2 + \epsilon$, $0 < \epsilon < 1$, reflecting the fact the error is a combination of the $O(k^3)$ wave and the $O(k)$ ray phase errors (by (2.3) and (2.10)).

4.3. Reflected Rays.

Table 4.5

Errors for different values of Δd_η, the size of the extension of ray domains in the η direction. The model problem is with the non-homogeneous RHS defined by (4.3), the RBC are homogeneous, $d = 64$*,* $h = 0.125$*. The errors are calculated in the entire computational domain* Ω*.*

Δd_η	0	5	10	20
error	2.53e-03	2.52e-03	2.52e-03	2.52e-03

The results in Table 4.5 show no difference for different values of Δd_η, and this is because of the character of the solution near the boundary: the solution decays as it propagates toward boundaries, and therefore the size of the reflected waves is small and does not affect the total error. This, however, changes if the solution amplitudes remain large near the boundary, as in Table 4.6.

Table 4.6

Errors for different values of Δd_η, the size of the extension of ray domains in the η direction. The model problem is with the homogeneous RHS; the RBC is non-homogeneous for $(\cos(\pi/12), \sin(\pi/12))$*,* $d = 64$*,* $h = 0.125$*. The errors are calculated in the entire computational domain* Ω*.*

Δd_η	0	5	10	20
error	1.57e-01	9.61e-02	7.16e-02	7.16e-02

Table 4.6 shows the influence of Δd_η (Section 3.1) on the accuracy of the solution. The result confirm that in order to avoid influence of the reflected rays, the side boundaries of ray domains should be sufficiently extended.

4.4. Backward Residual Propagation. As discussed at the end of Section 3.2, the results in Table 4.7 show no influence of the number of extra mesh-sizes in the direction opposite to the propagation direction ξ on the solution accuracy.

Table 4.7

Errors for different values of $m_\xi = \Delta d_\xi / h_\xi$ is the number of additional mesh-sizes in the negative ξ direction for ray domains. The model problem is with the non-homogeneous RHS defined by (4.3), the RBC are homogeneous, $d = 64$, $h = 0.125$. The error is calculated in the entire computational domain Ω.

m_ξ	0	5	10
error	2.52e-03	2.52e-03	2.52e-03

4.5. Ray Phase Errors. There are two ways to decrease ray phase errors. First, one can employ a high order discretization for the ray equation (1.7). In the next two Tables the accuracy of the solution obtained by the solver that employs the second- and the fourth-order discretization schemes for the ray equations are presented and compared.

Table 4.8

Errors for the solver that employs ray discrete operators of second order (2.5); the solver employs cubic interpolation, both in the interior and on the boundaries; the RHS is homogeneous, and the RBC are non-homogeneous for the principal component $(k_1, k_2) = (\cos(\alpha), \sin(\alpha))$.

h	d	$\alpha = 0.0$	$\alpha = \pi/16$	$\alpha = \pi/8$
0.125	64	2.09e-02	3.47e-02	2.91e-01
0.25	64	8.69e-02	9.13e-02	3.45e-01
0.25	128	1.61e-01	1.74e-01	4.79e-01

Table 4.9

Errors for the solver that employs ray discrete operators of fourth order; the solver employs cubic interpolation, both in the interior and on the boundaries; the RHS is homogeneous, and the RBC are non-homogeneous for the principal component $(k_1, k_2) = (\cos(\alpha), \sin(\alpha))$.

h	d	$\alpha = 0.0$	$\alpha = \pi/16$	$\alpha = \pi/8$
0.125	64	2.08e-02	2.22e-02	1.60e-01
0.25	64	8.67e-02	8.34e-02	2.03e-01
0.25	128	1.61e-01	1.74e-01	3.41e-01

Comparing the results in Tables 4.8 and 4.9 shows that the higher-order discretization does not improve the accuracy for $\alpha = 0$, the expected results since no discretization error is committed for a pure lattice component. The dominant error in this case is the wave phase error. For $\alpha = \pi/16$, the improvement is significant, and the dominant part of the remaining error is also clearly of the wave nature: it depends almost quadratically on h and linearly on d. For $\alpha = \pi/8$, however, even the reduced error consists mostly of the ray errors. This shows the limitations of the *higher-order discretization* approach: at given sizes of k and computational domain d for sufficiently small finest wave mesh-sizes h the ray error will still dominate. This happens because the wave phase errors are reduced as h becomes smaller, and the ray phase errors remain the same.

Another, much more general approach is to reduce ray phase errors when needed. To reduce the ray phase errors *without* increasing discretization orders, one needs to employ not one, as in the experiments discussed above, but several ray grids, with increasingly finer lattice discretization on the coarser grids. The finer is the lattice dicretization, the smaller are the ray phase errors (2.10). The goal of the intermediate (finer) ray grids is to reduce errors to the almost pure rays. The overall accuracy of the ray representation is then defined by the coarsest ray grid with the largest number of lattice components, $L = 2^{N+2}$, where N is the number of ray levels employed. There the radiation boundary conditions are imposed, facilitated by the nearly pure ray representation at that grid. With the right choice of N and, correspondingly, L, one can make the ray phase errors arbitrary small.

This approach is perfectly suitable for the problems considered on large computational domains: it requires significant coarsening in space, as L grows (1.10), which makes the ray approximation very inexpensive even for large L and d.

4.6. Interpolation of Boundary Conditions. In Tables 4.10–4.12 the influence of the interpolation orders is tested. The difference between the tables is in the choice of the inclination angle α of the principal component for which the non-homogeneous RBC are imposed. The influence of the interpolation orders on the accuracy of the solution is tested for different values of h and d, as well. Fourth order discretization is employed to the ray operators in order to reduce phase ray errors, by this making the influence of interpolation errors visible.

We start with the RBC non-homogeneous for the lattice principal component with $\alpha = 0$. In this case the ray functions are constant, and therefore the accuracy of the ray interpolation does not matter, since interpolation of any order accurately interpolates constant functions. In the wave representation, however, this e^{ix} component is the most oscillatory principal component that yields the highest truncation (phase) and boundary interpolation errors.

The results in Table 4.10 show that as long as the cubic interpolation ($p = 4$) is employed in the wave interpolation the errors depend quadratically on h and linearly on d, meaning that the wave phase errors is the main error source. The influence of the interpolation errors is insignificant (for instance, increasing the interpolation order on the boundary barely improves the accuracy). This, however, changes for $p = 2$ in the wave interpolation, for which the interpolation errors become visible. The ray interpolation order does not influence the accuracy of the solution for this α.

Table 4.10

Errors for interpolation operators of different orders; here $(p_w(p_w^b), p_r(p_r^b))$ are the order of wave and ray interpolation operators used in the algorithm, both in the interior and on the boundaries; the numbers in parenthesizes are the orders of the operators used to interpolate to the boundaries from the wave and the ray coarser grids, respectively, if they are different from the orders used for the interior interpolation; the RHS is homogeneous, and the RBC are non-homogeneous for the principal component $(\cos(0), \sin(0))$.

h	d	$(4,4)$	$(4,2)$	$(2,2)$	$(4(6),4)$	$(2(4),2)$
0.125	64	2.08e-02	2.08e-02	6.24e-02	2.05e-02	6.03e-02
0.25	64	8.67e-02	8.69e-02	1.00e-01	8.64e-02	9.98e-02
0.25	128	1.60e-01	1.61e-01	1.70e-01	1.60e-01	1.69e-01

Next we consider the principal component with $\alpha = \pi/16$. Linear interpolation $p = 2$ from the ray grids sufficient for the previous example ($\alpha = 0$) introduces a significant error here. For $\alpha = \pi/16$, $|\omega_\eta h_\eta| \approx 0.4$ (Sec. 3.3.1), and therefore interpolation from the ray grid can be accurately performed using cubic interpolation ($p = 4$). For the wave representation, however, on the reconstruction level $|wh| \approx 1$ (Sec. 3.3.2), and the higher order interpolation ($p = 4$ or $p = 6$) from the wave grids (especially on the boundaries) leads to a better accuracy.

Table 4.11

Like Table 4.10, except that the RBC are non-homogeneous for the principal component $(\cos(\pi/16), \sin(\pi/16))$.

h	d	$(4,4)$	$(4,2)$	$(2,2)$	$(4(6),4)$	$(2(6),2(6))$
0.125	64	2.22e-02	2.44e-02	4.53e-02	2.18e-02	4.50e-02
0.25	64	8.34e-02	9.36e-02	9.81e-02	8.30e-02	9.35e-02
0.25	128	1.61e-01	1.63e-01	1.74e-01	1.61e-01	1.73e-01

In Table 4.12 the non-homogeneous RBC are imposed for the principal component $(\cos(\pi/8), \sin(\pi/8))$, the mid-point between the neighboring lattice components $(\cos(0), \sin(0))$ and $(\cos(\pi/4), \sin(\pi/4))$. For this component, both the ray principal components (—$\omega_\eta h_\eta| \approx 0.8$) and the wave principal components (—$\omega h| \approx 1$ at the reconstruction level) are rather oscillatory, and it seems that the algorithm could benefit from increasing interpolation orders, at least for the boundary interpolation. This, however, is not reflected in the results shown in Table 4.12.

Table 4.12

Like Table 4.10, except that the RBC are non-homogeneous for the principal component $(\cos(\pi/8), \sin(\pi/8))$.

h	d	$(4,4)$	$(2,2)$	$(4(6),4(6))$
0.125	64	1.60e-01	1.82e-01	1.58e-01
0.25	64	2.03e-01	2.14e-01	1.98e-01
0.25	128	3.41e-01	3.43e-01	3.39e-01

The errors in Table 4.12 are almost independent of the interpolation accuracy. They, however, depend on the size of the computational domain, meaning that the ray phase errors discussed in Sec. 4.5 are the main reason of inaccurate approximations.

5. Conclusions and Future Application. For most multigrid procedures the accuracy of the solution is defined by the discretization accuracy at the finest grid. This is not always so for problems on unbounded domains, where it is computationally beneficial to use progressively extended coarse grid domains, so that in its outer part each coarse grid provides not a correction to the fine grid solution, but the solution itself. This feature is present in the wave-ray algorithm for solving indefinite Helmholtz equations with radiation boundary conditions. The developed solver shows a textbook multigrid efficiency for highly indefinite problems on unbounded domains. With the right choice of parameters (such as computational domains, wave and ray mesh-sizes, order of interpolation operators) the produced solution has the accuracy defined by the finest grid discretization. The analysis offered here can be applied to other problems given on infinite domains with boundary conditions defined at infinity. When restricted to finite computational domains, the numerical solvers for such problems have to deal with interpolation and other errors, similar to the ones discussed in this paper.

The wave-ray algorithm discussed in the paper is obviously very technical and employs several special techniques such as rotated coordinate systems, staggered grids, special weighting operators, etc. Most of these technicalities however could be avoided, and the obtained simplified algorithm would still be an efficient (though less efficient) solver for the Helmholtz equation. The main restriction that cannot be lifted in this *geometric* **multigrid solver is a necessity to represent principal components in their analytical form, and this is available only for constant or very smooth**

k. These principal components are used to obtain differential equations for the ray functions, which are then discretized and solved.

Our next step (currently in progress, preliminary results in [7]) is to use the wave-ray approach in the *algebraic* multigrid framework, with only numerical approximation to the principal components are needed and used to obtain discrete ray equations (skipping on differential equations step). This modified approach allows application of the wave-ray algorithm to problems with non-smooth wave numbers.

Acknowledgments. The final part of the work on this paper was completed while the authors were visiting Center for Applied Scientific Computing at Lawrence Livermore National Laboratory.

Appendix. Pseudocode. A brief pseudocode of the wave-ray algorithm is given next. It describes the solver that was used in all experiments presented in this paper.

```
Begin wave ray cycle
  Wave sub cycle: from the finest to the coarsest wave grid
  Finest wave grid $m = M_t$
      Relaxation  $L^m u^m = f^m$
      Residual calculation by full weighting $r^{m-1} = W_m^{m-1}(f^m - L^m u^m)$
      Solution initialization by injection $u^{m-1} = I_m^{m-1} u^m$
  for $m = M-1, \dots, 2$
      FAS right-hand-side $f^m = r^m + L^m u^m$
      if $(kh_m \not\approx 2)$ Relaxation  $L^m u^m = f^m$
      Residual calculation by full weighting  $r^{m-1} = W_m^{m-1}(r^m - L^m u^m)$
      Solution initialization by injection $u^{m-1} = I_m^{m-1} u^m$
  Coarsest wave grid $(m = 1)$ Relaxation  $L^m u^m = f^m$
  Wave sub cycle: from the coarsest to some fine wave grid
  for $m = 2, \dots, M_f$,  $M_f \leq M$ is the wave grid with a small phase error
      Interpolation $u^m = u^m + I_{m-1}^m v^{m-1}$
      if $(kh_m \not\approx 2)$ Relaxation  $L^m u^m = f^m$
  Wave residual calculation for ray sub cycle $(m = M_f)$ $r^m = f^m - L^m u^m$
  for $(m = M_f - 1, \dots, M_c)$,    $M_c$ is the wave grid with $kh_c \approx 1$
      Residual transfer by full weighting $r^m = W_{m+1}^m r^{m+1}$
  Ray sub cycle with $N$ ray levels
  Finest ray grid $n = 1$
  for $(\ell = 1, \dots, L_1)$
      Residual separation $\hat{r}^{1,\ell} = W_{M_c}^{1,\ell}(r^{M_c} \overline{e}^{1,\ell})$,
      $\overline{e}^{1,\ell}$ is the complex conjugate of $e^{1,\ell}$, the $\ell^{th}$ component on the $1^{st}$ lattice
      Ray FAS right-hand-side
          $\hat{f}^{1,\ell} = \hat{r}^{1,\ell} + \hat{L}^1 \hat{u}^{1,\ell}$, where $(\hat{r}^{1,\ell} \neq 0)$ and $\hat{f}^{1,\ell} = 0$, elsewhere
  for $(n = 2, \dots, N)$
      Residual separation
      for $(\ell = 1, \dots, N, \ell = \ell + 2)$      for odd $\ell$
          $\hat{r}^{n,\ell} = W_{n-1,(\ell+1)/2}^{n,\ell}(\hat{r}^{n-1,(\ell+1)/2})$
      for $(\ell = 2, \dots, N, \ell = \ell + 2)$      for even $\ell$
          $\hat{r}^{n,\ell} = W_{n-1,\ell/2}^{n,\ell}(\hat{r}^{n-1,\ell/2} e^{n-1,\ell/2} \overline{e}^{n,\ell}) + W_{n-1,\ell/2+1}^{n,\ell}(\hat{r}^{n-1,\ell/2+1} e^{n-1,\ell/2+1} \overline{e}^{n,\ell})$,
          where $e^{n,\ell}$ is the $\ell^{th}$ component on the $n^{th}$ lattice
```

```
    for $(\ell = 1, \dots, L_n)$
        Ray FAS right-hand-side
        $\hat{f}^{n,\ell} = \hat{r}^{n,\ell} + \hat{L}^n \hat{u}^{n,\ell}$ where $(\hat{r}^{n,\ell} \neq 0)$ and $\hat{f}^{n,\ell} = 0$ elsewhere
        if $(n \neq N)$ Relaxation $L^n u^{n,\ell} = f^{n,\ell}$
        else
            Imposition of RBC at entrance and side boundaries
            $\hat{u}^{n,\ell}(\xi, \eta) = \hat{u}_0(\eta)$ for  $\xi = \xi_0$; $\hat{u}^{n,\ell}(\xi, \eta) = \hat{u}_\pm(\xi)$ for  $\eta = \pm\eta_0$
            Solving $\hat{L}^n \hat{u}^{n,\ell} = \hat{f}^{n,\ell}$
for $(n = N-1, \dots, 1)$
    Interpolation to the next finer ray grid
    for $(\ell = 1, \dots, L_n)$
        Interior interpolation: $\hat{u}^{n,\ell} = \hat{u}^{n,\ell} + I^{n,\ell}_{n+1,2\ell-1} \hat{v}^{n+1,2\ell-1} +$
        $\frac{1}{2} I^{n,\ell}_{n+1,2\ell-2} (\hat{v}^{n+1,2\ell-2} e^{n+1,2\ell-2} / e^{n,\ell}) + \frac{1}{2} I^{n,\ell}_{n+1,2\ell} (\hat{v}^{n+1,2\ell} e^{n+1,2\ell} / e^{n,\ell})$
        Boundary interpolation: $\hat{u}^{n,\ell} = 0$ on $\hat{\Gamma}^{n,\ell}$
        $\hat{u}^{n,\ell} = I^{n,\ell}_{n+1,2\ell-1} \hat{u}^{n+1,2\ell-1} + \frac{1}{2} I^{n,\ell}_{n+1,2\ell-2} (\hat{u}^{n+1,2\ell-2} e^{n+1,2\ell-2} / e^{n,\ell}) +$
        $\frac{1}{2} I^{n,\ell}_{n+1,2\ell} (\hat{u}^{n+1,2\ell} e^{n+1,2\ell} / e^{n,\ell})$
Wave solution reconstruction $m = M_c$:
Zero initialization at boundaries $u^m = 0$ on $\Gamma_m$
for $(\ell = 1, \dots, L^1)$
    In interior $u^m = u^m + e^\ell I^m_{1,\ell} \hat{v}^{1,\ell}$; At boundaries $u^m = u^m + e^\ell I^m_{1,\ell} \hat{u}^{1,\ell}$
End wave reconstruction
for $(m = M_c + 1, \dots, M)$
    Zero initialization at boundaries $u^m = 0$ on $\Gamma_m$
    In interior $u^m = u^m + I^m_{m-1} v^{m-1}$; At boundaries $u^m = u^m + I^m_{m-1} u^{m-1}$
    Relaxation $L^m u^m = f^m$
```

REFERENCES

[1] A. Bayliss and E. Turkel, *Radiation boundary coniditionds for wave-like equations*, Comm. Pure Applied Math, 33, 1980, pp. 707-725.

[2] A. Bayliss, C.I. Goldstein, and E. Turkel, *On accuracy conditions for the numerical computations of waves*, J. Computational Physics, 59, 1985, pp. 396-404.

[3] I. M. Babushka, S. S. Sauter, *Is the Pollution Effect of the FEM Avoidable for the Helmholtz Equations Cosnidering High Wave Numbers ?,* SIAM Review, 42(3), pp. 451-484.

[4] A. Brandt, *Multigrid Techniques: 1984 Guide with applications to fluid dynamics*, Monograph, GMD-Studie 85, GMD-FIT, Postfach 1240, D-5205, St. Augustin 1, W.Germany, 1985.

[5] A. Brandt, I. Livshits, *Wave-Ray Multigrid Method for Standing Wave Equations*, ETNA (Electronic Transactions on Numerical Analysis), 6, 1997, pp. 162-181.

[6] A. Brandt, *The Gauss Center Research in Multiscale Scientific Computation*, ETNA (Electronic Transactions on Numerical Analysis), 6, (1997), pp. 1-34.

[7] I. Livshits, *An algebraic multigrid wave-ray algorithm to solve eigenvalue problems for the Helmholtz operator*, Numerical Linear Algebra with Applications, 11(2-3), 2004, pp. 229-239.

[8] A. Brandt and S. Ta'asan, *Multigrid Methods for Nearly Singular and Slightly Indefinite Problems*, In Multigrid Methods II Lecture Notes in Mathematics 1228, Hackbush W., Trottenberg U. (eds), Springer-Verlag, 1985, pp. 100-122.

[9] A. Deraemaker, I. Babuska, and P. Bouillard, *Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions,* International Journal for Numerical Methods in Engineering, 46, 1999, pp. 471-499.

[10] H. C. Elman, O. G. Ernst, and D. P. O'Leary, *A Multigrid Method Enhanced by Krylov Subspace Iterations for Discrete Helmholtz Equations*, SIAM Journal on Scientific Computing,

23(4), 2001, pp. 1291-1315.

[11] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical solutions of waves*, Math. Comp., 31, 1977, pp. 629-651.

[12] B. Engquist and A. Majda, *Radiation boundary conditions for acoustic and elastic wave calculations*, Comm. Pure Appl. Math., 32 , 1979, pp. 313-357.

[13] F. Ihlenburg and I. Babuska, *Finite element solution to the Helmholtz equation with high wave numbers - part I: the h-version of the FEM*, Comp. Math. Applic., 30, 1995, pp. 9–37.

[14] B. Lee, T. Manteuffel, S. McCormick, and J. Ruge, *First-Order System Least Squares for Helmholtz Equations*, SIAM Journal on Scientific Computing, 21(5), 2001, pp. 1927-1949.

[15] P. Vanek, J. Mandel, and M. Brezina, *Two Level Algebraic Multigrid for the Helmholtz Problem*, Contemporary Mathematics, 218, 1998, pp. 349-356.