Available online at www.sciencedirect.com
ScienceDirect
Applied Numerical Mathematics 57 (2007) 194–212
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

# Numerical methods for viscous and nonviscous wave equations

Hyeona Lim [a,*], Seongjai Kim [a,1], Jim Douglas Jr [b]

[a] *Department of Mathematics and Statistics, Mississippi State University, State, Mississippi, MS 39762-5921, USA*
[b] *Department of Mathematics, Purdue University, W. Lafayette, IN 47097-1395, USA*

Available online 22 March 2006

**Abstract**

This article is concerned with accurate and efficient numerical methods for solving viscous and nonviscous wave equations. A three-level second-order implicit algorithm is considered without introducing auxiliary variables. As a perturbation of the algorithm, a locally one-dimensional (*LOD*) procedure which has a splitting error not larger than the truncation error is suggested to solve problems of diagonal diffusion tensors in cubic domains efficiently. Both the three-level algorithm and its *LOD* procedure are proved to be unconditionally stable. An error analysis is provided for the numerical solution of viscous waves. Numerical results are presented to show the accuracy and efficiency of the new algorithms for the propagation of acoustic waves and of microscale heat transfer.


*MSC:* 65M06; 65M12; 65M15

*Keywords:* Viscous and nonviscous waves; Microscale heat transfer; Absorbing boundary condition; Locally one-dimensional method

## 1. Introduction

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

$$\begin{aligned}
&\text{(a)} \quad \gamma_1 \frac{\partial u}{\partial t} + \gamma_2 \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (A \nabla u) - \frac{\partial}{\partial t} \nabla \cdot (Q \nabla u) = S, \quad (\mathbf{x}, t) \in \Omega \times J, \\
&\text{(b)} \quad u = 0, \quad (\mathbf{x}, t) \in \Gamma \times J, \\
&\text{(c)} \quad u(\mathbf{x}, 0) = g_0(\mathbf{x}), \quad \frac{\partial u}{\partial t}(\mathbf{x}, 0) = g_1(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad t = 0,
\end{aligned} \tag{1}$$

where $\gamma_k = \gamma_k(\mathbf{x})$, $k = 1, 2$, are nonnegative coefficients, $S$ denotes the source, and $A = A(\mathbf{x})$ and $Q = Q(\mathbf{x})$ are symmetric, nonnegative diffusion tensors:

$$\begin{aligned}
&A = A^{\mathrm{T}}, \quad \xi \cdot A\xi \geqslant 0, \\
&Q = Q^{\mathrm{T}}, \quad \xi \cdot Q\xi \geqslant 0,
\end{aligned} \quad \forall \mathbf{x} \in \Omega, \quad \forall \xi \in \mathbb{R}^m. \tag{2}$$

* Corresponding author.
*E-mail addresses:* hlim@math.msstate.edu (H. Lim), skim@math.msstate.edu (S. Kim), douglas@math.purdue.edu (J. Douglas).
[1] The work of this author is supported in part by NSF grants DMS-0107210 and ITR-0312223.

0168-9274/$30.00 © 2006 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2006.02.004

Here, we have selected the homogeneous Dirichlet boundary condition for simplicity. It can be adjusted to accommodate the physical problem under consideration; see Section 2.

Eq. (1) governs various physical phenomena by properly choosing the coefficients. For example, when $\gamma_2 = 0$ and $Q = 0$, the equation represents standard heat transfer. Since there are many efficient algorithms known for the heat equation, we exclude its consideration. That is, we choose

$$\gamma_2(\mathbf{x}) > 0, \quad \forall \mathbf{x} \in \Omega. \tag{3}$$

Other important problems of the form (1) are given by the propagation of acoustic waves and microscale heat transfer; see Section 2 for details.

Some conventional numerical approaches for solving wave equations introduce auxiliary variables to rewrite the equations as first order hyperbolic systems; however, these approaches introduce new unknowns which results in an increase in the number of variables in the discrete problems. Thus, there are advantages in keeping the formulation (1) involving the second time-derivative and a scalar unknown. It has been the case that it is hard to construct methods combining good stability with high accuracy with this formulation [21]. The main goal of the paper is to introduce and analyze *stable* three-level numerical methods for solving (1) in the $m$-dimensional ($m$-D) space, $1 \leqslant m \leqslant 3$, which are second-order in time and incorporate no auxiliary variable. In order to solve problems with diagonal diffusion tensors in cubic domains $[0, 1]^m$ with a great efficiency, a locally one-dimensional (*LOD*) method is introduced and studied.

An outline of the paper is as follows. In Section 2, the equations describing acoustic waves and microscale heat transfer are briefly reviewed. Two-level methods that introduce auxiliary variables also are reviewed in the same section. Then, Section 3 presents a three-level implicit time-stepping algorithm. The three-level and two-level algorithms are compared and found to be equivalent for a set of selected parameters. In Section 4, an *LOD* procedure is presented as an efficient perturbation of the implicit algorithm. In Section 5, for problems in cubic domains we demonstrate stability for both the three-level implicit algorithm and its *LOD* procedure. In Section 6, an error analysis is added for viscous waves; i.e., either $\gamma_1 > 0$ or $Q$ is positive definite, or both. In Section 7, a second-order initialization scheme for $u^1$, the solution at the first time level is discussed and implementation details for a Clayton–Engquist-type absorbing boundary condition are presented for the simulation of acoustic waves. Then, we consider an alternative strategy for the computation of $u^1$, in particular when $\Delta t$ is not sufficiently small or $u^0$ is discontinuous. In Section 8, we present numerical results showing the accuracy and efficiency of our new algorithm in comparison with conventional methods; Section 9 includes conclusions.

## 2. Preliminaries

### 2.1. Acoustic waves

An important example of a nonviscous physical problem is given by the propagation of acoustic waves ($\gamma_1 = 0$, $\gamma_2 = 1/c^2$, $A = I$, the identity matrix, and $Q = 0$):

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \Delta u = S, \quad (\mathbf{x}, t) \in \Omega \times J, \tag{4}$$

where $c = c(\mathbf{x}) > 0$ denotes the normal velocity of a wavefront. (Here, the material density is assumed to be constant.) 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 domain, and then solving the resulting problem using discretization methods (e.g., finite differences, finite elements, and spectral methods). The classical *ABC*, which has been used for decades, is given by

$$\frac{1}{c}\frac{\partial u}{\partial t} + u_\nu = 0, \quad (\mathbf{x}, t) \in \Gamma \times J, \tag{5}$$

where $\nu$ denotes the unit outer normal from $\Gamma$. Plane waves arriving normally to the boundary are passed through without reflection by (5). See [6].

Eq. (4) has been studied as a model problem for second-order hyperbolic problems by many authors. See, e.g., [2,4,7,25,30]; also see [20] for unconditionally stable explicit schemes for the Maxwell's equations.

### 2.2. Microscale heat transfer

A new viscous wave problem has appeared in simulating "microscale heat transfer". As a side effect of device-size reduction, microtechnologies based on high-rate heating of thin film structures have received attention due to the advancement of laser technologies and their applications to micromanufacturing processes [3,23,27], and understanding the thermal behavior of thin films is an essential component in successfully designing such microstructures [22,23,29, 32]. The numerical solution of the microscale heat equation has been studied by a few authors; see, e.g., [8,9,16,35,36] and references therein.

The standard thermal conductivity equation is formulated from the continuity of the energy density and Fourier's law [24]:

$$\begin{aligned} &\widehat{C}\tfrac{\partial u}{\partial t} + \nabla \cdot \mathbf{J} = f, \\ &\mathbf{J} = -K\nabla u, \end{aligned} \tag{6}$$

where $u$ is the temperature, $\widehat{C}$ denotes the heat capacity per unit volume, $K$ is the thermal conductivity, $f$ is the heat source, and $\mathbf{J} = (J_1, J_2, J_3)$ is called the energy flux density (or, the heat flux). Here, $\widehat{C}$ and $K$ are assumed to be time-independent.

When thermal conduction is performed in thin films having a microscale size in one direction (say, the $z$-direction), the heat flux and the temperature gradient enter Fourier's law at different times. This physical phenomenon is called the *lagging effect* [32] and is expressed by

$$J_3(\mathbf{x}, t + \tau_J) = -K\frac{\partial u}{\partial z}(\mathbf{x}, t + \tau_K), \tag{7}$$

where $\tau_J$ and $\tau_K$ are positive constants that represent time lags for the heat flux and the temperature gradient, respectively, in the $z$-direction.

Assuming $\tau_J$ and $\tau_K$ are small, we can approximate (7) as

$$J_3 + \tau_J \frac{\partial J_3}{\partial t} \cong -K\left(u_z + \tau_K \frac{\partial u_z}{\partial t}\right). \tag{8}$$

It follows from (6) that

$$(J_3)_z = -\widehat{C}\frac{\partial u}{\partial t} + (Ku_x)_x + (Ku_y)_y + f, \tag{9}$$

and the $z$-derivative of (8) becomes

$$(J_3)_z + \tau_J \frac{\partial}{\partial t}(J_3)_z = -(Ku_z)_z - \tau_K \frac{\partial}{\partial t}(Ku_z)_z. \tag{10}$$

Thus, from (9) and (10) the microscale heat equation reads

$$\widehat{C}\frac{\partial u}{\partial t} + \tau_J \widehat{C}\frac{\partial^2 u}{\partial t^2} - \nabla \cdot (K\nabla u) - \frac{\partial}{\partial t}\nabla \cdot (Q\nabla u) = S, \tag{11}$$

where

$$Q = \operatorname{diag}(\tau_J K, \tau_J K, \tau_K K), \qquad S = f + \tau_J \frac{\partial f}{\partial t}.$$

A no-flux boundary condition is often imposed for the standard heat equation. Thus, it is quite reasonable for (11) to impose the following, corresponding boundary condition:

$$\frac{\partial}{\partial t}(Q\nabla u) \cdot \nu + Ku_\nu = 0. \tag{12}$$

The object of this article is to introduce and analyze a three-level, unconditionally stable numerical algorithm and its *LOD* procedure for solving wave equations of the form (1) in $m$-D, $1 \leqslant m \leqslant 3$, including (4)–(5) and (11)–(12).

### 2.3. Conventional numerical approaches

It is worth reviewing conventional methods for solving second-order hyperbolic equations. Let $\mathcal{A}$ denotes an approximation of $-\Delta$. Then, the semi-discrete equation for the acoustic wave equation (4) is given by

$$\frac{1}{c^2}v_{tt} + \mathcal{A}v = S. \tag{13}$$

(Hereafter we omit expressing the boundary and initial conditions, for a simpler presentation.) To discretize (13) with respect to time, it is convenient to rewrite the system as follows:

$$\begin{aligned} &\eta_t + \mathcal{A}v = S, \\ &\frac{1}{c^2}v_t - \eta = 0, \end{aligned} \tag{14}$$

where $\eta$ is an auxiliary variable. Let $\Delta t$ denote the time step and set $t^n = n\Delta t$ and $u^n(\mathbf{x}) = u(\mathbf{x}, t^n)$. Then, a popular time discretization [21] is given by

$$\begin{aligned} &\text{(a)} \quad \frac{\eta^{n+1} - \eta^n}{\Delta t} + \mathcal{A}\big[\alpha v^{n+1} + (1-\alpha)v^n\big] = S^{n+\alpha}, \\ &\text{(b)} \quad \frac{1}{c^2}\frac{v^{n+1} - v^n}{\Delta t} - \big[\beta\eta^{n+1} + (1-\beta)\eta^n\big] = 0, \end{aligned} \tag{15}$$

where $0 \leqslant \alpha, \beta \leqslant 1$ are algorithm parameters and

$$S^{n+\alpha} = \alpha S^{n+1} + (1-\alpha)S^n.$$

It is known [21, §9.11] that

- The algorithm (15) is unconditionally stable when $\alpha, \beta \geqslant 0.5$.
- It is second-order correct if $(\alpha, \beta) = (0.5, 0.5)$, for example.
- It coincides with the explicit second-order central scheme when $(\alpha, \beta) = (0, 1)$.

This scheme works well for smooth solutions, but otherwise it can introduce severe nonphysical oscillations.

The case $(\alpha, \beta) = (0.5, 0.5)$ is particularly interesting, because it allows the algorithm to be both second-order correct and unconditionally stable. For implementation purposes, it is convenient to substitute $\eta^{n+1}$ from (15b) into (15a). Then, we obtain the equivalent system

$$\begin{aligned} &\text{(a)} \quad \left(\frac{1}{c^2} + \frac{(\Delta t)^2}{4}\mathcal{A}\right)v^{n+1} = \left(\frac{1}{c^2} - \frac{(\Delta t)^2}{4}\mathcal{A}\right)v^n + \Delta t\eta^n + \frac{(\Delta t)^2}{2}S^{n+1/2}, \\ &\text{(b)} \quad \eta^{n+1} = -\eta^n + \frac{2}{c^2\Delta t}\big(v^{n+1} - v^n\big). \end{aligned} \tag{16}$$

Note that solving (16a) requires inverting a matrix, presumably large. However, since the matrix is symmetric and *strongly* diagonally dominant, most iterative algebraic solvers should converge rapidly unless the time step size $\Delta t$ is selected unreasonably large. It follows from (14) that (16) can also be derived as follows:

$$\begin{aligned} \eta^n &\cong \eta^{n+1/2} - \frac{\Delta t}{2}\eta_t^{n+1/2} + \mathcal{O}\big((\Delta t)^2\big) \\ &= \eta^{n+1/2} - \frac{\Delta t}{2}\big(-\mathcal{A}v^{n+1/2} + S^{n+1/2}\big) + \mathcal{O}\big((\Delta t)^2\big) \\ &\cong \frac{1}{c^2}\frac{v^{n+1} - v^n}{\Delta t} + \frac{\Delta t}{4}\mathcal{A}\big(v^{n+1} + v^n\big) - \frac{\Delta t}{2}S^{n+1/2} + \mathcal{O}\big((\Delta t)^2\big), \\ \eta^{n+1/2} &\cong \frac{\eta^{n+1} + \eta^n}{2} + \mathcal{O}\big((\Delta t)^2\big) \cong \frac{1}{c^2}\frac{v^{n+1} - v^n}{\Delta t} + \mathcal{O}\big((\Delta t)^2\big), \end{aligned} \tag{17}$$

where "$\cong$" denotes the introduction of a truncation error, second-order in $\Delta t$.

The microscale heat equation (11) can be approximated numerically in fashions analogous to the procedure (15); see [8,36]. Here, we begin by presenting the corresponding algorithm for (1), for which the auxiliary variable is

$$\eta = \gamma_1 v + \gamma_2 v_t. \tag{18}$$

By analogy to (17),

$$\begin{aligned}
&\text{(a)} \quad \left(\frac{\Delta t}{2}\gamma_1 + \gamma_2 + \frac{(\Delta t)^2}{4}\mathcal{A} + \frac{\Delta t}{2}\mathcal{Q}\right)v^{n+1} \\
&\qquad = \left(-\frac{\Delta t}{2}\gamma_1 + \gamma_2 - \frac{(\Delta t)^2}{4}\mathcal{A} + \frac{\Delta t}{2}\mathcal{Q}\right)v^n + \Delta t\eta^n + \frac{(\Delta t)^2}{2}S^{n+1/2}, \\
&\text{(b)} \quad \eta^{n+1} = -\eta^n + \left(\gamma_1 + \frac{2\gamma_2}{\Delta t}\right)v^{n+1} + \left(\gamma_1 - \frac{2\gamma_2}{\Delta t}\right)v^n,
\end{aligned} \tag{19}$$

where

$$\mathcal{A}v \approx -\nabla\cdot(A\nabla v) \quad \text{and} \quad \mathcal{Q}v \approx -\nabla\cdot(Q\nabla v)$$

are proper spatial approximations such as defined in (21) below. Algorithm (19) is second-order correct and unconditionally stable.

## 3. Discretization schemes

In this section, we introduce a three-level numerical algorithm for (1) which does not introduce an auxiliary variable; it will be the natural generalization of a standard time-stepping procedure [12,17] for wave equations not including the time-differentiated diffusive term. The three-level algorithm is equivalent to the conventional two-level method (15) for some selected algorithm parameters; the equivalence is discussed below.

### *3.1. The implicit time-stepping algorithm*

We first define two difference operators

$$\bar{\partial}_t u^n = \frac{u^{n+1} - u^{n-1}}{2\Delta t}, \qquad \bar{\partial}_{tt} u^n = \frac{u^{n+1} - 2u^n + u^{n-1}}{(\Delta t)^2}. \tag{20}$$

Let $h$ denote the spatial grid size. Define two approximation operators corresponding to the diffusion components in (1) such that

$$\begin{aligned}
&\mathcal{A}\phi = -\nabla\cdot(A\nabla\phi) + \mathcal{O}\big(h^p\big), \\
&\mathcal{Q}\phi = -\nabla\cdot(Q\nabla\phi) + \mathcal{O}\big(h^p\big), \quad p \geqslant 2,
\end{aligned} \tag{21}$$

for smooth functions $\phi$. The operators $\mathcal{A}$ and $\mathcal{Q}$ can be based on finite differences, finite elements, or finite volumes; i.e., the subject at hand is time discretization, not spatial discretization.

Then, an implicit, second-order time-stepping algorithm for (1) can be defined as follows. Given $v^0, \ldots, v^n$, $n \geqslant 1$, find $v^{n+1}$ by solving

$$\gamma_1\bar{\partial}_t v^n + \gamma_2\bar{\partial}_{tt}v^n + \mathcal{A}\big(\theta v^{n+1} + (1-2\theta)v^n + \theta v^{n-1}\big) + \bar{\partial}_t(\mathcal{Q}v)^n = S^n, \tag{22}$$

where $\theta \in [0, 0.5]$. For a second-order scheme for the initialization of $v^1$, see Section 7.1. The no-flux boundary condition for (22) given by

$$\big(A\nabla\big[\theta v^{n+1} + (1-2\theta)v^n + \theta v^{n-1}\big]\big)\cdot\nu + \bar{\partial}_t\big(Q\nabla v^n\big)\cdot\nu = 0 \tag{23}$$

must be interpreted so as to be consistent with the method employed for spatial discretization; that is, its discrete form will be different depending on whether finite difference, finite element, or finite volume discretization is used.

It is easy to see that every approximation involved in (22) produces a truncation error of the form

$$\mathcal{O}\big((\Delta t)^2 + h^p\big), \quad p \geqslant 2.$$

The algebraic system for (22), incorporating (23), has the form

$$\left(\frac{\Delta t}{2}\gamma_1 + \gamma_2 + (\Delta t)^2\theta\mathcal{A} + \frac{\Delta t}{2}\mathcal{Q}\right)\mathbf{v}^{n+1} = \mathbf{b};$$

the matrix in the system is symmetric and positive definite when the boundary condition is properly discretized. When the coefficient $Q$ is small, the matrix becomes strongly diagonally dominant and most iterative algebraic solvers converge quickly.

When $\theta = 0.25$, the three-level implicit algorithm (22) has the same matrix to be inverted as the standard two-level method (19). As a matter of fact, the two algorithms are equivalent; see Section 3.2 below.

### *3.2. Equivalence between the two-level and three-level methods*

Now, we verify equivalence between the two-level and the three-level algorithms, for selected algorithm parameters. For simplicity, we begin with the acoustic wave equation (4).

We first consider the three-level algorithm (22) for (4):

$$\left(\frac{1}{c^2} + \theta(\Delta t)^2\mathcal{A}\right)v^{n+1} + \left(-\frac{2}{c^2} + (1-2\theta)(\Delta t)^2\mathcal{A}\right)v^n + \left(\frac{1}{c^2} + \theta(\Delta t)^2\mathcal{A}\right)v^{n-1} = (\Delta t)^2 S^n. \tag{24}$$

Below, we eliminate the auxiliary variables $\eta^n$ and $\eta^{n+1}$ from the two-level algorithm. Multiply (15a) by $\beta\Delta t$ and add the resulting equation to (15b):

$$\left(\frac{1}{c^2} + \alpha\beta(\Delta t)^2\mathcal{A}\right)v^{n+1} + \left(-\frac{1}{c^2} + (1-\alpha)\beta(\Delta t)^2\mathcal{A}\right)v^n - \Delta t\eta^n = \beta(\Delta t)^2 S^{n+\alpha}. \tag{25}$$

The equation for $\eta^{n+1}$ can be obtained by multiplying (15a) by $(1-\beta)\Delta t$ and subtracting (15b) from the resulting equation:

$$\eta^{n+1} = (1-\beta)\Delta t S^{n+\alpha} - (1-\beta)\Delta t\mathcal{A}\left[\alpha v^{n+1} + (1-\alpha)v^n\right] + \frac{1}{\Delta t c^2}\left(v^{n+1} - v^n\right). \tag{26}$$

Substitute $\eta^n$ from (26) into (25):

$$\begin{aligned}&\left[\frac{1}{c^2} + \alpha\beta(\Delta t)^2\mathcal{A}\right]v^{n+1} + \left[-\frac{2}{c^2} + (\alpha+\beta-2\alpha\beta)(\Delta t)^2\mathcal{A}\right]v^n \\ &\quad + \left[\frac{1}{c^2} + (1-\alpha)(1-\beta)(\Delta t)^2\mathcal{A}\right]v^{n-1} = (\Delta t)^2 S^{n,\alpha,\beta},\end{aligned} \tag{27}$$

where

$$S^{n,\alpha,\beta} = \beta S^{n+\alpha} + (1-\beta)S^{n-1+\alpha}.$$

A necessary and sufficient condition for the equivalence of (24) and (27) is that

$$\alpha\beta = (1-\alpha)(1-\beta) = \theta. \tag{28}$$

(Here we have ignored the difference contributed from the source term, because it does not affect the mathematical stability and the difference is ignorable compared to the truncation error.) The first equality in (28) implies that $\alpha + \beta = 1$, so that $\alpha\beta \leqslant 0.25$. It is known that (27) is stable when $\alpha, \beta \geqslant 0.5$. On the other hand, stability holds for (24) when $\theta \in [0.25, 0.5]$; see Theorem 5.1 below. Thus, for unconditionally stable algorithms for the acoustic wave equation, the choice $\alpha = \beta = 0.5$ is the only case for the two-level algorithm to be equivalent to the three-level algorithm ($\theta = 0.25$).

Now, consider the two-level algorithm for solving the general equation (1):

$$\begin{aligned}&\frac{\eta^{n+1} - \eta^n}{\Delta t} + \mathcal{A}\left[\alpha v^{n+1} + (1-\alpha)v^n\right] + \frac{\mathcal{Q}v^{n+1} - \mathcal{Q}v^n}{\Delta t} = S^{n+\alpha}, \\ &\gamma_1\left[\beta v^{n+1} + (1-\beta)v^n\right] + \gamma_2\frac{v^{n+1} - v^n}{\Delta t} - \left[\beta\eta^{n+1} + (1-\beta)\eta^n\right] = 0.\end{aligned} \tag{29}$$

Again eliminate the auxiliary variables $\eta^n$ and $\eta^{n+1}$ to obtain the relation

$$\begin{aligned}&\big[\beta\Delta t\gamma_1+\gamma_2+\alpha\beta(\Delta t)^2\mathcal{A}+\beta\Delta t\mathcal{Q}\big]v^{n+1}\\&\quad+\big[(1-2\beta)\Delta t\gamma_1-2\gamma_2+(\alpha+\beta-2\alpha\beta)(\Delta t)^2\mathcal{A}+(1-2\beta)\Delta t\mathcal{Q}\big]v^n\\&\quad+\big[-(1-\beta)\Delta t\gamma_1+\gamma_2+(1-\alpha)(1-\beta)(\Delta t)^2\mathcal{A}-(1-\beta)\Delta t\mathcal{Q}\big]v^{n-1}\\&=(\Delta t)^2S^{n,\alpha,\beta}.\end{aligned}\tag{30}$$

The algorithms (22) and (30) are clearly equivalent when $\theta=0.25$ and $\alpha=\beta=0.5$, which again is the unique case for the equivalence under unconditional stability.

## 4. The *LOD* procedure

In this section, we assume that the problem is posed on the cubic domain $[0,1]^m$ with diagonal diffusion tensors:

$$A=\operatorname{diag}(a_1,\dots,a_m),\quad a_\ell=a_\ell(x)\geqslant 0;\qquad Q=\operatorname{diag}(q_1,\dots,q_m),\quad q_\ell=q_\ell(x)\geqslant 0.$$

Then, the solution of (1a) can be approximated efficiently by employing locally one-dimensional (*LOD*) procedures such as alternating-direction (*AD*) and fractional-step (*FS*) [18,26,33,34] methods, and doing so is the object of this section. The *AD* method was first introduced, as a perturbation of Crank–Nicolson difference equation for the numerical solution of the heat equation, in three papers [10,15,28] by Douglas, Peaceman, and Rachford. It has been applied to mildly nonlinear problems [11], nonsymmetric problems [31], and various physical problems [1,5,19]. A general formulation for *AD* methods for parabolic and hyperbolic problems can be found in [13]. Recently, Douglas and Kim [14] suggested a unified formulation for *AD* and *FS* in which the splitting error is *virtually* eliminated.

Recall (22)–(23). Let $\nabla_h=(D_{x_1,h},\dots,D_{x_m,h})^{\mathrm{T}}$, where $D_{x_\ell,h}$ are the half step central difference in the $x_\ell$-direction. Define $\mathcal{A}_\ell$ and $\mathcal{Q}_\ell$ as follows:

$$\begin{aligned}&\mathcal{A}_\ell u^n=-D_{x_\ell,h}\big(a_\ell D_{x_\ell,h}u^n\big),\\&\mathcal{Q}_\ell u^n=-D_{x_\ell,h}\big(q_\ell D_{x_\ell,h}u^n\big),\end{aligned}\qquad \ell=1,\dots,m,$$

where $A=\operatorname{diag}(a_1,\dots,a_m)$ and $Q=\operatorname{diag}(q_1,\dots,q_m)$, so that

$$\mathcal{A}=\mathcal{A}_1+\cdots+\mathcal{A}_m,\qquad \mathcal{Q}=\mathcal{Q}_1+\cdots+\mathcal{Q}_m.$$

The following *LOD* procedure would be efficient when $\mathcal{A}_\ell$ and $\mathcal{Q}_\ell$ are easy to invert; the matrices can be also obtained by other numerical methods (e.g., the linear finite element method on Cartesian grids).

As a perturbation of (22), 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:

$$\gamma_1\frac{w^{n+1,0}-w^{n-1}}{2\Delta t}+\gamma_2\frac{w^{n+1,0}-2w^n+w^{n-1}}{(\Delta t)^2}+\mathcal{A}w^n+\frac{\mathcal{Q}(w^n-w^{n-1})}{\Delta t}=S^n,\tag{31}$$

and then apply the implicit stepping

$$\begin{aligned}&\gamma_1\frac{w^{n+1,\ell}-w^{n+1,\ell-1}}{2\Delta t}+\gamma_2\frac{w^{n+1,\ell}-w^{n+1,\ell-1}}{(\Delta t)^2}+\theta\mathcal{A}_\ell\big(w^{n+1,\ell}-2w^n+w^{n-1}\big)\\&\qquad+\frac{\mathcal{Q}_\ell(w^{n+1,\ell}-2w^n+w^{n-1})}{2\Delta t}=0,\quad \ell=1,\dots,m,\\&w^{n+1}=w^{n+1,m}.\end{aligned}\tag{32}$$

The boundary conditions for (31) and (32) can be imposed respectively as

$$\begin{aligned}&\big(A\nabla_h w^n\big)\cdot\nu+\frac{(Q\nabla_h(w^n-w^{n-1}))\cdot\nu}{\Delta t}=0,\\&\theta a_\ell D_{x_\ell,h}\big(w^{n+1,\ell}-2w^n+w^{n-1}\big)+\frac{q_\ell D_{x_\ell,h}(w^{n+1,\ell}-2w^n+w^{n-1})}{2\Delta t}=0,\quad \ell=1,\dots,m,\end{aligned}\tag{33}$$

which correspond to (23).

In the following lemma, we reformulate the *LOD* algorithm (31)–(32) to eliminate the intermediate values. The equivalent reformulation will be helpful for the stability and error analysis.

**Lemma 4.1.** *Define* $\gamma_T$ *and* $\mathcal{E}_\ell$ *as*

$$\gamma_T := \gamma_2 + \gamma_1 \Delta t/2; \qquad \mathcal{E}_\ell := \mathcal{Q}_\ell + 2\theta\Delta t\mathcal{A}_\ell, \quad \ell = 1, \ldots, m.$$

*Then, the LOD procedure* (31)–(32) *is equivalent to*

$$\gamma_1 \bar{\partial}_t w^n + \gamma_2 \bar{\partial}_{tt} w^n + \mathcal{A}\big[\theta w^{n+1} + (1-2\theta)w^n + \theta w^{n-1}\big] + \bar{\partial}_t(\mathcal{Q}w)^n + \mathcal{B}\big(w^{n+1} - 2w^n + w^{n-1}\big) = S^n, \tag{34}$$

*where*

$$\mathcal{B} := \frac{1}{4\gamma_T} \sum_{1 \leqslant \ell_1 < \ell_2 \leqslant m} \mathcal{E}_{\ell_1}\mathcal{E}_{\ell_2} + \cdots + \frac{(\Delta t)^{m-2}}{2^m \gamma_T^{m-1}} \mathcal{E}_1\mathcal{E}_2 \cdots \mathcal{E}_m. \tag{35}$$

**Proof.** We sketch the proof for three spatial variables ($m = 3$); the general case follows straightforward. Add the four $(1+3)$ equations in the *LOD* procedure to see

$$\begin{aligned} &\gamma_1 \bar{\partial}_t w^n + \gamma_2 \bar{\partial}_{tt} w^n + \mathcal{A}\big[\theta w^{n+1} + (1-2\theta)w^n + \theta w^{n-1}\big] + \bar{\partial}_t(\mathcal{Q}w)^n \\ &\quad + \frac{1}{2\Delta t}\mathcal{E}_1\big(w^{n+1,1} - w^{n+1}\big) + \frac{1}{2\Delta t}\mathcal{E}_2\big(w^{n+1,2} - w^{n+1}\big) = S^n. \end{aligned} \tag{36}$$

Note that the second and third cases ($\ell = 2, 3$) in (32) read

$$\begin{aligned} w^{n+1,1} - w^{n+1,2} &= \frac{\Delta t}{2\gamma_T}\mathcal{E}_2\big(w^{n+1,2} - 2w^n + w^{n-1}\big), \\ w^{n+1,2} - w^{n+1} &= \frac{\Delta t}{2\gamma_T}\mathcal{E}_3\big(w^{n+1} - 2w^n + w^{n-1}\big). \end{aligned} \tag{37}$$

Add the two equations in (37) to have

$$\begin{aligned} w^{n+1,1} - w^{n+1} &= \left[\frac{\Delta t}{2\gamma_T}\mathcal{E}_2 + \frac{\Delta t}{2\gamma_T}\mathcal{E}_3\right]\big(w^{n+1} - 2w^n + w^{n-1}\big) + \frac{\Delta t}{2\gamma_T}\mathcal{E}_2\big(w^{n+1,2} - w^{n+1}\big) \\ &= \left[\frac{\Delta t}{2\gamma_T}\mathcal{E}_2 + \frac{\Delta t}{2\gamma_T}\mathcal{E}_3 + \left(\frac{\Delta t}{2\gamma_T}\right)^2 \mathcal{E}_2\mathcal{E}_3\right]\big(w^{n+1} - 2w^n + w^{n-1}\big), \end{aligned}$$

and therefore

$$\mathcal{E}_1\big(w^{n+1,1} - w^{n+1}\big) + \mathcal{E}_2\big(w^{n+1,2} - w^{n+1}\big) = 2\Delta t\mathcal{B}_3\big(w^{n+1} - 2w^n + w^{n-1}\big), \tag{38}$$

where

$$\mathcal{B}_3 = \frac{1}{4\gamma_T}(\mathcal{E}_1\mathcal{E}_2 + \mathcal{E}_1\mathcal{E}_3 + \mathcal{E}_2\mathcal{E}_3) + \frac{\Delta t}{8\gamma_T^2}\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3.$$

Here $\mathcal{B}_3$ coincides with $\mathcal{B}$ in 3-D. Eq. (34) follows from (36) and (38) for three spatial variables. □

The last term in the left side of (34), the one including $\mathcal{B}$, is called the *splitting error* which has been introduced into the algorithm as an extra error. Compare (34) with (22). Since $\mathcal{B} = \mathcal{O}(1)$ when $\mathcal{E}_\ell = \mathcal{O}(1)$, the splitting error of the *LOD* procedure (31)–(32) is $\mathcal{O}((\Delta t)^2)$, the same order in $\Delta t$ as the truncation error, for which in fact we have dared to bring a *slight* complication in the *LOD* procedure (31)–(32).

When the coefficients $\tau_J$ and $\tau_K$ in the microscale thermal equation (11) is small (e.g., $\tau_J, \tau_K = \mathcal{O}(\Delta t)$), which is the case in most computation environments in realistic media, we have $\mathcal{E}_\ell = \mathcal{O}(\Delta t)$ and $\gamma_T = \mathcal{O}(\Delta t)$ and therefore the splitting error of the *LOD* algorithm becomes $\mathcal{O}(\Delta t^3)$ (In this case, $\mathcal{B} = \mathcal{O}(\Delta t)$.). Thus the splitting error in the solution of the *LOD* algorithm must not be apparent for reasonable time step sizes $\Delta t$. The argument applies to the *LOD* solution of the acoustic wave equation, which has a splitting error of $\mathcal{O}(\Delta t^4)$.

Note that in each substep of (32), the matrix to be inverted is tridiagonal for the second-order spatial FD schemes; the *LOD* method (31)–(32) requires $\mathcal{O}(N := n_t n_{x_1} \cdots n_{x_m})$ flops, where $n_t$ and $n_{x_\ell}$, $\ell = 1, \ldots, m$, are the number of grid points in the $t$- and $x_\ell$-directions, respectively.

The *LOD* procedure (31)–(32) can be implemented as ($n \geqslant 1$)

$$
\begin{aligned}
&w^{n+1,0} = \frac{(\Delta t)^2}{\gamma_T}\left[S^n + \frac{\gamma_1}{2\Delta t} w^{n-1} + \frac{\gamma_2}{(\Delta t)^2}\widetilde{w}^n - \mathcal{A}w^n - \frac{1}{\Delta t}\mathcal{Q}\big(w^n - w^{n-1}\big)\right],\\
&\left(\gamma_T + \frac{\Delta t}{2}\mathcal{E}_\ell\right) w^{n+1,\ell} = \gamma_T w^{n+1,\ell-1} + \frac{\Delta t}{2}\mathcal{E}_\ell \widetilde{w}^n, \quad \ell = 1, \ldots, m,\\
&w^{n+1} = w^{n+1,m},
\end{aligned}
\tag{39}
$$

where $w^0 = g_0$, the initial value $w^1$ can be computed as in (54) below, and

$$
\widetilde{w}^n := 2w^n - w^{n-1}; \qquad \gamma_T := \gamma_2 + \gamma_1 \frac{\Delta t}{2}; \qquad \mathcal{E}_\ell := \mathcal{Q}_\ell + 2\theta\Delta t \mathcal{A}_\ell, \quad \ell = 1, \ldots, m.
$$

(Here we have collected the definitions.)

## 5. Stability analysis

In this section, we analyze stability of the three-level implicit algorithm (22) and its *LOD* procedure (34) for solving wave equations of the form (1). In the analysis, we will assume that $\mathcal{B}$ is symmetric and $(\gamma_2 + (\Delta t)^2\mathcal{B})$ is nonnegative. The nonnegativity assumption is reasonable from the definition of $\mathcal{B}$ in (35) and $\gamma_2 > 0$. In particular, $\mathcal{B}$ itself is nonnegative when $\mathcal{A}_\ell$ and $\mathcal{Q}_\ell$ commute.

**Theorem 5.1.** *Suppose that $\mathcal{B}$ is symmetric and $(\gamma_2 + (\Delta t)^2\mathcal{B})$ is nonnegative. Let $\theta \in [0.25, 0.5]$. Then* (22) *and its LOD procedure* (31)–(32) *are unconditionally stable.*

**Proof.** It suffices to show the stability of the *LOD* procedure (34); stability of (22) follows by replacing $w^n$ by $v^n$ and setting $\mathcal{B} = 0$.

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.) Choose $\bar{\partial}_t w^n$ as a test function for (34) and set $S^n = 0$. Then, for $n \geqslant 1$,

$$
\begin{aligned}
&\big((\gamma_1 + \mathcal{Q})\bar{\partial}_t w^n, \bar{\partial}_t w^n\big) + \big(\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)\bar{\partial}_{tt} w^n, \bar{\partial}_t w^n\big)\\
&\quad + \big(\mathcal{A}\big[\theta w^{n+1} + (1-2\theta)w^n + \theta w^{n-1}\big], \bar{\partial}_t w^n\big) = 0.
\end{aligned}
\tag{40}
$$

Multiply (40) by $\Delta t$ and sum beginning at $n = 1$ to have

$$
\begin{aligned}
&\sum_{j=1}^{n} \big\|(\gamma_1 + \mathcal{Q})^{1/2}\bar{\partial}_t w^j\big\|^2 \Delta t + \sum_{j=1}^{n}\big(\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)\bar{\partial}_{tt} w^j, \bar{\partial}_t w^j\big)\Delta t\\
&\quad + \sum_{j=1}^{n}\big(\mathcal{A}\big[\theta w^{j+1} + (1-2\theta)w^j + \theta w^{j-1}\big], \bar{\partial}_t w^j\big)\Delta t = 0.
\end{aligned}
\tag{41}
$$

Consider the following equalities

$$
\begin{aligned}
\big(\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)\bar{\partial}_{tt} w^j, \bar{\partial}_t w^j\big)\Delta t &= \frac{1}{2}\big(\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)\big(\partial_t^+ - \partial_t^-\big)w^j, \big(\partial_t^+ + \partial_t^-\big)w^j\big)\\
&= \frac{1}{2}\big\|\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)^{1/2}\partial_t^+ w^j\big\|^2 - \frac{1}{2}\big\|\big(\gamma_2 + (\Delta t)^2\mathcal{B}\big)^{1/2}\partial_t^- w^j\big\|^2,\\
\big(\mathcal{A}\big[\theta w^{j+1} + (1-2\theta)w^j + \theta w^{j-1}\big], \bar{\partial}_t w^j\big)\Delta t &= \frac{1}{2}\big(\mathcal{A}\big[\theta\big(w^{j+1} + w^{j-1}\big) + (1-2\theta)w^j\big], w^{j+1} - w^{j-1}\big)\\
&= \frac{1}{2}\mathcal{P}^+\big(w^j, \theta\big) - \frac{1}{2}\mathcal{P}^-\big(w^j, \theta\big),
\end{aligned}
\tag{42}
$$

where

$$\partial_t^{\pm} w^j := \pm\big(w^{j\pm1} - w^j\big)/\Delta t,$$
$$\mathcal{P}^{\pm}\big(w^j,\theta\big) := \theta\big(\mathcal{A}w^j, w^j\big) + \theta\big(\mathcal{A}w^{j\pm1}, w^{j\pm1}\big) + (1-2\theta)\big(\mathcal{A}w^j, w^{j\pm1}\big).$$

Note that

$$\partial_t^{-} w^j = \partial_t^{+} w^{j-1}, \qquad \mathcal{P}^{-}\big(w^j,\theta\big) = \mathcal{P}^{+}\big(w^{j-1},\theta\big), \tag{43}$$

and $\mathcal{P}^{\pm}(w^j,\theta) > 0$, for $\theta \in (0.25, 0.5]$, unless $w^j = w^{j\pm1} \equiv 0$. Thus, it follows from (41), (42), and (43) that

$$\begin{aligned} &2\sum_{j=1}^{n} \big\|(\gamma_1+\mathcal{Q})^{1/2}\bar{\partial}_t w^j\big\|^2\Delta t + \big\|\big(\gamma_2+(\Delta t)^2\mathcal{B}\big)^{1/2}\partial_t^{+} w^n\big\|^2 + \mathcal{P}^{+}\big(w^n,\theta\big) \\ &\quad\leqslant \big\|\big(\gamma_2+(\Delta t)^2\mathcal{B}\big)^{1/2}\partial_t^{+} w^0\big\|^2 + \mathcal{P}^{+}\big(w^0,\theta\big), \end{aligned} \tag{44}$$

which completes the proof. □

**Remark.** For the propagation of acoustic waves incorporating the *ABC* (5), the above analysis requires a minor modification. The left side of (40) must be added by the boundary integral

$$\big|c^{-1/2}\bar{\partial}_t w^n\big|_{\Gamma}^2,$$

where $|\cdot|_{\Gamma}$ denotes the trace $L^2$-norm on $\Gamma$. The corresponding sums should appear also in the left sides of (41) and (44), which does not alter the mathematical stability. For an approximation and implementation details for the *ABC*, see Section 7.2.

## 6. Convergence analysis

In this section, we analyze the errors associated with the algorithm (22) and its *LOD* procedure (34) for solving (1), when $(\gamma_1+\mathcal{Q})$ is positive definite. Again, we focus the analysis on (34).

The error equation associated with (34) is

$$\gamma_1\bar{\partial}_t e^n + \gamma_2\bar{\partial}_{tt} e^n + \mathcal{A}\big[\theta e^{n+1} + (1-2\theta)e^n + \theta e^{n-1}\big] + \bar{\partial}_t(\mathcal{Q}e)^n + \mathcal{B}\big(e^{n+1} - 2e^n + e^{n-1}\big) = \delta^n, \tag{45}$$

where $\delta^n$ denotes the sum of the discretization/truncation error ($\tau^n$) and *LOD*-perturbation error ($\sigma^n$) on the $n$th time level,

$$\delta^n = \tau^n + \sigma^n = \mathcal{O}\big((\Delta t)^2 + h^p\big), \quad p \geqslant 2. \tag{46}$$

Choose $\bar{\partial}_t e^n$ as a test function. Then, for $n \geqslant 1$,

$$\begin{aligned} &\big((\gamma_1+\mathcal{Q})\bar{\partial}_t e^n, \bar{\partial}_t e^n\big) + \big(\big(\gamma_2+(\Delta t)^2\mathcal{B}\big)\bar{\partial}_{tt} e^n, \bar{\partial}_t e^n\big) \\ &\qquad + \big(\mathcal{A}\big[\theta e^{n+1} + (1-2\theta)e^n + \theta e^{n-1}\big], \bar{\partial}_t e^n\big) = \big(\delta^n, \bar{\partial}_t e^n\big). \end{aligned} \tag{47}$$

An application of Cauchy–Schwarz inequality to the right side of (47) reads

$$\big(\delta^n, \bar{\partial}_t e^n\big) \leqslant \frac{1}{2}\big\|(\gamma_1+\mathcal{Q})^{-1/2}\delta^n\big\|^2 + \frac{1}{2}\big\|(\gamma_1+\mathcal{Q})^{1/2}\bar{\partial}_t e^n\big\|^2, \tag{48}$$

where we have assumed that $(\gamma_1+\mathcal{Q})$ is positive. As in the stability analysis in Section 5, multiply (47) by $\Delta t$, sum beginning at $n=1$, and apply the equalities in (42) and (48) to have the following theorem.

**Theorem 6.1.** *Assume that $(\gamma_1+\mathcal{Q})$ is positive definite, $\mathcal{B}$ is symmetric, and $(\gamma_2+(\Delta t)^2\mathcal{B})$ is nonnegative. Let $e^n = u^n - w^n$ and $\theta \in [0.25, 0.5]$. Then the LOD procedure* (31)–(32) *is unconditionally stable and its numerical error is bounded as*

$$\sum_{j=1}^{n} \left\| (\gamma_1 + \mathcal{Q})^{1/2} \bar{\partial}_t e^j \right\|^2 \Delta t + \left( \left(\gamma_2 + (\Delta t)^2 \mathcal{B}\right) \partial_t^+ e^n, \partial_t^+ e^n \right) + \mathcal{P}^+ \left(e^n, \theta\right)$$
$$\leqslant \sum_{j=1}^{n} \left\| \left(\gamma_1 + \mathcal{Q}\right)^{-1/2} \delta^j \right\|^2 \Delta t + \left( \left(\gamma_2 + (\Delta t)^2 \mathcal{B}\right) \partial_t^+ e^0, \partial_t^+ e^0 \right) + \mathcal{P}^+ \left(e^0, \theta\right). \tag{49}$$

Thus, the estimation of the error generated by the *LOD* method is reduced to bounding the errors in $w^0$ and $w^1$; it is advisable to obtain $w^1$ *accurately*; see Section 7.1 below.

When $\gamma_1 > 0$, Eq. (48) can be replaced by

$$\left(\delta^n, \bar{\partial}_t e^n\right) \leqslant \frac{1}{2} \left\| \gamma_1^{-1/2} \delta^n \right\|^2 + \frac{1}{2} \left\| \gamma_1^{1/2} \bar{\partial}_t e^n \right\|^2$$

and therefore (49) can be rewritten as

$$\sum_{j=1}^{n} \left\| \gamma_1^{1/2} \bar{\partial}_t e^j \right\|^2 \Delta t + 2 \sum_{j=1}^{n} \left\| \mathcal{Q}^{1/2} \bar{\partial}_t e^j \right\|^2 \Delta t + \left( \left(\gamma_2 + (\Delta t)^2 \mathcal{B}\right) \partial_t^+ e^n, \partial_t^+ e^n \right) + \mathcal{P}^+ \left(e^n, \theta\right)$$
$$\leqslant \sum_{j=1}^{n} \left\| \gamma_1^{-1/2} \delta^j \right\|^2 \Delta t + \left( \left(\gamma_2 + (\Delta t)^2 \mathcal{B}\right) \partial_t^+ e^0, \partial_t^+ e^0 \right) + \mathcal{P}^+ \left(e^0, \theta\right). \tag{50}$$

The operator $\mathcal{Q}^{1/2} \bar{\partial}_t$ works as a genuine viscosity when $\gamma_1 > 0$. That is, the energy decreases strictly when it is nonzero; see also (44).

The arguments introduced in this section can be applied to the error estimate for the algorithm (22), by replacing $\delta^j$ by $\tau^j$ (the truncation error in the $j$th time level) and by setting $\mathcal{B} = 0$.

**Corollary 6.2.** *Assume that $(\gamma_1 + \mathcal{Q})$ is positive definite. Let $E^n = u^n - v^n$ and $\theta \in [0.25, 0.5]$. Then the algorithm* (22) *is unconditionally stable and its numerical error is bounded as*

$$\sum_{j=1}^{n} \left\| (\gamma_1 + \mathcal{Q})^{1/2} \bar{\partial}_t E^j \right\|^2 \Delta t + \left( \gamma_2 \partial_t^+ E^n, \partial_t^+ E^n \right) + \mathcal{P}^+ \left(E^n, \theta\right)$$
$$\leqslant \sum_{j=1}^{n} \left\| (\gamma_1 + \mathcal{Q})^{-1/2} \tau^j \right\|^2 \Delta t + \left( \gamma_2 \partial_t^+ E^0, \partial_t^+ E^0 \right) + \mathcal{P}^+ \left(E^0, \theta\right), \tag{51}$$

*where $\tau^j = \mathcal{O}((\Delta t)^2 + h^p)$, $p \geqslant 2$, denotes the truncation error in the $j$th time level.*

## 7. Implementation issues

### *7.1. Initialization of $u^1$*

It is important to initialize $u^1$ accurately for wave equations, because the error at later time levels is often bounded by the error introduced in the first two levels ($n = 0$ and $n = 1$); see the error analysis presented in Section 6 below.

We first recall the initial conditions given in (1c). Note that

$$u(\mathbf{x}, \Delta t) = u(\mathbf{x}, 0) + \Delta t \frac{\partial u}{\partial t}(\mathbf{x}, 0) + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, 0) + \mathcal{O}\left((\Delta t)^3\right) \tag{52}$$

and

$$\frac{\partial^2 u}{\partial t^2} = \frac{1}{\gamma_2} \left[ S - \gamma_1 \frac{\partial u}{\partial t} + \nabla \cdot (A \nabla u) + \frac{\partial}{\partial t} \nabla \cdot (Q \nabla u) \right]. \tag{53}$$

Thus, from (1c), (52), and (53), we may approximate $u^1$ as follows:

$$u^1 \approx g_0 + \Delta t g_1 + \frac{(\Delta t)^2}{2\gamma_2} \left[ S^0 - \gamma_1 g_1 - \mathcal{A} g_0 - \mathcal{Q} g_1 \right] + \mathcal{O}\left((\Delta t)^3 + h^p (\Delta t)^2\right). \tag{54}$$

See Section 7.3 for an alternative strategy for the computation of $u^1$.

### 7.2. The absorbing boundary condition

We first rewrite the implicit algorithm (22) and its *LOD* procedure (31)–(32) for the acoustic wave equation (4)–(5):

$$\frac{1}{c^2}\bar{\partial}_{tt}v^n + \mathcal{A}\big[\theta v^{n+1} + (1-2\theta)v^n + \theta v^{n-1}\big] = S^n, \tag{55}$$

where $\mathcal{A} \cong -\Delta$, and

$$\begin{aligned}
&\frac{1}{c^2}\frac{w^{n+1,0} - 2w^n + w^{n-1}}{(\Delta t)^2} + \mathcal{A}w^n = S^n,\\
&\frac{1}{c^2}\frac{w^{n+1,\ell} - w^{n+1,\ell-1}}{(\Delta t)^2} + \theta\mathcal{A}_\ell\big(w^{n+1,\ell} - 2w^n + w^{n-1}\big) = 0, \quad \ell = 1,\dots,m,\\
&w^{n+1} = w^{n+1,m}.
\end{aligned} \tag{56}$$

For both (55) and (56), the *ABC* is

$$\frac{1}{c}\bar{\partial}_t v^n + \big(\nabla_h v^n\big)\cdot\nu = 0. \tag{57}$$

Then, it should be noticed that the *ABC* (57) incorporates no spatial difference approximations on the $(n+1)$th time level. Thus, the *ABC* can be imposed in the explicit step of *LOD*, and the implicit directional sweeps are carried out with a Dirichlet boundary condition. To see details of this strategy, let $\Omega = (0,1)^2$ in 2-D (for simplicity) and consider a uniform mesh of grid size $h(:= \Delta x = \Delta y)$. Choose a grid point on an edge, e.g., $\mathbf{x}_{0j} = (x_0, y_j) \in \{\mathbf{x}\colon\ x = 0,\ 0 < y < 1\}$; we momentarily assume that the grid point is not one of the vertices.

The explicit step of the *LOD* algorithm (56) at $\mathbf{x}_{0j}$ reads

$$\frac{w_{0j}^{n+1,0} - \widetilde{w}_{0j}^n}{c_{0j}^2(\Delta t)^2} + \mathcal{A}w_{0j}^n = S_{0j}^n, \tag{58}$$

where $\widetilde{w}_{0j}^n = 2w_{0j}^n - w_{0j}^{n-1}$ and

$$\mathcal{A}w_{0j}^n = \frac{1}{h^2}\big(4w_{0j}^n - w_{-1,j}^n - w_{1j}^n - w_{0,j-1}^n - w_{0,j+1}^n\big).$$

At the same point, the *ABC* (57), incorporating a second-order approximation, becomes

$$\frac{w_{0j}^{n+1,0} - w_{0j}^{n-1}}{2c_{0j}\Delta t} + \frac{w_{-1,j}^n - w_{1j}^n}{2h} = 0. \tag{59}$$

Multiply (59) by $2/h$ and add the result to (58) to have

$$\left(\frac{1}{c_{0j}h\Delta t} + \frac{1}{c_{0j}^2(\Delta t)^2}\right)w_{0j}^{n+1,0} = S_{0j}^n + \frac{w_{0j}^{n-1}}{c_{0j}h\Delta t} + \frac{\widetilde{w}_{0j}^n}{c_{0j}^2(\Delta t)^2} - \frac{1}{h^2}\big(4w_{0j}^n - 2w_{1j}^n - w_{0,j-1}^n - w_{0,j+1}^n\big), \tag{60}$$

from which one can get $w_{0j}^{n+1,0}$.

One can apply the same technique to other boundary grid points, incorporating 1–2 *ABC* equations depending on whether the grid point is on one of edges or vertices. Here the boundary values obtained from the explicit step of *LOD*, on the $(n+1)$th time level, satisfy the *ABC* on the $n$th time level and have no further relations to fulfill; they can be fixed in the remainder of the computation. That is, the implicit directional sweeps of the *LOD* algorithm can be carried out with the boundary values fixed, a Dirichlet boundary condition. Thus $w^{n+1}|_\Gamma = w^{n+1,0}|_\Gamma$. Application of the arguments to a higher-dimensional space is straightforward.

When (55) is to be solved by adopting an iterative algebraic solver, one would better provide the initial value as accurate to the solution $v^{n+1}$ as possible. A good but inexpensive initial value is the solution obtained from the application of the explicit step of the *LOD* procedure. In this case, since the boundary values of $v^{n+1,0}$ already satisfy the *ABC*, the iteration can be carried out again with a Dirichlet boundary condition.

### 7.3. An alternative initialization of $u^1$

It should be noticed that when $u^1$ is derived *explicitly* as in (54), it may cause a nonphysical oscillation (instability) when $\Delta t$ is not sufficiently small or $u^0$ is discontinuous. In the case, we can modify the initialization scheme of $u^1$ in the following way. Let

$$c_1 = \frac{\Delta t}{2}\gamma_1 + \gamma_2, \qquad c_2 = -\frac{\Delta t}{2}\gamma_1 + \gamma_2.$$

Then (19a) equivalently reads (when $n = 0$)

$$\left(1 + \frac{(\Delta t)^2}{4c_1}\mathcal{A} + \frac{\Delta t}{2c_1}\mathcal{Q}\right)v^1 = \left(\frac{c_2}{c_1} - \frac{(\Delta t)^2}{4c_1}\mathcal{A} + \frac{\Delta t}{2c_1}\mathcal{Q}\right)v^0 + \frac{1}{c_1}F^0, \tag{61}$$

where

$$F^0 = \Delta t\eta^0 + \frac{(\Delta t)^2}{2}S^{1/2}.$$

Consider the case $m = 2$ (2D) for simplicity. Define

$$\mathcal{G}_\ell := \frac{(\Delta t)^2}{4c_1}\mathcal{A}_\ell + \frac{\Delta t}{2c_1}\mathcal{Q}_\ell, \quad \ell = 1, 2.$$

Then an efficient *LOD* procedure for (61) can be formulated as follows:

$$\begin{aligned}(1 + \mathcal{G}_1)w^* &= \left(\frac{c_2}{c_1} - \frac{(\Delta t)^2}{4c_1}\mathcal{A} + \frac{\Delta t}{2c_1}\mathcal{Q} + \mathcal{G}_1\mathcal{G}_2\right)w^0 + \frac{1}{c_1}F^0,\\ (1 + \mathcal{G}_2)w^1 &= w^*.\end{aligned} \tag{62}$$

By eliminating the intermediate solution $w^*$ from the above algorithm, one can easily see that the difference between the resulting equation and (61) is

$$\mathcal{G}_1\mathcal{G}_2\big(w^1 - w^0\big).$$

Thus the *LOD* procedure (62) has a splitting error of $\mathcal{O}((\Delta t)^2)$, when the solution is sufficiently smooth.

## 8. Numerical experiments

This section reports some of numerical experiments for the three-level algorithm (22) and its *LOD* procedure (31)–(32), in two spatial variables. The algorithms are written in F77, with the main and input/output routines in C++ for a dynamic memory allocation and a flexible data manipulation. The computation is carried out on a 860 MHz personal computer of a Linux operating system. In the following, CPU denotes the user time on the machine measured in second. Note that when $\theta = 0$, the *LOD* algorithm becomes the conventional second-order explicit scheme; it is also denoted as "*LOD*$[\theta = 0]$".

### 8.1. Acoustic wave propagation

For the purpose of checking numerical errors of the algorithms solving (4)–(5), we first set the domain and a constant sound velocity as

$$\Omega = (-a, a)^2, \quad a > 0; \qquad c(x, y) \equiv c_0 > 0.$$

Select a true solution $u$ as follows:

$$u(x, y, t) = \frac{\sin(\omega[x^2 + y^2 - 2ac_0t])}{\omega}, \quad (x, y, t) \in \Omega \times J, \tag{63}$$

where $\omega(:= 2\pi f)$, with $f$ being the frequency, is the angular frequency. Note that the solution $u$ in (63) satisfies the *ABC* (5) on the boundary, $t \geqslant 0$. The error $E_\infty(T)$ in the maximum norm at $t = T$ is defined as

Table 1
The error and computation time of *LOD* for various $\theta$'s and grid sizes for acoustic waves

| $(\Delta t, h)$ | $LOD[\theta = 0]$ | | $LOD[\theta = 0.25]$ | | $LOD[\theta = 0.5]$ | |
|---|---|---|---|---|---|---|
| | $E_\infty(T = 1)$ | CPU | $E_\infty(T = 1)$ | CPU | $E_\infty(T = 1)$ | CPU |
| (0.02, 0.04) | 2.47E−02 | 0.09 | 9.13E−02 | 0.18 | 1.57E−01 | 0.17 |
| (0.01, 0.02) | 6.31E−03 | 0.90 | 2.29E−02 | 1.77 | 3.51E−02 | 1.77 |
| (0.005, 0.01) | 1.57E−03 | 7.40 | 5.64E−03 | 14.62 | 8.93E−03 | 14.61 |

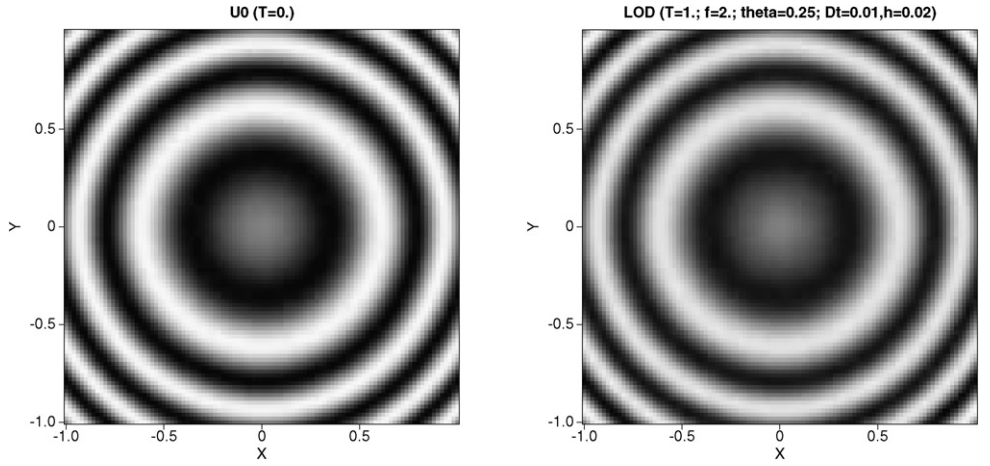


Fig. 1. The initial value $u^0$ (left) and the numerical solution of $LOD[\theta = 0.25]$ at $t = 1$ (right). In the setting, the analytic value at $t = 1$ is the same as the initial value.

$$E_\infty(T) = \max_{i,j} \left| u(\mathbf{x}_{ij}, T) - w(\mathbf{x}_{ij}, T) \right|,$$

where $w$ denotes the numerical solution for given $\Delta t$ and $h(:= \Delta x = \Delta y)$.

Table 1 presents the error and computation time of *LOD*, for various $\theta$'s and grid sizes for the acoustic waves. We set $a = 1$, $c_0 = 1$, and $\omega = 4\pi$ in Eq. (63); the final time $T = 1$. Fig. 1 contains the initial solution (left) and the numerical solution of $LOD[\theta = 0.25]$ at $t = 1$ (right). As one can see from the table, the algorithm shows a second-order accuracy and $\mathcal{O}(N)$ computation cost, for all parameters. The implicit cases take about twice the computation time of the explicit scheme. Also the explicit scheme results in the best accuracy for the constant coefficient problem; for example, $LOD[\theta = 0.25]$ has about four times larger error than the explicit scheme. Such a degeneration in accuracy seems coming from a "smoothing effect" of those implicit algorithms. However, when the time step size $\Delta t$ is set large, the explicit scheme becomes easily unstable, while the implicit scheme stays stable. For example, when we set $\Delta t$ so that $\frac{\Delta t}{h} \cdot c_0 = 0.715$, for $h = 0.02$ and $c_0 = 1$, the explicit scheme does not result in an accurate solution; $LOD[\theta = 0.25]$ still gives an accurate solution of error 3.32E−02.

It would be interesting to compare the accuracy of the $LOD[\theta]$ scheme with (22). It has been numerically verified that the difference of errors between the two methods (for the same $\theta$) is negligible ($< 1\%$) for most cases, as we have shown that the splitting error of $LOD[\theta]$ is $\mathcal{O}(\Delta t^4)$.

In Fig. 2, we investigate the quality of *ABC* (5). Set $\Omega = (0, 1)^2$, $c \equiv 1$, $\Delta t = 0.00125$, and $h = 0.0025$ ($\frac{\Delta t}{h} \cdot c = 0.5$). At the center of the domain, we locate a source of 20 Hz *Ricker* wavelet

$$S(\mathbf{x}_c, t) = \lambda^2 \left(1 - 2\pi^2\lambda^2 t^2\right) e^{-\pi^2\lambda^2 t^2},$$

where $\lambda$ is the frequency ($\lambda = 20$) and $\mathbf{x}_c$ denotes the center of the domain. Thus, the number of grid points per wavelength is ten. The figure contains two snapshots of $LOD[\theta = 0.25]$: $t = 0.6$ (left) and $t = 0.72$ (right). The earlier one corresponds to the moment when the wavefront passes the boundary with smaller incident angles; no apparent reflection from the artificial boundary is observed. When $t = 0.72$, the wavefront is about to pass the four corners of the domain. A certain degree of reflection is now observable, but not much. The reflection is not due to the implementation strategy in Section 7.2, but it is an intrinsic property of the Clayton–Engquist-type *ABC*s [6].

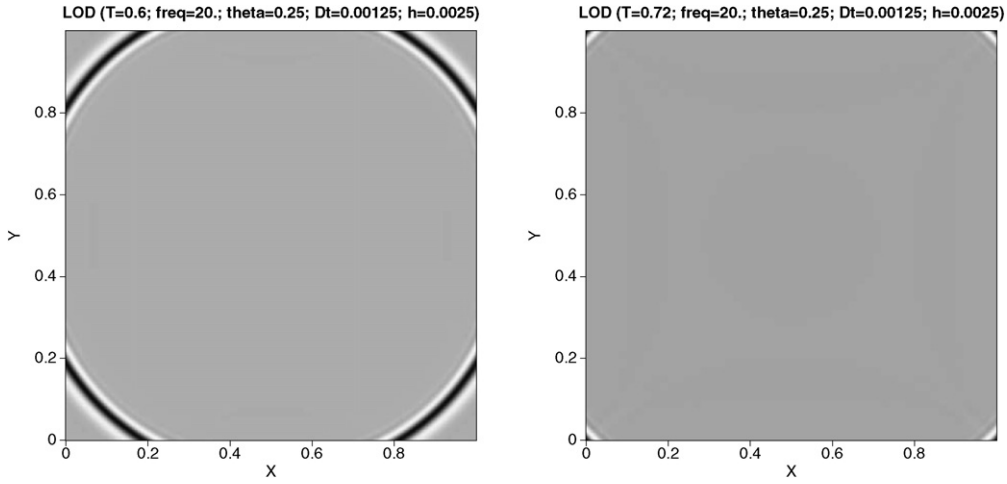


Fig. 2. Numerical solutions in different times: $t = 0.6$ (left) and $t = 0.72$ (right).

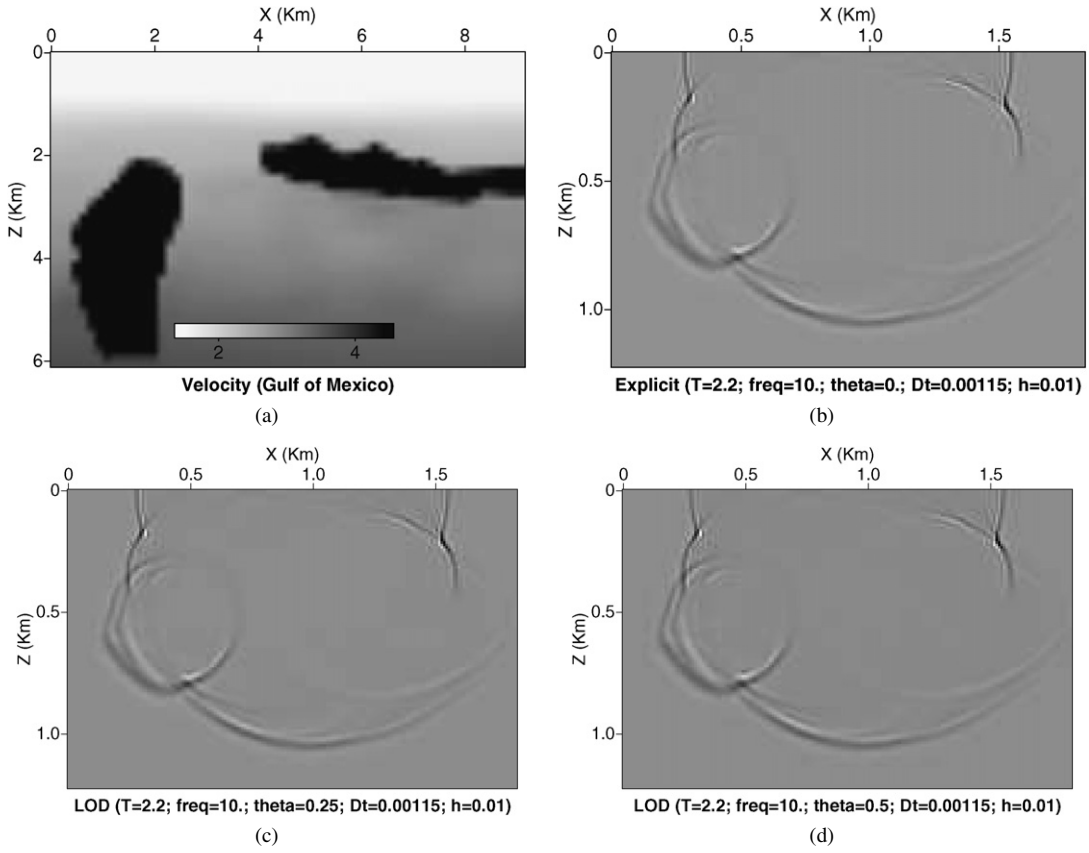


Fig. 3. A vertical section of a real velocity in the Gulf of Mexico (a) and the snapshots at $t = 2.2$ for $LOD[\theta = 0]$ (b), $LOD[\theta = 0.25]$ (c), and $LOD[\theta = 0.5]$ (d).

Fig. 3 shows a vertical section of the velocity in the Gulf of Mexico (a), provided from Shell Offshore Inc., and the snapshots of the numerical solution at $t = 2.2$ for $LOD[\theta = 0]$ (b), $LOD[\theta = 0.25]$ (c), and $LOD[\theta = 0.5]$ (d). A source of 10 Hz Ricker wavelet is located at the center of the top edge ($z = 0$). Since the velocity $c(\mathbf{x}) \in [1.5, 4.5]$ (km/sec), the wavelength varies between 150 and 450 meters. The original velocity model contains $60 \times 40$ cells of $500^2$ square-

feet each; for an accurate numerical simulation, the cells are refined $4 \times 4$ subcells of which the edge length becomes approximately 38.1 meters and therefore 4–12 grid points are chosen per wavelength. As one can see from the figure, the algorithm exhibits similar resolution for different algorithm parameters $\theta$. The solutions show physical reflection on the discontinuities of the velocity; rarefaction, in particular, inside the high-velocity zones; and focusing along the edges of the high-velocity areas. However, we have found from various experiments that the solution has a trend to become smoother as $\theta$ increases. A sharp resolution of wavefronts is often required in Geophysical applications. The explicit second-order method seems still comparable with the implicit stable algorithms for the simulation of acoustic waves in 2-D, although it exhibits nonphysical oscillations unless the time step size is sufficiently small.

### 8.2. Microscale heat transfer

We now check the numerical behavior of the microscale heat transfer model (11)–(12). Set the spatial and time domain as

$$\Omega \times J = (0,1)^2 \times (0,1]$$

and select the parameters $\tau_J = \tau_K = \widehat{C} = 1$ and $K = I$ for the purpose of error analysis. Choose the following function

$$u(x,y,t) = \left(1 + 0.5\sin\left(\pi^2 t\right)\right)\sin(5\pi x)\sin(3\pi y), \quad (x,y,t) \in \Omega \times J \tag{64}$$

as a true solution. One can easily verify that the solution $u$ in (64) satisfies the no-flux boundary condition (12) on the boundary for $t \geqslant 0$ and solves Eq. (11) when the source is given as

$$S(x,y,t) = \pi^2 \sin(5\pi x)\sin(3\pi y)\left(34 + \left(0.5 + 17\pi^2\right)\cos\left(\pi^2 t\right) + \left(17 - 0.5\pi^2\right)\sin\left(\pi^2 t\right)\right).$$

For the examples considered in this subsection, we utilize (62) for the initialization of $u^1$.

Table 2 presents the error and computation time for $LOD[\theta = 0.25]$ and $LOD[\theta = 0.5]$ for the microscale heat transfer (11)–(12) when $T = 1$. As the table shows, the algorithms have a second-order accuracy and $\mathcal{O}(N)$ computation cost. The *LOD* procedure of the three-level algorithm is efficient and accurate for the microscale heat transfer.

Now, we apply our algorithm for a set of real data. Consider a double-layered microscale thin film where a gold layer is on top of a chromium layer [9]. From the top view, the film is a square having a side 1000 μm. Each layer has a thickness of 5.E−02 μm. We consider this example as a 2D problem by setting

$$\Omega = (0, 1000) \times (0, 0.1).$$

Table 3 lists the parameter values for both gold and chromium layers.

In Fig. 4, we begin with an easier example, in order to see the heat propagation in the thin film; we set the domain $\Omega' = (0,4) \times (0,1)$ and the final time $T = 0.03$ ns. Utilize $LOD[\theta = 0.25]$ and choose the spatial grid size $h = 0.02$ and the time step size $\Delta t = 6.\text{E}{-}05$. A line source of 1 is located at the center of the top edge of the domain. The

Table 2
The error and computation time of *LOD* for $\theta = 0.25$ and 0.5, and various grid sizes for the microscale heat transfer

| $(\Delta t, h)$ | $LOD[\theta = 0.25]$ | | $LOD[\theta = 0.5]$ | |
|---|---|---|---|---|
| | $E_\infty(T = 1)$ | CPU | $E_\infty(T = 1)$ | CPU |
| (0.02, 0.02) | 4.40E−02 | 0.52 | 4.55E−02 | 0.51 |
| (0.01, 0.01) | 1.05E−02 | 4.58 | 1.08E−02 | 4.58 |
| (0.005, 0.005) | 2.60E−03 | 37.57 | 2.65E−03 | 37.39 |

Table 3
Parameter values of gold and chromium

| Parameters | Gold | Chromium |
|---|---|---|
| $\widehat{C}$ (kJ/mm$^3$/K) | 2.4897E−03 | 3.21484E−03 |
| $K$ (W/mm/K) | 3.17E−01 | 9.3E−02 |
| $\tau_J$ (ns) | 8.5E−03 | 1.36E−04 |
| $\tau_K$ (ns) | 9.E−02 | 7.86E−03 |

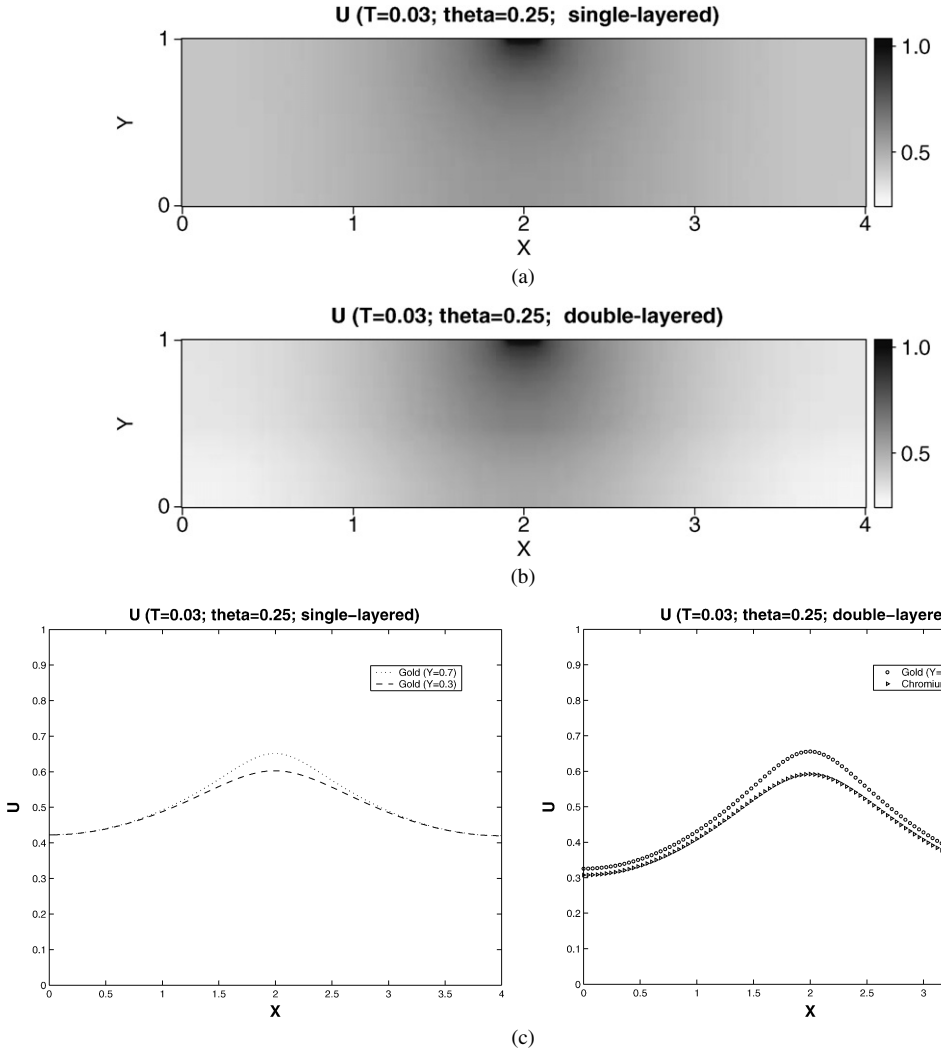


Fig. 4. The numerical solutions (a) of single-layered (Gold), (b) of double-layered (Gold and Chromium), and (c) on $\{y = 0.3\} \cup \{y = 0.7\}$ for single- (left) and double- (right) layered thin films at $t = 0.03$ ns.

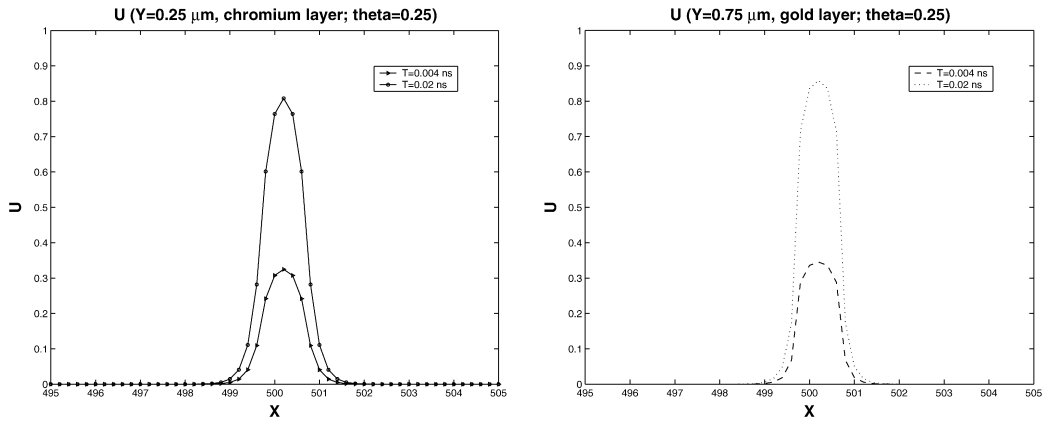


Fig. 5. The numerical solution of double-layered thin film on $\{y = 0.25\}$ (left) and on $\{y = 0.75\}$ (right) when $t = 0.004, 0.02$ ns.

solutions of single-layered and double-layered thin films in the domain $\Omega'$ are compared. Fig. 4(a) is a solution for the single-layered film with gold and Fig. 4(b) shows a solution of double-layered film of gold and chromium. As one can see from the figures, the heat propagates faster in the single-layer than in the double-layer; the diffusion coefficient is smaller in chromium (the lower layer). In Fig. 4(c), the temperatures on $\{y = 0.3\} \cup \{y = 0.7\}$ are compared for the single-layer film (left) and the double-layer film (right). On $\{y = 0.7\}$, the solutions in the single- and double-layered films are almost the same below the source. On the other hand, on $\{y = 0.3\}$, the solution of the double-layered film, (i.e., the solution in the chromium layer) is smaller than the solution in the single-layered film.

Fig. 5 shows the numerical solution on the double-layered microscale thin film of the actual size. Since it is too thin to look at the solution in the whole domain, we select a region near the heat source $\{(x, y)\colon\ x \in [495, 505]\}$ to show the solutions on two slices in it: $\{y = 0.25\}$ (left) and $\{y = 0.75\}$ (right), for $t = 0.004$ ns and $t = 0.02$ ns. Since the material property is different each other on $\{y = 0.25\}$ (chromium) and $\{y = 0.75\}$ (gold), the solution shows a different behavior, but not much.

## 9. Conclusions

We have considered efficient and accurate numerical algorithms for solving viscous and nonviscous wave equations, e.g., the microscale heat transfer and the propagation of acoustic waves. A three-level implicit algorithm, second-order correct in $\Delta t$, has been introduced along with an accurate initialization scheme. The algorithm is compared with conventional two-level algorithms; they turn out to be equivalent for a set of selected algorithm parameters. As a perturbation of the algorithm, a locally one-dimensional (*LOD*) procedure is suggested in order to simulate waves with a great efficiency when the problem is formulated in rectangular/cubic domains and of diagonal diffusion tensors. The *LOD* procedure is shown to have a splitting error not larger than the truncation error; in particular, the splitting error is $\mathcal{O}(\Delta t^4)$, two-order higher than the truncation error, for acoustic waves. Unconditional stability of both the three-level algorithm and its *LOD* procedure has been analyzed. An error analysis is added for viscous waves. An effective strategy for the implementation of the first-order absorbing boundary condition has been suggested for the propagation of acoustic waves. Numerical results are presented to show the accuracy and efficiency of the new algorithms in the case of both acoustic waves and microscale heat transfer.

## Acknowledgement

Seongjai Kim deeply appreciates research management of Shell Offshore Inc. for allowing him to use a real velocity model and to publish related results.

## References

[1] S. Abarbanel, D. Dwoyer, D. Gottlieb, Improving the convergence rate to steady state of parabolic ADI methods, J. Comput. Phys. 67 (1986) 236–239.
[2] W. Ames, D. Lee, Current development in the numerical treatment of ocean acoustic propagation, Appl. Numer. Math. 3 (1987) 25–47.
[3] I.W. Boyd, Laser Processing of Thin Films and Microstructures, Springer, New York, 1989.
[4] S. Candel, A review of numerical methods in acoustic wave propagation, in: Recent Advances in Aeroacoustics, Springer, New York, 1986, pp. 339–410.
[5] R. Chin, T. Manteuffel, J. De Pillis, ADI as preconditioning for solving the convection–diffusion equation, SIAM J. Sci. Statist. Comput. 5 (1984) 291–299.
[6] R. Clayton, B. Engquist, Absorbing boundary conditions for acoustic and elastic wave calculations, Bull. Seismol. Soc. Amer. 67 (1977) 1529–1540.
[7] G. Cohen, P. Joly, N. Tordjman, Construction and analysis of higher order finite elements with mass lumping for the wave equation, in: R. Kleinman, T. Angell, D. Colton, F. Santosa, I. Stakgold (Eds.), Second International Conference on Mathematical and Numerical Aspects of Wave Propagation, SIAM, Philadelphia, PA, 1993, pp. 152–160.
[8] W. Dai, R. Nassar, A finite difference method for solving the heat transport equation at the microscale, Numer. Methods PDEs 15 (1999) 697–708.
[9] W. Dai, R. Nassar, A hybrid finite element-finite difference method for solving three-dimensional heat transport equations in a double-layered thin film with microscale thickness, Numer. Heat Transfer, Part A 38 (2000) 573–588.
[10] 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) 42–65.
[11] J. Douglas Jr, Alternating direction iteration for mildly nonlinear elliptic differential equations, Numer. Math. 3 (1961) 92–98.
[12] J. Douglas Jr, T. Dupont, Alternating direction Galerkin methods on rectangles, in: B. Hubbard (Ed.), Numerical Solution of Partial Differential Equations—II, Academic Press, New York, 1971, pp. 133–214.

[13] J. Douglas Jr, J. Gunn, A general formulation of alternating direction methods Part I. Parabolic and hyperbolic problems, Numer. Math. 6 (1964) 428–453.
[14] J. Douglas Jr, S. Kim, Improved accuracy for locally one-dimensional methods for parabolic equations, Math. Models Methods Appl. Sci. 11 (2001) 1563–1579.
[15] J. Douglas Jr, D. Peaceman, Numerical solution of two-dimensional heat flow problems, Amer. Instit. Chem. Engrg. J. 1 (1955) 505–512.
[16] J. Douglas Jr, S. Kim, H. Lim, An improved alternating-direction method for a viscous wave equation, Contemp. Math. 329 (2003) 99–104.
[17] T. Dupont, $l^2$-estimates for Galerkin methods for second order hyperbolic equations, SIAM J. Numer. Anal. 10 (1973) 880–889.
[18] E. D'yakonov, Difference schemes with split operators for multidimensional unsteady problems, USSR Comput. Math. 3 (1963) 581–607 (English translation).
[19] D. Evans, G. Avelas, The solution of elliptic partial differential equations in $R$–$\theta$ geometry by extrapolated A.D.I. methods, Math. Comput. Simulation 23 (1981) 367–372.
[20] B. Fornberg, Some numerical techniques for Maxwell's equations in different types of geometries, in: Topics in Computational Wave Propagation, Lecture Notes Comput. Sci. Engrg., vol. 31, Springer, Berlin, 2003, pp. 265–299.
[21] C. Johnson, Numerical Solutions of Partial Differential Equations by the Finite Element Method, Cambridge University Press, New York, 1987.
[22] D. Joseph, L. Preziosi, Heat waves, Rev. Modern Phys. 61 (1989) 41–73.
[23] A. Joshi, A. Majumdar, Transient ballistic and diffusive phonon heat transport in thin films, J. Appl. Phys. 74 (1993) 31–39.
[24] C. Kittel, H. Kroemer, Thermal Physics, second ed., W.H. Freeman and Company, New York, 1980.
[25] G. Konyukh, Y. Krivtsov, B. Mikhailenko, Numerical–analytical algorithm of seismic wave propagation in inhomogeneous media, Appl. Math. Lett. 11 (1998) 99–104.
[26] G. Marchuk, Methods of Numerical Mathematics, Springer, New York, 1982.
[27] J. Narayan, V. Godbole, G. White, Laser method for synthesis and processing of continuous diamond films on nondiamond substrates, Science 52 (1991) 416–418.
[28] D. Peaceman, H. Rachford, The numerical solution of parabolic and elliptic equations, J. Soc. Indust. Appl. Math. 3 (1955) 28–41.
[29] T. Qiu, C. Tien, Heat transfer mechanisms during short-pulse laser heating of metals, ASME J. Heat Transfer 115 (1993) 835–841.
[30] A. Sei, W. Symes, Dispersion analysis of numerical wave propagation and its computational consequences, J. Sci. Comput. 10 (1995) 1–27.
[31] G. Starke, Alternating direction preconditioning for nonsymmetric systems of linear equations, SIAM J. Sci. Comput. 15 (1994) 369–384.
[32] D. Tzou, Experimental support for the lagging behavior in heat propagation, J. Thermophysics Heat Transfer 9 (1995) 689–693.
[33] N. Yanenko, Convergence of the method of splitting for the heat conduction equations with variable coefficients, USSR Comput. Math. 3 (1963) 1094–1100 (English translation).
[34] N. Yanenko, The Method of Fractional Steps, Springer, Berlin, 1971 (English translation; originally published in Russian, 1967).
[35] J. Zhang, J. Zhao, Finite difference scheme and iterative solution of two dimensional microscale heat transport equation, J. Comput. Phys. 169 (2001) 1–15.
[36] J. Zhang, J. Zhao, Iterative solution and finite difference approximations to 3D microscale heat transport equation, Math. Comput. Simulation 57 (2001) 387–404.