

# High-Order Schemes for Acoustic Waveform Simulation

Seongjai Kim\*

March 25, 2002

## Abstract

This article introduces a new fourth-order *implicit* time-stepping scheme for the numerical solution of the acoustic wave equation, as a variant of the conventional modified equation method. For an efficient simulation, the scheme incorporates a locally one-dimensional (LOD) procedure having the splitting error of  $\mathcal{O}(\Delta t^4)$ . Its stability and accuracy are compared with those of the standard explicit fourth-order scheme. It has been observed from various experiments for 2D problems that (a) the computational cost of the implicit LOD algorithm is only about 40% higher than that of the explicit method, for the problems of the same size, (b) the implicit LOD method produces slightly less dispersive solutions in heterogeneous media, and (c) its numerical stability and accuracy match well with those of the explicit method.

## 1. Introduction

Let  $\Omega \subset \mathbb{R}^m$ ,  $1 \leq m \leq 3$ , be a bounded domain with its boundary  $\Gamma = \partial\Omega$  and  $J = (0, T]$  the time interval,  $T > 0$ . Consider the following acoustic wave equation

$$\begin{aligned} \text{(a)} \quad & \frac{1}{c^2} u_{tt} - \Delta u = S(\mathbf{x}, t), & (\mathbf{x}, t) \in \Omega \times J, \\ \text{(b)} \quad & \frac{1}{c} u_t + u_\nu = 0, & (\mathbf{x}, t) \in \Gamma \times J, \\ \text{(c)} \quad & u(\mathbf{x}, 0) = g_0(\mathbf{x}), \quad u_t(\mathbf{x}, 0) = g_1(\mathbf{x}), & \mathbf{x} \in \Omega, \quad t = 0, \end{aligned} \tag{1.1}$$

where  $c = c(\mathbf{x}) > 0$  denotes the normal velocity of the wavefront,  $S$  is the wave source/sink,  $\nu$  denote the unit outer normal from  $\Gamma$ , and  $g_0$  and  $g_1$  are initial data.

---

\*Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA  
Email: skim@ms.uky.edu The work of the author is supported in part by NSF grant DMS-0107210.

Wave problems are often formulated in an unbounded domain. These problems can be solved numerically by first truncating the given unbounded domain, imposing a suitable absorbing boundary condition (ABC) on the (artificial) boundary of the truncated bounded domain, and then solving the resulting problem using discretization methods (e.g., finite differences, finite elements, and spectral methods). Equation (1.1.b) has been popular as a simple-but-effective ABC, since introduced by Clayton and Engquist [3]. Equation (1.1) has been extensively studied as a model problem for second-order hyperbolic problems by many authors; see e.g. [1, 2, 5, 11, 13]. It is often the case that the source is given in the following form

$$S(\mathbf{x}, t) = \delta(\mathbf{x} - \mathbf{x}_s)f(t),$$

where  $\mathbf{x}_s \in \Omega$  is the source point. For the function  $f$ , the Ricker wavelet of frequency  $\nu$  can be chosen, i.e.,

$$f(t) = \pi^2 \nu^2 (1 - 2\pi^2 \nu^2 t^2) e^{-\pi^2 \nu^2 t^2}.$$

In Geophysical applications, the wave equation (1.1) is often solved by explicit time-stepping schemes, which require to choose the time step size sufficiently small to satisfy the stability condition and to reduce numerical dispersion as well. Alternative conventional approaches for solving wave equations introduce an auxiliary variable to rewrite the equation as a first-order hyperbolic system. With these approaches one introduces new unknowns, which result in an increase in the number of variables in the discrete problems. Thus, there are good reasons to try to keep the formulation involving the second time-derivative and a scalar unknown. However, it has been known that with this formulation it is hard to construct methods combining good stability with high accuracy. In particular, it is hard to incorporate a high-order approximation of the ABC. In this paper we shall introduce a one-parameter family of three-level methods incorporating the locally one-dimensional (LOD) time-stepping procedure for an efficient simulation. It is analyzed to be unconditionally stable for the parameter in a certain range.

An outline of the article is as follows. In the next section, we first review the conventional methods: explicit (three-level) schemes and the two-level implicit scheme. Section 3 introduces a new three-level implicit scheme. A locally one-dimensional (LOD) perturbation having the splitting error in  $\mathcal{O}(\Delta t^4)$  is considered for an efficient simulation. Its stability and computational complexity are compared with those of the standard explicit fourth-order scheme in the same section. In Section 4, we present some numerical results showing numerical stability, efficiency, and accuracy of the new scheme. In Section 5, we discuss strategies of incorporating high-order approximations of the ABC. The last section included conclusions.

## 2. Preliminaries

In this section, we review conventional methods for the numerical solution of the wave equation (1.1). Let  $\mathcal{A}$  denote an approximation of  $-\Delta$  of order  $p$ , i.e.,

$$\mathcal{A}u \approx -\Delta u + \mathcal{O}(h^p),$$

where  $h$  is the grid size; in most cases,  $p$  is 2 or 4. Then, the semi-discrete equation for the acoustic wave equation reads

$$\frac{1}{c^2} v_{tt} + \mathcal{A}v = S. \quad (2.1)$$

(Here we have omitted the equations for the boundary and initial conditions, for a simpler presentation.)

It now remains to discretize the second-order system of ODEs (2.1) with respect to the time variable. Let  $\Delta t$  be the time step size and  $t^n = n\Delta t$ . Define  $v^n(\mathbf{x}) = v(\mathbf{x}, t^n)$ . For a simpler presentation, we define the following difference operator

$$\bar{\partial}_{tt}v^n := \frac{v^{n+1} - 2v^n + v^{n-1}}{\Delta t^2}.$$

### 2.1. Explicit schemes

Explicit methods are still popular in the simulation of waveforms. We begin with the second-order scheme (in time) formulated as

$$\frac{1}{c^2} \bar{\partial}_{tt}v^n + \mathcal{A}v^n = S^n. \quad (2.2)$$

As a stability constraint, the scheme requires to choose

$$\Delta t = \mathcal{O}(h).$$

The scheme (2.2) works well for smooth solutions, but otherwise it can introduce severe nonphysical oscillations.

To formulate the fourth-order scheme (in time), consider the Taylor expansion

$$v_{tt}(t^n) \approx \bar{\partial}_{tt}v^n - \frac{\Delta t^2}{12} v_{tttt}(t^n) + \mathcal{O}(\Delta t^4). \quad (2.3)$$

It follows from (2.1) that

$$v_{tttt}(t^n) = c^2(S_{tt}^n - \mathcal{A}v_{tt}^n) = c^2[S_{tt}^n - \mathcal{A}(c^2(S^n - \mathcal{A}v^n))]. \quad (2.4)$$

Utilizing the above identity, (2.3) can be rewritten as

$$\begin{aligned} v_{tt}(t^n) &\approx \bar{\partial}_{tt}v^n - \frac{\Delta t^2}{12}c^2[S_{tt}^n - \mathcal{A}(c^2(S^n - \mathcal{A}v^n))] + \mathcal{O}(\Delta t^4) \\ &\approx \bar{\partial}_{tt}v^n - \frac{\Delta t^2}{12}c^2[\bar{\partial}_{tt}S^n - \mathcal{A}(c^2(S^n - \mathcal{A}v^n))] + \mathcal{O}(\Delta t^4), \end{aligned} \quad (2.5)$$

where we have used  $S_{tt}^n \approx \bar{\partial}_{tt}S^n + \mathcal{O}(\Delta t^2)$ . Thus the explicit fourth-order algorithm reads

$$\frac{1}{c^2}\bar{\partial}_{tt}v^n + \mathcal{A}\left(v^n - \frac{\Delta t^2}{12}c^2\mathcal{A}v^n\right) = S^n + \frac{\Delta t^2}{12}(\bar{\partial}_{tt}S^n - \mathcal{A}c^2S^n) \quad (2.6)$$

or, equivalently,

$$\begin{aligned} v^{n+1} &= 2v^n - v^{n-1} - \Delta t^2c^2\mathcal{A}\left(v^n - \frac{\Delta t^2}{12}c^2\mathcal{A}v^n\right) \\ &\quad + \Delta t^2c^2\left[S^n + \frac{\Delta t^2}{12}(\bar{\partial}_{tt}S^n - \mathcal{A}c^2S^n)\right]. \end{aligned} \quad (2.7)$$

See [4, 5, 14] for details.

*Remark.* In the wave simulation using (2.7), the quantity  $\Delta t^2c^2$  can be pre-computed and saved in a vector. When the fourth-order central FD scheme is adopted for the Laplace operator in 3D, the matrix  $\mathcal{A}$  has 13 nonzero entries in a row. Thus a matrix-vector multiplication,  $\mathcal{A}v$ , requires 13 flops per point. It is not difficult to see that the algorithm needs 30 flops per point (in a time level) for 3D problems.

## 2.2. Two-level implicit schemes

Rewrite the system (2.1) as

$$\begin{aligned} \eta_t + \mathcal{A}v &= S, \\ \frac{1}{c^2}v_t - \eta &= 0, \end{aligned} \quad (2.8)$$

where  $\eta$  is an auxiliary variable. Then, the two-level implicit scheme can be formulated as follows [9]:

$$\begin{aligned} \text{(a)} \quad &\frac{\eta^{n+1} - \eta^n}{\Delta t} + \mathcal{A}[\alpha v^{n+1} + (1 - \alpha)v^n] = S^{n+\alpha}, \\ \text{(b)} \quad &\frac{1}{c^2}\frac{v^{n+1} - v^n}{\Delta t} - [\beta\eta^{n+1} + (1 - \beta)\eta^n] = 0, \end{aligned} \quad (2.9)$$

where  $0 \leq \alpha, \beta \leq 1$  are algorithm parameters and  $S^{n+\alpha} = \alpha S^{n+1} + (1 - \alpha)S^n$ . In the literature, the following is well known for the two-level algorithm (see e.g. [9, §9.11]):

- The algorithm (2.9) is unconditionally stable when  $\alpha, \beta \geq 0.5$ .

- It is second-order if  $(\alpha, \beta) = (0.5, 0.5)$ , for example.
- It coincides with the explicit second-order scheme (2.2) when  $(\alpha, \beta) = (0, 1)$ . Thus it works well for smooth solutions, but it can introduce severe numerical oscillations when the solution is nonsmooth.

The case  $(\alpha, \beta) = (0.5, 0.5)$  is particularly interesting, because it allows the algorithm to be both second-order accurate (in time) and unconditionally stable. For an efficient implementation, (2.9) can be reformulated as follows. Multiply (2.9.a) and (2.9.b) by  $\beta\Delta t^2$  and  $\Delta t$ , respectively, and add the resulting equations to have

$$\left(\frac{1}{c^2} + \alpha\beta\Delta t^2\mathcal{A}\right)v^{n+1} = \left(\frac{1}{c^2} - (1-\alpha)\beta\Delta t^2\mathcal{A}\right)v^n + \Delta t\eta^n + \beta\Delta t^2S^{n+\alpha}. \quad (2.10)$$

Along with (2.9.b) and  $\eta^0 = v_t^0/c^2 = g_1/c^2$ , the above equation solves the problem. When  $\alpha = \beta = 1/2$ , the resulting algorithm reads

$$\begin{aligned} \text{(a)} \quad & \left(\frac{1}{c^2} + \frac{\Delta t^2}{4}\mathcal{A}\right)v^{n+1} = \left(\frac{1}{c^2} - \frac{\Delta t^2}{4}\mathcal{A}\right)v^n + \Delta t\eta^n + \frac{\Delta t^2}{2}S^{n+1/2}, \\ \text{(b)} \quad & \eta^{n+1} = -\eta^n + \frac{2}{c^2\Delta t}(v^{n+1} - v^n). \end{aligned} \quad (2.11)$$

Note that solving (2.11.a) requires inverting a matrix, presumably large. However, since the matrix is symmetric and *strongly* diagonally dominant, most iterative algebraic solvers must converge fast unless  $\Delta t$  is selected unreasonably large.

For a purpose of comparison with the three-level algorithms to be presented in the next section, we shall reformulate (2.9) by eliminating  $\eta^n$  and  $\eta^{n+1}$ . Multiply (2.9.a) by  $(1-\beta)\Delta t$  and subtract (2.9.b) from the resulting equation to have

$$\eta^{n+1} = (1-\beta)\Delta t\left(S^{n+\alpha} - \mathcal{A}[\alpha v^{n+1} + (1-\alpha)v^n]\right) + \frac{1}{c^2}\frac{v^{n+1} - v^n}{\Delta t}. \quad (2.12)$$

Take  $\eta^n$  from (2.12) to plug in (2.10). Then the two-level algorithm (2.9) equivalently reads, for  $n \geq 1$ ,

$$\begin{aligned} \frac{1}{c^2}\bar{\partial}_{tt}v^n + \mathcal{A}\left(\alpha\beta v^{n+1} + (\alpha + \beta - 2\alpha\beta)v^n + (1-\alpha)(1-\beta)v^{n-1}\right) \\ = \beta S^{n+\alpha} + (1-\beta)S^{n-1+\alpha}. \end{aligned} \quad (2.13)$$

### 3. New Approaches

In this section, we introduce a three-level implicit method for solving the acoustic wave equation (1.1); see §3.1. For an efficient simulation, a locally one-dimensional (LOD) method is adopted for the problem in rectangular or cubic domains (§3.2). Later subsections of this section include stability analysis, fourth-order accuracy in time, an efficient treatment of ABC, and comparisons of computational complexity between the conventional explicit method and the new implicit method.

### 3.1. The three-level implicit method

We suggest a three-level implicit time-stepping algorithm for the acoustic wave equation (1.1) as follows: Given  $v^0, \dots, v^n$ ,  $n \geq 1$ , find  $v^{n+1}$  by solving

$$\frac{1}{c^2} \bar{\partial}_{tt} v^n + \mathcal{A}(\theta v^{n+1} + (1 - 2\theta)v^n + \theta v^{n-1}) = S^n + \theta \Delta t^2 \bar{\partial}_{tt} S^n, \quad (3.1)$$

where  $\theta$  is an algorithm parameter to be selected in  $[0, 0.5]$ . Note that

$$S^n + \theta \Delta t^2 \bar{\partial}_{tt} S^n = \theta S^{n+1} + (1 - 2\theta)S^n + \theta S^{n-1}.$$

One can verify the following:

- The truncation error of (3.1) is  $\mathcal{O}(\Delta t^2 + h^p)$ ,  $p \geq 2$ , for every  $\theta \in [0, 0.5]$ .
- When  $\theta = 0$ , (3.1) turns out to be the second-order explicit scheme (2.2).
- When  $\theta = 1/12$ , the truncation error of (3.1) becomes  $\mathcal{O}(\Delta t^4 + h^p)$ ; see §3.4.
- The algorithm is unconditionally stable when  $\theta \in [0.25, 0.5]$ . (See §3.3 below.)
- The three-level algorithm is equivalent to the two-level method (2.9) when  $\alpha = \beta = 0.5$  and  $\theta = 0.25$ . Also they are equivalent when  $\alpha = r_1$ ,  $\beta = r_2$ , and  $\theta = 1/12$ , where  $r_1$  and  $r_2$  are the two zeros of  $x^2 - x + 1/12 = 0$ . (The latter example is such that the algorithms are fourth-order accurate in time.)

The implicit method (3.1) requires an appropriate initialization for  $v^1$ . Recall the initial conditions given in (1.1.c) and the Taylor series expansion

$$u^1 = u^0 + \Delta t u_t^0 + \frac{\Delta t^2}{2} u_{tt}^0 + \frac{\Delta t^3}{3!} u_{ttt}^0 + \frac{\Delta t^4}{4!} u_{tttt}^0 + \mathcal{O}(\Delta t^5). \quad (3.2)$$

Consider the equalities

$$\begin{aligned} u_t^0 &= g_1, \\ u_{tt}^0 &= c^2(S^0 - \mathcal{A}g_0), \\ u_{ttt}^0 &= c^2(S_t^0 - \mathcal{A}g_1), \\ u_{tttt}^0 &= c^2[S_{tt}^0 - \mathcal{A}(c^2(S^0 - \mathcal{A}g_0))], \end{aligned} \quad (3.3)$$

and approximations

$$\begin{aligned} S_t^0 &\approx \frac{-3S^0 + 4S^1 - S^2}{2\Delta t} + \mathcal{O}(\Delta t^2), \\ S_{tt}^0 &\approx \frac{S^0 - 2S^1 + S^2}{\Delta t^2} + \mathcal{O}(\Delta t). \end{aligned} \quad (3.4)$$

Then, it follows from (3.2)-(3.4) that

$$\begin{aligned}
\text{(a)} \quad v^1 &\approx g_0 + \Delta t g_1 + \frac{\Delta t^2 c^2}{2} (S^0 - \mathcal{A}g_0) + \mathcal{O}(\Delta t^3), \\
\text{(b)} \quad v^1 &\approx g_0 + \Delta t g_1 + \frac{\Delta t^2 c^2}{2} \left[ \frac{7S^0 + 6S^1 - S^2}{12} \right. \\
&\quad \left. - \mathcal{A} \left( g_0 + \frac{\Delta t}{3} g_1 + \frac{\Delta t^2 c^2}{12} (S^0 - \mathcal{A}g_0) \right) \right] + \mathcal{O}(\Delta t^5).
\end{aligned} \tag{3.5}$$

The initial values in (3.5.a) and (3.5.b) can be adopted respectively for the second- and fourth-order methods in time.

### 3.2. The LOD procedure

In many applications including Geophysical ones, the domain is often rectangular or cubic. To solve the implicit algorithm (3.1) efficiently in these regular domains, we can adopt a locally one-dimensional (LOD) method, in particular, the alternating direction implicit (ADI) method [6, 7, 8, 12]. We will formulate the LOD procedure for 3D problems. Decompose  $\mathcal{A}$  into the three directional operators  $\mathcal{A}_\ell$ ,  $\ell = 1, 2, 3$ , i.e.,

$$\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3,$$

where  $\mathcal{A}_\ell$  is the  $p$ th-order FD approximation of  $-\partial_{x_\ell x_\ell}$ . Then, as a perturbation of (3.1), an LOD time-stepping procedure can be constructed as follows. Given  $w^0, \dots, w^n$ , first approximate the solution at  $t^{n+1}$  by the explicit procedure:

$$\frac{1}{c^2} \frac{w^{n+1,0} - 2w^n + w^{n-1}}{\Delta t^2} + \mathcal{A}w^n = S^n + \theta \Delta t^2 \bar{\partial}_{tt} S^n, \tag{3.6}$$

and then apply the implicit directional sweeps

$$\begin{aligned}
\frac{1}{c^2} \frac{w^{n+1,1} - w^{n+1,0}}{\Delta t^2} + \theta \mathcal{A}_1 (w^{n+1,1} - 2w^n + w^{n-1}) &= 0, \quad (x\text{-sweep}) \\
\frac{1}{c^2} \frac{w^{n+1,2} - w^{n+1,1}}{\Delta t^2} + \theta \mathcal{A}_2 (w^{n+1,2} - 2w^n + w^{n-1}) &= 0, \quad (y\text{-sweep}) \\
\frac{1}{c^2} \frac{w^{n+1,3} - w^{n+1,2}}{\Delta t^2} + \theta \mathcal{A}_3 (w^{n+1,3} - 2w^n + w^{n-1}) &= 0. \quad (z\text{-sweep})
\end{aligned} \tag{3.7}$$

The LOD algorithm, (3.6)-(3.7), can be implemented as follows:

$$\begin{aligned}
\tilde{w}^n &= 2w^n - w^{n-1}, \\
w^{n+1,0} &= \tilde{w}^n + \Delta t^2 c^2 (S^n + \theta \Delta t^2 \bar{\partial}_{tt} S^n - \mathcal{A}w^n), \\
(I + \theta \Delta t^2 c^2 \mathcal{A}_1) w^{n+1,1} &= w^{n+1,0} + \theta \Delta t^2 c^2 \mathcal{A}_1 \tilde{w}^n, \\
(I + \theta \Delta t^2 c^2 \mathcal{A}_2) w^{n+1,2} &= w^{n+1,1} + \theta \Delta t^2 c^2 \mathcal{A}_2 \tilde{w}^n, \\
(I + \theta \Delta t^2 c^2 \mathcal{A}_3) w^{n+1,3} &= w^{n+1,2} + \theta \Delta t^2 c^2 \mathcal{A}_3 \tilde{w}^n.
\end{aligned} \tag{3.8}$$

To find the error involved by the LOD-perturbation, we will eliminate the intermediate values in (3.6)-(3.7). Adding the four equations in (3.6)-(3.7), followed by some algebra, reads

$$\begin{aligned} \frac{1}{c^2} \bar{\partial}_{tt} w^n + \mathcal{A}(\theta w^{n+1} + (1 - 2\theta)w^n + \theta w^{n-1}) \\ + \mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1}) = S^n + \theta \Delta t^2 \bar{\partial}_{tt} S^n, \end{aligned} \quad (3.9)$$

where

$$\mathcal{B}_\theta = \theta^2 \Delta t^2 c^2 (\mathcal{A}_1 \mathcal{A}_2 + \mathcal{A}_1 \mathcal{A}_3 + \mathcal{A}_2 \mathcal{A}_3) + \theta^3 (\Delta t^2 c^2)^2 \mathcal{A}_1 \mathcal{A}_2 \mathcal{A}_3.$$

Compared with (3.1), the LOD algorithm (3.6)-(3.7) incorporates an extra term  $\mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1})$ , which results from the LOD operator splitting and is called the *splitting error*. Since  $(w^{n+1} - 2w^n + w^{n-1}) = \mathcal{O}(\Delta t^2)$  for sufficiently smooth solutions, the splitting error turns out to be fourth-order in time, i.e.,

$$\mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1}) = \mathcal{O}(\Delta t^4). \quad (3.10)$$

Thus the LOD algorithm (3.6)-(3.7) solves the three-level implicit difference equation (3.1) accurately, with an extra error (splitting error) in  $\mathcal{O}(\Delta t^4)$ .

For 2D problems, the  $z$ -sweep in (3.7) must be eliminated and  $w^{n+1,2}$  becomes the solution in the new time level, i.e.,  $w^{n+1} = w^{n+1,2}$ . The splitting error for 2D problems turns out to be  $\mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1})$ , where

$$\mathcal{B}_\theta = \theta^2 \Delta t^2 c^2 \mathcal{A}_1 \mathcal{A}_2. \quad (2D)$$

Thus the splitting error is again  $\mathcal{O}(\Delta t^4)$  for sufficiently smooth solutions.

As investigated in [7], the splitting error of the LOD algorithm can be much larger than the truncation error (although they show the same order in  $\Delta t$ ) unless the time step size is sufficiently small. However, one can *virtually* eliminate the splitting error as follows. If we could add  $\mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1})$  to the right hand side of (3.6), then we could cancel the perturbation term in (3.9) completely; but since we do not know  $w^{n+1}$  at the moment of solving (3.6), we cannot make this modification in the algorithm. Our best estimate for  $\mathcal{B}_\theta(w^{n+1} - 2w^n + w^{n-1})$  is  $\mathcal{B}_\theta(w^n - 2w^{n-1} + w^{n-2})$ , which we can add to the right side of (3.6). Then, the overall splitting error for this modified LOD algorithm becomes

$$\mathcal{B}_\theta(w^{n+1} - 3w^n + 3w^{n-1} - w^{n-2}),$$

which is  $\mathcal{O}(\Delta t^5)$  for sufficiently smooth solutions. We will not try to address further details in that direction; see [7] for the *virtual elimination* of the splitting error for the numerical solution of the heat equation.

A few more comments are in order for the LOD algorithm (3.6)-(3.7).

- Solving the directional sweeps requires to invert a series of tri- or quanta-diagonal matrices, which is inexpensive.
- The LOD algorithm is unconditionally stable when  $\theta \in [0.25, 0.5]$ ; see §3.3.
- Higher-order spatial difference schemes can be incorporated by simply replacing the LOD operators  $\mathcal{A}_\ell$ ,  $\ell = 1, 2, 3$ , by higher-order schemes.

### 3.3. Stability analysis

In this subsection, we present a stability analysis for the three-level implicit algorithm (3.1) and its LOD procedure (3.6)-(3.7).

**Theorem 3.1.** *Let  $\theta \in [0.25, 0.5]$ . Then (3.1) and its LOD procedure (3.6)-(3.7) are unconditionally stable.*

**Proof.** It suffices to show the stability of the LOD procedure (3.9); stability of (3.1) follows by replacing  $w^n$  by  $v^n$  and setting  $\mathcal{B}_\theta = 0$ .

Let  $\|\cdot\|$  denote the  $L^2(\Omega)$  or  $\ell^2$  norm, as appropriate. (That is, depending on the spatial discretization by finite elements or finite differences.) Define difference operators

$$\partial_t^- w^n := \frac{w^n - w^{n-1}}{\Delta t}, \quad \partial_t^+ w^n := \frac{w^{n+1} - w^n}{\Delta t}, \quad \bar{\partial}_t := (\partial_t^+ + \partial_t^-)/2.$$

Then

$$\bar{\partial}_{tt} = (\partial_t^+ - \partial_t^-)/\Delta t.$$

Choose  $\bar{\partial}_t w^n$  as a test function for (3.9). Then, for  $n \geq 1$ ,

$$\begin{aligned} & \left( \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right) \bar{\partial}_{tt} w^n, \bar{\partial}_t w^n \right) \\ & + \left( \mathcal{A}[\theta w^{n+1} + (1 - 2\theta)w^n + \theta w^{n-1}], \bar{\partial}_t w^n \right) = 0, \end{aligned} \quad (3.11)$$

where we have set  $S \equiv 0$ .

Multiply (3.11) by  $\Delta t$  and sum beginning at  $n = 1$  to have

$$\begin{aligned} & \sum_{j=1}^n \left( \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right) \bar{\partial}_{tt} w^j, \bar{\partial}_t w^j \right) \Delta t \\ & + \sum_{j=1}^n \left( \mathcal{A}[\theta w^{j+1} + (1 - 2\theta)w^j + \theta w^{j-1}], \bar{\partial}_t w^j \right) \Delta t = 0. \end{aligned} \quad (3.12)$$

Consider the following identities

$$\begin{aligned}
& \left( \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right) \bar{\partial}_{tt} w^j, \bar{\partial}_t w^j \right) \Delta t \\
&= \frac{1}{2} \left( \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right) (\partial_t^+ - \partial_t^-) w^j, (\partial_t^+ + \partial_t^-) w^j \right) \\
&= \frac{1}{2} \left\| \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right)^{1/2} \partial_t^+ w^j \right\|^2 - \frac{1}{2} \left\| \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right)^{1/2} \partial_t^- w^j \right\|^2, \\
& \left( \mathcal{A}[\theta w^{j+1} + (1 - 2\theta)w^j + \theta w^{j-1}], \bar{\partial}_t w^j \right) \Delta t \\
&= \frac{1}{2} \left( \mathcal{A}[\theta(w^{j+1} + w^{j-1}) + (1 - 2\theta)w^j], w^{j+1} - w^{j-1} \right) \\
&= \frac{1}{2} \mathcal{P}^+(w^j, \theta) - \frac{1}{2} \mathcal{P}^-(w^j, \theta),
\end{aligned} \tag{3.13}$$

where

$$\mathcal{P}^\pm(w^j, \theta) := \theta(\mathcal{A}w^j, w^j) + \theta(\mathcal{A}w^{j\pm 1}, w^{j\pm 1}) + (1 - 2\theta)(\mathcal{A}w^j, w^{j\pm 1}).$$

Since

$$\partial_t^- w^j = \partial_t^+ w^{j-1}, \quad \mathcal{P}^-(w^j, \theta) = \mathcal{P}^+(w^{j-1}, \theta),$$

it follows from (3.12) and (3.13) that

$$\begin{aligned}
& \left\| \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right)^{1/2} \partial_t^+ w^n \right\|^2 + \mathcal{P}^+(w^n, \theta) \\
& \leq \left\| \left( \frac{1}{c^2} + \Delta t^2 \mathcal{B}_\theta \right)^{1/2} \partial_t^+ w^0 \right\|^2 + \mathcal{P}^+(w^0, \theta).
\end{aligned} \tag{3.14}$$

Note that  $\mathcal{P}^\pm(w^j, \theta) \geq 0$ , for  $\theta \in [0.25, 0.5]$ , which completes the proof.  $\square$

### 3.4. Fourth-order accuracy in time ( $\theta = 1/12$ )

When  $\theta = 1/12$ , the algorithms (3.1) and (3.6)-(3.7) become fourth-order in time. To see this, recall the Taylor expansion for  $v_{tt}(t^n)$  in (2.3). Utilize the identity

$$v_{ttt}(t^n) = c^2(S_{tt}^n - \mathcal{A}v_{tt}^n) \tag{3.15}$$

to rewrite (2.3) as

$$\begin{aligned}
v_{tt}(t^n) &\approx \bar{\partial}_{tt} v^n - \frac{\Delta t^2}{12} c^2 (S_{tt}^n - \mathcal{A}v_{tt}^n) + \mathcal{O}(\Delta t^4) \\
&\approx \bar{\partial}_{tt} v^n - \frac{\Delta t^2}{12} c^2 \bar{\partial}_{tt} S^n + \frac{c^2}{12} \mathcal{A}(v^{n+1} - 2v^n + v^{n-1}) + \mathcal{O}(\Delta t^4),
\end{aligned} \tag{3.16}$$

where the central second-order approximations are applied for  $S_{tt}^n$  and  $v_{tt}^n$ . Thus a fourth-order time-stepping algorithm can be formulated as

$$\frac{1}{c^2} \bar{\partial}_{tt} v^n + \frac{1}{12} \mathcal{A}(v^{n+1} - 2v^n + v^{n-1}) + \mathcal{A}v^n = S^n + \frac{\Delta t^2}{12} \bar{\partial}_{tt} S^n, \tag{3.17}$$

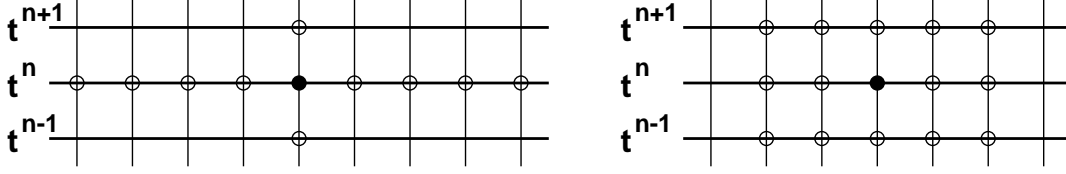


Figure 1: The FD stencils, depicted in one spatial variable, for the fourth-order explicit scheme (left) and the fourth-order implicit scheme (right).

which is identical to (3.1) when  $\theta = 1/12$ . The LOD variant of (3.17) clearly reads

$$\begin{aligned} \frac{1}{c^2} \bar{\partial}_{tt} v^n + \frac{1}{12} \mathcal{A}(w^{n+1} - 2w^n + w^{n-1}) + \mathcal{A}w^n \\ + \mathcal{B}_{1/12}(w^{n+1} - 2w^n + w^{n-1}) = S^n + \frac{\Delta t^2}{12} \bar{\partial}_{tt} S^n, \end{aligned} \quad (3.18)$$

which is equivalent to (3.6)-(3.7) when  $\theta = 1/12$ .  $\square$

*Remark.* The conventional fourth-order explicit scheme utilizes the identity (2.4) for the approximation of  $v_{tttt}$ , while the new implicit method employs (3.15). As a result, the implicit method adopts a more compact set of grid points in the FD approximation. See Figure 1, where the FD stencils are depicted for the fourth-order explicit scheme (left) and the fourth-order implicit scheme (right), in one spatial variable. The fourth-order explicit scheme utilizes values at 33 and 73 grid points in the  $n$ -th time level for 2D and 3D problems, respectively.

### 3.5. The absorbing boundary condition

The ABC in its second-order approximation can be formulated as

$$\frac{1}{c} \frac{w^{n+1} - w^{n-1}}{2\Delta t} + (\nabla_h w^n) \cdot \nu = 0, \quad (3.19)$$

where  $\nabla_h$  denotes the second-order FD approximation of the gradient  $\nabla$ . It should be noticed that the ABC (3.19) involves no spatial approximations on the  $(n+1)$ -th time level. Thus, the LOD algorithm can incorporate the ABC in its explicit step *only*, and the implicit directional sweeps are carried out with a Dirichlet boundary condition. This strategy clearly imposes the second-order approximation of the ABC on all time levels. For higher-order approximations, see discussions in Section 5 below.

### 3.6. Computational complexity

One of main goals in this article is to compare accuracy and efficiency of the fourth-order algorithms, (2.7) and (3.18), with  $\mathcal{A}_\ell$  being the fourth-order FD approximations of  $-\partial^2/\partial x_\ell^2$ .

In the implementation of the LOD algorithm (3.8) for 3D problems, one can pre-compute  $(I + \theta\Delta t^2 c^2 \mathcal{A}_\ell)$ ,  $\ell = 1, 2, 3$ , and save their  $LU$ -factorizations. When the central fourth-order scheme is used for the spatial derivatives, the explicit step in (3.8) needs 14 flops per point and each directional step can be carried out for 11 flops per point (6 for the right side and 5 for the inversion of the  $L$  and  $U$  matrices). Thus the total number of required operations is 48 per point, which is only 60% more than that of the fourth-order explicit method (2.7).

For 2D problems, the matrix  $\mathcal{A}$  contains 9 nonzero entries in a row for the central fourth-order scheme in space. Thus the required flops per point for (2.7) and (3.6)-(3.7) are 22 and 33, respectively. The LOD (implicit) method is only 50% more expensive than the explicit method per point.

Since the LOD algorithm shows implicit features, hopefully it can be either more flexible in choosing the time step size or less dispersive than the explicit method. (Note that when  $\theta \in [0.25, 0.5]$ , the LOD algorithm is unconditionally stable, although it turns out to be second-order accurate in time.) Employing various numerical examples, we will investigate if the implicit method allows a larger time step size for a stable solution. Also we will numerically verify and compare nonphysical oscillations introduced by these two fourth-order schemes for the acoustic wave propagation in realistic media (dispersion analysis).

## 4. Numerical Experiments

The fourth-order explicit method (2.7) and the LOD algorithm (3.8) are implemented for the acoustic wave equation in two space variables. For the spatial derivatives, the second-order and fourth-order FD schemes are adopted for both algorithms. The core computational routines are written in F77, with the main and input/output routines in C++ for a dynamic memory allocation and a flexible data management. The computation is carried out on a 400MHz Laptop having a Linux operating system. In the following, CPU denotes the user time on the machine measured in seconds.

Figure 2 presents a vertical section of a real velocity in the Gulf of Mexico (a), provided from Shell Offshore Inc., and the snapshots of the numerical solution at  $t = 2.2$  for the fourth-order explicit method (b), LOD with  $\theta = 0.25$  (c), and the fourth-order LOD (d). The fourth-order central FD scheme is applied for the spatial derivatives for all cases. For the point source, a Ricker wavelet of 10Hz ( $\nu = 10$ ) is located at the center of the top edge ( $\mathbf{x}_s = (4.57, 0)$ ). Since the velocity  $c(\mathbf{x}) \in [1.50, 4.42]$  (Km/sec), the wavelength ( $:= c/\nu$ ) varies between 150 and 442 meters. The velocity model contains  $240 \times 160$  cells of the edge length 38.1 meters ( $h = 38.1$ ). Thus the *grid frequency*  $G_f$  (the number of grid points per wavelength) becomes

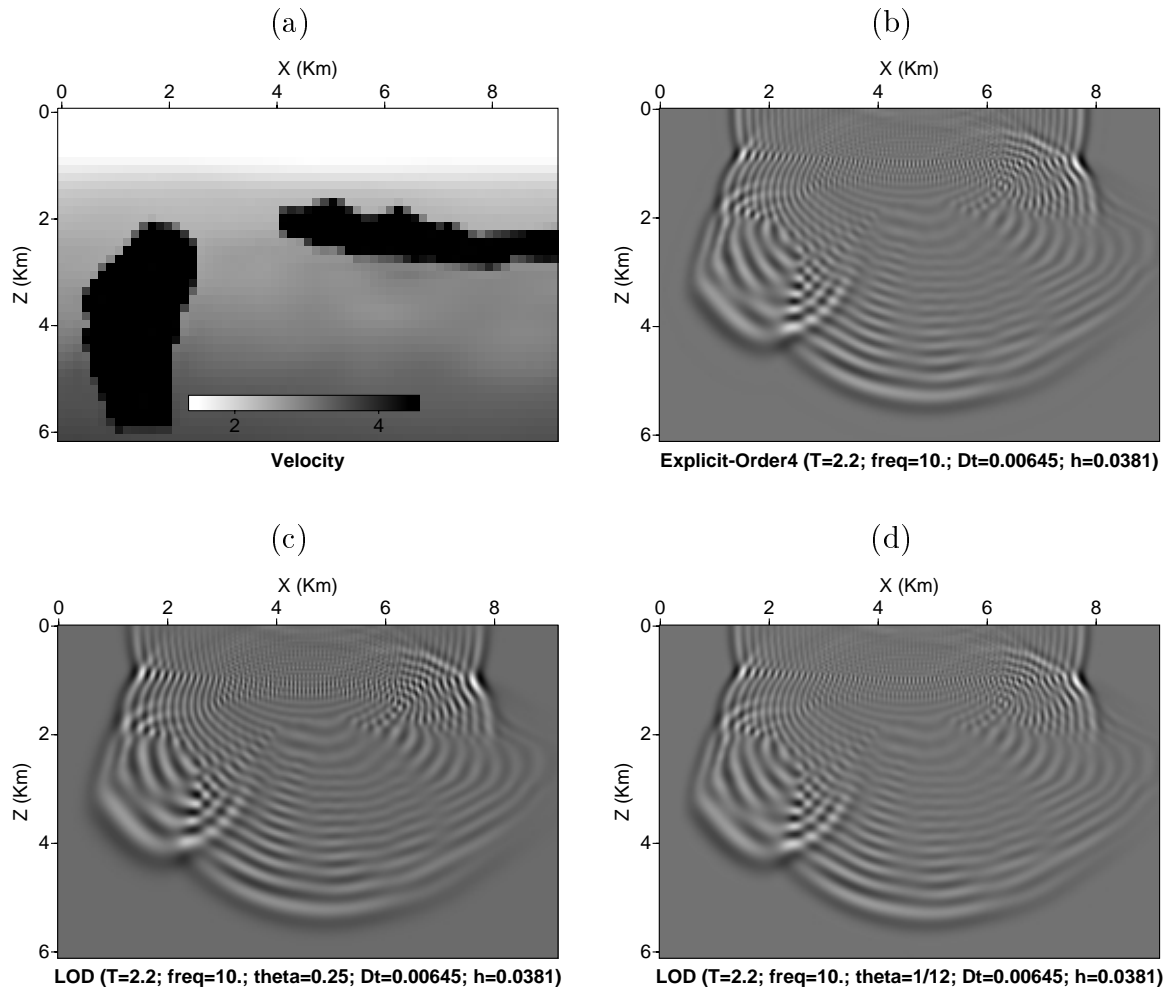


Figure 2: The velocity (a) and the snapshots at  $t = 2.2$  for the fourth-order explicit method (b), LOD with  $\theta = 0.25$  (c), and the fourth-order LOD ( $\theta = 1/12$ ) (d). The fourth-order central FD scheme is applied for the spatial derivatives for all cases.

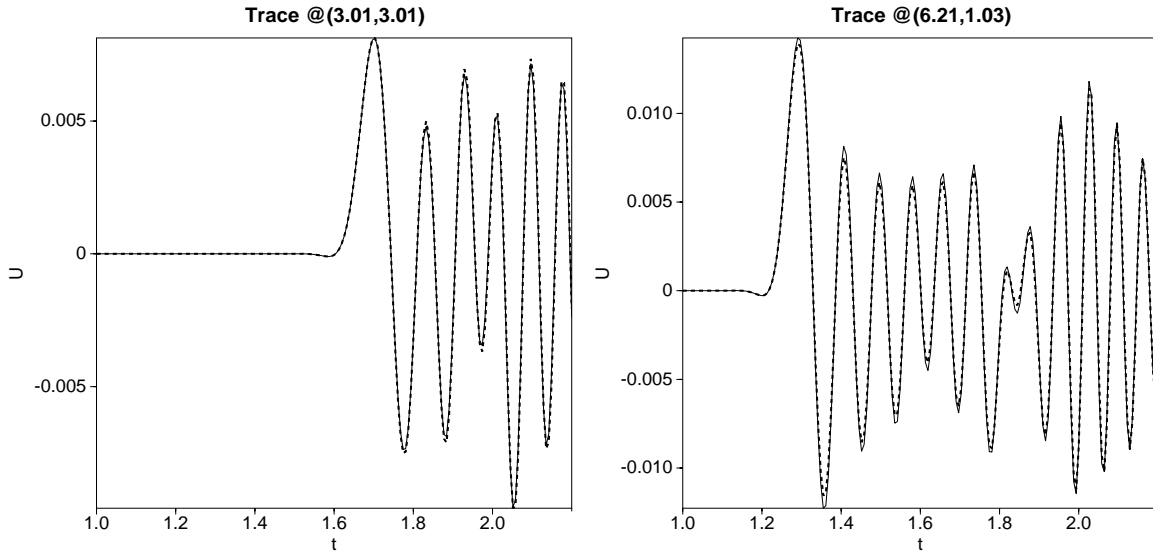


Figure 3: The traces seen at (3.01,3.01) (left) and (6.21,1.03) (right). The solid and dashed curves correspond to  $\text{LOD}(\theta = 1/12)$  and the fourth-order explicit methods, respectively.

3.94  $\sim$  11.60. The time step size  $\Delta t$  is selected for the Courant number  $\sigma$  near to 0.75 such that 2.2 is an integer multiple of  $\Delta t$ , i.e.,

$$\sigma := \frac{\Delta t \|c\|_{\infty}}{h} \approx 0.75,$$

where  $\|c\|_{\infty}$  denotes the maximum of the velocity  $c$ . ( $2.2 = 341\Delta t$ ) The following has been observed

| Methods                 | CPU  | $\min_{\mathbf{x}} u(\mathbf{x}, 2.2)$ | $\max_{\mathbf{x}} u(\mathbf{x}, 2.2)$ |
|-------------------------|------|--|--|
| Explicit-Order4         | 28.4 | -0.0250                                | 0.0204                                 |
| LOD ( $\theta = 0.25$ ) | 39.3 | -0.0233                                | 0.0169                                 |
| LOD ( $\theta = 1/12$ ) | 39.4 | -0.0255                                | 0.0208                                 |

The min-max of the computed solution is not a standard indicator in measuring the numerical error. However, it is often smaller in modulus for more dissipative methods; the second-order scheme seems to be dissipative quite a bit. As one can see, the solution in Figure 2(c) is smoother than other solutions. The solutions from the fourth-order methods, Figures 2(b) and 2(d), hardly differ from each other in their maximum and minimum. The solution from LOD ( $\theta = 1/12$ ) seems slightly smoother than that of the fourth-order explicit method.

To see the differences in detail, the traces are observed and compared at a few points. Figure 3 contains the traces recorded at (3.01,3.01) (left) and (6.21,1.03)

(right), where the waveform is expected to oscillate a lot due to sudden changes in velocity and therefore strong reflections. The solid and dashed curves correspond to LOD( $\theta = 1/12$ ) and the fourth-order explicit methods, respectively. As one can see from the figure, the solutions obtained from the two fourth-order methods match to each other quite well. It has been observed from various experiments that

- The implicit method shows a similar stability property as the explicit scheme. That is, instability has been observed for a similarly large  $\Delta t$  for both methods. Thus the implicit method may not gain efficiency by choosing a larger time step size than the explicit method.
- The implicit method takes about 40% more computation time than the explicit method for 2D problems of the same size. Note that the implicit method requires 50% more flops; see §3.6.
- The fourth-order LOD method is slightly less dispersive; it produces a less oscillatory solution than the fourth-order explicit scheme.
- Second-order methods produce more dissipative solutions than fourth-order methods. Thus second-order methods are less attractive; a sharp resolution of wavefronts is often very important in wave simulation.

## 5. Discussion on Accuracy of the ABC

In this section, we discuss some issues in fourth-order approximations of the ABC; see [10] for details.

Recall the ABC in (1.1.b). For the time-derivative  $u_t$ , the fourth-order approximation begins with

$$\begin{aligned}
 u_t(t^n) &\approx \bar{\partial}_t u^n - \frac{\Delta t^2}{6} u_{ttt}(t^n) + \mathcal{O}(\Delta t^4) \\
 &= \bar{\partial}_t u^n - \frac{\Delta t^2 c^2}{6} (-\mathcal{A}u + S)_t(t^n) + \mathcal{O}(\Delta t^4) \\
 &\approx \bar{\partial}_t u^n + \frac{\Delta t^2 c^2}{6} \mathcal{A}u_t(t^n) - \frac{\Delta t^2 c^2}{6} \bar{\partial}_t S^n + \mathcal{O}(\Delta t^4),
 \end{aligned} \tag{5.1}$$

where we have utilized the identity (2.1). For a fourth-order approximation, the term  $\mathcal{A}u_t(t^n)$  must be approximated by a second-order approximation. One can utilize either of the following:

$$\begin{aligned}
 \mathcal{A}u_t(t^n) &\approx \mathcal{A}\bar{\partial}_t u^n + \mathcal{O}(\Delta t^2), \\
 \mathcal{A}u_t(t^n) &\approx \mathcal{A} \frac{3u^n - 4u^{n-1} + u^{n-2}}{2\Delta t} + \mathcal{O}(\Delta t^2).
 \end{aligned} \tag{5.2}$$

It follows from (5.1) and (5.2) that fourth-order approximations of the ABC are

$$u_t(t^n) \approx \begin{cases} \bar{\partial}_t \left( 1 + \frac{\Delta t^2 c^2}{6} \mathcal{A} \right) u^n - \frac{\Delta t^2 c^2}{6} \bar{\partial}_t S^n + \mathcal{O}(\Delta t^4), \\ \bar{\partial}_t u^n + \frac{\Delta t^2 c^2}{6} \mathcal{A} \frac{3u^n - 4u^{n-1} + u^{n-2}}{2\Delta t} - \frac{\Delta t^2 c^2}{6} \bar{\partial}_t S^n + \mathcal{O}(\Delta t^4). \end{cases} \quad (5.3)$$

The normal derivative  $u_\nu$  can be naturally incorporated in FE methods. For FD methods, we need to find a detailed expression for its fourth-order approximation: either the one-sided approximation or the one as for  $u_t$ . However, it does not seem possible to incorporate the overall fourth-order approximation of the ABC for an *outer-bordering* with the difference equations in (2.7) and (3.8). Thus, FE methods are preferred for high-order approximations of the solution up to the boundary. The mass matrix can be simplified, without a loss of accuracy, by adopting the tensorization of the Gauss-Lobatto points for the nodal points and the corresponding mass-lumping quadratures [5]. In particular, for implicit schemes, the mass matrix can be expressed as a sum of sub-matrices of directional derivatives, for which one can adopt an efficient LOD procedure [10].

## 6. Conclusions

We have introduced a new implicit fourth-order FD scheme for the numerical solution of the acoustic wave equation. For an efficient simulation, a locally one-dimensional (LOD) procedure, having the splitting error not larger than the truncation error, has been adopted. The new implicit method is compared with the standard explicit method in stability, accuracy, and efficiency:

- The implicit method shows a similar stability condition as the explicit scheme.
- The implicit method turns out to be 40% more expensive than the explicit method for 2D problems of the same size.
- The implicit method is slightly less dispersive. Due to this property, the implicit scheme can show advantages over the explicit scheme for the waveform simulation in very oscillatory media.

Finally, we have discussed how to incorporate the fourth-order accuracy for the ABC.

## Acknowledgment

The computational code utilized in this article has been implemented for the modelcode library at the University of Kentucky: *Graduate Research and Applica-*

tions for *Differential Equations* (GRADE), and is available through internet access to [www.ms.uky.edu/~skim/GRADE/](http://www.ms.uky.edu/~skim/GRADE/) or by asking the author ([skim@ms.uky.edu](mailto:skim@ms.uky.edu)).

## References

- [1] W. AMES AND D. LEE, *Current development in the numerical treatment of ocean acoustic propagation*, Appl. Numer. Math., 3 (1987), pp. 25–47.
- [2] S. CANDEL, *A review of numerical methods in acoustic wave propagation*, in Recent Advances in Aeroacoustics, Springer, New York, 1986, pp. 339–410.
- [3] R. CLAYTON AND B. ENGQUIST, *Absorbing boundary conditions for acoustic and elastic wave calculations*, Bull. Seismol. Soc. Amer., 67 (1977), pp. 1529–1540.
- [4] G. COHEN AND P. JOLY, *Fourth order schemes for the heterogeneous acoustics equation*, Comp. Meth. in Appl. Mech. and Engin., 80 (1990), pp. 397–407.
- [5] G. COHEN, P. JOLY, AND N. TORDJMAN, *Construction and analysis of higher order finite elements with mass lumping for the wave equation*, in Second International Conference on Mathematical and Numerical Aspects of Wave Propagation, R. Kleinman, T. Angell, D. Colton, F. Santosa, and I. Stakgold, eds., SIAM, Philadelphia, 1993, pp. 152–160.
- [6] J. DOUGLAS, JR., *On the numerical integration of  $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$  by implicit methods*, J. Soc. Indust. Appl. Math., 3 (1955), pp. 42–65.
- [7] J. DOUGLAS, JR. AND S. KIM, *Improved accuracy for locally one-dimensional methods for parabolic equations*, Mathematical Models and Methods in Applied Sciences, 11 (2001), pp. 1563–1579.
- [8] J. DOUGLAS, JR. AND D. PEACEMAN, *Numerical solution of two-dimensional heat flow problems*, American Institute of Chemical Engineering Journal, 1 (1955), pp. 505–512.
- [9] C. JOHNSON, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, New York, New Rochelle, Melbourne, Sydney, 1987.
- [10] S. KIM, *Implicit high-order finite element solutions of the acoustic wave equation*. (in preparation).

- [11] G. KONYUKH, Y. KRIVTSOV, AND B. MIKHAILENKO, *Numerical-analytical algorithm of seismic wave propagation in inhomogeneous media*, Appl. Math. Lett., 11 (1998), pp. 99–104.
- [12] D. PEACEMAN AND H. RACHFORD, *The numerical solution of parabolic and elliptic equations*, J. Soc. Indust. Appl. Math., 3 (1955), pp. 28–41.
- [13] A. SEI AND W. SYMES, *Dispersion analysis of numerical wave propagation and its computational consequences*, J. Sci. Comput., 10 (1995), pp. 1–27.
- [14] G. SHUBIN AND J. BELL, *A modified equation approach to construction fourth order methods for acoustic wave equations*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 135–151.