

NUMERICAL SOLUTION OF 3D EXTERIOR UNSTEADY WAVE PROPAGATION PROBLEMS USING BOUNDARY OPERATORS*

SERGEY PETROPAVLOVSKY[†], SEMYON V. TSYNKOV[‡], AND ELI TURKEL[§]

Abstract. We propose a boundary method for the numerical simulation of time-dependent waves in three-dimensional (3D) exterior regions. The order of accuracy can be either second or fourth in both space and time. The method reduces a given initial boundary value problem for the wave equation to a set of operator equations at the boundary of the original domain. The reduction is based on a reformulation of the method of difference potentials. The resulting operator equations relate the solution and its normal derivative at the boundary. To solve these equations, one relies on the Huygens' principle. This yields an algorithm that works on a sliding time window of a finite nonincreasing duration. As a result, it allows one to avoid the ever increasing backward dependence of the solution on time. The major advantages of the proposed methodology are its reduced computational complexity (grid-independent on the boundary and sublinear in the volume), the capacity to handle curvilinear geometries using Cartesian finite difference time domain (FDTD) methods, and automatic and exact accounting for the far-field radiation conditions. In addition, the methodology facilitates solution of multiple similar problems at low individual cost per problem and guarantees uniform performance over arbitrarily long time intervals.

Key words. method of difference potentials, time-dependent wave (d'Alembert) equation, Huygens' principle, initial boundary value problem

AMS subject classifications. 35L05, 65M06, 65M12, 65M22, 65M99, 65Z05, 78A40

DOI. 10.1137/19M1269269

1. Introduction. We propose a methodology for computing the propagation of unsteady waves outside a given three-dimensional (3D) shape. Mathematically, it is an exterior initial boundary value problem (IBVP) for the hyperbolic PDE (or system) that governs the propagation, e.g., the wave (d'Alembert) equation or Maxwell's equations. The relevant physical settings include acoustics, electromagnetism, and elasticity.

The key challenges associated with these types of problems are their explicit time dependence, possible complex geometry of the scatterer, and unboundedness of the domain of propagation. The corresponding numerical methods include the finite difference time domain method (FDTD), time domain finite elements (TD FEM) that employ the weak formulation of the problem, the techniques that combine frequency domain solutions, and boundary methods.

Standard FDTD methods are fairly easy to implement, including their high order versions that are known to better control the dispersion error. Their main limitation

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 19, 2019; accepted for publication (in revised form) August 3, 2020; published electronically October 27, 2020.

<https://doi.org/10.1137/19M1269269>

Funding: This research was supported by the US Army Research Office (ARO) under grant W911NF-16-1-0115, and by the US Israel Binational Science Foundation (BSF) under grant 2014048.

[†]National Research University Higher School of Economics, Moscow 101000, Russia, and North Carolina State University, Raleigh, NC 27695 USA (spetrov@ncsu.edu).

[‡]Corresponding author. Department of Mathematics, North Carolina State University, Raleigh, NC 27695 USA (tsynkov@math.ncsu.edu, <https://stsynkov.math.ncsu.edu>).

[§]School of Mathematical Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv, 69978, Israel (turkel@tauex.tau.ac.il, <http://www.math.tau.ac.il/~turkel/>).

is their poor ability to handle curvilinear geometries on regular grids (e.g., Cartesian) that may lead to reduction in accuracy, and severe stability constraints due to small cut cells near the boundary [35, 15]. The immersed boundary method [20] may help attenuate the deterioration of accuracy. Both FDTD and TD FEM require artificial boundary conditions (ABCs) to truncate the computational domain [36].

Integral transforms can be employed to solve the problem in the dual space and subsequently reconstruct the solution by returning to the time domain. The transform may be applied after the discretization in time, e.g., the discrete Laplace or Z-transform [21]. However, the required computational resources grow as the time elapses, so it is increasingly difficult to compute the solution on longer intervals [33]. In particular, after the Fourier transform in time, one needs to solve a set of independent equations that correspond to different frequencies. Yet for longer times, the integrand in the inverse Fourier transform becomes more oscillatory, which requires more sampling points in the frequency domain. Then, the overall cost scales quadratically with time [29]. A strategy to alleviate this difficulty is presented in [4].

Boundary methods offer the advantage of having to solve the problem only on the surface of the scatterer and thus reducing the dimension by one. Moreover, they automatically account for the radiation of waves in the far field. However, their direct application in the time domain encounters the same difficulty as that of the methods based on integral transforms—the cost of computing the convolutions grows in time. Specifically, the retarded potential boundary integral equations (RPBIE) [9] involve convolutions of the boundary data with the fundamental solution that extend backward to the initial moment of time. To efficiently compute these convolutions, a number of time-stepping techniques (convolution quadratures sometimes are preceded by a Laplace transform) have been proposed [14]. Yet many of these methods have been reported to generate instabilities in the course of integration; see [14, 10]. Another implementation of RPBIE is based on the full Galerkin discretization in both space and time so that no finite difference approximation of the temporal convolutions is required. The fundamental solution of the 3D d'Alembert equation is $G(\mathbf{x}, t) = \frac{1}{4\pi|\mathbf{x}|} \delta(t - |\mathbf{x}|/c)$. Then, the integration is performed over the intersection of the light cone of the past with the space-time boundary. For scatterers of finite size it has finite duration in time. Thus, the backward dependence on time is truncated, and the algorithm has only a finite temporal “tail” [14, 33]. In doing so, the matrices of the resulting linear system contain weakly singular integrals over the aforementioned intersection of the light cone with the boundary, which may have complicated geometry. The errors in computing those integrals may degrade the accuracy of the solution [33]. Recent progress toward alleviating this difficulty is reported in [5].

In [25, 27, 24], we proposed a technique that combines the advantages of boundary methods and FDTD. It is based on Calderon's operators and the method of difference potentials (MDP) [32]. A given IBVP is reduced to the operator equation with respect to the unknown boundary sources. The radiation of waves at infinity is guaranteed automatically. Instead of boundary integrals, the MDP relies on a volumetric auxiliary problem (AP). The latter is set on a simple computational domain and can be solved by a standard FDTD scheme. This provides considerable flexibility and ease, in particular, for high order accuracy, which can be achieved with no additional overheads typical, for example, of TD FEM. Another important feature of the MDP is that it employs physical boundary sources (the solution and its normal derivative) rather than artificial densities, and treats all types of boundary conditions on the

surface of the scatterer universally, i.e., without having to reformulate the boundary problem every time a new boundary condition is introduced.

When a given IBVP is reduced from its domain to the boundary, the resulting Calderon's boundary equations involve the entire history of the solution in time. To truncate it, we employ the (strong) Huygens' principle [12]. The implementation of this part of the algorithm is fundamentally different from the full Galerkin version of RPBIE [14, 33]. It relies on lacunae of the solution, i.e., voids behind aft fronts of the waves [28]. Lacunae allow us to limit the backward dependence of the MDP algorithm in time and thus solve the AP only over a finite nonincreasing time interval. The latter depends on the size of the scatterer and speed of propagation, but does not depend on the frequency/wavelength. In our earlier work [26, 23], lacunae-based time marching has shown a robust stabilizing effect on the treatment of artificial outer boundaries for the wave equation and Maxwell's equations. The use of lacunae for the design of ABCs was pioneered by Ryaben'kii [31].

Our algorithm consists of two stages. In the premarching stage, we use a second or fourth order accurate FDTD scheme to solve a series of 3D APs that correspond to a spectral representation of the unknown boundary sources on a specially chosen finite time interval of the (2+1)-D space-time boundary. The required number of APs is not fixed—it depends on the solution that is being computed. For solutions that contain higher frequencies, this number will be larger. However, all APs are independent of one another and can be solved in parallel, which can be thought of as parallelization in time (section 6.4). The resulting solutions of the APs are used to form the matrix of a discrete Calderon's projection operator. The inversion of this matrix is another important component of the premarching stage of the algorithm. It is done in the weak sense using a QR factorization. Efficient parallel implementation of QR is also available [16]. Specific details on computational costs and scaling are given in section 6.

At the subsequent time-marching stage, we consecutively update the coefficients of the boundary spectral expansion on one time interval after another. The cost of any of these updates does not depend on the dimension of the FDTD grid. As a result, for a given range of frequencies, the proposed methodology appears superior to any volumetric time-marching scheme provided that the integration time is sufficiently long. Indeed, while the premarching cost increases with frequency, for a predetermined frequency band it becomes fixed. Then, as soon as the integration time exceeds the number of APs times the AP time interval (plus the cost of QR prorated accordingly), the boundary time marching becomes more efficient than a volumetric one (sections 6.3 and 6.4). If, in addition, the solution needs to be known in the volume at a given moment of time, the corresponding cost scales cubically with respect to the grid dimension in one direction. This is sublinear complexity which surpasses that of a regular volumetric time marching by means of an explicit FDTD scheme. Moreover, our algorithm enables a solution of a range of similar problems at low individual cost per problem (section 6.2) and yields a nondeteriorating performance over arbitrarily long time intervals (section 6.5).

2. Governing equations. We simulate the scattering of unsteady waves about a finite obstacle $\Omega \subset \mathbb{R}^3$. The physical quantity of interest u is assumed scalar so that the propagation is governed by the 3D wave equation. In applications, u may represent the acoustic pressure or any Cartesian component of the electric or magnetic field. The scattering problem is formulated as an exterior IBVP on the unbounded

domain $\mathbb{R}^3 \setminus \Omega$:

$$(1a) \quad \square_c u \equiv \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} - \Delta u = 0, \quad (\mathbf{x}, t) \in \mathbb{R}^3 \setminus \Omega \times [0, T],$$

$$(1b) \quad \mathbf{l}_\Gamma u = \phi, \quad (\mathbf{x}, t) \in \Gamma \equiv \partial\Omega \times [0, T],$$

$$(1c) \quad u|_{t=0} = \left. \frac{\partial u}{\partial t} \right|_{t=0} = 0,$$

where c is the constant speed of light (or sound), and the final time T can be chosen arbitrarily. The operator \mathbf{l}_Γ and the data ϕ in (1b) account for the specific boundary condition on the surface of the scatterer $\partial\Omega$. For example, (1b) can be a Dirichlet boundary condition $u|_\Gamma = -u^{\text{inc}}$ or a Neumann boundary condition $\frac{\partial u}{\partial n}|_\Gamma = -\frac{\partial u^{\text{inc}}}{\partial n}$, where u^{inc} is the given incident field. Other, more complex, boundary conditions can also be considered. The initial conditions (1c) are homogeneous, which means that the simulation starts the moment the incident wave reaches the obstacle (inhomogeneous compactly supported initial conditions can be incorporated as well). As u represents only the scattered field, we assume that u is outgoing in the sense of [18]. This means that $u = 0$ for all $t \geq 0$ and $|\mathbf{x}| \geq \rho + ct$, where ρ is sufficiently large so that the scatterer Ω belongs to the ball of radius ρ centered at the origin.

3. Discrete equations. To discretize the IVP (1), we introduce a computational domain $\Omega' \supset \Omega$ that is bounded in space; see Figure 1(a). For convenience, and with no loss of generality, we take Ω' as a sufficiently large parallelepiped. The treatment of the outer boundary $\partial\Omega'$ will be discussed later. The discrete solution to system (1) will be defined on $\Omega' \setminus \Omega$. However, to obtain this solution, we will need to build a discretization on the full space-time domain $\Omega' \times [0, T]$, which includes the scatterer Ω . Thus, we introduce a uniform Cartesian grid with size h in space on Ω' . The grid is aligned with the shape of Ω' , but the physical boundary $\partial\Omega$ does not, generally speaking, have to conform to this grid. To discretize (1a) on this Cartesian grid, we use equally spaced time levels with step τ and employ two finite difference schemes. The first is the standard explicit central difference scheme, which is second order accurate in space and time. The stencil of this scheme contains nine nodes; it is schematically shown in Figure 1(b). The other scheme is compact fourth order accurate in space and time; it is implicit in time with the stencil that contains a total of $3 \times 3 \times 3 \times 3 = 81$ nodes. The fourth order semidiscretization of (1a) in time is as follows:

$$(2) \quad \Delta u^{n+1} - \frac{u^{n+1}}{\theta \tau^2 c^2} = 2 \left(\Delta u^n - \frac{u^n}{\theta \tau^2 c^2} \right) - \left(\Delta u^{n-1} - \frac{u^{n-1}}{\theta \tau^2 c^2} \right) - \frac{1}{\theta} \Delta u^n,$$

where $\theta = \frac{1}{12}$. The resulting modified Helmholtz equation $\Delta u^{n+1} - \kappa^2 u^{n+1} = f^{n+1}$ on the upper time level is subsequently discretized by fourth order compact finite differences in space; see [34] for further details.

Let N_m be the stencil of the scheme (either second order central difference scheme or fourth order compact scheme (2)) centered at the node m . Next, we introduce various subsets of the space-time Cartesian grid on $\Omega' \times [0, T]$ that we will need for the discretization of problem (1). Denote by M^- and M^+ the subsets of grid nodes that belong to the domains $\Omega' \setminus \Omega \times [\tau, T - \tau]$ and $\Omega \times [\tau, T - \tau]$, respectively. In other words, M^- and M^+ contain the nodes that lie outside and inside the scatterer Ω for all moments of time $t \in [\tau, T - \tau]$; the nodes residing precisely on the boundary $\partial\Omega$, if

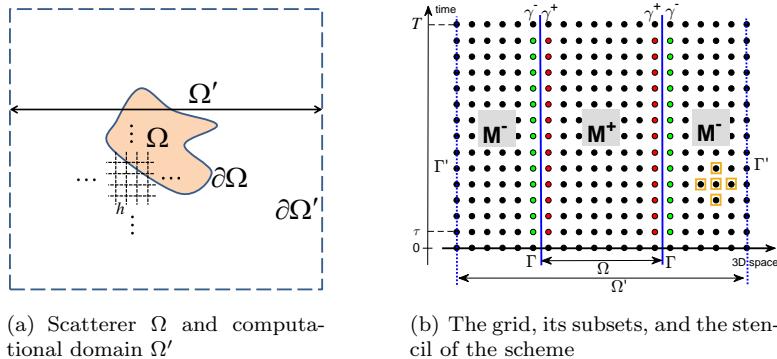


FIG. 1. Computational domain and discretization grid. Panel (b): the sets γ^+ and γ^- are shown in red and green, respectively. The stencil N_m is symbolically shown by orange rectangles. (Figure in color online.)

any, are combined into M^+ . The center of the stencil N_m is always inside the domains $\Omega' \setminus \Omega \times [0, T]$ and $\Omega \times [0, T]$ as long as $m \in M^-$ and $m \in M^+$, respectively.

Let γ^+ denote the subset of nodes that belong to the stencil N_m with $m \in M^-$ and lie inside Ω ; see Figure 1(b). Likewise, let γ^- be the subset of nodes that belong to the stencil N_m with $m \in M^+$ and lie outside Ω . The nodes γ^+ and γ^- form a thin fringe that straddles the continuous space-time boundary Γ of the scatterer. In addition to the sets M^+ and M^- , we introduce $N^+ = \cup_{m \in M^+} N_m$ and $N^- = \cup_{m \in M^-} N_m$ by adding the outermost points of the stencil N_m while its center sweeps over M^+ or M^- . For example, N^- also contains the nodes on the artificial boundary $\partial\Omega'$, the first and the last time levels $t = \{0, T\}$ inside $\Omega' \setminus \Omega$, as well as γ^+ . The sum $N^0 = N^+ \cup N^-$ is the set of all grid nodes touched by the stencil N_m when $m \in M^+ \cup M^-$.

Let the grid function u be defined on N^- and consider discretized equations (1):

$$(3a) \quad \square_c^{(h)} u = 0 \quad \text{on } M^-,$$

$$(3b) \quad u = w^+ \quad \text{on } \gamma^+,$$

$$(3c) \quad u = 0 \quad \text{on } N^- \cap \{t = 0, \tau\},$$

$$(3d) \quad l_{\Gamma'}^{(h)} u = 0 \quad \text{on } \Gamma' \equiv \partial\Omega' \times [0, T].$$

In (3a), $\square_c^{(h)}$ is the 9-point central difference discrete d'Alembert operator or the compact fourth order accurate discrete d'Alembert operator built in accordance with (2). Note that, in general, compact schemes involve modifications to the right-hand side of the equation as well [34]. As, however, (1a) is homogeneous, no modification is needed. Equation (3d) is the far-field radiation boundary condition that we have to include because, unlike in (1), the domain Ω' where we build the discretization (3) is finite. The boundary condition (3d) is homogeneous. Its role is to guarantee that the solution of (3) will be outgoing (see the discussion after (1)), i.e., that there will be no spurious reflections of waves from the outer boundary $\partial\Omega'$ back to the computational domain; see [36].

Condition (3b) is a discrete counterpart of the boundary condition (1b) on the surface of the scatterer. However, for a nonconforming boundary $\partial\Omega$, the analytic boundary data in (1b) and the discrete boundary data in (3b) are specified at different loci in space. Adjusting the spatial grid near the boundary so that $\gamma^+ \subset \partial\Omega \times [0, T]$

may reduce accuracy and impose constraints on the time step due to cut cells [35, 15]. As an alternative, in section 4 we employ the governing equation (1a) to extrapolate the exact boundary data (1b) from Γ to γ^+ and thus obtain w^+ without modifying the grid and with no adverse effect on the accuracy.

4. Description of the method.

4.1. Equation-based extension. Let A be a node of γ^+ at a given time level $t = n\tau$: $A \in \gamma^+ \cap \{t = n\tau\}$. Consider a local Cartesian coordinate system (x', y', z') such that the axis z' is aligned with the normal \mathbf{n}_A to the surface $\partial\Omega$ pointing toward A ; see Figure 2. If there are multiple such normals \mathbf{n}_A (e.g., for nonconvex shapes, see the inset in Figure 2), then the one to be chosen should correspond to the shortest distance h_A from $\partial\Omega$ to A . The origin $O_A = (0, 0, 0)$ of the system (x', y', z') coincides with the foot of the normal \mathbf{n}_A , and the point A lies on the axis z' , i.e., $A = (0, 0, h_A)$. For some simple shapes of Ω , e.g., spheres or ellipsoids, the local Cartesian system (x', y', z') at every point of $\partial\Omega$ can be taken so as to have its axes aligned with the coordinate directions of the corresponding curvilinear orthogonal system (spherical or ellipsoidal). Note, that this local coordinate system does not depend on time and remains the same for all n .

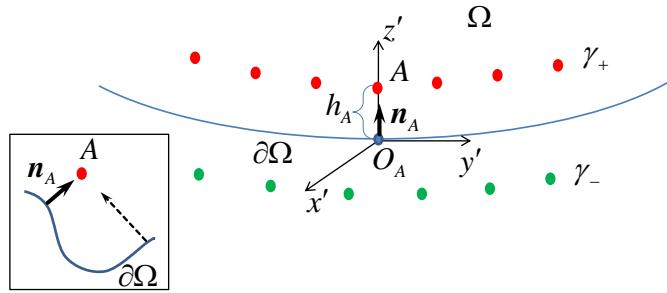


FIG. 2. Local frame of reference for the equation-based extension. The inset: the normal that corresponds to the shortest distance from $\partial\Omega$ to the node A .

The value of w^+ at the node A can be approximated by Taylor's formula:

$$(4) \quad w^+|_A = u|_{O_A} + h_A \frac{\partial u}{\partial z'}\Big|_{O_A} + \frac{h_A^2}{2} \frac{\partial^2 u}{\partial z'^2}\Big|_{O_A} + \dots,$$

where u and its derivatives with respect to z' are evaluated at the foot of the normal O_A at $t = n\tau$. The specific number of terms to be taken in formula (4) will be discussed in section 4.3.

We introduce a finite-dimensional basis composed of functions $\xi_i(\mathbf{x}, t)$ on Γ and expand the solution u and its normal derivative $\partial u / \partial \mathbf{n}$ with respect to this basis:

$$(5a) \quad u|_\Gamma = \sum_{i=1}^N c_{0,i} \xi_i(\mathbf{x}, t),$$

$$(5b) \quad \left. \frac{\partial u}{\partial \mathbf{n}} \right|_\Gamma = \sum_{i=1}^N c_{1,i} \xi_i(\mathbf{x}, t).$$

In practice, formulae (5) can be thought of as approximations rendered by truncated expansions that otherwise converge rapidly, e.g., Chebyshev, Legendre, etc. Future applications of formulae (5) to more complex settings are discussed in section 7.

Higher order normal derivatives in (4) (beyond $\frac{\partial u}{\partial z'}|_{O_A}$) can be obtained using the governing differential equation (1a). For example,

$$(6) \quad \frac{\partial^2 u}{\partial z'^2} \Big|_{O_A} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} \Big|_{O_A} - \frac{\partial^2 u}{\partial x'^2} \Big|_{O_A} - \frac{\partial^2 u}{\partial y'^2} \Big|_{O_A}.$$

Substituting expansion (5a) into the right-hand side of (6) and differentiating it term-wise, we can express the time derivative and tangential derivatives of u via the coefficients $c_{0,i}$, because the basis functions $\xi_i = \xi_i(\mathbf{x}, t)$ are known. To take into account the next term in expansion (4), one needs the third derivative of u . It can be obtained by differentiating equation (1a) with respect to z' :

$$(7) \quad \frac{\partial^2 u}{\partial z'^3} \Big|_{O_A} = \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \frac{\partial u}{\partial z'} \Big|_{O_A} - \frac{\partial^2}{\partial x'^2} \frac{\partial u}{\partial z'} \Big|_{O_A} - \frac{\partial^2}{\partial y'^2} \frac{\partial u}{\partial z'} \Big|_{O_A}.$$

Then, $\partial u / \partial z'$ is substituted on the right-hand side of (7) in the form (5b). Likewise, the fourth derivative can be computed by differentiating (7) and then substituting (6). Altogether, this yields a systematic way of using formula (4) to extend the Cauchy data $(u, \frac{\partial u}{\partial n})$ on the left-hand side of (5) from the continuous boundary Γ to the discrete set of nodes γ^+ . Hereafter, we will refer to it as the equation-based extension. It is applied individually to each point of space-time boundary γ^+ (assuming that none of the derivatives on the right-hand side of formula (4) is singular at O_A).

Combining (4), (5), and (6) (expressions for higher order derivatives, such as (7), may also be needed) and rearranging the terms, we can write

$$(8a) \quad w^+ = \sum_{i=1}^N c_{0,i} w_{0,i}^+ + \sum_{i=1}^N c_{1,i} w_{1,i}^+,$$

where $w_{0,i}^+$ and $w_{1,i}^+$ in (8a) represent the contributions from (5a) and (5b) to (4), respectively. A similar equation-based extension can be developed for w^- on γ^- :

$$(8b) \quad w^- = \sum_{i=1}^N c_{0,i} w_{0,i}^- + \sum_{i=1}^N c_{1,i} w_{1,i}^-.$$

4.2. Expansion of the boundary conditions. Applying expansions (5) to $u|_\Gamma$ and $\partial u / \partial \mathbf{n}|_\Gamma$ in the boundary condition (1b), we recast it as

$$(9) \quad \mathbf{B}_0 \mathbf{c}_0 + \mathbf{B}_1 \mathbf{c}_1 = \boldsymbol{\phi},$$

where \mathbf{B}_0 and \mathbf{B}_1 are square matrices, $\mathbf{c}_0 \equiv (c_{0,1}, c_{0,2}, \dots, c_{0,N})^T$, $\mathbf{c}_1 \equiv (c_{1,1}, c_{1,2}, \dots, c_{1,N})^T$, and $\boldsymbol{\phi}$ is the vector of coefficients obtained by expanding ϕ of (1b) with respect to the basis $\{\xi_i\}$. For a Dirichlet problem, $\mathbf{B}_0 = \mathbf{I}$ and $\mathbf{B}_1 = \mathbf{0}$. For a Neumann problem, $\mathbf{B}_0 = \mathbf{0}$ and $\mathbf{B}_1 = \mathbf{I}$. For other boundary conditions (Robin or more general), neither \mathbf{B}_0 nor \mathbf{B}_1 vanishes. As, however, $u|_\Gamma$ and $\partial u / \partial \mathbf{n}|_\Gamma$ may not be specified at the same time, the coefficients \mathbf{c}_0 in (5a) and \mathbf{c}_1 in (5b) are not available simultaneously. Equation (9) alone cannot determine them unambiguously. Thus, equation-based extension (8a) cannot be used immediately for reconstructing w^+ . The remaining coefficients are determined using simultaneous approximation of the governing equation (1a) and boundary condition (1b) that involves extension (8b).

4.3. Simultaneous approximation. Let u be a solution to system (3), with the boundary data w^+ in (3b) given by (8a). By linearity, it can be represented as a superposition of $2N$ partial solutions $u_{0,i}$, $u_{1,i}$ driven by the boundary data $w_{0,i}^+$, $w_{1,i}^+$, respectively,

$$(10) \quad u = \sum_{i=1}^N c_{0,i} u_{0,i} + \sum_{i=1}^N c_{1,i} u_{1,i} \quad \text{on } \mathbb{N}^-.$$

The coefficients $c_{0,i}$ and $c_{1,i}$ in (10) are still undetermined. Next, we require that

$$(11) \quad u = w^- \quad \text{on } \gamma^-,$$

where w^- is given by (8b). Substituting (8b) and (10) into (11), we have

$$(12) \quad \sum_{i=1}^N c_{0,i} u_{0,i} + \sum_{i=1}^N c_{1,i} u_{1,i} = \sum_{i=1}^N c_{0,i} w_{0,i}^- + \sum_{i=1}^N c_{1,i} w_{1,i}^- \quad \text{on } \gamma^-,$$

where all $u_{0,i}$, $u_{1,i}$, $w_{0,i}^-$, $w_{1,i}^-$ are known.

Condition (11) requires that u of (10) coincide on γ^- , with w^- given by (8b). As both w^+ and w^- are obtained as equation-based extensions of $(u, \frac{\partial u}{\partial n})|_\Gamma$, the resulting equation (12) yields a relation between $u|_\Gamma$ and $\frac{\partial u}{\partial n}|_\Gamma$ enforced by the governing PDE (1a) on $\mathbb{R}^3 \setminus \Omega$, i.e., by its discrete counterpart (3a) on \mathbb{N}^- . However, this relation alone is not sufficient for obtaining the coefficients \mathbf{c}_0 and \mathbf{c}_1 . It needs to be combined with the boundary condition (9). Systems (12) and (9) determine the coefficients \mathbf{c}_0 and \mathbf{c}_1 unambiguously and thus simultaneously approximate both the governing PDE (1a) and boundary condition (1b) at the boundary Γ . Once the coefficients \mathbf{c}_0 and \mathbf{c}_1 have been determined by solving (12) and (9), one can obtain the solution on γ via (8) and on the rest of the grid \mathbb{N}^- by computing with the scheme (3). The proposed algorithm does not involve the approximation of the boundary condition (1b) on the grid.

To maintain the overall consistency, the accuracy of (11), or equivalently, that of the equation-based extension (8), should at least match the accuracy of the finite difference operator $\square_c^{(h)}$. The accuracy of the equation-based extension is determined by the number of terms in the Taylor formula (4). Theoretical estimates obtained for elliptic PDEs [30] require that this number be no less than the sum of the order of the differential operator and the order of accuracy of the scheme that renders its discrete approximation. Our practical experience for both elliptic and hyperbolic equations shows that taking fewer terms is typically sufficient. When (3a) represents a second order discretization, one can keep only the first three terms in (4). For a fourth order scheme (3a), two additional terms are needed in the Taylor formula (4), bringing the total number of terms to five.

In actual computations, the system of equations (9) and (12) is first solved on γ^- in the sense of least squares for the unknown coefficients \mathbf{c}_0 and \mathbf{c}_1 . Then, w^+ is restored on γ^+ by (8a), and the overall finite difference solution is computed on \mathbb{N}^- by time marching the discrete equation (3a).

In the rest of this section, we derive an equivalent formulation of the problem that makes its numerical solution easier. It also appears more amenable to the application of lacunae-based time marching (see section 5), which is of key importance for handling the outer boundary $\partial\Omega'$ (see Figure 1(a)), as well as for achieving the sublinear computational complexity.

4.4. Making the boundary conditions homogeneous. Consider a grid function w' defined on $\mathbb{N}^- = \cup_{m \in \mathbb{M}^-} \mathbb{N}_m$:

$$(13) \quad w' = \begin{cases} w^+ & \text{on } \gamma^+, \\ w^- & \text{on } \gamma^-, \\ 0 & \text{elsewhere on } \mathbb{N}^-, \end{cases}$$

where w^+ , w^- are given by (8). Introduce another grid function v' on \mathbb{N}^- such that

$$(14) \quad u = w' - v',$$

where u is the solution to (3) subject to (11). Then, in accordance with (13) and (14), we have

$$(15) \quad v' = 0 \quad \text{on } \gamma^+ \cup \gamma^-.$$

Moreover, $v' = 0$ on $\mathbb{N}^- \cap t = \{0, \tau\}$ and $\mathcal{L}_{\Gamma'}^{(h)} v' = 0$, because $v' = -u$ at the boundary Γ' , and (3d) holds. Substitution of (14) into (3) yields a linear problem for v' :

$$(16a) \quad \square_c^{(h)} v' = \square_c^{(h)} w' \quad \text{on } \mathbb{M}^-,$$

$$(16b) \quad v' = 0 \quad \text{on } \gamma^+,$$

$$(16c) \quad v' = 0 \quad \text{on } \mathbb{N}^- \cap \{t = 0, \tau\},$$

$$(16d) \quad \mathcal{L}_{\Gamma'}^{(h)} v' = 0 \quad \text{on } \Gamma' \equiv \partial \Omega' \times [0, T].$$

The requirement (11) translates into

$$(16e) \quad v' = 0 \quad \text{on } \gamma^-.$$

Equations (16b) and (16e) together are equivalent to (15).

Problem (16) has the exact same structure as problem (3) except that the inhomogeneous boundary condition (3b) on γ^+ has been replaced with the homogeneous boundary condition (16b), while instead of the homogeneous governing equation (3a) we have the inhomogeneous equation (16a). Note that the right-hand side $\square_c^{(h)} w'$ of (16a) depends on the undetermined coefficients \mathbf{c}_0 and \mathbf{c}_1 , because w^+ and w^- in (13) are specified via (8). This right-hand side is defined on \mathbb{M}^- , but may differ from zero only next to the boundary Γ , because $w' = 0$ on most of \mathbb{N}^- ; see (13). The solutions to (3) and (16) are converted into one another by means of (14).

4.5. Problem formulation on the entire Ω' . We introduce the functions w and v on the entire grid \mathbb{N}^0 :

$$(17) \quad w = \begin{cases} w' & \text{on } \mathbb{N}^-, \\ 0 & \text{on } \mathbb{N}^0 \setminus \mathbb{N}^-, \end{cases} \quad v = \begin{cases} v' & \text{on } \mathbb{N}^-, \\ 0 & \text{on } \mathbb{N}^0 \setminus \mathbb{N}^-. \end{cases}$$

Applying the discrete d'Alembert operator $\square_c^{(h)}$ to v , recalling (16), and taking into account that $v = 0$ on $\mathbb{N}^0 \setminus \mathbb{N}^-$, we conclude that v solves the following AP on \mathbb{N}^0 :

$$(18a) \quad \square_c^{(h)} v = \mu_{\mathbb{M}^-} \square_c^{(h)} w \quad \text{on } \mathbb{M}^- \cup \mathbb{M}^+,$$

$$(18b) \quad v = 0 \quad \text{on } \mathbb{N}^0 \cap \{t = 0, \tau\},$$

$$(18c) \quad \mathcal{L}_{\Gamma'}^{(h)} v = 0 \quad \text{on } \Gamma' \equiv \partial \Omega' \times [0, T].$$

In (18a), $\mu_{\mathbb{M}^-}$ is the characteristic function of the set \mathbb{M}^- . According to (17) and (15), $v = 0$ on \mathbb{N}^+ . Hence, the right-hand side of (18a) may indeed differ from zero only on \mathbb{M}^- . In addition to the AP (18), formula (16e) requires that

$$(19) \quad v = 0 \quad \text{on } \gamma^-.$$

Once the right-hand side of (18a) is written via the undetermined coefficients \mathbf{c}_0 and \mathbf{c}_1 , then (19) yields a constraint equivalent to (12); see section 4.6.

Compared to problem (3) or (16), the AP (18) is much easier to solve. Indeed, the domain Ω' is a parallelepiped, and the grid \mathbb{N}^0 is Cartesian. The boundary condition (3b) or (16b) that follows the geometry of Γ is replaced with a source term to equation (18a). Moreover, problem (18) allows one to apply the lacunae-based time marching (see section 5), which enables accurate treatment of the artificial outer boundary Γ' and helps substantially reduce the computational complexity.

4.6. Solution algorithm. Expansion (8) shows that the grid function w' of (13) and thus w of (17) can be represented as

$$(20) \quad w = \sum_{i=1}^N c_{0,i} w_{0,i} + \sum_{i=1}^N c_{1,i} w_{1,i}.$$

Accordingly, the AP (18) can be partitioned into $2N$ subproblems that correspond to the individual terms from the right-hand side of (20) substituted into the right-hand side of (18a):

$$(21) \quad \begin{cases} \square_c^{(h)} v_{0,i} = \mu_{\mathbb{M}^-} \square_c^{(h)} w_{0,i} & \text{on } \mathbb{M}^- \cup \mathbb{M}^+, \\ v_{0,i} = 0 & \text{on } \mathbb{N}^0 \cap \{t = 0, \tau\}, \\ l_{\Gamma'}^{(h)} v_{0,i} = 0 & \text{on } \Gamma' \equiv \partial \Omega' \times [0, T], \end{cases} \quad \begin{cases} \square_c^{(h)} v_{1,i} = \mu_{\mathbb{M}^-} \square_c^{(h)} w_{1,i} & \text{on } \mathbb{M}^- \cup \mathbb{M}^+, \\ v_{1,i} = 0 & \text{on } \mathbb{N}^0 \cap \{t = 0, \tau\}, \\ l_{\Gamma'}^{(h)} v_{1,i} = 0 & \text{on } \Gamma' \equiv \partial \Omega' \times [0, T]. \end{cases}$$

Constraint (19) then becomes

$$(22) \quad \sum_{i=1}^N c_{0,i} v_{0,i} + \sum_{i=1}^N c_{1,i} v_{1,i} = 0 \quad \text{on } \gamma^-,$$

where $v_{0,i}$, $v_{1,i}$ are solutions of (21). Equation (22) can be recast as

$$(23) \quad \mathbf{Q}_0 \mathbf{c}_0 + \mathbf{Q}_1 \mathbf{c}_1 = 0,$$

where the matrices of the operators \mathbf{Q}_0 and \mathbf{Q}_1 are composed of columns $v_{0,i}$ and $v_{1,i}$ that are traces of the solutions to the respective subproblems (21) on the grid set γ^- :

$$(24) \quad \mathbf{Q}_0 = [v_{0,1}|_{\gamma^-}, v_{0,2}|_{\gamma^-}, \dots, v_{0,N}|_{\gamma^-}], \quad \mathbf{Q}_1 = [v_{1,1}|_{\gamma^-}, v_{1,2}|_{\gamma^-}, \dots, v_{1,N}|_{\gamma^-}].$$

Equation (23) is equivalent to (12). The system of equations (23) and (9) determines the coefficients \mathbf{c}_0 and \mathbf{c}_1 . The matrices \mathbf{Q}_0 and \mathbf{Q}_1 are $|\gamma^-| \times N$, where $|\gamma^-|$ is the number of nodes in γ^- , and N is the dimension of the basis on Γ ; see (5). For sufficiently smooth boundaries and data, N can be chosen small so that $|\gamma^-| > N$ and system (23) is overdetermined. We solve it by least squares. However, as long as the original IBVP (1) has a unique solution, the quadratic cost function at the minimum will be zero within the accuracy of the discrete approximation.

The overall algorithm can now be summarized as follows:

1. Introduce the basis on Γ ; see formulae (5).
2. Recast the boundary condition (1b) into the form (9) or, in the case of a Dirichlet or Neumann boundary condition, compute the corresponding vector, \mathbf{c}_0 or \mathbf{c}_1 , explicitly.
3. Perform the equation-based extension (8) for each basis function ξ_i .
4. Solve $2N$ subproblems (21) and form the matrices \mathbf{Q}_0 and \mathbf{Q}_1 of (24) from the traces of the respective solutions on γ^- .
5. Solve the resulting system (23) and (9) for the undetermined coefficients \mathbf{c}_0 or/and \mathbf{c}_1 .
6. Once both \mathbf{c}_0 and \mathbf{c}_1 are available, compute w^+ via (8a), substitute it into (3b), and time march the finite difference IBVP (3) to obtain the discrete solution u .
7. Alternatively, one can save the solutions $v_{0,i}$, $v_{1,i}$ to subproblems (21) at the space-time locations of interest in \mathbb{N}^- and then reconstruct the overall solution u at these points using the same linear superposition as on the left-hand side of (22) with a substitution of (17) and (14).

This algorithm is equivalent to the method of difference potentials [32, supplement].

5. Lacunae-based time marching. A key limitation of the proposed methodology is that the boundary $\Gamma = \partial\Omega \times [0, T]$ extends as T becomes larger. Accordingly, the vertical dimension of the matrices \mathbf{Q}_0 and \mathbf{Q}_1 given by (24) increases linearly with time, and the solution of the boundary equation (23) becomes computationally infeasible already at modest T . To address this issue, we employ lacunae-based time marching. It replaces the solution algorithm described in section 4.6 by its equivalent version that solves the problem in a sequence of time increments. The equivalence is enabled by the Huygens' principle, i.e., the presence of lacunae in the solutions of the wave equation [28].

We partition the time span T into K subintervals, each of duration

$$(25) \quad T_0 \geq \frac{1}{c} \operatorname{diam} \Omega,$$

so that $T = KT_0$. Hereafter, T_0 will be considered fixed, while the increase of T will be accommodated by increasing K . The grid sets γ^+ and γ^- can be represented as sums of “shorter” subsets $\gamma_k^\pm \equiv \gamma^\pm \cap [(k-1)T_0, kT_0]$:

$$\gamma^+ = \bigcup_{k=1}^K \gamma_k^+ \quad \text{and} \quad \gamma^- = \bigcup_{k=1}^K \gamma_k^-.$$

Let $\Gamma_k \equiv \Gamma \cap [(k-1)T_0, kT_0]$, and let $\{\xi_i(\mathbf{x}, t)\}$ be the basis on Γ_k for expanding u and $\frac{\partial u}{\partial n}$ in the form (5). The basis $\{\xi_i(\mathbf{x}, t)\}$ is assumed to be independent of k . Applying the equation-based extension to each $\xi_i(\mathbf{x}, t)$ for a given Γ_k , we have (cf. formulae (8))

$$(26a) \quad w^{(+,k)} = \sum_{i=1}^N c_{0,i}^{(k)} w_{0,i}^+ + \sum_{i=1}^N c_{1,i}^{(k)} w_{1,i}^+,$$

$$(26b) \quad w^{(-,k)} = \sum_{i=1}^N c_{0,i}^{(k)} w_{0,i}^- + \sum_{i=1}^N c_{1,i}^{(k)} w_{1,i}^-.$$

The grid functions $w^{(+,k)}$ of (26a) and $w^{(-,k)}$ of (26b) are defined on γ_k^+ and γ_k^- , respectively. The quantities $w_{0,i}^\pm$, $w_{1,i}^\pm$ in (26) do not depend on k because they

correspond to the i th basis function $\xi_i(\mathbf{x}, t)$, and the basis functions are the same for all $k = 1, \dots, K$. However, the coefficients $c_{0,i}^{(k)}$ and $c_{1,i}^{(k)}$ in (26) may depend on k . We also extend the functions $w^{(+,k)}$ and $w^{(-,k)}$ to the entire γ^+ and γ^- , respectively, so that $w^{(+,k)}|_{\gamma_{k'}^+} = 0$ and $w^{(-,k)}|_{\gamma_{k'}^-} = 0$ for $k' \neq k$. Then, the equation-based extension (8) on γ^+ and γ^- is replaced with the sum of contributions (26):

$$w^+ = \sum_{k=1}^K w^{(+,k)}, \quad w^- = \sum_{k=1}^K w^{(-,k)},$$

where $w^{(\pm,k)} \neq 0$ only for $t \in [(k-1)T_0, kT_0]$. According to (13), (17), and (20), for each k we have

$$(27) \quad w^{(k)} = \sum_{i=1}^N c_{0,i}^{(k)} w_{0,i} + \sum_{i=1}^N c_{1,i}^{(k)} w_{1,i}.$$

Thus, the same partition in time holds for the grid function $w = \sum_{k=1}^K w^{(k)}$, and consequently, the AP (18) gets split into a combination of subproblems:

$$(28a) \quad \square_c^{(h)} v^{(k)} = \mu_{\mathbb{M}^-} \square_c^{(h)} w^{(k)} \quad \text{on } \mathbb{M}^- \cup \mathbb{M}^+,$$

$$(28b) \quad v^{(k)} = 0 \quad \text{on } \mathbb{N}^0 \cap \{t = 0, \tau\},$$

$$(28c) \quad l_{\Gamma'}^{(h)} v^{(k)} = 0 \quad \text{on } \Gamma' \equiv \partial\Omega' \times [0, T],$$

with the overall solution given by $v = \sum_{k=1}^K v^{(k)}$. Each partial solution $v^{(k)}$ in the sum has the following properties:

1. $v^{(k)} \equiv 0$ for $t < (k-1)T_0$ because the right-hand side of (28a) is zero before $t = (k-1)T_0$.
2. $v^{(k)} \approx 0$ on \mathbb{N}^+ for $t > (k+1)T_0$ under condition (25).
3. Combining 1 and 2, we conclude that $v^{(k)} \neq 0$ on \mathbb{N}^+ only for $(k-1)T_0 \leq t \leq (k+1)T_0$.

Property 2 on this list is of key importance. It holds because the discrete system (28) approximates, in free space, the continuous wave equation with a source term that is compactly supported in both space and time. Indeed, for the grid function (27) we have $w^{(k)} \neq 0$ only for $t \in [(k-1)T_0, kT_0]$ in time and at the nodes next to the boundary $\partial\Omega$ in space, the latter due to (13) and (17). A continuous solution to the 3D wave equation with a compactly supported source is known to have a (secondary) lacuna in the sense of Petrowsky [28]. The lacuna is a manifestation of the Huygens' principle. It is a region of space-time behind the aft (trailing) fronts of the waves propagating away from the source; see Figure 3(a). The solution inside the lacuna is zero. Thus, once the source compactly supported on Ω (including the boundaries) ceases to operate, the waves will completely leave the domain Ω after the time T_0 elapses (provided that (25) holds).

Earlier studies of the phenomenon of lacunae in the discrete context have shown that lacunae do exist in numerical solutions and can thus be exploited for the development of numerical methods. As the solution inside a lacuna is zero, one might expect that the numerical solution will be zero within the accuracy of the specific discrete approximation. Remarkably, however, there is superconvergence inside the lacunae (see [23, sect. 8.4]), so that property 2 in the previous list holds with accuracy higher than $\mathcal{O}(h^2)$ for the second order scheme or $\mathcal{O}(h^4)$ for the fourth order scheme (3a).

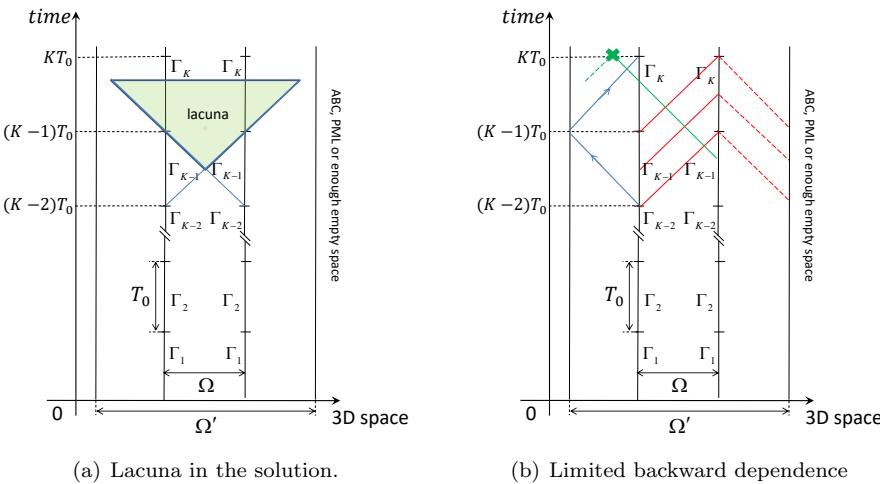


FIG. 3. Partition in time and lacuna of the solution to the wave equation.

The solution $v^{(k)}$ of problem (28) inherits its linear superposition form from (27):

$$(29) \quad v^{(k)} = \sum_{i=1}^N c_{0,i}^{(k)} v_{0,i} + \sum_{i=1}^N c_{1,i}^{(k)} v_{1,i}.$$

It is fully determined by the coefficients $c_{0,i}^{(k)}$ and $c_{1,i}^{(k)}$. The individual solutions $v_{0,i}$ and $v_{1,i}$ on the right-hand side of (29) correspond to the extensions $w_{0,i}$ and $w_{1,i}$ of the corresponding basis functions on Γ_k ; see (26) and (27). They are invariant with respect to the translations in time by increments of T_0 because the basis $\{\xi_i(\mathbf{x}, t)\}$ is the same for each Γ_k (and, therefore, the index k can be omitted). Properties 1–3 on the previous page apply to each term in the sum (29). Then, taking into account the translational invariance in time, for $v_{0,i}$ and $v_{1,i}$ on \mathbb{N}^+ we can write

$$(30) \quad v_{0,i} = \begin{cases} v_{0,i}^{(I)} & 0 \leq t < T_0, \\ v_{0,i}^{(II)} & T_0 \leq t < 2T_0, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad v_{1,i} = \begin{cases} v_{1,i}^{(I)} & 0 \leq t < T_0, \\ v_{1,i}^{(II)} & T_0 \leq t < 2T_0, \\ 0 & \text{otherwise.} \end{cases}$$

In (30), $v_{0,i}^{(I)}$ and $v_{1,i}^{(I)}$ are solutions to the respective APs (28) on the interval of time where the sources still operate, while $v_{0,i}^{(II)}$ and $v_{1,i}^{(II)}$ are solutions after the sources cease to operate but before the waves completely leave \mathbb{N}^+ . Otherwise, the grid set \mathbb{N}^+ entirely falls into the lacuna, where the solution is zero. Hence, the APs (28) never need to be integrated beyond the terminal time $2T_0$.

Let us now consider the boundary equation (19) on a given subset γ_k^- , $k = 1, \dots, K$, rather than on the entire γ^- . By causality, we have $v|_{\gamma_k^-} = \sum_{k'=1}^k v^{(k')}$, because subproblems (28) for $k' > k$ have not been initiated yet. Moreover, the subproblems with $k' < k - 1$ do not contribute to the sum either because the subset γ_k^- falls into the lacunae of the respective solutions. As a result, only two partial solutions are nonzero on γ_k^- , and instead of (19) we can write

$$(31) \quad v|_{\gamma_k^-} = v^{(k-1)} \left(\mathbf{c}_0^{(k-1)}, \mathbf{c}_1^{(k-1)} \right) \Big|_{\gamma_k^-} + v^{(k)} \left(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)} \right) \Big|_{\gamma_k^-} = 0.$$

The grid functions $v^{(k-1)}$ and $v^{(k)}$ in (31) are determined by the coefficients $\mathbf{c}_0^{(k-1)}$, $\mathbf{c}_1^{(k-1)}$ and $\mathbf{c}_0^{(k)}$, $\mathbf{c}_1^{(k)}$, respectively. Thus, (31) provides a recurrence relation between the coefficients from two consecutive time intervals T_0 . In the matrix-vector notation, (31) reads (cf. formula (23))

$$(32) \quad \mathbf{Q}_0^{(I)} \mathbf{c}_0^{(k-1)} + \mathbf{Q}_1^{(I)} \mathbf{c}_1^{(k-1)} + \mathbf{Q}_0^{(II)} \mathbf{c}_0^{(k)} + \mathbf{Q}_1^{(II)} \mathbf{c}_1^{(k)} = 0,$$

where the matrices of the boundary operators are given by (cf. formula (24))

(33a)

$$\mathbf{Q}_0^{(I)} = \left[v_{0,1}^{(I)}|_{\gamma^-}, v_{0,2}^{(I)}|_{\gamma^-}, \dots, v_{0,N}^{(I)}|_{\gamma^-} \right], \quad \mathbf{Q}_1^{(I)} = \left[v_{1,1}^{(I)}|_{\gamma^-}, v_{1,2}^{(I)}|_{\gamma^-}, \dots, v_{1,N}^{(I)}|_{\gamma^-} \right],$$

(33b)

$$\mathbf{Q}_0^{(II)} = \left[v_{0,1}^{(II)}|_{\gamma^-}, v_{0,2}^{(II)}|_{\gamma^-}, \dots, v_{0,N}^{(II)}|_{\gamma^-} \right], \quad \mathbf{Q}_1^{(II)} = \left[v_{1,1}^{(II)}|_{\gamma^-}, v_{1,2}^{(II)}|_{\gamma^-}, \dots, v_{1,N}^{(II)}|_{\gamma^-} \right].$$

The matrix equation (32) can be thought of as block bidiagonal. The column vectors in the matrices (33) correspond to notation (30) and have dimension $|\gamma_k^-|$, where $|\gamma_k^-|$ is the number of nodes in the grid set γ_k^- (the same for all k). To compute the matrices (33) one needs to solve numerically $2N$ APs (28), each over the interval $2T_0$, and save the traces of the respective solutions in the matrices (33a) (for $0 \leq t < T_0$) and (33b) (for $T_0 \leq t < 2T_0$). The overall algorithm can now be summarized as follows:

1. Precompute the matrices (33).
2. Initialization step, $k = 1$. Obtain the values of the undetermined coefficients $\mathbf{c}_0^{(1)}$ and $\mathbf{c}_1^{(1)}$ using the algorithm of section 4.6. This requires taking into account (9) for Γ_1 .
3. For each $k = 2, 3, \dots, K$, solve the boundary operator equation (32) combined with (9) for Γ_k in the sense of least squares with respect to the undetermined coefficients $\mathbf{c}_0^{(k)}$, $\mathbf{c}_1^{(k)}$.

The recursion in stage 3 can be interpreted as time marching with step T_0 .

The system (32) and (9) that is solved recursively can be written in the matrix form,

$$(34) \quad \begin{bmatrix} \mathbf{Q}_0^{(II)} & \mathbf{Q}_1^{(II)} \\ \mathbf{B}_0 & \mathbf{B}_1 \end{bmatrix} \begin{bmatrix} \mathbf{c}_0^{(k)} \\ \mathbf{c}_1^{(k)} \end{bmatrix} = - \begin{bmatrix} \mathbf{Q}_0^{(I)} & \mathbf{Q}_1^{(I)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{c}_0^{(k-1)} \\ \mathbf{c}_1^{(k-1)} \\ \phi^{(k)} \end{bmatrix}.$$

For a Dirichlet boundary condition (1b), $\mathbf{c}_0^{(k)} = \phi^{(k)}$, and system (34) simplifies to $\mathbf{Q}_1^{(II)} \mathbf{c}_1^{(k)} = -\mathbf{Q}_0^{(I)} \mathbf{c}_0^{(k-1)} - \mathbf{Q}_1^{(I)} \mathbf{c}_1^{(k-1)} - \mathbf{Q}_0^{(II)} \phi^{(k)}$. For Neumann, $\mathbf{c}_1^{(k)} = \phi^{(k)}$, and (34) reduces to $\mathbf{Q}_0^{(II)} \mathbf{c}_0^{(k)} = -\mathbf{Q}_0^{(I)} \mathbf{c}_0^{(k-1)} - \mathbf{Q}_1^{(I)} \mathbf{c}_1^{(k-1)} - \mathbf{Q}_1^{(II)} \phi^{(k)}$. In general, the dimension of the system matrix on the left-hand side of (34) is $(|\gamma_k^-| + N) \times 2N$. Let \mathcal{QR} be its factorization, where \mathcal{Q} is orthogonal $(|\gamma_k^-| + N) \times 2N$ and \mathcal{R} is upper triangular $2N \times 2N$. Then, the solution to system (34) is given by

$$(35) \quad \begin{bmatrix} \mathbf{c}_0^{(k)} \\ \mathbf{c}_1^{(k)} \end{bmatrix} = -\mathcal{R}^{-1} \mathcal{Q}^* \begin{bmatrix} \mathbf{Q}_0^{(I)} & \mathbf{Q}_1^{(I)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{c}_0^{(k-1)} \\ \mathbf{c}_1^{(k-1)} \\ \phi^{(k)} \end{bmatrix}.$$

On the right-hand side of (35), a $2N \times 3N$ matrix is applied to a vector of length $3N$. Of key importance is that this matrix is fixed, i.e., it does not depend on k . As

such, it can be precomputed right after computing the operators (33). Consequently, advancing the solution along the boundary by time step T_0 costs $\mathcal{O}(N^2)$ arithmetic operations, where N is the dimension of the basis $\{\xi_i\}$ at the boundary Γ ; see (5). *Remarkably, this cost does not depend on the dimension of the grid at all.*

Moreover, the matrices (33) do not depend on the specific boundary condition (1b) and, once computed, accommodate any boundary condition. Then, solving for a different boundary condition, e.g., another impinging wave or surface impedance, is inexpensive. It only requires a new recursive update (35) and may or may not require a new QR factorization depending on whether the matrices B_0 and B_1 have changed.

Knowing the solution only at the boundary may often be sufficient (e.g., for calculating scattering cross-sections). If, however, it needs to be known beyond the boundary, we have two options to compute it that are similar to steps 6 and 7 of the algorithm of section 4.6. We can obtain the boundary data $w^{(+,K-1)}$ and $w^{(+,K)}$ using the respective coefficients $c_0^{(K-1)}$, $c_1^{(K-1)}$ and $c_0^{(K)}$, $c_1^{(K)}$ and then compute the solution of the wave equation on the grid by means of the core scheme (3a) at $t = T \equiv KT_0$, and no further away from Ω than $\text{diam } \Omega$; see Figure 3(b). Alternatively, we can keep the values of the solutions $v_{0,i}$ and $v_{1,i}$ at the locations of interest outside the scatterer. Upon completion of step 3, the solution at those locations can be reconstructed using the same sum as in (31) and subsequently employing (17) and (14).

In our algorithm, we use the conventional FDTD only for computing the matrices of the boundary operators (33), which requires solving $2N$ APs (28) over the interval $2T_0 \ll T$. The number $2N$ is twice the dimension of the basis used for expansion (5). While typically much smaller than the dimension $|\gamma_k^-|$, the number $2N$ may not always be small itself, depending on the required accuracy of the data representation at the boundary. However, the $2N$ APs (28) can be efficiently solved in parallel because different APs are completely independent from one another (section 6.4).

Another important consideration is that, traditionally, some artificial boundary conditions must be set on Γ' to enable the reflectionless propagation of waves toward infinity; see formula (3d). The use of lacunae for computations allows for a simple and elegant alternative approach. The auxiliary domain Ω' can be chosen sufficiently large so that the waves reflected off the outer boundary $\partial\Omega'$ back toward the interior will not reach the boundary of the scatterer $\partial\Omega$ during the time interval $2T_0$, i.e.,

$$(36) \quad \text{dist}(\partial\Omega', \partial\Omega) > cT_0;$$

see Figure 3(b). For computing the solution near $\partial\Omega$, this is equivalent to the exact treatment of the artificial outer boundary regardless of the actual boundary condition set at Γ' . We have used this approach in [26, 23] for the wave and Maxwell's equations.

6. Numerical simulations. We consider scattering about a sphere of radius R_0 and three different prolate spheroids with aspect ratios $4/3$, 2 , and 4 .

6.1. Computational setting for the sphere. There is a well-known series solution for time-harmonic scattering of a plane wave about a sphere. However, to avoid a pure periodicity in time, we employ the sum of two plane waves with two incommensurate frequencies ω_1, ω_2 . A Dirichlet problem for the scattered field in the

case when $c = 1$ is formulated as follows (cf. problem (1)):

$$(37a) \quad \frac{\partial^2 u}{\partial t^2} = \Delta u, \quad (\mathbf{x}, t) \in \mathbb{R}^3 \setminus \Omega \times [0, T],$$

$$(37b) \quad u|_{\Gamma} = -\cos(\mathbf{k}_1 \cdot \mathbf{x} - \omega_1 t) - \cos(\mathbf{k}_2 \cdot \mathbf{x} - \omega_2 t), \quad (\mathbf{x}, t) \in \Gamma,$$

$$(37c) \quad u|_{t=0} = \left. \frac{\partial u}{\partial t} \right|_{t=0} = 0.$$

The boundary condition (37b) implies that the sum of the incident and scattered field on the surface of the scatterer is zero. This corresponds to sound-hard scattering in acoustics or a perfect electric conductor (PEC) in electromagnetism. The ratio of the frequencies is chosen to be $\omega_1/\omega_2 = \sqrt{2}$. The wave vectors \mathbf{k}_1 and \mathbf{k}_2 are parallel, and as $c = 1$, we have $k_{1,2} = \omega_{1,2}$. The exact solution to problem (37) involves two terms:

$$(38) \quad u^{\text{ref}} = \operatorname{Re} \{u^{\text{sc}}(\mathbf{x}, t, \omega_1)\} + \operatorname{Re} \{u^{\text{sc}}(\mathbf{x}, t, \omega_2)\},$$

where

$$(39) \quad u^{\text{sc}}(\mathbf{x}, t, \omega) = -e^{-i\omega t} \sum_{l=0}^{\infty} (2l+1)i^l P_l(\cos \theta) \frac{j_l(kR_0)}{h_l(kR_0)} h_l(k|\mathbf{x}|),$$

and θ is the angle between \mathbf{k} and \mathbf{x} . In (39), $j_l(\cdot)$, $h_l(\cdot)$, and $P_l(\cdot)$ denote the spherical Bessel and Hankel functions and Legendre polynomials, respectively, where

$$P_l(\mathbf{n}\mathbf{n}') = \frac{4\pi}{2l+1} \sum_{m=-l}^l Y_{lm}^*(\mathbf{n}') Y_{lm}(\mathbf{n}),$$

Y_{lm} are spherical harmonics, $\mathbf{n} = \mathbf{x}/\|\mathbf{x}\|$, and $\mathbf{n}' = \mathbf{k}/\|\mathbf{k}\|$.

As the boundary condition can be changed easily, we consider not only the Dirichlet condition (37b), but the corresponding Neumann and Robin boundary conditions as well. The normal derivatives involved are derived by differentiating series (39).

For the first set of simulations, the scatterer is a unit sphere, $R_0 = 1$. The frequencies are $\omega_1 = 3$ and $\omega_2 = \omega_1/\sqrt{2}$. We are interested in obtaining the solution near the scatterer, in the spherical layer $R_0 < |\mathbf{x}| < R_1$ where $R_1 = 1.5$.

The operators (33) are computed by solving $2N$ independent APs (28) over the time $2T_0$. Each of those APs is driven by its own compactly supported source term and formulated on the auxiliary cube $\Omega' = [-R_1, R_1]^3$ with no cavity for the scatterer Ω ; see Figure 4. At the outer boundary $\partial\Omega'$, we employ a perfectly matched layer (PML) [11] that has a quadratic damping profile. Its width was chosen ≈ 1.65 to achieve the best performance.¹ The approximate size of Ω^{PML} is $6.3 \times 6.3 \times 6.3$.

Alternatively, the lacunae-based time marching treats the artificial outer boundary exactly. It requires a somewhat larger augmented domain that we choose as a Cartesian cube $\Omega^{\text{aug}} = [-R_2, R_2]^3$. Its size R_2 is determined as in (36), specifically, the waves that travel outward from $r = R_0$ to $\partial\Omega^{\text{aug}}$ and get reflected back should not reach the boundary $r = R_1$ during the time interval $2T_0$:

$$(40) \quad 2R_2 - R_0 - R_1 > 2cT_0.$$

Hence, we can specify a zero Dirichlet boundary condition at $\partial\Omega^{\text{aug}}$ because it does not affect the solution on $R_0 \leq |\mathbf{x}| \leq R_1$. In practice, R_2 needs to be taken a bit

¹Layers narrower than the wavelength could not eliminate the error growth.

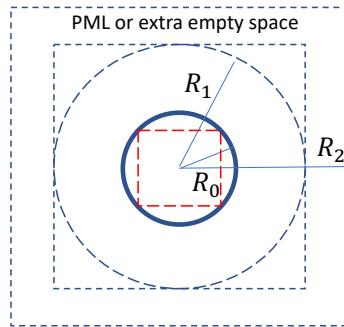


FIG. 4. A two-dimensional schematic of the computational domain. (Figure in color online.)

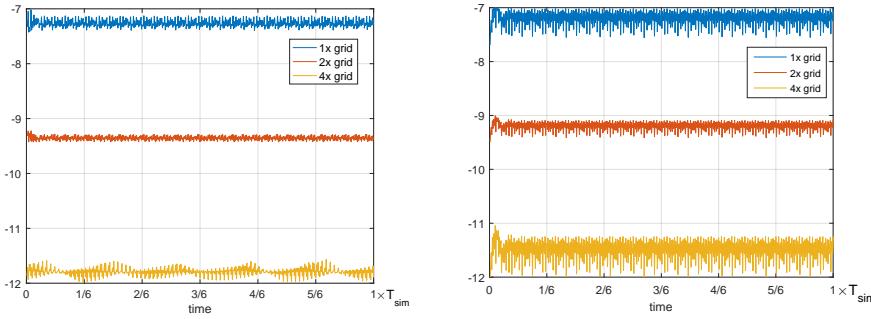
larger than prescribed by (40) to accommodate the transients in the formation of the discrete lacunae. This yields the size of Ω^{aug} as $9.6 \times 9.6 \times 9.6$.

The coarsest grid is uniform Cartesian with size $h = 2/15$ in all space directions and time step $\tau = h/3$, which satisfies the CFL condition. The corresponding grid dimensions in space are $47 \times 47 \times 47$ for Ω^{PML} and $73 \times 73 \times 73$ for Ω^{aug} . The wave equation (1a) is discretized by either the standard second order accurate central difference scheme or the fourth order compact scheme (2). The PML is discretized as in [11] with second order accuracy. In the fourth order case, we use lacunae for termination at the artificial outer boundary. To study the grid convergence outside the scatterer, we introduce a sequence of finer grids, each time dividing the spatial size h and time step τ in half. The grids in this sequence are denoted by $1\times$, $2\times$, and $4\times$.

The basis functions ξ_i in expansions (5) are products of spherical harmonics in space and Chebyshev polynomials in time: $\xi_i(\mathbf{x}, t) = Y_{lm}(\theta, \varphi)T_n(t)$. Y_{lm} are eigenfunctions of the Beltrami operator on the sphere, which provides extra convenience. To analyze the grid convergence for a given solution (38)–(39), the dimension of the basis $\{\xi_i\}$ is fixed. It is chosen so as to make sure that the accuracy of the truncated series representation (5) will be no worse than the finite difference accuracy on the finest grid. Specifically, we took $l = 0, \dots, L_{\max}$ and $n = 0, \dots, N_{\max}$ with $L_{\max} = 12$ and $N_{\max} = 10$. This has proven sufficient for all the grids used in our simulations.

6.2. Grid convergence for the sphere. Figures 5 and 6 show the ℓ_∞ error of the second order numerical solution, $\|u - u^{\text{ref}}\|_\infty$ over $R_0 < |\mathbf{x}| < R_1$. The proposed method converges with the design rate. The series (39) that yields reference solution (38) was truncated at $l = L_{\max} + 5$ to ensure its superior accuracy. The simulation time was taken very long, $T_{\text{sim}} = 6600R_0$. The specific Robin boundary condition was taken as $u + \frac{\partial u}{\partial n}|_\Gamma = 0$. We emphasize that the computations with three different boundary conditions were done without recomputing the operators (33).

For the fourth order scheme (2), the grid convergence is shown in Figure 7; it also reaches the design rate. The fourth order scheme requires normal derivatives up to order four in the Taylor's formula (4). The error of the boundary expansions (5) is always smaller than the error on the grid, because according to Figures 5–7 it does not hamper the grid convergence. The actual values of the error averaged over the simulation interval T_{sim} are presented in Tables 1 and 2. The rates in Tables 1 and 2 are defined as $\log_2(\text{coarser grid error}) - \log_2(\text{finer grid error})$. If the grid is refined by a factor of two, this quantity directly approximates the order of accuracy.



(a) Matrices (33) computed by lacunae-based termination
 (b) Matrices (33) computed using the PML

FIG. 5. Grid convergence in the case of a Dirichlet boundary condition (1b) and second order accurate central difference scheme—binary logarithm of ℓ_∞ error versus time.

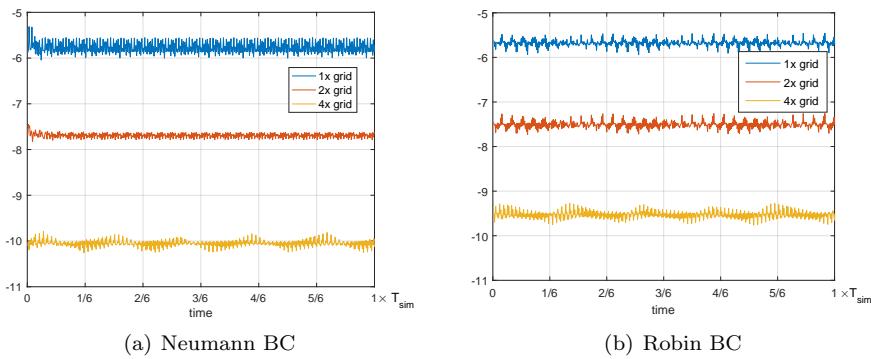


FIG. 6. Grid convergence in the case of Neumann and Robin boundary conditions and a second order accurate central difference scheme—binary logarithm of ℓ_∞ error versus time. Operators (33) are computed using lacunae-based termination.

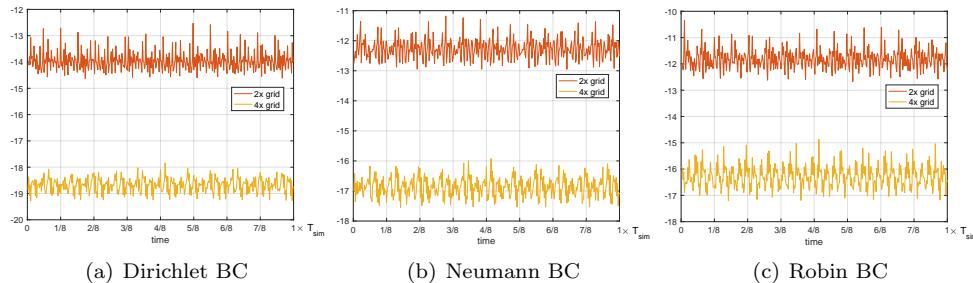


FIG. 7. Grid convergence in the case of a fourth order discretization scheme; see (2). Binary logarithm of ℓ_∞ error versus time.

To compute the operators (33), we have used both a PML and exact lacunae-based termination at the artificial outer boundary. The performance is compared in the second order case, because the PML [11] is discretized only to second order accuracy. We observed no significant difference between the two approaches for either the recursive time marching (35) or final accuracy on the grid. In Figure 5(a), we show the grid convergence in the case of exact lacunae-based termination, while Figure 5(b) corresponds to the PML. Apart from a somewhat stronger “wiggling” in Figure 5(b),

TABLE 1

ℓ_∞ error on the grid averaged over $0 \leq t \leq T_{\text{sim}}$ and grid convergence rates for the second order central difference scheme. The rate is \log_2 (coarser grid error) – \log_2 (finer grid error).

Grid	Dirichlet		Neumann		Robin	
	Average error	Rate	Average error	Rate	Average error	Rate
1×	$6.55 \cdot 10^{-3}$	-	$1.84 \cdot 10^{-2}$	-	$2.02 \cdot 10^{-2}$	-
2×	$1.52 \cdot 10^{-3}$	2.11	$4.84 \cdot 10^{-3}$	1.93	$5.74 \cdot 10^{-3}$	1.82
4×	$2.91 \cdot 10^{-4}$	2.38	$9.69 \cdot 10^{-4}$	2.32	$1.41 \cdot 10^{-3}$	2.03

TABLE 2

ℓ_∞ error on the grid averaged over $0 \leq t \leq T_{\text{sim}}$ and grid convergence rates for the fourth order compact scheme (2). The rate is \log_2 (coarser grid error) – \log_2 (finer grid error).

Grid	Dirichlet		Neumann		Robin	
	Average error	Rate	Average error	Rate	Average error	Rate
2×	$6.52 \cdot 10^{-5}$	-	$2.09 \cdot 10^{-4}$	-	$2.86 \cdot 10^{-4}$	-
4×	$2.44 \cdot 10^{-6}$	4.74	$8.60 \cdot 10^{-6}$	4.6	$1.31 \cdot 10^{-5}$	4.45

the results are the same. The graphs in Figures 6(a) and 6(b) correspond to the Neumann and Robin boundary conditions, respectively, with the matrices (33) computed using lacunae-based termination. The errors in Figure 6 are one to two binary orders of magnitude larger than those in Figure 5. We attribute this to derivatives in the boundary condition rather than the treatment of the outer boundary.

As the domain Ω^{aug} is somewhat larger than Ω^{PML} , while the governing equations in the PML are more complex and require more operations per grid node, the computation of the matrices (33) takes roughly the same time. The subsequent boundary time marching (35) does not depend on how the matrices (33) have been computed. Consequently, neither the PML nor exact lacunae-based termination currently offers a faster performance. Yet, in addition to its unimprovable accuracy, the lacunae-based time-marching has proven very efficient in suppressing the long-time instabilities that may characterize some PMLs [1, 2, 3, 26, 23].

6.3. Sublinear complexity. When computing solutions with a predetermined range of frequencies so that the dimension N of the basis $\{\xi_i\}$ is fixed, the reduced dimension of the boundary problem (32) and (9) yields a substantial improvement of numerical performance compared to the conventional volumetric time marching.

The reference volumetric problem is set up on the domain Ω^{PML} . The scatterer is shaped as a cube rather than a sphere (see Figure 4 (shown in dashed red)), because for this case we want to use only the Cartesian scheme (3a) over the volume with no MDP. Instead of (39), we use a different test solution [23], because it satisfies the homogeneous initial conditions everywhere including the PML. We specify the Dirichlet data on the surface of the scatterer from test solution [23] and solve the resulting IBVP with second order accuracy by volumetric time marching terminated with a PML. The observed numerical performance provides a reference point for comparison to the MDP+lacunae solver for the sphere of a comparable size; see Figure 4.

Table 3 summarizes the CPU times required for integrating the wave equation over the time T_0 with these two methods. For MDP+lacunae, it is very important to distinguish between the solution (35) only at the boundary and additional solution on the grid exterior to the scatter. In the case of only boundary integration, the cost does not depend on the dimension h^{-1} of the spatial grid at all (see section

TABLE 3

CPU times needed to advance the solution over the interval T_0 . In the case of MDP+lacunae, the solution on the 3D grid in space is computed once per T_0 .

Grid	Volumetric method+PML		MDP+lacunae	
	CPU time, seconds	Scaling, times	CPU time, seconds	Scaling, times
1x	1.26	-	$6.14 \cdot 10^{-2}$	-
2x	19.8	15.7	$5.16 \cdot 10^{-1}$	8.44
4x	322	16.3	4.44	8.61

5). It depends on the dimension N of the basis on Γ as $\mathcal{O}(N^2)$ and remains very small: $2.4 \cdot 10^{-3}$ sec, because for the specific values we have chosen, $L_{\max} = 12$ and $N_{\max} = 10$, $N = (L_{\max} + 1)^2(N_{\max} + 1)$ is orders of magnitude less than the FDTD dimension $\frac{(\text{diam } \Omega^{\text{aug}})^3}{h^3} \frac{T_0}{\tau}$, even for the coarsest grid that we are using. Therefore, the cost of boundary update is not included in Table 3. The CPU times reported in Table 3 for MDP+lacunae are the times needed to reconstruct the solution on the spatial grid exterior to the scatterer after the spectral expansion coefficients at the boundary have been determined. This reconstruction is performed once per T_0 , and the 8-fold scaling on successive grids corresponds to the grid dimension in 3D space. Taking uniform increments T_0 in physical units of time (as opposed to discretization steps τ) is adequate for computing the solution in the volume, especially since in many cases the volumetric solution may not be needed at all or may otherwise be needed only at the final time T . The 16-fold scaling of the cost for the volumetric method in Table 3 is commensurate with the grid dimension in the (3+1)D space-time. Thus, the proposed method offers grid-independent complexity at the boundary and sublinear complexity over the volume with respect to the total grid dimension. In doing so, the dimension of the basis $N = (L_{\max} + 1)^2(N_{\max} + 1)$ is fixed (see section 6.1).

6.4. Parallelization in time. The data in Table 3 do not take into account the cost of computing the operators (33) and the matrix on the right-hand side of (35). This is a one-time expense, but it may be large. To compute the operators (33), one needs to solve $2N$ APs (28) over the interval $2T_0$. Costwise, this is equivalent to solving one AP over the time $4NT_0$. As the APs (28) with right-hand sides $\mu_{M-} \square_c^{(h)} w^{(k)}$ that correspond to different $w_{0,i}^+$ and $w_{1,i}^-$, $i = 1, 2, \dots, N$, are completely independent, they can be efficiently solved in parallel. This can be thought of as parallelization in time. On 32 cores, we have achieved about 60% parallelization efficiency effortlessly. On a larger parallel system, the wall clock time of computing (33) will be small.

TABLE 4

CPU times and break-even times for the proposed boundary method with second order scheme. The recursive boundary update (see step 3 of the algorithm on page A3475) is performed for a smaller increment $T'_0 = \frac{1}{4}T_0$. The size of Ω^{aug} is reduced according to (36) and (40), which yields a three-fold cost reduction compared to having the increment T_0 .

Grid	CPU time for computing (33), (35), seconds	Break-even time, R_0
1x	414.93	329
2x	6717.63	353
4x	103483	323

In Table 4, we provide the CPU times required to precompute the operators (33) and the matrix on the right-hand side of (35) for the boundary method. The operators (33) are computed using the central difference second order scheme and the exact lacunae-based termination at the outer boundary Γ' . The matrix on the right-

hand side of (35) requires a QR factorization at a cost of approximately $\mathcal{O}(|\gamma_k^-| \cdot N^2)$ operations; see section 5. Every time the grid is refined by a factor of 2, the execution time in the second column of Table 4 increases by roughly a factor of 16, because the number of grid nodes in the (3+1)D computational domain increases by a factor of 16 (the dominant behavior), the number of nodes in the (2+1)D grid boundary $|\gamma_k^-|$ increases by a factor of 8, and the dimension of the basis N remains fixed. This setting corresponds to computing a chosen fixed solution on a sequence of grids. The results of a pollution study that involves refining the grid and increasing the frequency of the solution at the same time are presented in section 6.6. In Table 4, we also present the corresponding break-even times in the units R_0 (because $c = 1$). A break-even time is the time of integrating one problem by a volumetric method that would be equivalent to the time for integrating $2N$ APs (28). From Table 4 we see that the boundary method breaks even at computational times $\gtrsim 300R_0$. As we advance further in time, the boundary method becomes progressively more efficient, because the cost of time marching along the boundary is $\mathcal{O}(N^2)$ and does not depend on the dimension of the finite difference grid, while the subsequent reconstruction of the solution in the volume at select moments of time scales cubically with respect to h^{-1} . On the other hand, the volumetric method scales quartically (see Table 3 and all of section 6.3).

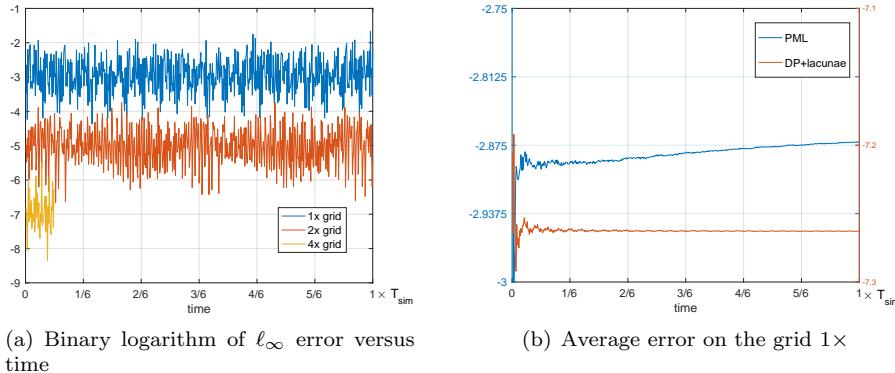


FIG. 8. Long-time performance of the DPM+lacunae method versus that of the reference scheme.

6.5. Long-time and nonreflecting performance. In Figure 8(a), we plot the error profiles for the second order volumetric method terminated with a PML on three consecutive grids.² The graphs in Figure 8(a) corroborate the second order grid convergence and do not point to any problems with long-time performance. However, the error averaged over the elapsed time interval gradually increases; see Figure 8(b). On the other hand, the average error profile for the boundary method is flat. The long-time error growth for the volumetric scheme itself (see, e.g., [22, 13, 37]) and/or the PML [11] that we have chosen. Yet the boundary method, for which the operators (33) were computed using the same PML to terminate the APs (28), shows no error growth at all; see Figure 8(b). This example illustrates the specific advantage that we have presented previously—stabilization of the long-time numerical performance with the help of lacunae; see [26, 23]. Regardless of what causes the error accumulation over long runs, the Huygens' principle allows one to partition the overall temporal evolution of the solution into a

²The 4 \times solution is not computed on the full interval T_{sim} , because that would take too long.

sequence of small increments (of length T_0 or shorter). Each of those is computed only over its own short duration so that the error does not accumulate noticeably. Due to the presence of lacunae, the overall solution at any given moment of time is affected only by a fixed and nonincreasing number of the preceding partition elements; hence the error does not accumulate at all.

6.6. Scaling with frequency and numerical pollution. For the Helmholtz equation, it is well known [6] that as the wavenumber k increases, one needs to keep the quantity $k^{p+1}h^p$ constant in order to guarantee that the error of a p th order accurate finite difference method will not increase. This means that for the grid dimension in each coordinate direction in space, we must have $h^{-1} \propto k^{\frac{p+1}{p}}$.

Similar analysis for time-dependent problems (hyperbolic) has been conducted in [17]. Time-dependent formulations usually involve a set or band of frequencies; see, e.g., our test solution (38) and (39). As $k = \omega/c$ and $\tau \propto h$, we have $k^{p+1}h^p \propto \omega^{p+1}\tau^p$. Hence, to make sure that the error doesn't grow, one must maintain the same scaling in time: $\tau^{-1} \propto \omega^{\frac{p+1}{p}}$, where ω is a typical or the maximum frequency in the band.

The boundary method that we are proposing requires a basis of dimension $N = (1 + L_{\max})^2(1 + N_{\max})$ on Γ (section 6.1). In Table 5, we show how N should increase in order to maintain a predetermined accuracy of the spectral expansion (5) at the boundary. Table 5 indicates that as the frequency doubles, the required number of APs (last column in Table 5) increases by a factor of 2^α ,³ where α is a number somewhat larger than 2. For example, for $\omega = 24$ and $\omega = 48$ we have $\alpha = 2.35$. Since the basis $\{\xi_i\}$ covers a (2+1)D surface in space-time, it is a very slow increase compared to what a finite difference scheme requires to maintain the accuracy: $(2^{\frac{3}{2}})^3 \approx 22.5$ as many nodes on a fixed 3D domain in space, and 2^6 as many on a given (3+1)D domain in space-time for second order and $(2^{\frac{5}{4}})^3 \approx 13.5$ and 2^5 , respectively, for fourth order.

TABLE 5
Spectral expansion of a plane wave on Γ .

Frequency	Error for u	Error for $\frac{\partial u}{\partial n}$	L_{\max}	N_{\max}	# of APs
$\omega = 3$	$2.5 \cdot 10^{-3}$	$5.2 \cdot 10^{-2}$	7	6	252
$\omega = 6$	$2.4 \cdot 10^{-3}$	$4.1 \cdot 10^{-2}$	11	8	702
$\omega = 12$	$3.5 \cdot 10^{-3}$	$4.9 \cdot 10^{-2}$	18	12	2470
$\omega = 16$	$3.1 \cdot 10^{-3}$	$5.7 \cdot 10^{-2}$	22	16	4692
$\omega = 20$	$3.7 \cdot 10^{-3}$	$7.1 \cdot 10^{-2}$	27	18	7714
$\omega = 24$	$5.6 \cdot 10^{-3}$	$4.8 \cdot 10^{-2}$	31	21	11616
$\omega = 36$	$3.4 \cdot 10^{-3}$	$6.8 \cdot 10^{-2}$	42	27	26488
$\omega = 48$	$1.1 \cdot 10^{-3}$	$4.4 \cdot 10^{-2}$	56	35	59508

The actual value of α cannot be determined accurately from Table 5. But even for a value larger than what we have observed, for example, $\alpha=3$, the growth in N that would guarantee the desired accuracy is still much slower than that of the dimension of an FDTD grid. The computational cost for an explicit finite difference scheme is proportional to the grid dimension. In the case of second order accuracy ($p=2$), we have $k^3h^2 = \text{const}$, which means that the grid dimension is $\frac{(\text{diam } \Omega^{\text{aug}})^3}{h^3} \frac{T_0}{\tau}$ and the cost scale as ω^6 . Even if we assume that $\alpha=3$, we have $N \propto \omega^3$ as the frequency increases. Accordingly, the cost of a boundary update, which is $\mathcal{O}(N^2)$, will also scale as ω^6 .

³Since the d'Alembert operator and Chebyshev polynomials are real, the number of APs solved to compute the boundary operators (33) is only $(1 + N_{\max})(1 + L_{\max})(2 + L_{\max})/2$; see [25, sect. 5.1].

Therefore, it will remain as small as is shown in section 6.3 compared to the cost of volumetric time marching. For a fourth order accurate scheme ($p = 4$), we have $k^5 h^4 = \text{const}$, so that the grid dimension and the cost of volumetric integration scale as ω^5 . If we assume that $\alpha = 3$, then the cost of a boundary update, $\mathcal{O}(N^2) \propto \omega^6$, increases faster. However, this cost will “catch up” only when $A\omega^5 = B\omega^6$, where A and B are constants (frequency independent). Consequently, the frequency for which the two costs become equal is $\omega = A/B$, where A/B is the relative cost of the two methods on the chosen reference grid for $\omega = 1$. As we have seen in section 6.3 (Table 3), this quantity could be anywhere between 10^3 and 10^5 . For a scatterer of size 1, the frequencies in the range around 10^3 to 10^5 are already approaching geometrical optics, since, in observations $\alpha < 3$, the actual parity between the methods appears even later. Therefore, for all practical purposes the cost of integration along the boundary will still be much lower than that of the volumetric time marching.

In Table 6, we present the results of computations for a sequence of frequencies that allow us to maintain the same level of error on the grids $1\times$, $2\times$, and $4\times$ according to the law $k^3 h^2 = \text{const}$ ($p = 2$). For each ω , we also show the minimum dimension of the basis that guarantees that the error at the boundary will be smaller than the error on the grid. The last column of Table 6 is the CPU time required to advance the solution at the boundary by $T_0/4$ (see caption for Table 4). From Table 6, we see that to maintain a given accuracy, N grows slower than the dimension of the grid in three dimensions. Indeed, as the latter increases 8-fold, the dimension of the basis in (2+1)D increases 2 to 2.5 times. Table 6 also corroborates the finding of section 5: the cost of advancing the solution in time at the boundary does not depend on the dimension of the grid. It is proportional to N^2 , where N is the dimension of the basis on Γ .

TABLE 6

Illustration of the numerical pollution effect for the second order scheme. CPU time in seconds.

ω	Error on three grids			Error $u _\Gamma$	Basis	APs	CPU time
	$1\times$	$2\times$	$3\times$				
3	$6.5 \cdot 10^{-3}$	—	—	$2.3 \cdot 10^{-3}$	7×6	252	$1.2 \cdot 10^{-4}$
$3\sqrt[3]{4}$	—	$5.8 \cdot 10^{-3}$	—	$3.7 \cdot 10^{-3}$	9×7	440	$4.5 \cdot 10^{-4}$
$3\sqrt[3]{16}$	—	—	$5 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	13×10	1155	$3 \cdot 10^{-3}$

Note also that the dimension $2N \times 3N$ of the matrix on the right-hand side of (35) is small compared to all typical dimensions associated with the grid. Hence, after the operators (33) and the matrix on the right-hand side of (35) have been computed, the runtime memory requirements of the proposed algorithm remain very low.

6.7. Prolate spheroids. We compute the solution to problem (1) outside a prolate spheroid that has its major axis aligned with the Cartesian coordinate z : $(x^2 + y^2)/b^2 + z^2/a^2 = 1$. For brevity, we only present the results of computations by means of the second order scheme. In our simulations, we keep the major semiaxis of the spheroid fixed, $a = 1$, and allow its aspect ratio a/b to vary. An outgoing test solution in the region exterior to the spheroid is obtained by placing a δ -source inside the spheroid (off-center at the point $\mathbf{x}' = (0.1, 0, 0)$). This source generates a spherical wave $G(\mathbf{x}, \mathbf{x}') = 1/4\pi \exp(ik|\mathbf{x} - \mathbf{x}'| - i\omega t)/|\mathbf{x} - \mathbf{x}'|$, which is interpreted as the scattered field outside the spheroid. Its trace $G|_\Gamma$ provides the Dirichlet data for an exterior problem, while the trace $\frac{\partial G}{\partial n}|_\Gamma$ provides the Neumann data. The frequency $\omega = 3$, so that the wavelength $2\pi c/\omega$ is of the same order of magnitude as the diameter $2a = 2$. The basis $\{\xi_i\}$ for representing the boundary data (see (5)) consists

TABLE 7

ℓ_∞ error on the grid averaged over $0 \leq t \leq T_{\text{sim}}$, and grid convergence rates for prolate spheroids with three different aspect ratios a/b ; computations by second order central difference scheme. The rate is $\log_2(\text{coarser grid error}) - \log_2(\text{finer grid error})$.

Grid	$a/b = 4/3$							
	L_{\max}	APs	Error $u _\Gamma$	Dirichlet		Neumann		
				Error	Rate	Error	Rate	
1×	6	378	$5.6 \cdot 10^{-3}$	$6.1 \cdot 10^{-3}$	—	$2.0 \cdot 10^{-2}$	—	
2×	9	567	$1.2 \cdot 10^{-4}$	$1.6 \cdot 10^{-3}$	1.9	$4.9 \cdot 10^{-3}$	2.0	
4×	11	693	$2.1 \cdot 10^{-5}$	$4.0 \cdot 10^{-4}$	2.0	$1.2 \cdot 10^{-3}$	2.0	
$a/b = 2$								
1×	12	756	$3.9 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	—	$3.9 \cdot 10^{-2}$	—	
2×	15	945	$5.8 \cdot 10^{-4}$	$2.1 \cdot 10^{-3}$	1.9	$1.0 \cdot 10^{-2}$	2.0	
4×	18	1134	$2.3 \cdot 10^{-4}$	$5.8 \cdot 10^{-4}$	1.8	$2.8 \cdot 10^{-3}$	1.8	
$a/b = 4$								
2×	24	1512	$5.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-3}$	—	$2.5 \cdot 10^{-2}$	—	
4×	34	2754	$7.1 \cdot 10^{-4}$	$1.9 \cdot 10^{-3}$	1.8	$6.8 \cdot 10^{-3}$	1.8	

of the angular spheroidal functions $S_{ml}(d, \eta)$ multiplied with $\cos(m\varphi)$, where $m = 0, 1, \dots, M_{\max}$, $l = m, m+1, \dots, m+L_{\max}-1$, $d = k\sqrt{a^2 - b^2}$, and η, φ are the angular coordinates in the prolate spheroidal system; see, e.g., [19]. Similarly to spherical harmonics on the sphere, the system of functions $\{S_{ml}(d, \eta) \cdot \cos(m\varphi)\}$ is complete on the surface of the spheroid (if not truncated). To account for the temporal dependence of the solution at the boundary, we still employ the Chebyshev polynomials $T_n(t)$, $n = 0, \dots, N_{\max}$. In our computations, we take $N_{\max} = 8$ and $M_{\max} = 6$ (except for the last row of Table 7 where $M_{\max} = 8$), while L_{\max} varies in order to maintain the accuracy of the boundary expansion (5) on the spheroid (column “Error $u|_\Gamma$ ” in Table 7). The overall dimension of the basis is $N = (1 + N_{\max})(1 + M_{\max})L_{\max}$; it increases as the spheroid gets more prolate (see column “APs” in Table 7).

As Table 7 suggests, the mean error converges with the design rate of the second order accurate scheme. The time of simulation is chosen comparably long to that of section 6.2, $T_{\text{sim}} = 6600a/c$. The error profiles remain flat throughout the interval $0 \leq t \leq T_{\text{sim}}$ and look very similar to those in Figures 5 and 6; that’s why we do not present the additional plots here. For the most prolate spheroid ($a/b = 4$), the grid $1\times$ appears to be too coarse (the spatial step $h \approx 0.13$, whereas the maximum transversal size $2b = 0.5$), which ruins the computation. Switching to finer grids $2\times$ and $4\times$ fixes the problem without any changes in the code.

7. Discussion. The proposed numerical method solves exterior initial boundary value problems for the 3D wave equation (e.g., scattering about a given shape). It has been built and tested successfully using both a second order accurate and fourth order accurate finite difference discretization. The method employs boundary operators that are similar to Calderon’s projections. The same set of operators allows us to handle any boundary condition on the surface of the scatterer, which means that multiple similar problems can be solved at a low individual cost per problem. The computations are conducted on a regular Cartesian grid, but the shape of the scatterer can be arbitrary (nonconforming). This does not result in any reduction of accuracy and does not cause any problems with stability.

The proposed method requires a one-time computation of the boundary operators that may entail a substantial cost. However, this initial computation parallelizes efficiently, and we expect that the associated execution time on a larger multiprocessor

platform will be small. Once the operators have been computed, the cost of advancing a given solution in time along the boundary becomes completely grid-independent. The cost of computing the solution on the grid exterior to the scatterer scales cubically with respect to the linear dimension, i.e., it increases slower than the overall number of grid nodes in space-time. At the same time, the cost of the conventional volumetric method scales quartically, i.e., directly proportional to the number of nodes. Thus, for a fixed range of frequencies, our method will outperform any volumetric scheme as long as the integration time is sufficiently long.

The proposed method involves a spectral expansion of the solution and its normal derivative with respect to a basis chosen at the continuous boundary; see formulae (5). As the frequency increases, the number of terms in this expansion needed to maintain the accuracy grows. However, this growth is much slower than that of the dimension of the finite difference grid. The cost of the boundary update increases with frequency as well. Yet it remains a small fraction of the cost of volumetric integration.

As the proposed methodology relies on the Huygens' principle, it is inherently three-dimensional. We presented it for the case of a constant propagation speed c in (1a). It can be extended to the case of a variable propagation speed provided that $0 < c_{\min} \leq c$. The Huygens' principle also enables the exact reflectionless treatment of artificial outer boundaries and provides a stabilizing effect for long-time integration regardless of what may have caused a deterioration of accuracy in the first place.

In the future, we plan to incorporate more sophisticated geometries/shapes, consider not only boundaries per se but also interfaces between domains and transmission/scattering problems, and generalize from the scalar wave equation to systems. Going beyond simple shapes such as a sphere or spheroid is likely to require patching and piecewise parametrization of the boundary, like, e.g., in [8, 7], which, in turn, will lead to choosing the systems of basis functions in (5) independently for different patches. This approach will also help accommodate nonsmooth surfaces, although addressing the corresponding singularities in the solution may require special care.

REFERENCES

- [1] S. ABARBANEL AND D. GOTTLIEB, *A mathematical analysis of the PML method*, J. Comput. Phys., 134 (1997), pp. 357–363.
- [2] S. ABARBANEL, D. GOTTLIEB, AND J. S. HESTHAVEN, *Long time behavior of the perfectly matched layer equations in computational electromagnetics*, J. Sci. Comput., 17 (2002), pp. 405–422.
- [3] S. ABARBANEL, H. QASIMOV, AND S. TSYNKOV, *Long-time performance of unsplit PMLs with explicit second order schemes*, J. Sci. Comput., 41 (2009), pp. 1–12, <https://doi.org/10.1007/s10915-009-9282-4>.
- [4] T. G. ANDERSON, O. P. BRUNO, AND M. LYON, *High-order, Dispersionless, “Fast-hybrid” Wave Equation Solver. Part I: $\mathcal{O}(1)$ Sampling Cost via Incident Field Windowing and Recentering*, preprint, <https://arxiv.org/abs/1807.02718>, 2020.
- [5] A. BARNETT, L. GREENGARD, AND T. HAGSTROM, *High-order discretization of a stable time-domain integral equation for 3D acoustic scattering*, J. Comput. Phys., 402 (2020), 109047, <https://doi.org/10.1016/j.jcp.2019.109047>.
- [6] A. BAYLISS, C. I. GOLDSTEIN, AND E. TURKEL, *On accuracy conditions for the numerical computation of waves*, J. Comput. Phys., 59 (1985), pp. 396–404.
- [7] O. P. BRUNO AND L. A. KUNYANSKY, *A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications*, J. Comput. Phys., 169 (2001), pp. 80–110, <https://doi.org/10.1006/jcph.2001.6714>.
- [8] O. P. BRUNO AND L. A. KUNYANSKY, *Surface scattering in three dimensions: An accelerated high-order solver*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 457 (2001), pp. 2921–2934, <https://doi.org/10.1098/rspa.2001.0882>.
- [9] M. COSTABEL AND F.-J. SAYAS, *Time-dependent Problems with the Boundary Integral Equation*

- Method*, Part 2, Fundamentals, Encyclopedia of Computational Mechanics, 2nd ed., John Wiley & Sons, 2017, New York, <https://doi.org/10.1002/9781119176817.ecm2022>.
- [10] C. L. EPSTEIN, L. GREENGARD, AND T. HAGSTROM, *On the stability of time-domain integral equations for acoustic wave propagation*, Discrete Contin. Dyn. Syst., 36 (2016), pp. 4367–4382, <https://doi.org/10.3934/dcds.2016.36.4367>.
 - [11] M. J. GROTE AND I. SIM, *Efficient PML for the Wave Equation*, preprint, <https://arxiv.org/abs/1001.0319>, 2010.
 - [12] P. GÜNTHER, *Huygens' Principle and Hyperbolic Equations*, with Appendices by V. Wünsch, Perspect. Math. 5, Academic Press, Boston, 1988.
 - [13] B. GUSTAFSSON AND P. WAHLUND, *Time compact high order difference methods for wave propagation, 2D*, J. Sci. Comput., 25 (2005), pp. 195–211, <https://doi.org/10.1007/s10915-004-4639-1>.
 - [14] T. HA-DUONG, *On retarded potential boundary integral equations and their discretisation*, in Topics in Computational Wave Propagation, Lect. Notes Comput. Sci. Eng. 31, Springer, Berlin, 2003, pp. 301–336, https://doi.org/10.1007/978-3-642-55483-4_8.
 - [15] I. HARARI AND U. ALBOCHER, *Complementary solutions of Nitsche's method*, J. Sci. Comput., 81 (2019), pp. 1472–1492, <https://doi.org/10.1007/s10915-019-01066-2>.
 - [16] *Developer Reference for Intel Math Kernel Library – Fortran*, <https://software.intel.com/content/www/us/en/develop/documentation/mkl-developer-reference-fortran/top.html>.
 - [17] H.-O. KREISS AND J. OLIGER, *Comparison of accurate methods for the integration of hyperbolic equations*, Tellus, 24 (1972), pp. 199–215, <https://doi.org/10.3402/tellusa.v24i3.10634>.
 - [18] P. D. LAX, C. S. MORAWETZ, AND R. S. PHILLIPS, *Exponential decay of solutions of the wave equation in the exterior of a star-shaped obstacle*, Comm. Pure Appl. Math., 16 (1963), pp. 477–486, <https://doi.org/10.1002/cpa.3160160407>.
 - [19] L.-W. LI, X.-K. KANG, AND M.-S. LEONG, *Spheroidal Wave Functions in Electromagnetic Theory*, John Wiley & Sons, New York, 2002.
 - [20] Z. LI AND K. ITO, *The Immersed Interface Method*, Frontiers Appl. Math. 33, SIAM, Philadelphia, 2006, <https://doi.org/10.1137/1.9780898717464>.
 - [21] C. LUBICH, *On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations*, Numer. Math., 67 (1994), pp. 365–389, <https://doi.org/10.1007/s002110050033>.
 - [22] J. NORDSTRÖM AND H. FRENANDER, *On long time error bounds for the wave equation on second order form*, J. Sci. Comput., 76 (2018), pp. 1327–1336, <https://doi.org/10.1007/s10915-018-0667-0>.
 - [23] S. PETROPAVLOVSKY AND S. TSYNKOV, *Non-deteriorating time domain numerical algorithms for Maxwell's electrodynamics*, J. Comput. Phys., 336 (2017), pp. 1–35, <https://doi.org/10.1016/j.jcp.2017.01.068>.
 - [24] S. PETROPAVLOVSKY AND S. TSYNKOV, *Method of difference potentials for evolution equations with lacunas*, Comp. Math. Math. Phys., 60 (2020), pp. 711–722.
 - [25] S. PETROPAVLOVSKY, S. TSYNKOV, AND E. TURKEL, *A method of boundary equations for unsteady hyperbolic problems in 3D*, J. Comput. Phys., 365 (2018), pp. 294–323, <https://doi.org/10.1016/j.jcp.2018.03.039>.
 - [26] S. V. PETROPAVLOVSKY AND S. V. TSYNKOV, *A non-deteriorating algorithm for computational electromagnetism based on quasi-lacunae of Maxwell's equations*, J. Comput. Phys., 231 (2012), pp. 558–585, <https://doi.org/10.1016/j.jcp.2011.09.019>.
 - [27] S. V. PETROPAVLOVSKY, S. V. TSYNKOV, AND E. TURKEL, *A method of boundary equations for unsteady hyperbolic problems in 3D*, in Continuum Mechanics, Applied Mathematics and Scientific Computing: Godunov's Legacy — A Liber Amicorum to Professor Godunov, G. V. Demidenko, E. Romenski, E. Toro, and M. Dumbser, eds., Springer, Cham, 2020, pp. 291–297.
 - [28] I. PETROWSKY, *On the diffusion of waves and the lacunas for hyperbolic equations*, Rec. Math. [Mat. Sbornik] N. S., 17 (1945), pp. 289–370.
 - [29] M. C. RECCHIONI AND F. ZIRILLI, *A new formalism for time-dependent electromagnetic scattering from a bounded obstacle*, J. Engrg. Math., 47 (2003), pp. 17–43, <https://doi.org/10.1023/A:1025570924371>.
 - [30] A. A. REZNIK, *Approximation of surface potentials of elliptic operators by difference potentials*, Dokl. Akad. Nauk SSSR, 263 (1982), pp. 1318–1321 (in Russian).
 - [31] V. S. RYABEN'KII, *Exact transfer of difference boundary conditions*, Funct. Anal. Appl., 24 (1990), pp. 251–253.
 - [32] V. S. RYABEN'KII, *Method of Difference Potentials and Its Applications*, Springer Ser. Comput. Math. 30, Springer-Verlag, Berlin, 2002.
 - [33] F.-J. SAYAS, *Retarded Potentials and Time Domain Boundary Integral Equations. A Road*

- Map*, Springer Ser. Comput. Math. 50, Springer, Cham, 2016, <https://doi.org/10.1007/978-3-319-26645-9>.
- [34] F. SMITH, S. TSYNKOV, AND E. TURKEL, *Compact high order accurate schemes for the three dimensional wave equation*, J. Sci. Comput., 81 (2019), pp. 1181–1209, <https://doi.org/10.1007/s10915-019-00970-x>.
 - [35] S. STICKO AND G. KREISS, *A stabilized Nitsche cut element method for the wave equation*, Comput. Methods Appl. Mech. Engrg., 309 (2016), pp. 364–387, <https://doi.org/https://doi.org/10.1016/j.cma.2016.06.001>.
 - [36] S. V. TSYNKOV, *Numerical solution of problems on unbounded domains. A review*, Appl. Numer. Math., 27 (1998), pp. 465–532, [https://doi.org/10.1016/S0168-9274\(98\)00025-7](https://doi.org/10.1016/S0168-9274(98)00025-7).
 - [37] D. W. ZINGG, *Comparison of high-accuracy finite-difference methods for linear wave propagation*, SIAM J. Sci. Comput., 22 (2000), pp. 476–502, <https://doi.org/10.1137/S1064827599350320>.