Journal of Computational Physics 460 (2022) 111161

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A high order compact time/space finite difference scheme for the 2D and 3D wave equation with a damping layer [☆]

Adar Kahana [a,*], Fouche Smith [b], Eli Turkel [a], Semyon Tsynkov [b]

[a] *Department of Applied Mathematics, Tel-Aviv University, Tel-Aviv 69978, Israel*
[b] *Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA*

ARTICLE INFO

Article history:
Received 20 October 2021
Received in revised form 18 March 2022
Accepted 20 March 2022
Available online 24 March 2022

Keywords:
Compact finite differences
Absorbing boundary layer
Unbounded domain
High-order accuracy
Wave equation

ABSTRACT

We consider fourth order accurate compact schemes, in both space and time, for the second order wave equation with a variable speed of sound. For unbounded domains we add a fourth order accurate sponge layer to damp the outgoing waves. We demonstrate that usually this is more efficient than lower order schemes despite being implicit and conditionally stable. Fast time marching of the implicit scheme is accomplished by iterative methods such as multi-grid. Computations confirm the design convergence rate for the inhomogeneous, variable wave speed equation.

1. Introduction

The acoustic wave equation describes the propagation of waves in the atmosphere or underwater and can also be used as a simplification for modeling electromagnetic waves or elastic waves in a solid. The methods developed here for the 2D and 3D acoustic wave equation may be extended to more complicated hyperbolic systems. In particular, the fourth order scheme in time is applicable to any second order equation in time (i.e., in which second time derivatives appear). If first order time derivatives also appear, then an extension of the method described here is straightforward. In the proposed scheme, the time and space portions are discretized separately. Hence, a fourth order approximation in space depends only on a similar fourth order discretization for the time harmonic version of the equation. The resulting spatial equation in this case is negative definite elliptic rather than non-definite as with the Helmholtz equation.

Even though the formulation is implicit, the overall scheme remains efficient due to the high order of accuracy and the use of rapidly converging iterative methods for time marching. Since the spatial equation at each time step is symmetric negative definite, conjugate gradient (CG) and multigrid (MG) can be used. Performance is improved by using a 2nd order accurate initial guess at each time step from the standard explicit scheme, and this further reduces the number of iterations per time step.

In [5,22] we developed a compact fourth order accurate scheme in space and time for the acoustic wave equation in an interior region. In this paper, we extend that scheme to an unbounded region. We generalize a damping layer suggested by Sochacki et al. [23] that prevents reflections back into the domain, see also [1]. A similar damping layer was proposed by Cerjan et al. [6]. In this layer, an additional term containing a first time derivative is added to the wave equation. Gao et al.

[☆] Work supported by the United States–Israel Binational Science Foundation (BSF) under grant 2020128.

[*] Corresponding author.
E-mail address: adarkahana@gmail.com (A. Kahana).

https://doi.org/10.1016/j.jcp.2022.111161
0021-9991/© 2022 Elsevier Inc. All rights reserved.

[11] found that the sponge layer is not as effective as a PML for second order accuracy. Lee and Joo-Min [15] suggested an improvement that contains a frequency term and so is most appropriate for wave equations in the frequency domain. The sponge layer is a simpler modification to the wave equation than that required by a PML layer and requires fewer variables. Thus, one can use a larger layer than the PML but still not have more degrees of freedom. In addition, we shall use an improved stretching function than found in [23,11]. In terms of CPU times, the additional term pu_t adds only a negligible overhead compared with u_{tt} and the calculation of the Laplacian. This is especially true when the more complicated formula for a fourth order accurate approximation to the Laplacian is used. On the other hand, the PML complicates the second order wave equation by the addition of several terms and requires the solution of several ODEs. So also from the viewpoint of computational time the sponge layer can be much thicker without requiring more time. As a further factor, we have seen that the addition of the sponge layer does not reduce the allowable time step for stability while a PML frequently requires a smaller time step. This smaller step is then required in the entire domain and not just the PML adding considerably to the cost of the PML approach.

More importantly, we are able to develop a compact formula that is still fourth order accurate in both space and time, in both the interior and the absorbing layer. We were not able to develop a stable fourth order accurate compact discretization of a PML for the second order acoustic wave equation. Computations we have performed demonstrate that using a fourth order accurate scheme in the interior and a second order accurate scheme for the PML yields a global second order accuracy. Hence, we do not present any comparisons between the sponge layer and a PML. We note that, a non-compact 25 point stencil for the 2D Helmholtz equation was developed by Chen et al. [7] and improved to a 13 point stencil by Dastour and Liao [8,9]. Fourth order accuracy is important to reduce the pollution error inherent in a wave equation [2]. Non-compact schemes cause difficulties near boundaries and for the Helmholtz equation increase the bandwidth of the matrix that is inverted. We note that, Duru and Kreiss [10] achieved fourth order accuracy in the interior for a two dimensional waveguide. They used a summation by parts SBP-SAT difference scheme to obtain estimates for the energy and accuracy. As such, the scheme is not compact and so the order of accuracy is reduced near boundaries. A final advantage of the sponge layer approach is that it is straightforward to apply to a large range of equations. Constructing a PML for the elastic equations or other physical systems is difficult especially if one needs high order accuracy.

2. High order discretization of the wave equation in time

2.1. Derivation of the time-marching scheme

We consider the variable coefficient wave equation:

$$u_{tt} = c^2 \Delta u + F, \tag{1}$$

where $c = c(\mathbf{x})$ and $F = F(\mathbf{x}, t)$ are given, $\mathbf{x} \in \mathbb{R}^2$ or $\mathbf{x} \in \mathbb{R}^3$. For our numerical demonstrations (see Sections 4 and 5), we choose a constant c. We wish to solve the initial value problem where equation (1) is supplemented with the initial conditions in $\mathbb{R}^2$ or $\mathbb{R}^3$:

$$\begin{aligned} u(x, y, t = 0) &= u^0(x, y), \\ u_t(x, y, t = 0) &= \psi(x, y). \end{aligned} \tag{2}$$

To reduce the infinite domain to finite size, we consider a bounded computational domain, which is a Cartesian square in $\mathbb{R}^2$ or cube in $\mathbb{R}^3$, and introduce a sponge layer at all its edges or faces that correspond to unbounded regions. We then replace (1) with the equation:

$$u_{tt} + p(x, y)u_t = c^2 \Delta u + F, \tag{3}$$

where $p = p(\mathbf{x})$ is non-zero in the layer, $p = 0$ at the interface with the interior domain, and p is given by an arctangent function to some maximum value. At the end of this layer either a homogeneous Dirichlet boundary condition is imposed or preferably a local absorbing boundary condition.

We consider a semi-discrete approach by first discretizing the equation in time and later in space. Denote by h_t the uniform time step so that $t_n = nh_t$. The first and second central difference operators are given respectively by

$$\begin{aligned} \delta_t u^n &= \frac{u^{n+1} - u^{n-1}}{2}, \\ \delta_{tt} u^n &= u^{n+1} - 2u^n + u^{n-1}. \end{aligned}$$

The first and second time derivatives are then approximated by

$$\begin{aligned} u_t^n &= \frac{1}{h_t}\delta_t u^n - \frac{h_t^2}{6} u_{ttt}^n + \mathcal{O}\left(h_t^4\right), \\ u_{tt}^n &= \frac{1}{h_t^2}\delta_{tt} u^n - \frac{h_t^2}{12} u_{tttt}^n + \mathcal{O}\left(h_t^4\right). \end{aligned}$$

Substituting into equation (3) we get

$$\frac{1}{h_t^2}\delta_{tt}u - \frac{h_t^2}{12}u_{tttt} = c^2\Delta u - p\left(\frac{1}{h_t}\delta_t u - \frac{h_t^2}{6}u_{ttt}\right) + F. \tag{4}$$

To find the higher derivatives we differentiate (3) to get

$$\begin{aligned} u_{ttt} &= c^2\Delta u_t - pu_{tt} + F_t, \\ u_{tttt} &= c^2\Delta u_{tt} - pu_{ttt} + F_{tt}. \end{aligned}$$

Substituting u_{tttt} into (4) we get

$$\frac{1}{h_t^2}\delta_{tt}u - \frac{h_t^2}{12}(c^2\Delta u_{tt} - pu_{ttt} + F_{tt}) = c^2\Delta u - \frac{p}{h_t}\delta_t u + \frac{ph_t^2}{6}u_{ttt} + F.$$

Combining u_{ttt} on both sides we get

$$\frac{1}{h_t^2}\delta_{tt}u - \frac{h_t^2}{12}c^2\Delta u_{tt} - \frac{h_t^2}{12}pu_{ttt} - \frac{h_t^2}{12}F_{tt} = c^2\Delta u - \frac{p}{h_t}\delta_t u + F.$$

Using the expression for u_{ttt} and merging the terms $\delta_{tt}u$ on both sides, we arrive at the scheme

$$\left(\frac{1}{h_t^2} + \frac{p^2}{12}\right)\delta_{tt}u - c^2\Delta u + \frac{p}{h_t}\delta_t u = \frac{h_t^2}{12}c^2\frac{\delta_{tt}\Delta u}{h_t^2} + \frac{ph_t^2}{12}\left(c^2\frac{\delta_t\Delta u}{h_t} + F_t\right) + F + \frac{h_t^2}{12}F_{tt}. \tag{5}$$

Next, we explicitly express the finite differences δ_t and δ_{tt}. Then (5) becomes

$$\begin{aligned} &\left(\frac{1}{h_t^2} + \frac{p^2}{12}\right)(u^{n+1} - 2u^n + u^{n-1}) - c^2\Delta u^n + p\frac{u^{n+1} - u^{n-1}}{2h_t} \\ &= \frac{1}{12}\left(c^2(\Delta u^{n+1} - 2\Delta u^n + \Delta u^{n-1}) + ph_tc^2\frac{\Delta u^{n+1} - \Delta u^{n-1}}{2} + h_t^2pF_t^n + h_t^2F_{tt}^n\right) + F^n. \end{aligned} \tag{6}$$

Bringing all the terms at step $n+1$ to the left hand side and those at steps n and $n-1$ to the right hand side we then have

$$\begin{aligned} &-\frac{1}{12}c^2\left(1 + \frac{ph_t}{2}\right)\Delta u^{n+1} + \frac{1}{12}\left(p^2 + \frac{6p}{h_t} + \frac{12}{h_t^2}\right)u^{n+1} = F_1^{n+1} = \\ &\frac{1}{12}\left(c^2(-2\Delta u^n + \Delta u^{n-1}) - ph_tc^2\frac{\Delta u^{n-1}}{2} + p^2(2u^n - u^{n-1}) + ph_t^2F_t^n + h_t^2F_{tt}^n\right) \\ &+ \frac{2u^n - u^{n-1}}{h_t^2} + c^2\Delta u^n + p\frac{u^{n-1}}{2h_t} + F^n. \end{aligned} \tag{7}$$

By construction, we find that the error is $\mathcal{O}\left(h_t^4\right)$, i.e. the scheme is fourth order accurate.

To put this in a more standard form we divide the equation by $-\frac{1}{12}c^2(1 + \frac{ph_t}{2})$. Then, (7) can be expressed as a modified Helmholtz equation

$$\Delta u^{n+1} - k^2u^{n+1} = G^{n+1}, \tag{8}$$

$$k^2 = \frac{1}{c^2h_t^2}\frac{1 + \frac{ph_t}{2} + \frac{p^2h_t^2}{12}}{\frac{1}{12}(1 + \frac{ph_t}{2})}, \tag{9}$$

$$G^{n+1} = -\frac{F_1^{n+1}}{\frac{c^2}{12}(1 + \frac{ph_t}{2})}.$$

We can also express k^2 as

$$k^2 = \frac{1}{\phi c^2h_t^2}, \quad \text{where} \tag{10}$$

$$\phi = \frac{\frac{1}{12}(1 + \frac{ph_t}{2})}{1 + \frac{ph_t}{2} + \frac{p^2h_t^2}{12}} = \frac{1}{12}\left(1 - \frac{\frac{p^2h_t^2}{12}}{1 + \frac{ph_t}{2} + \frac{p^2h_t^2}{12}}\right) = \frac{1}{12} + O(h_t^2).$$

F_1^{n+1} from (7) simplifies to

$$\begin{aligned} F_1^{n+1} &= \frac{5}{6}c^2\Delta u^n + \frac{c^2}{12}(1-\frac{ph_t}{2})\Delta u^{n-1} \\ &+ \left(\frac{p^2}{12}+\frac{1}{h_t^2}\right)(2u^n - u^{n-1}) + \frac{pu^{n-1}}{2h_t} + F + \frac{h_t^2}{12}(pF_t + F_{tt}). \end{aligned} \tag{11}$$

If F and its time derivatives are not given analytically then F_t and F_{tt}, which are multiplied by h_t^2, can be approximated by central finite differences with second order accuracy without extending the needed levels in time. The initial data u^0 and $u_t^0 = \psi$ are given (see (2)). One then obtains an approximation to u^1 by combining the Taylor expansion with an equation-based procedure as follows:

$$\begin{aligned} u^1 &= u^0 + h_t u_t^0 + \frac{h_t^2}{2}u_{tt}^0 + \frac{h_t^3}{6}u_{ttt}^0 + \frac{h_t^4}{24}u_{tttt}^0 + \mathcal{O}\left(h_t^5\right) \\ &= u^0 + h_t u_t^0 + \frac{h_t^2}{2}\left(c^2\Delta u^0 + F^0\right) + \frac{h_t^3}{6}(c^2\Delta u_t^0 + F_t^0) \\ &+ \frac{h_t^4}{24}\left(c^2\Delta\left(c^2\Delta u^0 + F^0\right) + F_{tt}^0\right) + \mathcal{O}\left(h_t^5\right). \end{aligned} \tag{12}$$

If u^0 and u_t^0 are given by explicit formulae, then the expressions of (12) can be computed exactly to give the desired approximation. Otherwise, (12) can be approximated by difference formulae. The initial conditions for a scheme of order ρ should be accurate, in time, to order $\rho + 1$ [14].

3. High order spatial discretization

We thus have a modified Helmholtz equation (8) with k^2 given by (9). Since $k^2 > 0$, equation (8) differs substantially from the conventional Helmholtz equation. The quantity k is not a physical wavenumber. It is rather a parameter of the discrete approximation that depends primarily on the time step h_t.

We approximate the equation by the compact finite difference scheme given in [21] on a Cartesian grid, which is equally spaced in all coordinate directions. The scheme is developed in [4] is for an equation with a variable coefficient Laplacian. We note that for the equation with constant coefficients in the Laplacian term the schemes of [21] and [4] coincide. In the rest of this section, we discuss the 2D, while the corresponding analysis in 3D is similar.

Let u_s and u_c denote, respectively, the sums of the four side and corner points:

$$\begin{aligned} u_s &= u_{m+1,n} + u_{m-1,n} + u_{m,n+1} + u_{m,n-1}, \\ u_c &= u_{m+1,n+1} + u_{m+1,n-1} + u_{m-1,n+1} + u_{m-1,n-1}. \end{aligned}$$

Let g_s and g_c denote the corresponding sums for the inhomogeneous term of (8). Then, any symmetric compact scheme for the Helmholtz equation (8) can be written in the form

$$A_0 u_{m,n} + A_s u_s + A_c u_c = B_0 g_{m,n} + B_s g_s + B_c g_c,$$

where A_0, A_s, and A_c represent, respectively, the coefficients of the center, side, and corner nodes of the compact stencil acting on the solution u. The 4th order scheme described in [21] (with k^2 replaced by $-k^2$) is given by

$$A_0 = -\frac{10}{3} - \frac{2}{3}k^2h_x^2, \quad A_s = \frac{2}{3} - \frac{k^2h_x^2}{12}, \quad A_c = \frac{1}{6}. \tag{13}$$

We define the stencil operating on the right-hand side of (8) by B_i corresponding to the A_i. Then,

$$B_0 = \frac{2h_x^2}{3}, \quad B_s = \frac{h_x^2}{12}, \quad B_c = 0. \tag{14}$$

Define $\lambda(x) = \frac{c(x)h_t}{h_x}$ which we shall refer to as the Courant-Friedrichs-Lewy (CFL) number. In [5], we showed that for the interior domain the stability condition is given by $\lambda^2 \le \frac{3}{8}$. In [22], we present the scheme in three dimensions and for a variable speed of sound.

To preserve fourth order spatial accuracy, the Laplacian that appears in (11) also needs to be approximated to fourth order accuracy. One option to do this is to use a fourth order accurate Padé approximation for each second order derivative. Thus, the second derivative in each direction would be approximated by

$$\frac{1}{12}(D_{j+1} + D_{j-1}) + \frac{5}{6}D_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2},$$

where D is an approximation to the second derivative bound by a Padé approximation (rather than central differences). This leads to a tridiagonal system which can be rapidly solved. An alternative approach is to note that we have already solved (8) for u^{n+1}. We then use (8) to find Δu^{n+1}.

$$\Delta u^{n+1} = k^2 u^{n+1} + G^{n+1}, \tag{15}$$

which is then stored (we need only store two steps). Thus, when evaluating (11) we use the previously stored Δu^n and Δu^{n-1} (known to fourth order accuracy). To complete this recursion, we need only Δu at the initial and first time step. Initially, we know u explicitly and so can calculate Δu^0 analytically. For the next time step, we have the Taylor series (12) which gives u^1 explicitly to fourth order accuracy. We can compute its Laplacian by either a Padé approximation or by an explicit fourth order finite difference formula with a larger stencil. Since the source is compact, u is zero near the boundary at time Δt and so the larger stencil causes no difficulties near the boundary. Computations show that the two approaches yield a similar error and require similar CPU times.

3.1. Stability analysis

We now present a generalization of the stability proof by Britt et al. [5], Smith et al. [22], or Z. Li [16]. The following analysis is general provided that the spatial scheme is self-adjoint negative definite. For example, the wave equation can be extended to include a self-adjoint Laplacian with variable coefficients. We now consider the effect of the dissipative term pu_t on the stability.

We consider the stability analysis for a generalized wave equation

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} + p\frac{\partial u}{\partial t} = Lu.$$

where both c and p are functions of the space variables and p is nonnegative. Let $\frac{1}{h_x^2}L_h$ be the numerical approximation to L (in Section 2.1, $L \equiv \Delta$), where h_x is the uniform spatial grid size and the operator L_h has the following properties:

1. L_h is negative definite, i.e., there exists a real inner product so that $(u, v) = (v, u)$ and $(-L_h u, u) \geq 0$. Further, we require that $0 < L_{\text{lower}}\|u\|^2 \leq (-L_h u, u) \leq L_{\text{upper}}\|u\|^2$.
2. L_h is self-adjoint, so that $(L_h u, v) = (u, L_h v)$. Hence there exists a symmetric or anti-symmetric matrix M which satisfies $M^2 = L_h$. Thus $(L_h u, u) = (Mu, Mu)$, showing that $(-L_h u, u)$ is a norm. Note that, in the case of a one-dimensional PDE, L_h is a second derivative while M is a first derivative which is anti-symmetric.

We consider a modified θ scheme, with $0 \leq \theta \leq 1$. We replace $\frac{1}{12}$ in (6) with θ i.e., the scheme is fourth order accurate when $\theta = \phi = \frac{1}{12} + O(h_t^2)$, see (10). We allow $c = c(\mathbf{x})$ and $p = p(\mathbf{x})$ and replace Δ in (6) with $\frac{L_h}{h_x^2}$. Dividing by c^2 and setting $F = 0$, we obtain from (6)

$$\begin{aligned}
&\left(\frac{1}{c^2 h_t^2} + \frac{\theta p^2}{c^2}\right)\left(u^{n+1} - 2u^n + u^{n-1}\right) + p\frac{u^{n+1} - u^{n-1}}{2c^2 h_t} \\
&= \frac{1}{h_x^2}L_h\left(\theta u^{n+1} + (1-2\theta)u^n + \theta u^{n-1}\right) + \frac{h_t}{2h_x^2}L_h\left(p\theta(u^{n+1} - u^{n-1})\right) \\
&= \frac{1}{h_x^2}L_h\left(\theta(u^{n+1} - u^n) - \theta(u^n - u^{n-1}) + u^n\right) + \frac{h_t}{2h_x^2}L_h\left(p\theta(u^{n+1} - u^{n-1})\right).
\end{aligned} \tag{16}$$

Define $v^{n+1} = u^{n+1} - u^n$. Observe that $v^{n+1} - v^n = u^{n+1} - 2u^n + u^{n-1}$ and $v^{n+1} + v^n = u^{n+1} - u^{n-1}$. Multiplying both sides of (16) by h_x^2 and taking the inner product with $u^{n+1} - u^{n-1} = v^{n+1} + v^n$, we obtain

$$\begin{aligned}
&\frac{h_x^2}{c^2 h_t^2}(1 + \theta p^2 h_t^2)(v^{n+1} - v^n, v^{n+1} + v^n) + \frac{h_x^2}{2c^2 h_t}(p(v^{n+1} + v^n), v^{n+1} + v^n) \\
&= (\theta L_h(v^{n+1} - v^n) + L_h u^n, v^{n+1} + v^n) + \frac{h_t}{2}(p\theta L_h(v^{n+1} + v^n), v^{n+1} + v^n).
\end{aligned}$$

As before, $\lambda(x) = \frac{c(x)h_t}{h_x}$. Simplifying, we obtain

$$\begin{aligned}
&\frac{1}{\lambda^2}(1 + \theta p^2 h_t^2)(\|v^{n+1}\|^2 - \|v^n\|^2) + \frac{h_x^2}{2c^2 h_t}(p(v^{n+1} + v^n), v^{n+1} + v^n) \\
&= \theta(L_h(v^{n+1} - v^n), v^{n+1} + v^n) + (L_h u^n, v^{n+1} + v^n) + \frac{h_t}{2}(p\theta L_h(v^{n+1} + v^n), v^{n+1} + v^n).
\end{aligned} \tag{17}$$

Since L_h is self-adjoint, the cross term in $(L_h(v^{n+1}-v^n), v^{n+1}+v^n)$ is zero, so that (17) reduces to

$$\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)(\|v^{n+1}\|^2-\|v^n\|^2)+\frac{h_x^2}{2c^2h_t}(p(v^{n+1}+v^n), v^{n+1}+v^n)$$
$$=\theta\left((L_h v^{n+1}, v^{n+1})-(L_h v^n, v^n)\right)+(L_h u^n, u^{n+1}+v^n)+\frac{h_t}{2}(p\theta L_h(v^{n+1}+v^n), v^{n+1}+v^n).$$

Rearranging yields

$$\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)\|v^{n+1}\|^2-\theta(L_h v^{n+1}, v^{n+1})+\frac{h_x^2}{2c^2h_t}(p(v^{n+1}+v^n), v^{n+1}+v^n) \tag{18}$$
$$=\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)\|v^n\|^2-\theta(L_h v^n, v^n)+(L_h u^n, v^{n+1}+v^n)+\frac{h_t}{2}(p\theta L_h(v^{n+1}+v^n), v^{n+1}+v^n).$$

We rewrite $(L_h u^n, v^{n+1}+v^n)$ using the following identity $v^{n+1}+v^n=u^{n+1}-u^{n-1}$ and

$$(L_h u^n, u^{n+1})-(L_h u^n, u^{n-1})=\frac{1}{4}\left[(L_h v^n, v^n)-(L_h v^{n+1}, v^{n+1})\right. \tag{19}$$
$$\left.-(L_h(u^n+u^{n-1}), u^n+u^{n-1})+(L_h(u^{n+1}+u^n), u^{n+1}+u^n)\right].$$

This is verified by expanding the right hand side. Combining (18) with (19) we get

$$\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)\|v^{n+1}\|^2+\frac{h_x^2}{2c^2h_t}(p(v^{n+1}+v^n), v^{n+1}+v^n)$$
$$+\left(\frac{1}{4}-\theta\right)(L_h v^{n+1}, v^{n+1})-\frac{1}{4}(L_h(u^{n+1}+u^n), u^{n+1}+u^n) \tag{20}$$
$$=\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)\|v^n\|^2+\left(\frac{1}{4}-\theta\right)(L_h v^n, v^n)-\frac{1}{4}(L_h(u^n+u^{n-1}), u^n+u^{n-1})$$
$$+\frac{h_t}{2}(p\theta L_h(v^{n+1}+v^n), v^{n+1}+v^n).$$

Define

$$S_n=\frac{1}{\lambda^2}(1+\theta p^2 h_t^2)\|v^n\|^2+\left(\frac{1}{4}-\theta\right)(L_h v^n, v^n)-\frac{1}{4}(L_h(u^n+u^{n-1}), u^n+u^{n-1}).$$

Then, (20) is equivalent to

$$S_{n+1}=S_n-\frac{h_x^2}{2c^2h_t}(p(v^{n+1}+v^n), v^{n+1}+v^n)+\frac{h_t}{2}(p\theta L_h(v^{n+1}+v^n), v^{n+1}+v^n).$$

However,

$$p_{\min}\|v^{n+1}+v^n\|^2\le\frac{h_x^2}{2h_t}(p(v^{n+1}+v^n)), v^{n+1}+v^n))\le p_{\max}\|v^{n+1}+v^n\|^2.$$

In addition, L_h is negative definite and so if p is constant then $(pL_h(v^{n+1}+v^n), v^{m+1}+v^n)$ is nonpositive. Hence, the quantity S_n is non-increasing during the calculation. To prove stability it remains to show that S_n is equivalent to a norm. We stress that the conditions for stability, i.e., the CFL condition, depend only on the properties of the spatial operator and are not affected by the additional term pu_t. Thus, the maximal time step allowed is independent of the dissipative layer that we have added.

We now check the stability of the scheme in two cases:

1. $\theta\ge\frac{1}{4}$: Since L_h is negative definite, every term in S_n is positive. Hence, defining $\|u\|_E^2=S_n$, we have energy loss of $\|u\|_E$, and the scheme is unconditionally stable.
2. $0\le\theta<\frac{1}{4}$: We now use the assumption that $0<L_{\text{lower}}\|u\|^2\le(-L_h u, u)\le L_{\text{upper}}\|u\|^2$. We then get

$$\left(\frac{1}{\lambda^2}-\left(\frac{1}{4}-\theta\right)L_{\text{upper}}\right)\|v^n\|^2+\frac{L_{\text{lower}}}{4}\|u^n+u^{n-1}\|^2\le S_n\le$$
$$\left(\frac{1}{\lambda^2}+\left(\frac{1}{4}-\theta\right)L_{\text{lower}}\right)\|v^n\|^2+\frac{L_{\text{upper}}}{4}\|u^n+u^{n-1}\|^2.$$

Therefore, S_n is equivalent to the norm $\|u^n - u^{n-1}\|^2 + \|u^n + u^{n-1}\|^2$ if and only if $\frac{1}{\lambda^2} - \left(\frac{1}{4} - \theta\right) L_{\text{upper}} \geq 0$. Also $\|u^n - u^{n-1}\|^2 + \|u^n + u^{n-1}\|^2 = 2(\|u^n\|^2 + \|u^{n-1}\|^2)$. Thus, when $\theta < \frac{1}{4}$ the scheme is stable provided that

$$\max_x \lambda(x)^2 \leq \frac{1}{\left(\frac{1}{4} - \theta\right) L_{\text{upper}}}.$$

For $\theta = \frac{1}{12}$, this yields $\lambda^2 \leq \frac{6}{L_{\text{upper}}}$ which is 50% larger than for the explicit scheme $\theta = 0$. For the five point second order central difference stencil, $L_{\text{upper}} = 8$ and the stability condition is $\lambda \leq \sqrt{0.75}$. For a fourth order spatial approximation to the modified Helmholtz equation, the value of L_{upper} depends on the details of the scheme.

While we have derived the stability limit on the time step, there are occasions when one wishes to use a larger time step. This may occur because there are regions with small grid cells that locally restrict the time step, which then implies a globally small time step. Another scenario is when there are several wave speeds. The time step is then limited by the fastest wave, which may be physically negligible. One way to extend the time step beyond the CFL limit is to perturb the unstable eigenvalues or, alternatively, use filtering of the high frequencies. Accuracy can be kept by using dispersion transforms. This allows a time step 2-3 times larger than the CFL limit, see Gao et al. [12,13] and also [17]. A different approach is to use deep learning to train the scheme. This allows a much larger time step without any loss of accuracy, see [19].

4. Numerical simulations in 2D

In this section we will present several 2D computations demonstrating the effect of the dissipative layer. The next section will present several 3D results. The two and three dimensional results were produced by two independent codes developed by the first two authors. Therefore, the details are not always identical.

4.1. Scheme implementation

In order to construct an exact solution it is sufficient to consider F as a function of x and y but not t. This simplifies (11). We rewrite (7) as:

$$\left(\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}\right) u^{n+1} - \left(\frac{c^2}{12} + \frac{ph_t c^2}{24}\right) \Delta u^{n+1} = \left(\frac{p^2}{6} + \frac{2}{h_t^2}\right) u^n - \left(\frac{p^2}{12} - \frac{p}{2h_t} + \frac{1}{h_t^2}\right) u^{n-1}$$
$$+ \left(\frac{5c^2}{6}\right) \Delta u^n + \left(\frac{c^2}{12} - \frac{ph_t c^2}{24}\right) \Delta u^{n-1} + F.$$

To ease the notation, divide by the coefficient of u^{n+1} and set:

$$\omega_1 = \frac{\frac{c^2}{12} + \frac{ph_t c^2}{24}}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}}, \qquad \omega_2 = \frac{\frac{p^2}{6} + \frac{2}{h_t^2}}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}}, \qquad \omega_3 = \frac{\frac{p^2}{12} + \frac{1}{h_t^2} - \frac{p}{2h_t}}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}},$$
$$\omega_4 = \frac{\frac{5c^2}{6}}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}}, \qquad \omega_5 = \frac{\frac{c^2}{12} - \frac{ph_t c^2}{24}}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}}, \qquad \omega_6 = \frac{1}{\frac{p^2}{12} + \frac{p}{2h_t} + \frac{1}{h_t^2}}.$$

Therefore,

$$u^{n+1} - \omega_1 \Delta u^{n+1} = \omega_2 u^n - \omega_3 u^{n-1} + \omega_4 \Delta u^n + \omega_5 \Delta u^{n-1} = G,$$

which conforms with (8). We use the 3×3 stencils from (13) and (14) to construct the following matrices:

$$A = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & \ldots & A_c & A_s & A_c & 0 & \ldots & A_s & A_0 & A_s & 0 & \ldots & A_c & A_s & A_c & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & A_c & A_s & A_c & \ldots & 0 & A_s & A_0 & A_s & \ldots & 0 & A_c & A_s & A_c & \ldots & 0 & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix},$$

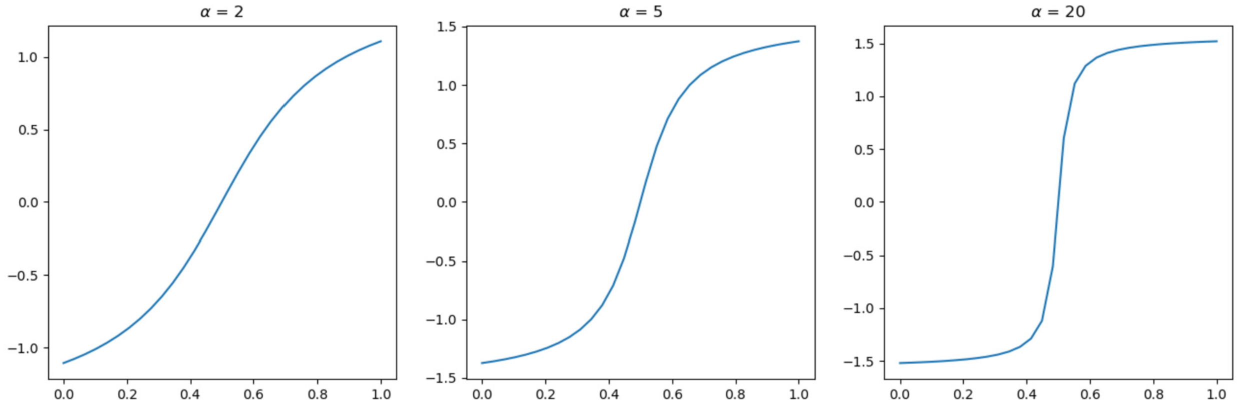


Fig. 1. Damping profile examples for different α values.

$$B=\begin{pmatrix}
1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0\\
0 & 1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0\\
\vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots\\
0 & 0 & \dots & B_c & B_s & B_c & 0 & \dots & B_s & B_0 & B_s & 0 & \dots & B_c & B_s & B_c & 0 & \dots & 0 & 0\\
0 & 0 & \dots & 0 & B_c & B_s & B_c & \dots & 0 & B_s & B_0 & B_s & \dots & 0 & B_c & B_s & B_c & \dots & 0 & 0\\
\vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 0\\
0 & & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 1
\end{pmatrix}.$$

Finally, we directly solve $Au^{n+1}=BG$ to get u^{n+1}. To ensure fourth order accuracy and reduce computational complexity, in the first time step we calculate Δu^0 and Δu^1 to 4th order accuracy. For the next time steps we calculate Δu^{n+1}. All $\omega_k,\ 1\le k\le 6$, are dependent on the damping profile p. In the interior we set $p{=}0$ and in the sponge layers we create a damping profile:

$$p(s)=p_0\arctan(\alpha(2s-1)),\qquad s\in[0,1].$$

This creates a one-dimensional damping profile. We use this profile for the sponge layers on each of the four boundaries. For example, for the top boundary where $x\in[0,\pi]$ and $y\in\left[-\frac{\pi}{2},0\right]$, we get

$$p(x,y)=p_0\arctan\left(\alpha\left(2\left(1+\frac{2}{\pi}y\right)-1\right)\right)$$

We use similar expressions for the other boundaries. We first normalize the damping profile so that the values of p are between 0 and 1, and then multiply the values by a factor p_0, so the final profile values are between 0 (on the interface with the interior) and p_0 (on the outer boundary, end of the sponge layers). Based on numerical tests, a small value of p_0 results in less absorption in the sponge layers and a large value of p_0 results in reflections from the interface between the sponge and the interior. To choose p_0 we conducted a series of numerical experiments, enlarging the value and monitoring the total energy in the system to determine the value that leads to the lowest energy (best absorption) and we achieved that in this setup $p_0{=}210^5$ was a good choice in this trade-off. On the corners, we arbitrarily set the damping profile as one of the horizontal boundaries and achieved sufficiently good results and there was no need to create a complex definition for the corners. The choice of α dictates how steep the damping profile is. A rather flat profile results in low absorption and a steep one produces numerical artifacts. We chose $\alpha{=}5$ using trial and error. For each side of the domain (upper, lower, left and right boundaries) we create an appropriate profile and adjust the coefficients ω_k accordingly. Examples of different one dimensional damping profiles are shown in Fig. 1 (before normalization between 0 and 1).

The initial condition is a compactly supported Gaussian of the form $Ae^{\frac{-\left((x-x_0)^2+(y-y_0)^2\right)}{2\sigma}}$ and for the numerical tests we set $A{=}\sigma{=}1$. The initial velocity is set to $u_t(x,y,t=0)\equiv 0$. We use the above implementation to iterate over $N_t{=}400$ time steps on a grid of size $N_x\times N_y$ with sponge size N_s. The values of N_x, N_y and N_s vary in different tests. We chose $\Delta t=10^{-5}$. to comply with the CFL condition, and $\Delta x=\Delta y=\frac{\pi}{N_x}$, so the spatial size is $[0,\pi]\times[0,\pi]$.

4.2. 2D results

As mentioned above we were not able to construct a stable compact fourth order accurate scheme for a PML that terminates the time dependent acoustic wave equation. The existing algorithms in the literature are not compact. As noted

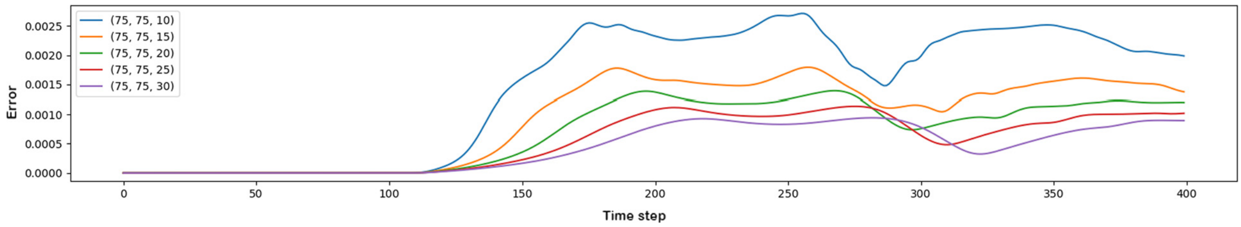
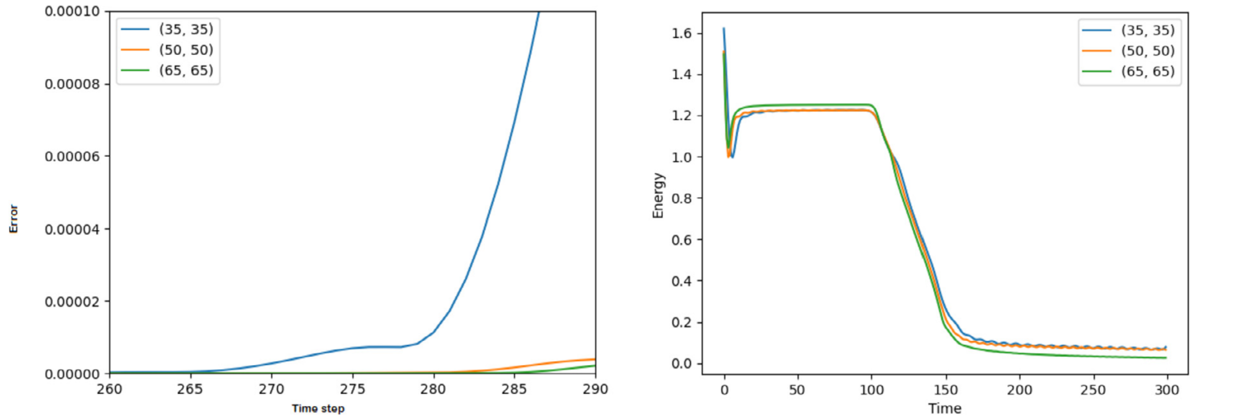


Fig. 2. Graph of the error over time for a single mesh and varying sponge layer thickness.


(a) The error over time for different meshes and scaling sponge layer thickness.

(b) The energy over time for different meshes and scaling sponge layer thickness.

Fig. 3. Graphs of the error and energy over time for different meshes and scaling sponge layer thickness.

above most of them are for the Helmholtz equation. The scheme of Duru and Kreiss [10] is also non-compact and requires a completely different scheme in the interior based on summation by parts. Hence, we do not present any comparisons between the fourth order accurate sponge layer and a PML alternative. To check the accuracy we compare the two dimensional numerical solution u_{ABC} on a grid with the sponge layers Ω_{ABC} to the solution u_L on a much larger grid, without sponge layers Ω_L. The large grid simulates open boundaries. When comparing the two grids, we compute the numerical artifacts that return from the sponge layers back to the domain. The grid Ω_L is of size $3N_x \times 3N_y$ and the spatial size is set to $[-\pi, 2\pi] \times [-\pi, 2\pi]$. The initial condition on both grids is the same. For Ω_L, we set $p(z) = 0$ so there is no absorption at all. We then measure the difference in the interior of both domains ($[0, \pi] \times [0, \pi]$). We calculate:

$$\text{error}(t) = \frac{1}{N_x \cdot N_y} \|u_{ABC}(x, y, t) - u_L(x, y, t)\|_2 , \qquad (x, y) \in [0, \pi] \times [0, \pi].$$

In Fig. 2, we display the error for a mesh of size 75 × 75 for different sponge layer thicknesses. The error is a function of time. We observe that the error decreases when the thickness of the sponge layer increases. We again stress that the sponge layer requires only one extra variable while a PML requires several new variables plus the solution of ODEs. So, for example, the sponge layer with 30 points requires less storage and CPU than a second order accurate PML with 10 points. Presumably a fourth order accurate PML would require even more storage and CPU time.

We wish to determine the order of accuracy of the proposed method. We compute the error only in the interior domain when using the sponge layer. In addition, we compute the error on the interior when using large grid Ω_L. Before the wavefront has an initial impact on the boundary of the interior, both errors are similar and the difference is 0. We are interested only in the error after the initial impact. To calculate the order of accuracy we first calculate the norm over all the time steps (per grid): $\text{error}_{mesh} = \frac{1}{N_t} \|\text{error}(t)\|_2$. We do this for multiple grids by increasing the grid size. Assuming the error is bounded by: $\text{error}_{mesh} \le \alpha_0 h_{mesh}^{k_0}$, we compute the error rate by: $\text{ErrorRate} = \frac{\frac{log(error_{mesh_1})}{log(error_{mesh_2})}}{\frac{log(h_{mesh_1})}{log(h_{mesh_2})}}$. Here, $mesh_1$ and $mesh_2$ are different grids (of different size). The mesh sizes we test have: 35 × 35, 50 × 50 and 65 × 65 interior points. The sponge size scales with the number of points, so the sizes are 17, 25 and 32 respectively (half of the number of points in the x direction). The error rate between grids 35 × 35 and 50 × 50 received is 10.47 and between 50 × 50 and 65 × 65 is 4.16, ensuring the desired fourth order of accuracy. In Fig. 3a, we display the error for the different grid sizes and Fig. 3b shows the decrease in the energy (L_2-norm) in time and shows how the absorption causes the energy to decrease.

5. Numerical simulations in 3D

5.1. Test solution

For the 3D numerical simulations, we consider the test problem described in Appendix A. Define the test solution as (A.3) with a compactly supported source term (A.5) on the ball of radius $\varepsilon=3.0$ generated by the speed $c(x,y,z)=1$. The smooth function S in (A.2) is given by $S(t)=\sin^{12}\left(\pi t/4\right)\chi_{(0,\infty)}$, the central point is $(x_0,y_0,z_0)=(0,0,0)$, the order of the smooth step function (A.4) is $m=12$, the translation is $t_0=1$, and the multi-index is $\boldsymbol{\beta}=(1,0,0)$.

By construction, the test solution is zero for a short time initially, and then it starts radiating from the origin outward. The shift $t_0=1$ guarantees the initial data are zero and the source term is zero in the beginning. Then, the test solution provides continuous output to the sponge layer and the multi-index $\boldsymbol{\beta}=(1,0,0)$ generates asymmetry with the test solution to further test the capabilities of the sponge layer.

5.2. Nonnegative function $p(x,y,z)$

The nonnegative function $p(x,y,z)$ is a linear combination of univariate functions. The construction is simple and avoids any special considerations in the corner regions of the sponge layer. Given the computational domain $[-l,l]^3$ and the sponge layer thickness dl, define the nonnegative function

$$p^{(x)}(x)=\begin{cases} p_0 s((x-l+dl)/dl) & \text{if } l-dl<x\le l,\\ 0 & \text{if } -l+dl\le x\le l-dl,\\ p_0 s((dl-l+x)/dl) & \text{if } -l\le x<-l+dl,\end{cases} \tag{21}$$

where $p_0>0$, the shape function is $s(\xi)=\frac{1}{2}+\frac{\arctan\left(\alpha(2\xi-1)\right)}{2\arctan\left(\alpha\right)}$ for $\xi\in[0,1]$, and $\alpha=5$. Then, we define the nonnegative function

$$p(x,y,z)=p^{(x)}(x)+p^{(y)}(y)+p^{(z)}(z),$$

where the univariate functions are defined in (21). In principle, the shape function $s(\xi)$ can be any smooth increasing function satisfying $0\le s(\xi)\le 1$ where $s(0)=0$ and $s(1)=1$. For example, we could use the smooth shape function $s(\xi)=\exp\left(1-1/\xi\right)$ for $\xi\in(0,1]$, which satisfies $0=\frac{d^k s}{d\xi^k}(0^+)$ for $k=0,1,2,\ldots$ However, we observed that the results are fairly insensitive to the shape of the profile. So, we will only present the results for the arctangent damping profile. For more details on this smooth shape function and how smooth damping profile affects the performance of a perfectly matched layer see [18].

5.3. 3D solver

To advance the time marching scheme, we need to solve a modified Helmholtz equation (8). This equation is solved by a multigrid method. See [22] for several fourth order compact discretizations of the modified Helmholtz equation on a uniformly discretized cube. The results presented here used the Padé approximation. For the multigrid method, we utilize V-cycles, Gauss-Seidel iterations, full weighting, tri-cubic interpolation, and standard coarsening for the cycle type, smoother, restriction operator, prolongation operator, and construction of the coarse grids, respectively. See [22] for a short summary of the multigrid method and application to the modified Helmholtz equation. See [24,3] for a thorough description of multigrid methods. Because the modified Helmholtz operator is symmetric and negative definite, the multigrid converges very rapidly.

5.4. 3D results

We now investigate the absorption properties of the sponge layer on various grids. We stress again that, there is no compact fourth order PML and so we are not able to compare the number of added variables needed by the compact 4th order sponge layer with a corresponding PML.

The error that we calculate at each time step is quantified using either the discrete L_2 norm or else the ∞ norm. The discrete L_2 error on the upper time level is given by

$$\text{error}_{L_2}(t_{n+1})=\sqrt{h^3\sum_{(x_i,y_j,z_k)\in\Omega_{\text{vac}}}\left(u^{n+1}(x_i,y_j,z_k)-u^{\text{true}}(x_i,y_j,z_k,t_{n+1})\right)^2},$$

and the infinity error on the upper time level is given by

$$\text{error}_{\infty}(t_{n+1})=\max_{(x_i,y_j,z_k)\in\Omega_{\text{vac}}}|u^{n+1}(x_i,y_j,z_k)-u^{\text{true}}(x_i,y_j,z_k,t_{n+1})|,$$

where h is the uniform grid size in space, u^{true} is the test solution described in Section 5.1, and Ω_{vac} is the computational domain without the sponge layer. The overall error is calculated as the supremum of either $\text{error}_{L_2}(t_{n+1})$ or $\text{error}_{\infty}(t_{n+1})$

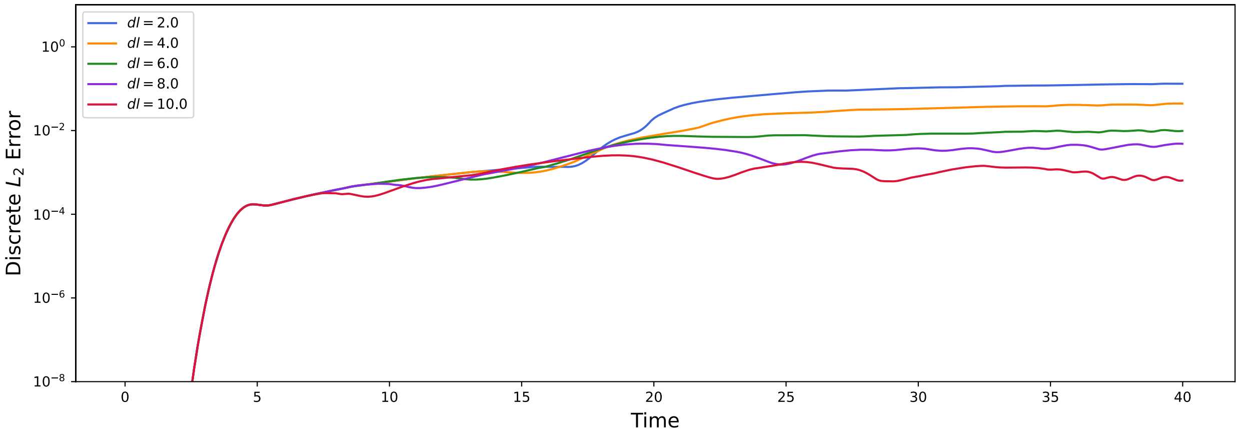


Fig. 4. Error history of the 3D test solution where $p_0 = 1.0$, the computational domain is the cube $[-15.0, 15.0]^3$, and the sponge layer thickness is allowed to vary. A uniform $257 \times 257 \times 257$ grid with size $h = 30/256$ and constant time step $h_t = 297/2560$ were utilized for each graph.

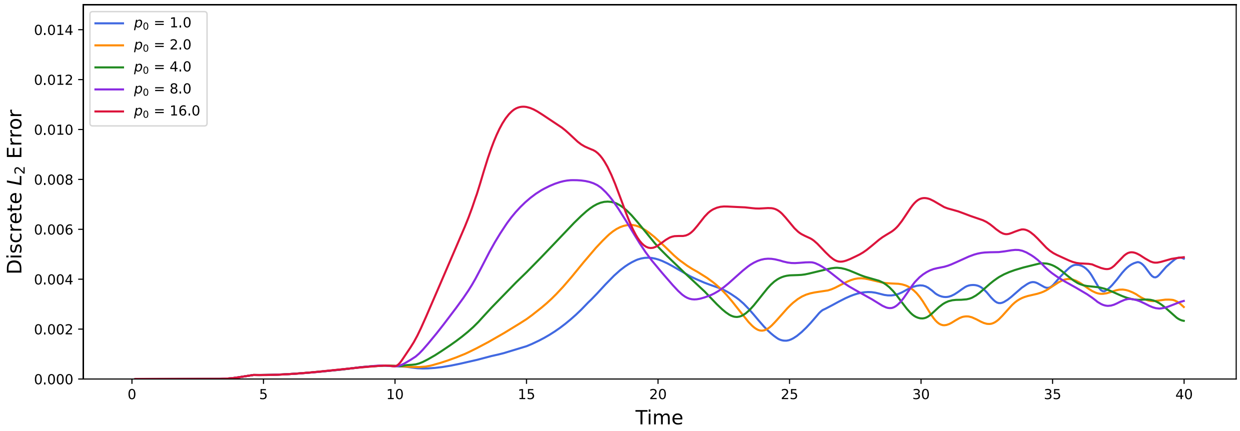


Fig. 5. Error history of the 3D test solution where $dl = 6.0$, the computational domain is the cube $[-15.0, 15.0]^3$, and p_0 is allowed to vary. A uniform $257 \times 257 \times 257$ grid with size $h = 30/256$ and constant time step $h_t = 297/2560$ were utilized for each graph.

over all time levels. Since our multigrid method uses standard coarsening, we always discretize the computational domain using a $(N+2) \times (N+2) \times (N+2)$ grid where $N = 2^J - 1$ for some integer J. We use a uniform grid size $h = 2l/(N+1)$ and uniform time step $h_t = 99h/100$ in all of the simulations. As a consequence, the convergence rate is measured using the quantity

$$\text{convergence rate} = {}^{\text{error}_{2h}}/_{\text{error}_h} / \log(2.0).$$

To increase the absorption in the sponge layer, we consider the parameter p_0 and the thickness of the sponge layer dl. Fig. 4 shows the error history where $p_0 = 1$ and the thickness of the sponge layer varies on the computational domain $[-15.0, 15.0]^3$. Between $t = 0$ and $t = 6$, all the curves coincide because the test solution (A.3) hasn't had enough time to penetrate the sponge layer. When $t > 20.0$, the remaining artificial reflections from the sponge layer have had enough time to reenter the interior region and pollute the solution. Clearly, as the sponge layer thickness increases, the error decreases. On the other hand, Fig. 5 shows the error history when $dl = 6.0$ and p_0 of (21) is allowed to vary on the computational domain $[-15.0, 15.0]^3$. When $t \approx 10.0$, the plots deviate from one another since the test solution is beginning to penetrate the sponge layer. Between $t = 10.0$ and $t = 20.0$, it is evident that as p_0 increases, the error grows. So, there doesn't appear to be any advantage to increasing p_0 beyond 1.0.

Fig. 6 shows the overall error when the thickness dl is fixed and p_0 varies for several values of dl on the computational domain $[-15.0, 15.0]^3$. One can see that changing p_0 carries only a very small benefit. When $dl \in \{2.0, 4.0, 6.0\}$, doubling p_0 from 1 to 2 reduces the error. Unfortunately, increasing p_0 beyond 2.0 produces no additional reduction in the overall error. When $dl \in \{8.0, 10.0\}$, increasing p_0 beyond 1.0 produces no benefit whatsoever since the error increases as p_0 increases. We emphasize that, regardless of the value of p_0, increasing the thickness of the layer always decreases the error. Hence, instead of trying to optimize the pair (p_0, dl), we set $p_0 = 1$ and then choose the best sponge thickness dl to produce the desired level of accuracy.

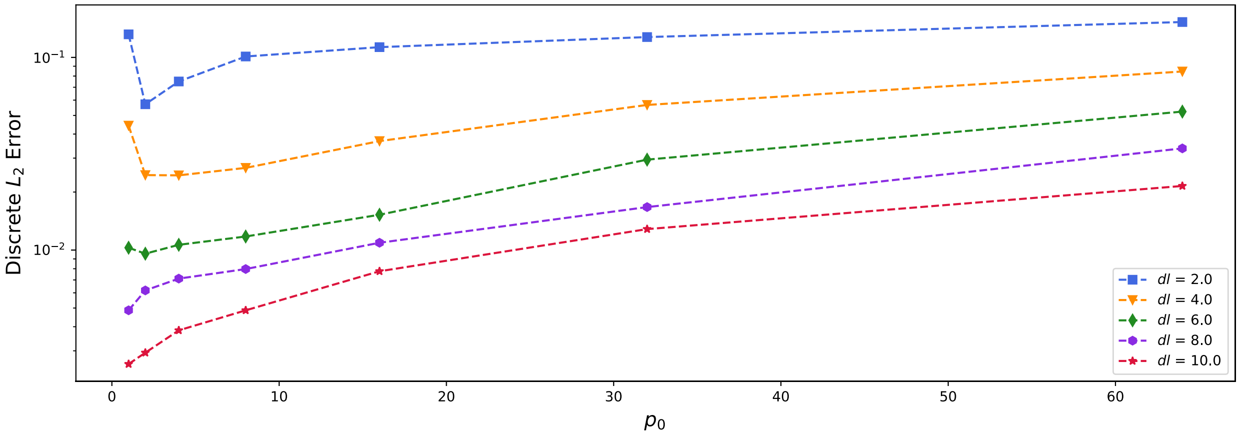


Fig. 6. Error versus p_0 of the 3D test solution on $[-15.0, 15.0]^3$ for various sponge thicknesses *dl*. A terminal time of 40.0, uniform 257 × 257 × 257 grid with size $h=30/256$, and constant time step $h_t=297/2560$ were utilized for each simulation.

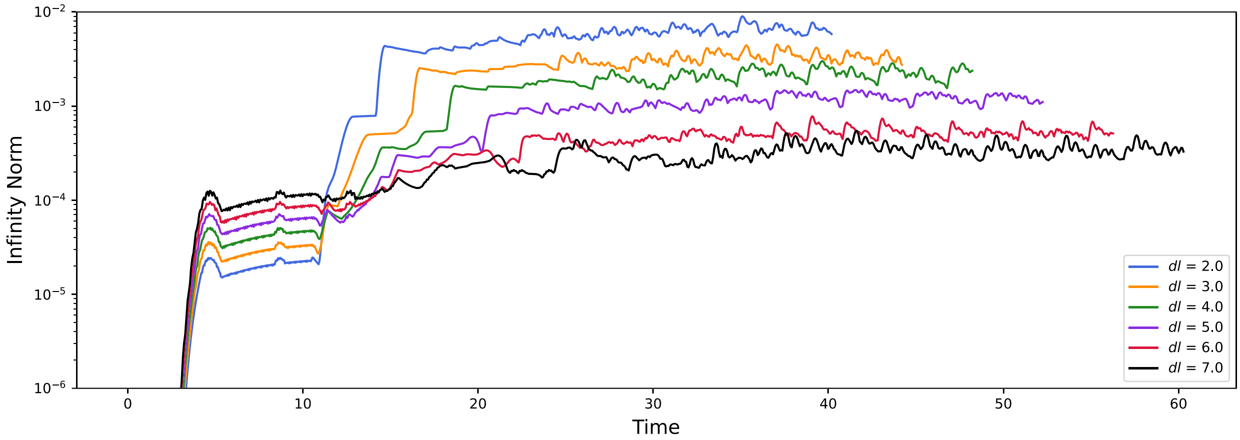


Fig. 7. Error history of the 3D test solution for $\Omega_{\text{vac}}=[-8.0, 8.0]^3$, $p_0=1$, and varying sponge layer thickness *dl* on a 257 × 257 × 257 grid. For each case, the uniform grid size is $h=(16+2dl)/256$ and uniform time step is $h_t=99h/100$ for a total of 900 time steps.

In our previous simulations, the computational domain was fixed and the thickness was allowed to vary. Then, we observed that increasing the thickness of the sponge layer decreased the error. However, this comes at the expense of shrinking the vacuum region, Ω_{vac}. We now consider the case where Ω_{vac} is fixed and $p_0=1$, while the sponge layer thickness varies. If the sponge layer is too thin, it will produce noticeable spurious reflections. Increasing the thickness of the sponge layer will decrease these spurious reflections. However, as we increase *dl* while keeping the Ω_{vac} and the grid dimension fixed, the discretization error of the fourth order compact scheme will increase because the grid size will increase. Therefore, we need to consider both these effects when choosing *dl*.

Fig. 7 shows the error history of the test solution on a 257 × 257 × 257 grid for various sponge thicknesses where $\Omega_{\text{vac}}=[-8.0, 8.0]^3$ is fixed. All error profiles for $t < 9.0$ have a similar shape. The solution hasn't had enough time to penetrate the sponge layer so there can't be any reflection errors, which means the overall error is due solely to the discretization error. After $t \approx 10.0$, the error for each *dl* rises due to the spurious reflections from the layer back into Ω_{vac}. Then, the error flattens. The thicker the layer, the slower the error grows and the smaller its value is for large times. This indicates that the sponge layer is absorbing more effectively as the thickness of the layer increases. However, when increasing the sponge layer thickness from $dl=6.0$ to $dl=7.0$, the improvement in the error becomes marginal. This suggests that the reduction in the reflection error has been balanced by the growth in the discretization error. If we further increase the thickness of the sponge layer while holding Ω_{vac} fixed, the increase in the discretization error will exceed the decrease in the reflection errors.

In Fig. 8, we let $\Omega_{\text{vac}}=[-8.0, 8.0]^3$ be fixed while the sponge layer varies in thickness for several different grids. Increasing the thickness of the sponge layer steadily decreases the error until the minimum error is reached at some optimal thickness. Beyond this optimal thickness, the additional increase in the discretization error due to a larger domain and hence, coarser grid, outweighs the decrease in the reflection error. Grids with higher dimension require a thicker layer to reach the optimal value of the error. Table 1 is a companion to Fig. 8 and lists the settings which produce the optimal error for each grid depicted in Fig. 8.

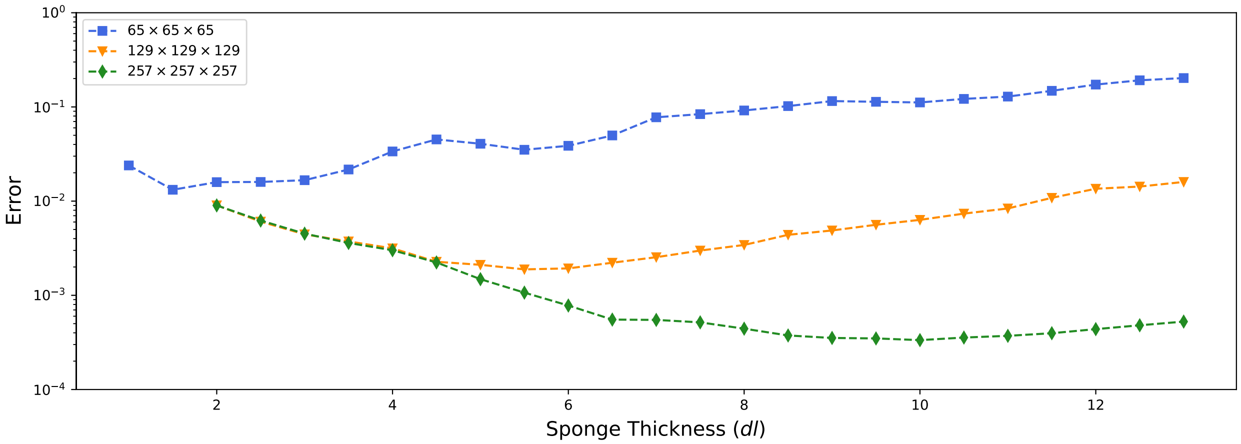


Fig. 8. Error of the 3D test solution for $\Omega_{vac}=[-8.0, 8.0]^3$, $p_0=1$, and varying sponge layer thickness dl on several grids. Given the $(N+2)\times(N+2)\times(N+2)$ grid, the uniform grid size is $h={}^{(16+2dl)}/_{N+1}$ and the uniform time step is $h_t=99h/100$. We use a total of 225, 450, and 900 time steps on the $65\times65\times65$, $129\times129\times129$, and $257\times257\times257$ grid, respectively. In space, we utilize the infinity norm.

Table 1
Optimal parameters of the 3D test solution for $\Omega_{vac}=[-8.0, 8.0]^3$ and the same settings as in Fig. 8.

Grid	Optimal thickness	Grid in vacuum region	Error
65 × 65 × 65	1.50	59 × 59 × 59	1.320e-02
129 × 129 × 129	5.50	77 × 77 × 77	1.886e-03
257 × 257 × 257	10.00	113 × 113 × 113	3.344e-04

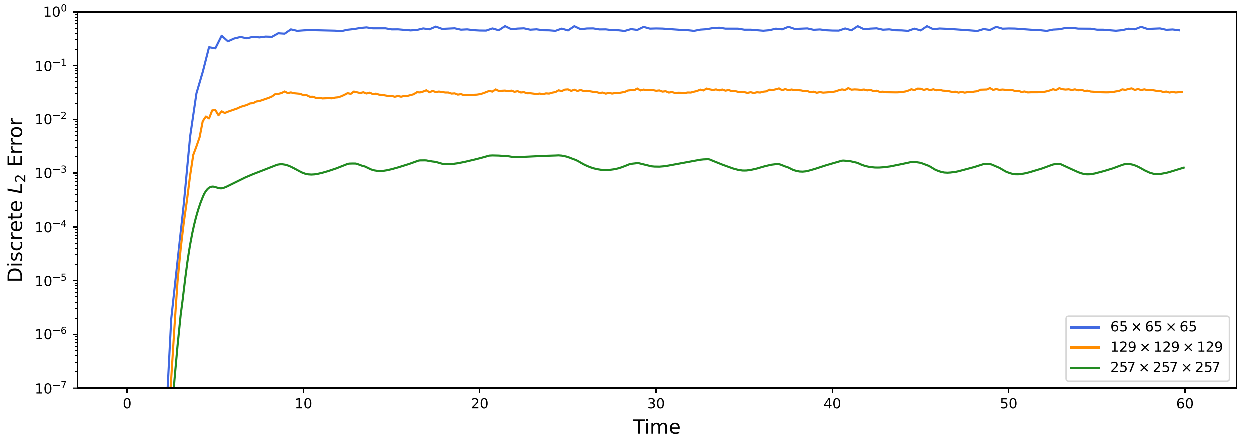


Fig. 9. Error history of the 3D test solution on $[-20.0, 20.0]^3$ for $dl=14.0$ for various grid dimensions. Given the $(N+2)\times(N+2)\times(N+2)$ grid on the computational domain, the uniform grid size $h=40/(N+1)$, and the uniform time step $h_t=99h/100$.

Based on the previous observations, to do a proper grid refinement study one needs a sufficiently thick layer to damp the artificial reflections on the finest grid. However, this thickness will greatly exceed the optimal thickness on coarser grids so that on those grids the discretization error will dominate. As the grid is refined, once can expect convergence with the rate $O(h^4)$.

To demonstrate fourth order convergence, we consider the test problem on $[-20.0, 20.0]^3$ with $dl=14.0$. Fig. 9 shows the error history of the test problem on a $65\times65\times65$, $129\times129\times129$, and $257\times257\times257$ grid. Table 2 demonstrates that fourth order convergence is indeed met on the three grids.

6. Conclusions

We have applied a damping layer to the time dependent acoustic wave equation in both two and three dimensions. We have adapted the previously developed compact fourth order accurate finite difference schemes for the wave equation to computing the solutions on exterior regions. To truncate the unbounded domain of the solution, we have extended the damping layer suggested by Sochacki et al. [23] to fourth order accuracy in both space and time. This requires only the

Table 2
Convergence of the 3D test solution on $[-20.0, 20.0]^3$ for $dl=14.0$. Given a $(N+2)\times(N+2)\times(N+2)$ grid on the computational domain, the uniform grid size $h = 40/(N+1)$, and the uniform time step $h_t = 99h/100$.

Grid domain	Grid vacuum region	Error	Conv. rate	CPUTIME (sec.)
65 × 65 × 65	21 × 21 × 21	5.436e-01	—-	65.2
129 × 129 × 129	39 × 39 × 39	3.824e-02	3.83	1281.0
257 × 257 × 257	79 × 79 × 79	2.140e-03	4.16	19255.6

addition of one variable per grid point within the layer. Thus, one can use a larger layer than the PML but still not have more degrees of freedom. In addition, we have used improved stretching function. In terms of CPU times, the additional term pu_t adds only a negligible overhead compared with u_{tt} and the calculation of the Laplacian. Moreover, we have thoroughly analyzed the stability of our fourth order accurate scheme as applied to the damped wave equation in the layer. It turns out that the addition of the sponge layer does not reduce the allowable time step for stability. At the same time, a PML usually requires a smaller time step. This smaller step is then required in the entire domain and not just the PML adding considerably to the cost of the PML approach. Computations demonstrate that one needs the same order of accuracy in an absorbing layer (PML or sponge) as in the interior in order to preserve the global accuracy. We again stress that we were not able to construct a stable compact fourth order approximation for the PML equations. Preliminary results show that this increase may be slightly greater than linear.

A final advantage of the sponge layer approach is that it is straightforward to apply to a large range of equations. Constructing a PML for the elastic equations or other physical systems is nontrivial. As we have remarked developing a compact fourth order accurate PML for the wave equation in second order form is very nontrivial. This becomes even more difficult for the elastic equations or Maxwell's equations. On the other hand, developing the sponge layer and extending it to higher order is much easier.

CRediT authorship contribution statement

Adar Kahana: Conceptualization, Formal analysis, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Fouche Smith:** Conceptualization, Formal analysis, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Eli Turkel:** Conceptualization, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing. **Semyon Tsynkov:** Conceptualization, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the United States–Israel Binational Science Foundation (BSF) under grant 2020128.

Appendix A. Generating compactly supported test problems

Consider the wave equation with a point source

$$\frac{\partial^2 u}{\partial t^2} - c^2\Delta u = \delta_x(x-x_0)\delta_y(y-y_0)\delta_z(z-z_0)S(t) \qquad (x,y,z,t)\in\mathbb{R}^3\times(0,\infty) \tag{A.1}$$

where the wave speed is constant, the initial conditions are zero, $\delta_\xi(\xi)$ is the one dimension Dirac delta function, and $(x_0, y_0, z_0)\in\mathbb{R}^3$ is the location of the point source. A solution to (A.1) is given by

$$u(x,y,z,t) = \frac{S(t-t_0-r/c)}{4\pi r} \quad \text{with} \quad r = \sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2}, \tag{A.2}$$

which is singular at $(x,y,z) = (x_0,y_0,z_0)$. We generate a smooth solution which resembles (A.2) without the singularity at (x_0, y_0, z_0) using the strategy shown in [20, Section 8.2].

Consider a general solution of the form

$$u^{\text{Test}}_{m,\varepsilon}(x,y,z,t) = D^{\boldsymbol\beta}\left(\frac{\phi_m(r/\varepsilon)S(t-t_0-r/c)}{4\pi r}\right), \tag{A.3}$$

where the multi-index $\boldsymbol\beta = (\beta_1,\beta_2,\beta_3)$, the derivative $D^{\boldsymbol\beta} = \partial_x^{\beta_1}\partial_y^{\beta_2}\partial_z^{\beta_3}$, and the smooth step function

$$\phi_m(x) = \begin{cases} 0 & \text{if } x \le 0, \\ x^{m+1} \sum_{k=0}^{m} \binom{m+k}{m}\binom{2m+1}{m-k}(-x)^k & \text{if } 0 < x < 1\,, \\ 1 & \text{if } x \ge 1. \end{cases} \tag{A.4}$$

This satisfies $\frac{d^k\phi_m}{dx^k}(0^+) = 0 = \frac{d^k\phi_m}{dx^k}(1^-)$ for $k = 1, \ldots, m$. By construction, $\phi(r/\varepsilon)/r = \mathcal{O}(r^m)$ as $r \downarrow 0$, which implies (A.3) is smooth in a neighborhood of (x_0, y_0, z_0) provided m is sufficiently large. In addition, (A.3) is equal to (A.2) on the complement of the ball of radius ε centered at (x_0, y_0, z_0) if $\boldsymbol{\beta} = (0, 0, 0)$. Finally, the smooth test solution (A.3) satisfies the constant speed wave equation, where the source term

$$F^{\text{Test}}_{m,\varepsilon}(x, y, z, t) \triangleq D^{\boldsymbol{\beta}}\left(\frac{c\phi'_m(r/\varepsilon)S'(t - t_0 - r/c)}{2\pi\varepsilon r} - \frac{c^2\phi''_m(r/\varepsilon)S(t - t_0 - r/c)}{4\pi\varepsilon^2 r}\right) \tag{A.5}$$

is compactly supported in space on the ball of radius ε centered at $(x_0, y_0, z_0) \in \mathbb{R}^3$ and t_0 is chosen sufficiently large to guarantee that the initial data is zero.

References

[1] S. Barros, F. Silva, P. Araújo, P.S. Peixoto, On absorbing boundary conditions and boundary layers for the acoustic wave equations on unbounded domains, J. Comput. Phys. 139 (1) (1998) 184–208.

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

[3] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second edition, Society for Industrial and Applied Mathematics, 2000.

[4] S. Britt, S.V. Tsynkov, E. Turkel, Numerical simulation of time-harmonic waves in inhomogeneous media using compact high order schemes, Commun. Comput. Phys. 9 (2011) 520–541.

[5] D.S. Britt, E. Turkel, S. Tsynkov, A high order compact time/space finite difference scheme for the wave equation in variable media, J. Sci. Comput. 76 (2018) 777–811.

[6] C. Cerjan, D. Kosloff, R. Kosloff, M. Reshhel, Nonreflecting boundary condition for discrete acoustic and elastic waves, Geophysics 50 (8) (1985) 705–708.

[7] Z. Chen, T. Wu, H. Yang, An optimal 25-point finite difference scheme for the Helmholtz equation with PML, J. Comput. Appl. Math. 236 (6) (2011) 1240–1258.

[8] H. Dastour, W. Liao, A fourth-order optimal finite difference scheme for the Helmholtz equation with PML, Comput. Math. Appl. 6 (78) (2019) 2147–2165.

[9] H. Dastour, W. Liao, An optimal 13-point finite difference scheme for a 2D Helmholtz equation with a perfectly matched layer boundary condition, Numer. Algorithms 6 (3) (2021) 1109–1141.

[10] K. Duru, G. Kreiss, On the accuracy and stability of the perfectly matched layer in transient waveguides, J. Sci. Comput. 53 (2012) 642–671.

[11] Y. Gao, H. Song, J. Zhang, Z. Yao, Comparison of artificial boundaries for acoustic wave equation modeling, Explor. Geophys. 48 (2017) 76–93.

[12] Y. Gao, J. Zhang, Z. Yao, Removing the stability limit of the explicit finite-difference scheme with eigenvalue perturbation, Geophysics 83 (6) (2018) A93–A98.

[13] Y. Gao, J. Zhang, Z. Yao, Extending the stability limit of explicit scheme with spatial filtering for solving wave equations, J. Comput. Phys. 397 (11) (2019) 108853.

[14] J.D. Lambert, Computational Methods in Ordinary Differential Equations, John Wiley and Sons, NY, 1973.

[15] Jun-Woo Lee, Dong-Joo Min, Improvement of sponge boundary condition for seismic wave modeling, in: SEG International Exposition and Annual Meeting, San Antonio, Texas, September 2019, Paper Number: SEG-2019-3211925.

[16] Z. Li, http://tinyurl.com/z5x7log, or http://www.math.pku.edu.cn.

[17] Chao Lyu, Yann Capdeville, Gang Lu, Liang Zhao, Removing the courant-Friedrichs-Lewy stability criterion of the explicit time-domain very high degree spectral-element method with eigenvalue perturbation, Geophysics 86 (5) (2021), https://doi.org/10.1190/geo2020-0623.1.

[18] A.F. Oskooi, L. Zhang Lei, Y. Avniel, S.G. Johnson, The failure of perfectly matched layers, and towards their redemption by adiabatic absorbers, Opt. Express 16 (15) (2008) 11376–11392.

[19] O. Ovadia, A. Kahana, E. Turkel, S. Dekel, Beyond the courant-Friedrichs-Lewy condition: numerical methods for the wave problem using deep learning, J. Comput. Phys. 442 (10) (2021) 110493.

[20] S. Petropavlovsky, S. Tsynkov, Non-deteriorating time domain numerical algorithms for Maxwell's electrodynamics, J. Comput. Phys. 336 (2017) 1–35.

[21] I. Singer, E. Turkel, High-order finite difference methods for the Helmholtz equation, Comput. Methods Appl. Mech. Eng. 163 (1–4) (1998) 343–358.

[22] F. Smith, S. Tsynkov, E. Turkel, Compact high order accurate schemes for the three dimensional wave equation, J. Sci. Comput. 81 (3) (2019) 1181–1209.

[23] J.Sochacki.R. Kubichek, J. George, W.R. Fletcher, S. Smithson, Absorbing boundary conditions and surface waves, Geophysics 52 (1) (1987) 60–71.

[24] U. Trottenberg, C.W. Oosterlee, A. Schuller, Multigrid, Elsevier, 2000.