

Research Article

Higher-Order Splitting Method for Elastic Wave Propagation

Jürgen Geiser

Institut für Mathematik, Humboldt Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

Correspondence should be addressed to Jürgen Geiser, geiser@mathematik.hu-berlin.de

Received 30 July 2008; Accepted 10 November 2008

Recommended by Thomas Witelski

Motivated by seismological problems, we have studied a fourth-order split scheme for the elastic wave equation. We split in the spatial directions and obtain locally one-dimensional systems to be solved. We have analyzed the new scheme and obtained results showing consistency and stability. We have used the split scheme to solve problems in two and three dimensions. We have also looked at the influence of singular forcing terms on the convergence properties of the scheme.

Copyright © 2008 Jürgen Geiser. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

We are motivated by seismological problems that can be studied with higher-order splitting methods scheme. Because of the decomposition, we can save memory and computational time, which is important to study realistic elastic wave propagation. The ideas behind are to split in the spatial directions and obtain locally one-dimensional systems to be solved. Traditionally, using the classical operator splitting methods, we decouple the differential equation into more basic equations. These methods are often not sufficiently stable while also neglecting the physical correlations between the operators. Inspired by the work for the scalar wave equation presented in [1], we devise a fourth-order split scheme for the elastic wave equation. From there on, we are going to develop new efficient methods based on a stable variant by coupling new operators and deriving new strong directions. We are going to examine the stability and consistency analyses for these methods and adopt them to linear acoustic wave equations (seismic waves). Numerical experiments can validate our theoretical results and show the possibility to apply our methods.

The paper is organized as follows. A mathematical model based on the wave equation is introduced in Section 2. The utilized discretization methods are described in Section 3. The splitting method for the scalar and vectorial wave equations are discussed in Section 4 and the stability and consistency analyses are given. We discuss the numerical experiments in

Section 5 with respect to scalar and vectorial problems. Finally, in Section 6, we foresee our future works in the area of splitting and decomposition methods.

2. Mathematical model

The mathematical models are studied in the following subsection. We introduce a scalar and also a vectorial model to distinguish the splitting methods.

2.1. Scalar wave equation

The motivation for the study presented below is coming from a computational simulation of earthquakes, see [2], and the examination of seismic waves, see [3, 4].

We concentrate on the scalar wave equation, see [1], for which the mathematical equations are given by

$$\begin{aligned} \partial_{tt}u &= \mathbf{D}\nabla\cdot\nabla u, \quad \text{in } \Omega, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}), \quad u_t(\mathbf{x}, 0) = u_1(\mathbf{x}), \quad \text{in } \Omega. \end{aligned} \tag{2.1}$$

The unknown function $u = u(\mathbf{x}, t)$ is considered to be in $\Omega \times (0, T) \subset \mathbb{R}^d \times \mathbb{R}$, where the spatial dimension is given by d .

For three dimensions, we define the diffusion tensor as

$$\mathbf{D} = \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{pmatrix} \in \mathbb{R}^{3,+} \times \mathbb{R}^{3,+}, \tag{2.2}$$

which describes the wave propagation. Further, the diffusion tensor \mathbf{D} is given anisotropic, with $D_1, D_2, D_3 \in \mathbb{R}^+$ for $D_1, D_2 \geq D_3$. The functions $u_0(\mathbf{x})$ and $u_1(\mathbf{x})$ are the initial conditions for the wave equation.

We deal with the following boundary conditions:

$$\begin{aligned} u(\mathbf{x}, t) &= u_3, \quad \text{Dirichlet boundary condition,} \\ \frac{\partial u(\mathbf{x}, t)}{\partial n} &= 0, \quad \text{Neumann boundary condition,} \\ \mathbf{D}\nabla u(\mathbf{x}, t) &= u_{\text{out}}, \quad \text{Outflow boundary condition,} \end{aligned} \tag{2.3}$$

where all boundary conditions are on $\partial\Omega \times T$.

2.2. Elastic wave propagation

We consider the initial-value problem for the elastic wave equation for constant coefficients, given as

$$\rho \partial_{tt} \mathbf{U} = \mu \nabla^2 \mathbf{U} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{U}) + \mathbf{f}, \quad (2.4a)$$

$$\mathbf{U}(t = 0, \mathbf{x}) = \mathbf{g}_0(\mathbf{x}), \quad (2.4b)$$

$$\partial_t \mathbf{U}(t = 0, \mathbf{x}) = \mathbf{g}_1(\mathbf{x}), \quad (2.4c)$$

where $\mathbf{U} \equiv \mathbf{U}(\mathbf{x}, t)$ is the displacement vector with components $(u, v)^T$ or $(u, v, w)^T$ in two and three dimensions, \mathbf{f} , \mathbf{g}_0 , and \mathbf{g}_1 are known initial functions, and finally $\mathbf{x} = (x, y, z)^T$. This equation is commonly used to simulate seismic events.

In seismology, it is common to use spatially singular forcing terms which can look like

$$\mathbf{f} = \mathbf{F} \delta(\mathbf{x}) g(t), \quad (2.5)$$

where \mathbf{F} is a constant vector. A numeric method for (2.4a) needs to approximate the Dirac function correctly in order to achieve full convergence.

3. Discretization methods

In this section, we discuss the discretization methods, both for time and space, to construct higher order methods. Because of the combination of both discretization, we can further show also higher-order methods for the splitting schemes, see also [1].

3.1. Discretization of the scalar equation

At first, we underly finite difference schemes for the time and spatial discretization.

For the classical wave equation, this discretization is the well-known discretization in time and space.

Based on this discretization, the time is discretized as

$$U_{tt,i} = \frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{\Delta t^2}, \quad (3.1)$$

$$U(t^n) = u(\mathbf{x}, t), \quad U_i(t^n) = u_t(\mathbf{x}, t),$$

where the index i refers to the space point x_i and $\Delta t = t^{n+1} - t^n$ is the time step. We apply finite difference methods for the spatial discretization. The spatial terms and the initial conditions are given as

$$U_{xx,n} = \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}, \quad (3.2)$$

$$U(t^n) = u(\mathbf{x}, t), \quad U_i(t^n) = u_t(\mathbf{x}, t),$$

where the index n refers to the time t_n and $\Delta x = x_{i+1} - x_i$ is the grid width.

Then the two-dimensional equation,

$$\begin{aligned} u_{tt} &= D_1 u_{xx} + D_2 u_{yy} \quad \text{in } \Omega, \\ u(x, y, 0) &= u_0(x, y), \quad u_t(x, y, 0) = u_1(x, y), \\ u(x, y, t) &= u_2 \quad \text{on } \partial\Omega, \end{aligned} \quad (3.3)$$

is discretized with the unconditionally stable implicit η -method, see [5],

$$\begin{aligned} &\frac{U_{i,j}^{n+1} - 2U_{i,j}^n + U_{i,j}^{n-1}}{\Delta t^2} \\ &= \frac{D_1}{\Delta x^2} (\eta(U_{i+1,j}^{n+1} - 2U_{i,j}^{n+1} + U_{i-1,j}^{n+1}) + (1-2\eta)(U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n) + \eta(U_{i+1,j}^{n-1} - 2U_{i,j}^{n-1} + U_{i-1,j}^{n-1})) \\ &\quad + \frac{D_2}{\Delta y^2} (\eta(U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n) + (1-2\eta)(U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n) + \eta(U_{i,j+1}^{n-1} - 2U_{i,j}^{n-1} + U_{i,j-1}^{n-1})), \end{aligned} \quad (3.4)$$

where Δx and Δy are the grid width in x and y and $0 \leq \eta \leq 1$. The initial conditions are given by $U(x, y, t^n) = u(x, y, t^n)$ and $U(x, y, t^{n-1}) = u(x, y, t^n) - \Delta t u_t(x, y, t^n)$.

These discretization schemes are adopted to the operator splitting schemes.

On the finite differences grid, Δt corresponds to the time step, and h_x, h_y are the grid sizes in the different spatial directions. The time $n\Delta t$ is denoted by t^n , and i, j refer to the spatial coordinates of the grid point (ih_x, jh_y) . Let \mathbf{u}^n denote the grid function on the time level n , and $u_{i,j}^n$ be the specific value of \mathbf{u}^n at point i, j .

In Section 3.2, we describe the traditional splitting methods for the wave equation.

3.2. Discretization of the vectorial equation

One of the first practical difference scheme with central differences used everywhere was introduced in [3]. To save space we exemplify it and some newer schemes in two dimensions first.

If we discretize uniformly in space and time on the unit square, we get a grid with grid points $x_j = jh, y_k = kh, t_n = n\Delta t$, where $h > 0$ is the spatial grid size and Δt the time step. Defining the grid function $\mathbf{U}_{j,k}^n = \mathbf{U}(x_j, y_k, t_n)$, the basic explicit scheme is

$$\rho \frac{\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}}{\Delta t^2} = \mathcal{M}_2 \mathbf{U}_{j,k}^n + \mathbf{f}_{j,k}^n, \quad (3.5)$$

where \mathcal{M}_2 is a difference operator

$$\mathcal{M}_2 = \begin{pmatrix} (\lambda + 2\mu)D^{x^2} + \mu D^{y^2} & (\lambda + \mu)D_0^x D_0^y \\ (\lambda + \mu)D_0^x D_0^y & (\lambda + 2\mu)D^{y^2} + \mu D^{x^2} \end{pmatrix}, \quad (3.6)$$

and we use the standard difference operator notation:

$$D_+^x v_{j,k} = \frac{1}{h}(v_{j+1,k} - v_{j,k}), \quad D_-^x v_{j,k} = D_+^x v_{j-1,k}, \quad D_0^x = \frac{1}{2}(D_+^x + D_-^x), \quad D^{x^2} = D_+^x D_-^x. \quad (3.7)$$

\mathcal{M}_2 is a second-order difference approximation of the right-hand side operator of (2.4a). This explicit scheme is stable for time steps satisfying [6]

$$\Delta t < \frac{h}{\sqrt{\lambda + 3\mu}}. \quad (3.8)$$

Replacing \mathcal{M}_2 with \mathcal{M}_4 , a fourth-order difference operator given by

$$\mathcal{M}_4 = \begin{pmatrix} (\lambda + 2\mu) \left(1 - \frac{h^2}{12} D^{x^2}\right) D^{x^2} + \mu \left(1 - \frac{h^2}{12} D^{y^2}\right) D^{y^2} \\ (\lambda + \mu) \left(1 - \frac{h^2}{6} D^{x^2}\right) D_0^x \left(1 - \frac{h^2}{6} D^{y^2}\right) D_0^y \\ (\lambda + \mu) \left(1 - \frac{h^2}{6} D^{x^2}\right) D_0^x \left(1 - \frac{h^2}{6} D^{y^2}\right) D_0^y \\ (\lambda + 2\mu) \left(1 - \frac{h^2}{12} D^{y^2}\right) D^{y^2} + \mu \left(1 - \frac{h^2}{12} D^{x^2}\right) D^{x^2} \end{pmatrix}, \quad (3.9)$$

and using the modified equation approach to eliminate the lower-order error terms in the time difference [6], we obtain the explicit fourth-order scheme

$$\rho \frac{\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}}{\Delta t^2} = \mathcal{M}_4 \mathbf{U}_{j,k}^n + \mathbf{f}_{j,k}^n + \frac{\Delta t^2}{12} (\mathcal{M}_2^2 \mathbf{U}_{j,k}^n + \mathcal{M}_2 \mathbf{f}_{i,j}^n + \partial_{tt} \mathbf{f}_{i,j}^n), \quad (3.10)$$

where \mathcal{M}_2^2 is a second-order approximation to the squared right-hand side operator in (2.4a). As it only needs to be second-order accurate, \mathcal{M}_2^2 has the same extent in space as \mathcal{M}_4 and no more grid points are used. This scheme has the same time step restriction as (3.8).

In [1] the following implicit scheme for the scalar wave equation was introduced:

$$\rho \frac{\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}}{\Delta t^2} = \mathcal{M}_4 (\theta \mathbf{U}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}) + \theta \mathbf{f}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{f}_{j,k}^n + \theta \mathbf{f}_{j,k}^{n-1}. \quad (3.11)$$

When $\theta = 1/12$, the error of this scheme is fourth-order in time and space. For this θ value, it is, however, only conditionally stable, allowing a time step approximately 45% larger than (3.8) (for $\theta \in [0.25, 0.5]$, it is unconditionally stable).

In order to make it competitive with the explicit scheme (3.10), we provide an operator split version of the implicit scheme (3.11). This is made complicated by the presence of the mixed derivative terms that couple different coordinate directions.

4. Higher-order splitting method for the wave equations

In this section, we discuss the splitting methods for the wave equations. The higher order results as a combination between the spatial and time discretization method and the weighting factors in the splitting schemes.

4.1. Traditional splitting methods for the scalar wave equation

Our classical method is based on the splitting method of [5, 7].

The classical splitting methods alternating direction methods (ADIs) are based on the idea of computing the different directions of the given operators. Each direction is computed independently by solving more basic equations. The result combines all the solutions of the elementary equations. So we obtain more efficiency by decoupling the operators.

The classical splitting method for the wave equation starts from

$$\begin{aligned} \partial_{tt}u(t) &= (A + B + C)u(t) + f(t), \quad t \in (t^n, t^{n+1}), \\ u(t^n) &= u_0, \quad u'(t^n) = u_1, \end{aligned} \quad (4.1)$$

where the initial functions u_0 and u_1 are given. We could also apply for u_1 that $u'(t^n) = (u(t^n) - u(t^{n-1})) / \Delta t + O(\Delta t) = u_1$. Consequently, we have $u(t^{n-1}) \approx u_0 - \Delta t u_1$. The right-hand side $f(t)$ is given as a force term.

The spatial discretization terms are given by

$$A = \frac{\partial^2}{\partial x^2}, \quad B = \frac{\partial^2}{\partial y^2}, \quad C = \frac{\partial^2}{\partial z^2}, \quad (4.2)$$

where the approximated discretization is given by

$$\begin{aligned} Au(x, y, z) &\approx \frac{u(x + \Delta x, y, z) - 2u(x, y, z) + u(x - \Delta x, y, z)}{\Delta x^2}, \\ Bu(x, y, z) &\approx \frac{u(x, y + \Delta y, z) - 2u(x, y, z) + u(x, y - \Delta y, z)}{\Delta y^2}, \\ Cu(x, y, z) &\approx \frac{u(x, y, z + \Delta z) - 2u(x, y, z) + u(x, y, z - \Delta z)}{\Delta z^2}. \end{aligned} \quad (4.3)$$

We could decouple the equation into 3 simpler equations obtaining a method of second order:

$$\begin{aligned}\tilde{\tilde{u}} - 2u(t^n) + u(t^{n-1}) &= \Delta t^2 A(\eta \tilde{\tilde{u}} + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 B u(t^n) + \Delta t^2 C u(t^n) \\ &\quad + \Delta t^2 (\eta f(t^{n+1}) + (1 - 2\eta)f(t^n) + \eta f(t^{n-1})),\end{aligned}\tag{4.4a}$$

$$\begin{aligned}\tilde{u} - 2u(t^n) + u(t^{n-1}) &= \Delta t^2 A(\eta \tilde{u} + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 B(\eta \tilde{u} + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 C u(t^n) + \Delta t^2 (\eta f(t^{n+1}) + (1 - 2\eta)f(t^n) + \eta f(t^{n-1})),\end{aligned}\tag{4.4b}$$

$$\begin{aligned}u(t^{n+1}) - 2u(t^n) + u(t^{n-1}) &= \Delta t^2 A(\eta \tilde{\tilde{u}} + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 B(\eta \tilde{u} + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 C(\eta u(t^{n+1}) + (1 - 2\eta)u(t^n) + \eta u(t^{n-1})) \\ &\quad + \Delta t^2 (\eta f(t^{n+1}) + (1 - 2\eta)f(t^n) + \eta f(t^{n-1})),\end{aligned}\tag{4.4c}$$

where the result is given as $u(t^{n+1})$ with the initial conditions $u(t^n) = u_0$, $u(t^{n-1}) = u_0 - \Delta t u_1$, and $\eta \in (0, 0.5)$. A fully coupled method is given for $\eta = 0$ and for $0 < \eta \leq 1$ the decoupled method consists of a composition of explicit and implicit Euler methods.

We have to compute the first equation (4.4a) and get the result $\tilde{\tilde{u}}$ that is a further initial condition for the second equation (4.4b); after whose computation we obtain \tilde{u} . In the third equation (4.4c), we have to put \tilde{u} as a further initial condition and get the result $u(t^{n+1})$.

The underlying idea consists of the approximation of the pairwise operators:

$$\begin{aligned}\Delta t^2 A \eta (\tilde{\tilde{u}} - 2u(t^n) + u(t^{n-1})) &\approx 0, \\ \Delta t^2 B \eta (\tilde{u} - 2u(t^n) + u(t^{n-1})) &\approx 0,\end{aligned}\tag{4.5}$$

which we can raise to second order.

4.2. Boundary splitting method for the scalar wave equation

The time-dependent boundary conditions also have to be taken into account for the splitting method. Let us consider the three-operator example with the equations

$$\begin{aligned}\partial_{tt}u(t) &= (A + B + C)u(t) + h(t), \quad t \in (t^n, t^{n+1}), \\ u(t^n) &= g(t), \quad u'(t^n) = f(t),\end{aligned}\tag{4.6}$$

where $A = D_1(x, y, z)(\partial^2/\partial x^2)$, $B = D_2(x, y, z)(\partial^2/\partial y^2)$, and $C = D_3(x, y, z)(\partial^2/\partial z^2)$ are the spatial operators. The wave-propagation functions are as follows:

$$D_1(x, y, z), D_2(x, y, z), D_3(x, y, z) : \mathbb{R}^3 \longrightarrow \mathbb{R}^+. \quad (4.7)$$

Hence, for 3 operators, we have the following second-order splitting method:

$$\begin{aligned} \tilde{u} - 2\tilde{u}(t^n) + \tilde{u}(t^{n-1}) &= \Delta t^2 A(\eta\tilde{u} + (1 - 2\eta)\tilde{u}(t^n) + \eta\tilde{u}(t^{n-1})) \\ &\quad + \Delta t^2 B\tilde{u}(t^n) + \Delta t^2 C\tilde{u}(t^n) \\ &\quad + \Delta t^2 (\eta h(t^{n+1}) + (1 - 2\eta)h(t^n) + \eta h(t^{n-1})), \\ \tilde{u} - 2\tilde{u}(t^n) + \tilde{u}(t^{n-1}) &= \Delta t^2 A(\eta\tilde{u} + (1 - 2\eta)\tilde{u}(t^n) + \eta\tilde{u}(t^{n-1})) \\ &\quad + \Delta t^2 B(\eta\tilde{u} + (1 - 2\eta)\tilde{u}(t^n) + \eta\tilde{u}(t^{n-1})) \\ &\quad + \Delta t^2 C\tilde{u}(t^n) + \Delta t^2 (\eta h(t^{n+1}) + (1 - 2\eta)h(t^n) + \eta h(t^{n-1})), \\ u(t^{n+1}) - 2\hat{u}(t^n) + \hat{u}(t^{n-1}) &= \Delta t^2 A(\eta\tilde{u} + (1 - 2\eta)\hat{u}(t^n) + \eta\hat{u}(t^{n-1})) \\ &\quad + \Delta t^2 B(\eta\tilde{u} + (1 - 2\eta)\hat{u}(t^n) + \eta\hat{u}(t^{n-1})) \\ &\quad + \Delta t^2 C(\eta u(t^{n+1}) + (1 - 2\eta)\hat{u}(t^n) + \eta\hat{u}(t^{n-1})) \\ &\quad + \Delta t^2 (\eta h(t^{n+1}) + (1 - 2\eta)h(t^n) + \eta h(t^{n-1})), \end{aligned} \quad (4.8)$$

where the result is given as $u(t^{n+1})$.

The boundary values are given by the following.

- (1) Dirichlet values. We have to use the same boundary values for all 3 equations.
- (2) Neumann values. We have to decouple the values into the different directions:

$$\frac{\partial \tilde{u}}{\partial n} = 0 \quad (4.9)$$

is split in

$$\frac{\partial \tilde{u}}{\partial x} n_x + \frac{\partial \tilde{u}}{\partial y} n_y + \frac{\partial \tilde{u}}{\partial z} n_z = 0; \quad (4.10)$$

$$\frac{\partial \tilde{u}}{\partial n} = 0$$

is split in

$$\begin{aligned} \frac{\partial \tilde{u}}{\partial x} n_x + \frac{\partial \tilde{u}}{\partial y} n_y + \frac{\partial \tilde{u}}{\partial z} n_z &= 0; \\ \frac{\partial u(t^{n+1})}{\partial n} &= 0 \end{aligned} \quad (4.11)$$

is split in

$$\frac{\partial \tilde{u}}{\partial x} n_x + \frac{\partial \tilde{u}}{\partial y} n_y + \frac{\partial u^{n+1}}{\partial z} n_z = 0. \quad (4.12)$$

(3) Outflowing values, we have to decouple the values into the different directions:

$$\mathbf{nD}\nabla\tilde{u} = u_{\text{out}} \quad (4.13)$$

is split in

$$\begin{aligned} D_1 \partial_x \tilde{u} n_x + D_2 \partial_y \tilde{u} n_y + D_3 \partial_z \tilde{u} n_z &= u_{\text{out}}; \\ \mathbf{nD}\nabla\tilde{u} &= u_{\text{out}} \end{aligned} \quad (4.14)$$

is split in

$$\begin{aligned} D_1 \partial_x \tilde{u} n_x + D_2 \partial_y \tilde{u} n_y + D_3 \partial_z \tilde{u} n_z &= u_{\text{out}}; \\ \mathbf{nD}\nabla u^{n+1} &= u_{\text{out}} \end{aligned} \quad (4.15)$$

is split in

$$D_1 \partial_x \tilde{u} n_x + D_2 \partial_y \tilde{u} n_y + D_3 \partial_z u^{n+1} n_z = u_{\text{out}}, \quad (4.16)$$

where \mathbf{n} is the outer normal vector and the anisotropic diffusion \mathbf{D} , see (2.2), is the parameter matrix to the wave-propagations.

We have the following initial conditions for the three equations:

$$\begin{aligned} u(t^n) &= u_0, \\ u(t^{n-1}) &= u_0 - \Delta t u_1 + \frac{\Delta t^2}{2} ((A+B+C)u_0 + f(t^n)) + O(\Delta t^3), \\ u(t^{n-1}) &= u_0 - \Delta t u_1 + \frac{\Delta t^2}{2} \left((A+B+C) \left(u_0 - \frac{\Delta t}{3} u_1 + \frac{\Delta t^2}{12} (A+B+C)u_0 \right) \right) \\ &\quad + \frac{\Delta t^2}{2} f(t^n) - \frac{\Delta t^3}{6} \frac{\partial f(t^n)}{\partial t} + \frac{\Delta t^4}{24} \frac{\partial^2 f(t^n)}{\partial t^2} + O(\Delta t^5). \end{aligned} \quad (4.17)$$

Remark 4.1. By solving the two or three splitting steps, it is important to mention that each solution \tilde{u} , \tilde{u} , and u is corrected only once by using the boundary conditions.

Otherwise, an “overdoing” of the boundary conditions takes place.

4.3. LOD method: locally one-dimensional method for the scalar wave equation

In the following, we introduce the LOD method as an improved splitting method while using prestepping techniques.

The method was discussed in [1] and is given by

$$\begin{aligned} u^{n+1,0} - 2u^n + u^{n-1} &= \Delta t^2(A + B)u^n, \\ u^{n+1,1} - u^{n+1,0} &= \Delta t^2\eta A(u^{n+1} - 2u^n + u^{n-1}), \\ u^{n+1} - u^{n+1,1} &= \Delta t^2\eta B(u^{n+1} - 2u^n + u^{n-1}), \end{aligned} \quad (4.18)$$

where $\eta \in (0.0, 0.5)$ and A, B are the spatial discretized operators.

If we eliminate the intermediate values in (4.18), we obtain

$$u^{n+1} - 2u^n + u^{n-1} = \Delta t^2(A + B)(\eta u^{n+1} + (1 - 2\eta)u^n + \eta u^{n-1}) + B_\eta(u^{n+1} - 2u^n + u^{n-1}), \quad (4.19)$$

where $B_\eta = \eta^2 \Delta t^4(AB)$ and thus $B_\eta(u^{n+1} - 2u^n + u^{n-1}) = O(\Delta t^6)$.

So, we obtain a higher-order method.

Remark 4.2. For $\eta \in (0.25, 0.5)$, we have unconditionally stable methods and for higher order we use $\eta = 1/12$. Then, for sufficiently small time steps, we get a conditionally stable splitting method.

4.4. Stability and consistency analysis for the LOD method of the scalar wave equation

The consistency of the fourth-order splitting method is given in the next theorem.

Hence, we assume discretization orders of $O(h^p)$, $p = 2, 4$, for the discretization in space, where $h = h_x = h_y$ is the spatial grid width.

Then we obtain the following consistency result for our method (4.18).

Theorem 4.3. *The consistency of the LOD method is given by*

$$u_{tt} - Au - (\overline{\partial_{tt}}u - \tilde{A}u) = O(\Delta t^2), \quad (4.20)$$

where $\overline{\partial_{tt}}$ is a second-order discretization in time and \tilde{A} is the discretized fourth-order spatial operator.

Proof. We add (4.18) and obtain the following, see also [1]:

$$\overline{\partial_{tt}u^n} - \tilde{A}(\theta u^{n+1} + (1-2\theta)u^n + \theta u^{n-1}) - \tilde{B}(u^{n+1} - 2u^n + u^{n-1}) = 0, \quad (4.21)$$

where $\tilde{B} = \theta^2 \Delta t^2 \tilde{A}_1 \tilde{A}_2$.

Therefore, we obtain a splitting error of $\tilde{B}(u^{n+1} - 2u^n + u^{n-1})$.

Sufficient smoothness assumed that we have $(u^{n+1} - 2u^n + u^{n-1}) = O(\Delta t^2)$, and we obtain $\tilde{B}(u^{n+1} - 2u^n + u^{n-1}) = O(\Delta t^4)$.

Thus, we obtain a fourth-order method if the spatial operators are also discretized as fourth-order terms. \square

The stability of the fourth-order splitting method is given in the following theorem.

Theorem 4.4. *The stability of the method is given by*

$$\|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^+ u^n\|^2 + \mathcal{P}^+(u^n, \theta) \leq \|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^+ u^0\|^2 + \mathcal{P}^+(u^0, \theta), \quad (4.22)$$

where $\theta \in [0.25, 0.5]$ and $\mathcal{P}^\pm(u^j, \theta) := \theta(\tilde{A}u^j, u^j) + \theta(\tilde{A}u^{j\pm 1}, u^{j\pm 1}) + (1-2\theta)(\tilde{A}u^j, u^{j\pm 1})$.

Proof. We have to proof the theorem for a test function $\overline{\partial_t}u^n$, where $\overline{\partial_t}$ denotes the central difference.

For $j \geq 1$, we have

$$((1 - \Delta t^2 \tilde{B}) \overline{\partial_{tt}u^j}, \overline{\partial_t}u^j) + (\tilde{A}(\theta u^{j+1} - (1-2\theta)u^j + \theta u^{j-1}), \overline{\partial_t}u^j) = 0. \quad (4.23)$$

Multiplying with Δt and summarizing over j yield

$$\sum_{j=1}^n ((1 - \Delta t^2 \tilde{B}) \overline{\partial_{tt}u^j}, \overline{\partial_t}u^j) \Delta t + (\tilde{A}(\theta u^{j+1} - (1-2\theta)u^j + \theta u^{j-1}), \overline{\partial_t}u^j) \Delta t = 0. \quad (4.24)$$

We can derive the identities

$$\begin{aligned} ((1 - \Delta t^2 \tilde{B}) \overline{\partial_{tt}u^j}, \overline{\partial_t}u^j) \Delta t &= \frac{1}{2} \|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^+ u^j\|^2 - \frac{1}{2} \|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^- u^j\|^2, \\ (\tilde{A}(\theta u^{j+1} - (1-2\theta)u^j + \theta u^{j-1}), \overline{\partial_t}u^j) \Delta t &= \frac{1}{2} (\mathcal{P}^+(u^j, \theta) - \mathcal{P}^-(u^j, \theta)), \end{aligned} \quad (4.25)$$

and obtain the result

$$\|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^+ u^n\|^2 + \mathcal{P}^+(u^n, \theta) \leq \|(1 - \Delta t^2 \tilde{B})^{1/2} \partial_t^+ u^0\|^2 + \mathcal{P}^+(u^0, \theta), \quad (4.26)$$

see also the idea of [1]. \square

Remark 4.5. For $\theta = 1/12$, we obtain a fourth-order method.

To compute the error of the local splitting, we have to use the multiplier $\tilde{A}_1 \tilde{A}_2$, thus for large constants, we have an unconditional small time step.

Remark 4.6. (1) The unconditional stable version of LOD method is given for $\theta \in [0.25, 0.5]$.
 (2) The truncation error is $O(\Delta t^2 + h^p)$, $p \geq 2$, for $\theta \in [0, 0.5]$.
 (3) $\theta = 1/12$, we have a fourth-order method in time $O(\Delta t^2 + h^p)$, $p \geq 2$.
 (4) $\theta = 0$ we have a second-order explicit scheme.
 (5) For the stable version of the LOD method, the CFL condition should be taken into account for all $\theta \in [0, 0.25]$ with $CFL = \Delta t^2 / \Delta x_{\max}^2 D_{\max}$, where x_{\max} are the maximal spatial step and D_{\max} are the maximal wave-propagation parameter in space.

In the next subsections, we discuss the higher-order splitting methods for the vectorial wave equations.

4.5. Higher-order splitting method for the vectorial wave equation

In the following, we present a fourth-order splitting method based on the basic scheme (3.11). We split the operator \mathcal{M}_4 into three parts: \mathcal{M}_{xx} , \mathcal{M}_{yy} , and \mathcal{M}_{xy} , where we have

$$\begin{aligned} \mathcal{M}_{xx} &= \begin{pmatrix} (\lambda + 2\mu) \left(1 - \frac{h^2}{12} D^{x^2}\right) D^{x^2} & 0 \\ 0 & \mu \left(1 - \frac{h^2}{12} D^{x^2}\right) D^{x^2} \end{pmatrix}, \\ \mathcal{M}_{yy} &= \begin{pmatrix} \mu \left(1 - \frac{h^2}{12} D^{y^2}\right) D^{y^2} & 0 \\ 0 & (\lambda + 2\mu) \left(1 - \frac{h^2}{12} D^{y^2}\right) D^{y^2} \end{pmatrix}, \\ \mathcal{M}_{xy} &= \mathcal{M}_4 - \mathcal{M}_{xx} - \mathcal{M}_{yy}. \end{aligned} \tag{4.27}$$

Our proposed split method has the following steps:

$$\begin{aligned} (1) \quad \rho \frac{\mathbf{U}_{j,k}^* - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}}{\Delta t^2} &= \mathcal{M}_4 \mathbf{U}_{j,k}^n + \theta \mathbf{f}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{f}_{j,k}^n + \theta \mathbf{f}_{j,k}^{n-1}, \\ (2) \quad \rho \frac{\mathbf{U}_{j,k}^{**} - \mathbf{U}_{j,k}^*}{\Delta t^2} &= \theta \mathcal{M}_{xx} (\mathbf{U}_{j,k}^{**} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) + \frac{\theta}{2} \mathcal{M}_{xy} (\mathbf{U}_{j,k}^* - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}), \\ (3) \quad \rho \frac{\mathbf{U}_{j,k}^{n+1} - \mathbf{U}_{j,k}^{**}}{\Delta t^2} &= \theta \mathcal{M}_{yy} (\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) + \frac{\theta}{2} \mathcal{M}_{xy} (\mathbf{U}_{j,k}^{**} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}). \end{aligned} \tag{4.28}$$

Here, the first step is explicit, while the second and third steps treat the derivatives along the coordinate axes implicitly and the mixed derivatives explicitly. This is similar to how the mixed case is handled for parabolic problems [8].

Note that each of the equation systems that needs to be solved in steps (2) and (3) is actually two decoupled tri-diagonal systems that can be solved independently.

4.6. Stability and consistency of the higher-order splitting method of the vectorial wave equations

The consistency of the fourth-order splitting method is given in the following theorem.

We have for all sufficiently smooth functions $\mathbf{U}(\mathbf{x}, t)$ the following discretization order:

$$\mathcal{M}_4 \mathbf{U} = \mu \nabla^2 \mathbf{U} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{U}) + \mathcal{O}(h^4). \quad (4.29)$$

Furthermore, the split operators are also discretized with the same order of accuracy.

Then, we obtain the following consistency result for the split method (4.28).

Theorem 4.7. *The split method has a splitting error which for smooth solutions \mathbf{U} is $\mathcal{O}(\Delta t^4)$, where it is assumed that $\Delta t = \mathcal{O}(h)$.*

Proof. We assume in the following that $\mathbf{f} = (0, 0)^T$. We add (4.28) and obtain, like in the scalar case [1], the following result for the discretized equations

$$D_+^t D_-^t \mathbf{U}_{j,k}^n - \mathcal{M}_4(\theta \mathbf{U}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}) - \mathcal{N}_{4,\theta}(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) = \mathbf{0}, \quad (4.30)$$

where $\mathcal{N}_{4,\theta} = \theta^2 \Delta t^2 (\mathcal{M}_{xx} \mathcal{M}_{yy} + \mathcal{M}_{xx} \mathcal{M}_{xy} + \mathcal{M}_{xy} \mathcal{M}_{yy}) + \theta^3 \Delta t^4 \mathcal{M}_{xx} \mathcal{M}_{yy} \mathcal{M}_{xy}$.

We, therefore, obtain a splitting error of $\mathcal{N}_{4,\theta}(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1})$.

For sufficient smoothness, we have $(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) = \mathcal{O}(\Delta t^2)$ and we obtain $\mathcal{N}_{4,\theta}(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) = \mathcal{O}(\Delta t^4)$.

It is important that the influence of the mixed terms can be also be discretized as fourth-order method and, therefore, the terms are canceled in the proof. \square

For the stability, we have to denote an appropriate norm, which is in our case the $L_2(\Omega)$.

In the following, we introduce the notation of the norms.

Remark 4.8. For our system, we extend the L_2 -norm as

$$\|\mathbf{U}\|_{L_2}^2 = (\mathbf{U}, \mathbf{U})_{L_2} = \int_{\Omega} \mathbf{U}^2 dx, \quad (4.31)$$

where $\mathbf{U}^2 = u^2 + v^2$ or $\mathbf{U}^2 = u^2 + v^2 + w^2$ in two and three dimensions.

Remark 4.9. For a discrete grid function $\mathbf{U}_{j,k}^n$, the L_2 -norm is given as

$$\int_{\Omega} (\mathbf{U}_{j,k}^n)^2 dx = \Delta x^2 \sum_{j,k}^M U_{j,k}^n, \quad (4.32)$$

where Δx is the uniform grid length in x and y , M is the number of grid points in x and y . Further, $U_{j,k}^n$ is the solution at grid point (j, k) and at time t^n .

Remark 4.10. The matrix

$$\mathcal{N}_{4,\theta} = \theta^2 \Delta t^2 (\mathcal{M}_{xx} \mathcal{M}_{yy} + \mathcal{M}_{xx} \mathcal{M}_{xy} + \mathcal{M}_{xy} \mathcal{M}_{yy}) + \theta^3 \Delta t^4 \mathcal{M}_{xx} \mathcal{M}_{yy} \mathcal{M}_{xy}, \quad (4.33)$$

where \mathcal{M}_{xx} , \mathcal{M}_{yy} , and \mathcal{M}_{xy} are symmetrical and positive-definite matrices, therefore, the matrix $\mathcal{N}_{4,\theta}$ is also symmetrical and positive-definite.

Furthermore, we can estimate the norms and define a weighted norm, see [9, 10].

Remark 4.11. The energy norm is given as

$$(\mathcal{N}_{4,\theta} \mathbf{U}, \mathbf{U})_{L_2} = \int_{\Omega} (\mathcal{N}_{4,\theta} \mathbf{U} \mathbf{U}) dx. \quad (4.34)$$

Consequently, we can denote

$$\|\mathcal{N}_{4,\theta} \mathbf{U}\| \leq \omega \|\mathbf{U}\| \quad \forall \mathbf{U} \in H^d, \quad (4.35)$$

where $\omega \in \mathbb{R}^+$ is the weight and $\mathcal{N}_{4,\theta}$ is bounded. d is the dimension, and H is Sobolev-space, see [11].

The stability of the fourth-order splitting method is given in the following theorem.

Theorem 4.12. *Let $\theta \in [0.25, 0.5]$, then the implicit time-stepping algorithm, see (3.5), and the split procedure, see (4.28), are unconditionally stable. One can estimate the split procedure iteratively as*

$$\|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_+^t \mathbf{U}_{j,k}^n\|^2 + \rho^+(\mathbf{U}_{j,k}^n, \theta) \leq \|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_+^t \mathbf{U}_{j,k}^0\|^2 + \rho^+(\mathbf{U}_{j,k}^0, \theta), \quad (4.36)$$

where one has $\rho^\pm(\mathbf{U}_{j,k}^n, \theta) := \theta(\mathcal{M}_4 \mathbf{U}_{j,k}^n, \mathbf{U}_{j,k}^n) + \theta(\mathcal{M}_4 \mathbf{U}_{j,k}^{n\pm 1}, \mathbf{U}_{j,k}^{n\pm 1}) + (1 - 2\theta)(\mathcal{M}_4 \mathbf{U}_{j,k}^n, \mathbf{U}_{j,k}^{n\pm 1})$ and $\rho^\pm \geq 0$ for $\theta \in [0.25, 0.5]$. Further, $1 - \Delta t^2 \tilde{\omega} \in \mathbb{R}^+$ is the factor for the weighted norm $(\mathcal{D} - \Delta t^2 \mathcal{N}_{4,\theta}) \mathbf{U} \leq \tilde{\omega} \mathbf{U}$ for all $\mathbf{U} \in H^d$.

We have to prove the iterative estimate for the split procedure and the proof is given as follows.

Proof. To obtain an energy estimate for the scheme, we multiply with a test-function $D_0^t \mathbf{U}_{j,k}^n$.

The following result is given for the discretized equations, see also (4.30):

$$(\mathcal{D} - \Delta t^2 \mathcal{N}_{4,\theta}) D_+^t D_-^t \mathbf{U}_{j,k}^n - \mathcal{M}_4 (\theta \mathbf{U}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}) = \mathbf{0}. \quad (4.37)$$

So for $n \geq 1$, we can rewrite (4.37) for the stability proof as follows:

$$((\mathcal{D} - \Delta t^2 \mathcal{N}_{4,\theta}) D_+^t D_-^t \mathbf{U}_{j,k}^n, D_0^t \mathbf{U}_{j,k}^n) - (\mathcal{M}_4 (\theta \mathbf{U}_{j,k}^{n+1} + (1 - 2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}), D_0^t \mathbf{U}_{j,k}^n) = \mathbf{0}. \quad (4.38)$$

Multiplying with Δt and summarizing over the time levels, we obtain

$$\sum_n ((\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}) D_+^t D_-^t \mathbf{U}_{j,k}^n, D_0^t \mathbf{U}_{j,k}^n) \Delta t - \sum_n (\mathcal{M}_4(\theta \mathbf{U}_{j,k}^{n+1} + (1-2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}), D_0^t \mathbf{U}_{j,k}^n) \Delta t = 0, \quad (4.39)$$

for each term of the sum, one can derive the following identities. So for $\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}$, we have

$$\begin{aligned} ((\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}) D_+^t D_-^t \mathbf{U}_{j,k}^n, D_0^t \mathbf{U}_{j,k}^n) \Delta t &= \frac{1}{2} ((\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}) (D_+^t - D_-^t) \mathbf{U}_{j,k}^n, (D_+^t + D_-^t) \mathbf{U}_{j,k}^n) \\ &= \int_{\Omega} ((\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}) (D_+^t - D_-^t))^T (D_+^t + D_-^t) \mathbf{U}_{j,k}^n dx \\ &\leq (1 - \Delta t^2 \tilde{\omega}) \int_{\Omega} (D_+^t \mathbf{U}_{j,k}^n)^2 (D_-^t \mathbf{U}_{j,k}^n)^2 dx, \end{aligned} \quad (4.40)$$

where the operator $\mathcal{D} - \Delta t^2 \mathcal{M}_{4,\theta}$ is symmetric and positive-definite and we can apply the weighted norm, see Remark 4.11 and [11].

We obtain the following result:

$$(1 - \Delta t^2 \tilde{\omega}) \int_{\Omega} (D_+^t \mathbf{U}_{j,k}^n)^2 (D_-^t \mathbf{U}_{j,k}^n)^2 dx = \frac{1}{2} \|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_+^t \mathbf{U}_{j,k}^n\|^2 - \frac{1}{2} \|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_-^t \mathbf{U}_{j,k}^n\|^2. \quad (4.41)$$

Further, for $-\mathcal{M}_4$, we have

$$(-\mathcal{M}_4(\theta \mathbf{U}_{j,k}^{n+1} + (1-2\theta) \mathbf{U}_{j,k}^n + \theta \mathbf{U}_{j,k}^{n-1}), D_0^t \mathbf{U}_{j,k}^n) \Delta t = \frac{1}{2} (\rho^+(\mathbf{U}_{j,k}^n, \theta) - \rho^-(\mathbf{U}_{j,k}^n, \theta)). \quad (4.42)$$

Due to the result of the operators:

$$\rho^-(\mathbf{U}_{j,k}^n, \theta) = \rho^+(\mathbf{U}_{j,k}^{n-1}, \theta), \quad D_-^t \mathbf{U}_{j,k}^n = D_+^t \mathbf{U}_{j,k}^{n-1}, \quad (4.43)$$

we can recursively derive the following result:

$$\|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_+^t \mathbf{U}_{j,k}^n\|^2 + \rho^+(\mathbf{U}_{j,k}^n, \theta) \leq \|(1 - \Delta t^2 \tilde{\omega})^{1/2} D_+^t \mathbf{U}_{j,k}^0\|^2 + \rho^+(\mathbf{U}_{j,k}^0, \theta), \quad (4.44)$$

where for $\theta \in [0.25, 0.5]$, we have $\rho^+(\mathbf{U}_{j,k}^n, \theta) \geq 0$ for all $n \in \mathbb{N}^+$, and, therefore, we have the unconditional stability. The scalar proof is also presented in the work of [1]. \square

Remark 4.13. For $\theta = 1/12$, the split method is fourth-order accurate in time and space.

See the following theorem.

Theorem 4.14. *One obtains a fourth-order accurate scheme in time and space for the split method, see (4.28), when $\theta = 1/12$. That reads*

$$D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{1}{12} \mathcal{M}_4(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) + \mathcal{M}_4 \mathbf{U}_{j,k}^n + \mathcal{N}_{4,\theta}(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) = \mathbf{0}, \quad (4.45)$$

where \mathcal{M}_4 is a fourth-order discretization scheme in space.

The proof is given as follows.

Proof. We consider the following Taylor-expansion:

$$\partial_{tt} \mathbf{U}_{j,k}^n = D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{\Delta t^2}{12} \partial_{tttt} \mathbf{U}_{j,k}^n + \mathcal{O}(\Delta t^4). \quad (4.46)$$

Furthermore, we have

$$\partial_{tttt} \mathbf{U}_{j,k}^n \approx \mathcal{M}_4 \partial_{tt} \mathbf{U}_{j,k}^n, \quad (4.47)$$

and we can rewrite (4.46) as

$$\begin{aligned} \partial_{tt} \mathbf{U}_{j,k}^n &\approx D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{\Delta t^2}{12} \mathcal{M}_4 \partial_{tt} \mathbf{U}_{j,k}^n + \mathcal{O}(\Delta t^4) \\ &\approx D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{\Delta t^2}{12} \mathcal{M}_4(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) + \mathcal{O}(\Delta t^4). \end{aligned} \quad (4.48)$$

So the fourth-order time-stepping algorithm can be formulated as

$$D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{1}{12} \mathcal{M}_4(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) - \mathcal{M}_4 \mathbf{U}_{j,k}^n = \mathbf{0}. \quad (4.49)$$

The split method (4.28) becomes

$$D_+^t D_-^t \mathbf{U}_{j,k}^n - \frac{1}{12} \mathcal{M}_4(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) - \mathcal{M}_4 \mathbf{U}_{j,k}^n - \mathcal{N}_{4,1/12}(\mathbf{U}_{j,k}^{n+1} - 2\mathbf{U}_{j,k}^n + \mathbf{U}_{j,k}^{n-1}) = \mathbf{0}, \quad (4.50)$$

and we obtain a fourth-order split scheme (cf. the scalar case [1]). \square

Remark 4.15. As follows from Theorem 4.14, we obtain a fourth-order in time for $\theta = 1/12$. For the stability analysis, the method is conditional stable for $\theta \in (0, 0.25)$. So the splitting method will not restrict our stability condition for the fourth-order method with $\theta = 1/12$.

Our theoretical results are verified by the following numerical examples.

5. Numerical experiments

In this section, we present the numerical experiments for scalar and vectorial wave equations. The benefit of the splitting methods is discussed.

5.1. Numerical examples of the scalar wave equation

To test examples for the scalar wave equations, we discussed numerical experiments, which are based on analytical solutions. We present various boundary conditions and also spatial-dependent propagation functions. The benefit of the splitting method to reduce the computational amount is discussed with respect to the approximation errors.

5.1.1. Test example 1: problem with analytical solution and Dirichlet boundary condition

We deal with a two-dimensional example with constant coefficients where we can derive an analytical solution:

$$\begin{aligned}
 \partial_{tt}u &= D_1^2 \partial_{xx}u + D_2^2 \partial_{yy}u, \quad \text{in } \Omega \times (0, T), \\
 u(x, y, 0) &= u_0(x, y) = \sin\left(\frac{1}{D_1}\pi x\right) \sin\left(\frac{1}{D_2}\pi y\right), \quad \text{on } \Omega, \\
 \partial_t u(x, y, 0) &= u_1(x, y) = 0, \quad \text{on } \Omega, \\
 u(x, y, t) &= \sin\left(\frac{1}{D_1}\pi x\right) \sin\left(\frac{1}{D_2}\pi y\right) \cos(\sqrt{2}\pi t), \quad \text{on } \partial\Omega \times (0, T),
 \end{aligned} \tag{5.1}$$

where the initial conditions can be written as $u(x, y, t^n) = u_0(x, y)$ and $u(x, y, t^{n-1}) = u(x, y, t^{n+1}) = u(x, y, \Delta t)$.

The analytical solution is given by

$$u_{\text{ana}}(x, y, t) = \sin\left(\frac{1}{D_1}\pi x\right) \sin\left(\frac{1}{D_2}\pi y\right) \cos(\sqrt{2}\pi t). \tag{5.2}$$

For the approximation error, we choose the L_1 -norm.

The L_1 -norm is given by

$$\text{err}_{L_1} := \sum_{i,j=1,\dots,m} V_{i,j} |u(x_i, y_j, t^n) - u_{\text{ana}}(x_i, y_j, t^n)|, \tag{5.3}$$

where $u(x_i, y_j, t^n)$ is the numerical and $u_{\text{ana}}(x_i, y_j, t^n)$ is the analytical solution and $V_{i,j} = \Delta x \Delta y$.

Our test examples are organized as follows.

- (1) The non-stiff case. We choose $D_1 = D_2 = 1$ with a rectangle as our model domain $\Omega = [0, 1] \times [0, 1]$. We discretize with $\Delta x = 1/16$ and $\Delta y = 1/16$ and $\Delta t = 1/32$ and choose our parameter η between $0 \leq \eta \leq 1$. The exemplary function values u_{num} and u_{ana} are taken from the center of our domain.

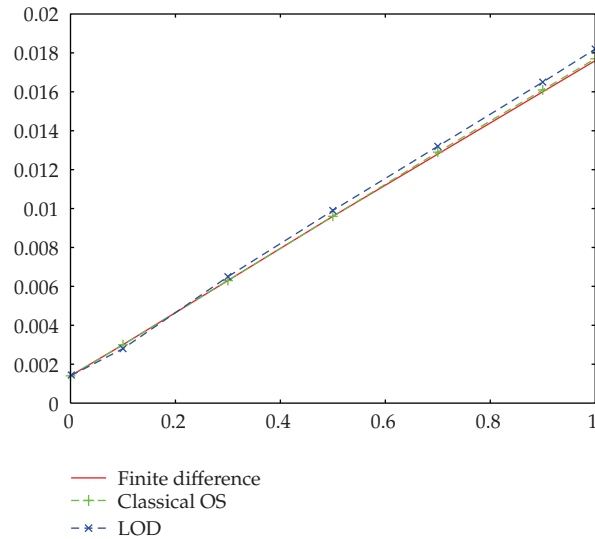


Figure 1: Numerical error for standard and modified methods, with respect to the η parameter and Dirichlet boundary conditions.

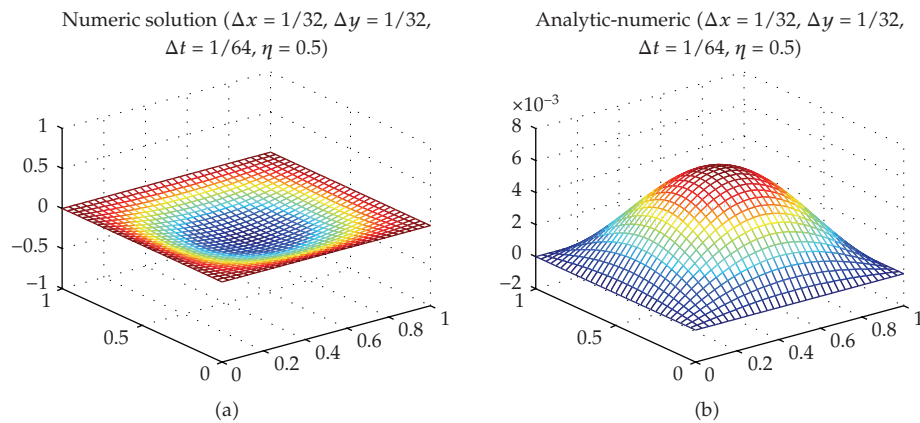


Figure 2: Numerical resolution of the wave equation: numerical approximation (a) and error functions (b) for the Dirichlet boundary conditions ($\Delta x = \Delta y = 1/32$, $\Delta t = 1/64$, $D_1 = 1$, $D_2 = 1$) (classical method).

- (2) The stiff case. We choose $D_1 = D_2 = 0.01$ with a rectangle as our model domain $\Omega = [0, 1] \times [0, 1]$. We discretize with $\Delta x = 1/32$ and $\Delta y = 1/32$ and $\Delta t = 1/64$ and choose our parameter η between $0 \leq \eta \leq 1$. The exemplary function values u_{num} and u_{ana} are taken from the point $(0.5, 0.5625)$.

The experiments are done with the uncoupled standard discretization method (i.e., the finite differences methods for time and space, and with the operator splitting methods, i.e., the classical operator splitting method and the LOD method).

The numerical errors for the non-stiff case with Dirichlet boundary conditions are presented in Figure 1 and their results in Figure 2.

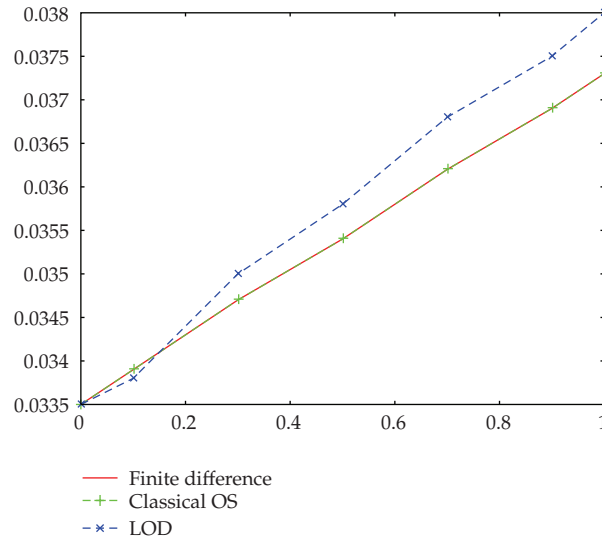


Figure 3: Numerical error for standard and modified methods, with respect to the η parameter, Dirichlet boundary conditions and stiff case ($\Delta x = \Delta y = 1/32$, $\Delta t = 1/64$, $D_1 = 1$, $D_2 = 0.01$).

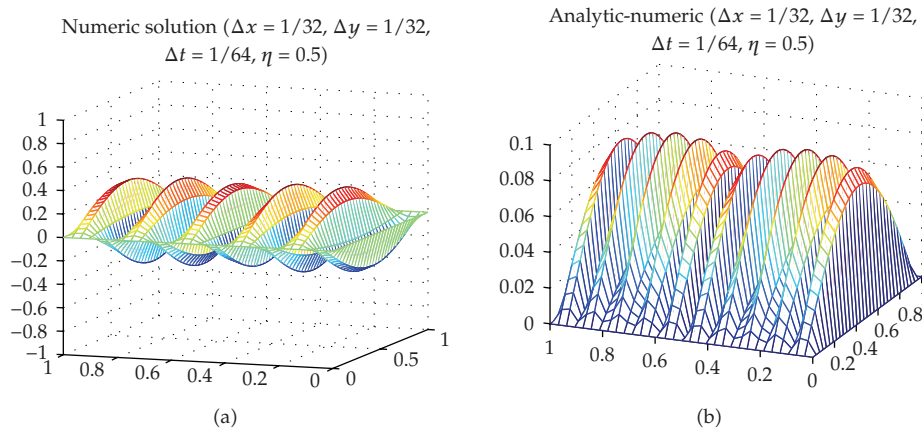


Figure 4: Numerical approximation and error function for the Dirichlet boundary in the stiff case ($\Delta x = \Delta y = 1/32$, $\Delta t = 1/64$, $D_1 = 1$, $D_2 = 0.01$).

The numerical errors for the stiff case with Dirichlet boundary conditions are presented in Figure 3 and their results in Figure 4.

Remark 5.1. In the experiments, we compare the non-splitting with the splitting methods. We obtain nearly the same results and could see improved results for the LOD method, which is for $\eta = 1/12$ a fourth-order method.

In the next test example, we study the Neumann boundary conditions.

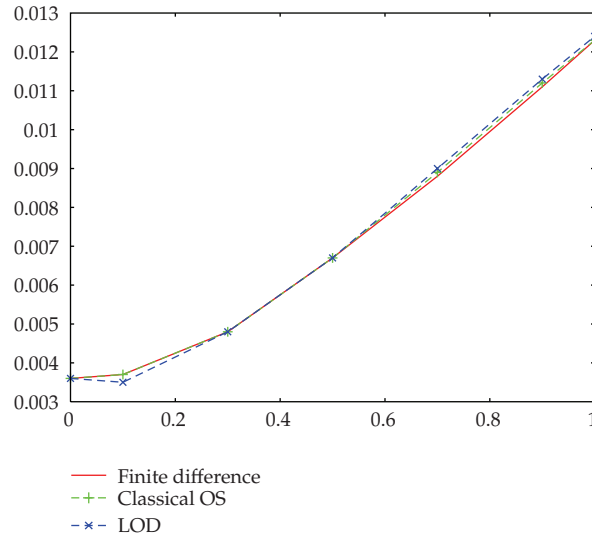


Figure 5: Numerical error for standard and modified methods, with respect to the η parameter and Neumann boundary conditions.

5.1.2. Test example 2: problem with analytical solution and Neumann boundary condition

In this example, we modify our boundary conditions with respect to the Neumann boundary. We deal with our 2D example where we can derive an analytical solution:

$$\begin{aligned}
 \partial_{tt}u &= D_1^2 \partial_{xx}u + D_2^2 \partial_{yy}u, \quad \text{in } \Omega \times (0, T), \\
 u(x, y, 0) &= u_0(x, y) = \sin\left(\frac{1}{D_1}\pi x\right) \sin\left(\frac{1}{D_2}\pi y\right), \quad \text{on } \Omega, \\
 \partial_t u(x, y, 0) &= u_1(x, y) = 0, \quad \text{on } \Omega, \\
 \frac{\partial u(x, y, t)}{\partial n} &= \frac{\partial u_{\text{analy}}(x, y, t)}{\partial n} = 0, \quad \text{on } \partial\Omega \times (0, T),
 \end{aligned} \tag{5.4}$$

where $\Omega = [0, 1] \times [0, 1]$. $D_1 = 1$, $D_2 = 0.5$ and the initial conditions can be written as $u(x, y, t^n) = u_0(x, y)$ and $u(x, y, t^{n-1}) = u(x, y, t^{n+1}) = u(x, y, \Delta t)$.

The analytical solution is given as

$$c_{\text{analy}}(x, y, t) = \sin\left(\frac{1}{D_1}\pi x\right) \sin\left(\frac{1}{D_2}\pi y\right) \cos(\sqrt{2}\pi t). \tag{5.5}$$

We have the same discretization methods as in test example 1.

The numerical errors for the non-stiff case with Neumann boundary conditions are presented in Figure 5 and their results in Figure 6.

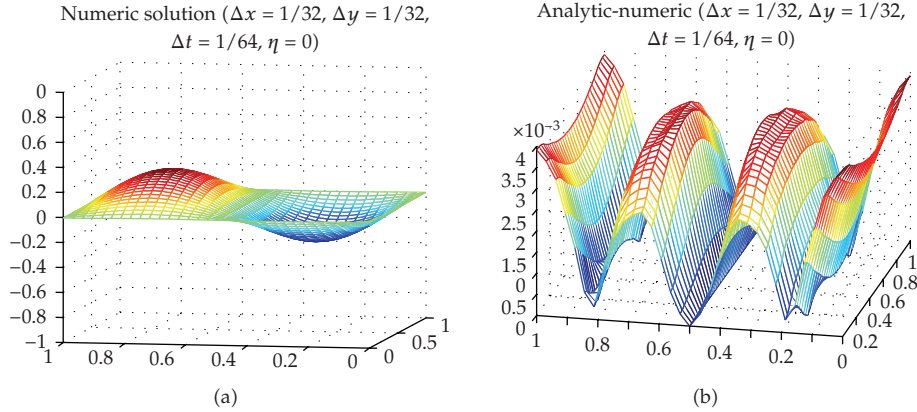


Figure 6: Numerical resolution of the wave equation: numerical approximation (a) and error functions (b) for the Neumann boundary condition ($\Delta x = \Delta y = 1/32, \Delta t = 1/64, D_1 = 1, D_2 = 1$) (classical method).

Remark 5.2. In the experiments, we can obtain the same accuracy as for the Dirichlet boundary conditions. More accurate results are gained by the LOD method with small η . We obtain also stable results in our computations.

5.1.3. Test example 3: spatial-dependent wave equation

In this experiment, we apply our method to the spatial-dependent problem, given by

$$\begin{aligned} \partial_{tt}u &= D_1(x, y)\partial_{xx}u + D_2(x, y)\partial_{yy}u, & \text{in } \Omega \times (0, T), \\ u(x, y, t^n) &= u_0, \quad \partial_t u(x, y, t^n) = u_1, & \text{on } \partial\Omega \times (0, T), \\ u(x, y, t) &= u_2, & \text{on } \partial\Omega \times (0, T), \end{aligned} \tag{5.6}$$

where $D_1(x, y) = 0.1x + 0.01y + 0.01, D_2(x, y) = 0.01x + 0.1y + 0.1$.

To compare the numerical results, we cannot use an analytical solution, that is why in a first prestep we are computing a reference solution. The reference solution is done with the finite difference scheme with fine time and space steps.

Concerning the choice of the time steps, it is important to consider the CFL condition, that is now based on the spatial coefficients.

Remark 5.3. We have assumed the following CFL condition:

$$\Delta t < \frac{0.5 \min(\Delta x, \Delta y)}{\max_{x,y \in \Omega} (D_1(x, y), D_2(x, y))}. \tag{5.7}$$

For the test example, we define our model domain as a rectangle $\Omega = [0, 1] \times [0, 1]$.

The reference solution is obtained by executing the finite differences method and setting $\Delta x = 1/256, \Delta y = 1/256$, and the time step $\Delta t = 1/256 < 0.390625$.

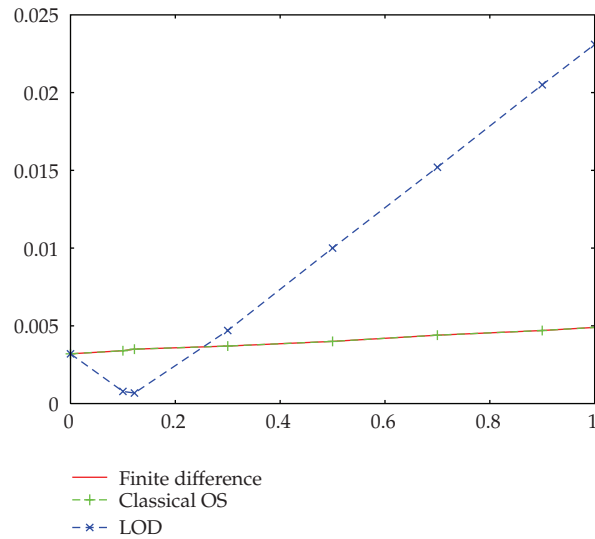


Figure 7: Numerical error for standard and modified methods, with respect to the η parameter, spatial-dependent parameters, and Dirichlet boundary conditions.

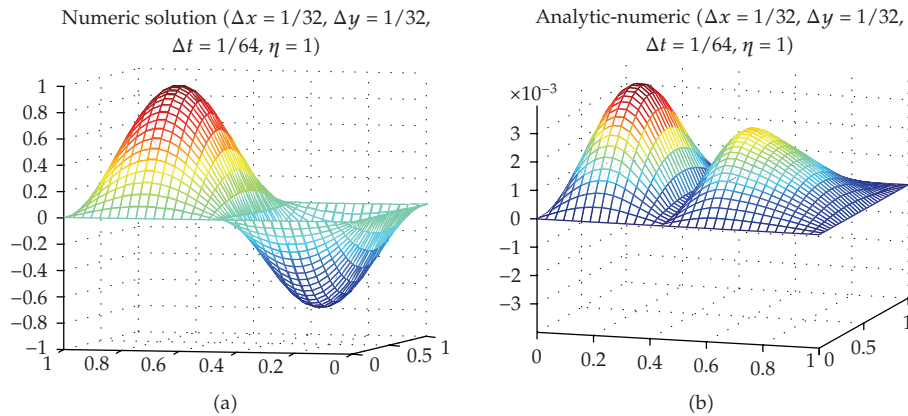


Figure 8: Dirichlet boundary condition: numerical solution and error function for the spatial-dependent test example.

The model domain is given by a rectangle with $\Delta x = 1/16$ and $\Delta y = 1/32$. The time steps are given by $\Delta t = 1/16$ and $0 \leq \eta \leq 1$.

The numerical errors for the spatial-dependent parameters with Dirichlet boundary conditions are presented in Figure 7 and their results in Figure 8.

The numerical errors for the spatial-dependent parameters with Neumann boundary conditions are presented in Figure 9 and their results in Figure 10.

Remark 5.4. In the experiments, we analyze the classical operator splitting and the LOD method and show that the LOD method yields yet more accurate values.

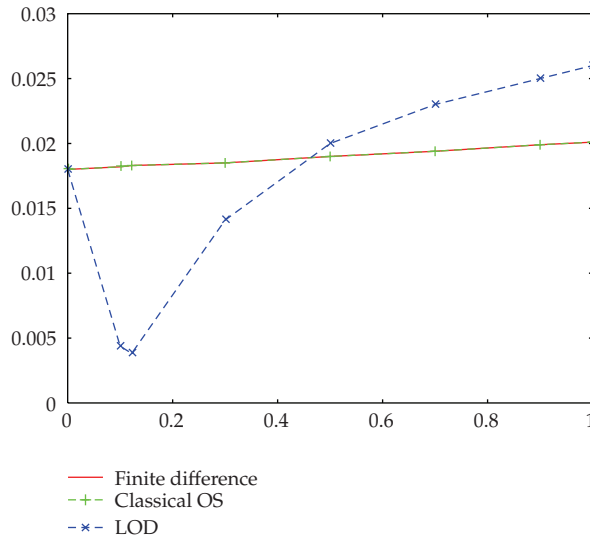


Figure 9: Numerical error for standard and modified methods with respect to the η parameter, spatial-dependent parameters, and Neumann boundary conditions.

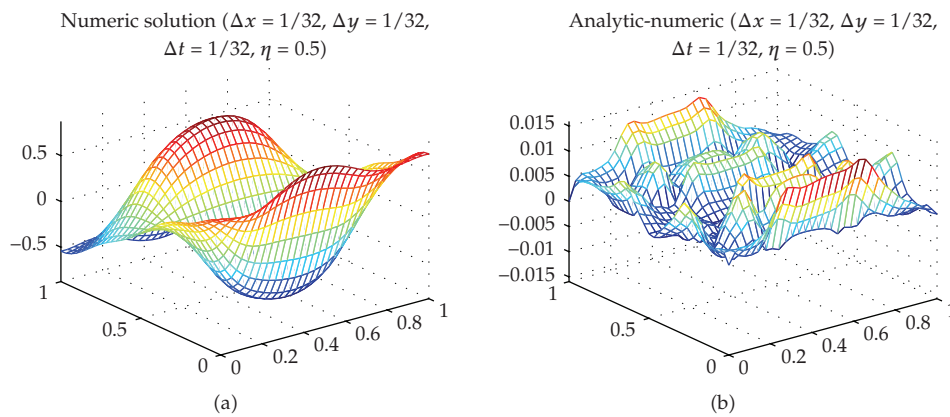


Figure 10: Neumann boundary condition: numerical solution and error function for the spatial-dependent test example.

5.2. Numerical experiments of the elastic wave propagation

To test a fourth-order split method, we have done grid convergence studies on two types of problems. For the first, we impose a smooth solution of (2.4a) using a specific form of the forcing function \mathbf{f} and check the error of the numerical solution against the known solution as the grid is refined.

For the second problem, we use a singular forcing function (2.5), and compare the numerical solution to a solution computed using the Green's function for the free space elastodynamic problem. The convergence for this case is dependent not only on the approximations of time and space derivatives but also on how the Dirac function is approximated.

During the numerical testing, we have observed a need to reduce the allowable time step when the ration of λ over μ became too large. This is likely from the influence of the explicitly treated mixed derivative. For really high ratios (>20), a reduction of 35% was necessary to avoid numerical instabilities.

5.2.1. Initial values and boundary conditions

In order to start the time stepping scheme, we need to know the values at two earlier time levels. Starting at time $t = 0$, we know the value at level $n = 0$ as $\mathbf{U}^0 = \mathbf{g}_0$. The value at level $n = -1$ can be obtained by Taylor expansion as

$$\mathbf{U}^{-1} = \mathbf{U}^0 - \Delta t \partial_t \mathbf{U}^0 + \frac{\Delta t^2}{2} \partial_{tt} \mathbf{U}^0 - \frac{\Delta t^3}{6} \partial_{ttt} \mathbf{U}^0 + \frac{\Delta t^4}{24} \partial_{tttt} \mathbf{U}^0 + O(\Delta t^5), \quad (5.8)$$

where we use

$$\partial_t \mathbf{U}_{j,k}^0 = \mathbf{g}_{1,j,k}, \quad (5.9a)$$

$$\partial_{tt} \mathbf{U}_{j,k}^0 \approx \frac{1}{\rho} (\mathcal{M}_4 \mathbf{g}_{0,j,k}) + \mathbf{f}_{j,k}, \quad (5.9b)$$

$$\partial_{ttt} \mathbf{U}_{j,k}^0 \approx \frac{1}{\rho} (\mathcal{M}_4 \mathbf{g}_{1,j,k}) + \partial_t \mathbf{f}_{j,k}^0, \quad (5.9c)$$

$$\partial_{tttt} \mathbf{U}_{j,k}^0 \approx \frac{1}{\rho} (\mathcal{M}_2^2 \mathbf{g}_{0,j,k}) + \mathcal{M}_4 \mathbf{f}_{j,k}^0 + \partial_{tt} \mathbf{f}_{j,k}^0, \quad (5.9d)$$

and also for (5.9c) and (5.9d),

$$\partial_t \mathbf{f}_{j,k}^0 \approx \frac{\mathbf{f}_{j,k}^1 - \mathbf{f}_{j,k}^{-1}}{2\Delta t}, \quad (5.9e)$$

$$\partial_{tt} \mathbf{f}_{j,k}^0 \approx \frac{\mathbf{f}_{j,k}^1 - 2\mathbf{f}_{j,k}^0 + \mathbf{f}_{j,k}^{-1}}{\Delta t^2}. \quad (5.9f)$$

We are not considering the boundary value problem in this paper and so will not be concerned with constructing proper difference stencils at grid points close to the boundaries of the computational domain. We have simply added a two-point-thick layer of extra-grid points at the boundaries of the domain and assigned the correct analytical solution at all points in the layer every time step.

Remark 5.5. For the Dirichlet boundary conditions, the splitting method (see (4.28)) conserves also the conditions. We can use for the 3 equations (see (4.28)), so for \mathbf{U}^* , \mathbf{U}^{**} , and for \mathbf{U}^{n+1} , the same conditions.

For the Neumann boundary conditions and other boundary conditions of higher order, we have also to split the boundary conditions with respect to the split operators, see [12].

Table 1: Errors in max-norm for decreasing h and smooth analytical solution \mathbf{U}^{true} . Convergence rate indicates a fourth-order convergence for the split scheme.

h	$t = 2$		$e_h = \ \mathbf{U}^n - \mathbf{U}^{\text{true}}\ _\infty$	
	case 1	$\log_2(e_{2h}/e_h)$	case 2	$\log_2(e_{2h}/e_h)$
0.05	1.7683e-07	—	2.5403e-07	—
0.025	1.2220e-08	3.855	2.1104e-08	3.589
0.0125	7.9018e-10	3.951	1.4376e-09	3.876
0.006125	5.0013e-11	3.982	9.2727e-11	3.955

Table 2: Errors in max-norm for decreasing h and smooth analytical solution \mathbf{U}^{true} and using the non-split scheme. Comparing with Table 1, we see that the splitting error is very small for this case.

h	$t = 2, e_h = \ \mathbf{U}^n - \mathbf{U}^{\text{true}}\ _\infty$	
	case 1	case 2
0.05	1.6878e-07	2.4593e-07
0.025	1.1561e-08	2.0682e-08
0.0125	7.4757e-10	1.4205e-09
0.006125	4.8112e-11	9.2573e-11

5.2.2. Test example

For the first test case, we use a forcing function

$$\mathbf{f} = \begin{pmatrix} \sin(t-x)\sin(y) - 2\mu\sin(t-x)\sin(y) - (\lambda + \mu)(\cos(x)\cos(t-y) + \sin(t-x)\sin(y)), \\ \sin(t-y)\sin(x) - 2Vs^2\sin(x)\sin(t-y) - (\lambda + \mu)(\cos(t-x)\cos(y) + \sin(y)\sin(t-y)) \end{pmatrix}^T, \quad (5.10)$$

giving the analytical solution

$$\mathbf{U}^{\text{true}} = (\sin(x-t)\sin(y), \sin(y-t)\sin(x))^T. \quad (5.11)$$

Using the split method we solved (2.4a) on a domain $x \times y = [-11] \times [-11]$ up to $t = 2$. We used two sets of material parameters; for the first case ρ , λ , and μ were all equal to 1, for the second case ρ and μ were 1 and λ was set to 14. Solving on four different grids with a refinement factor of two in each direction between the successive grids we obtained the results shown in Table 1. The errors are measured in the ∞ -norm defined as $\|\mathbf{U}_{j,k}\| = \max(\max_{j,k}|u_{j,k}|, \max_{j,k}|v_{j,k}|)$. As can be seen we get the expected 4th order convergence for problems with smooth solutions.

To check the influence of the splitting error $\mathcal{N}_{4,\theta}$ on the error we solved the same problems using the non-split scheme (3.11). The results are shown in Table 2. The errors are only marginally smaller than for the split scheme.

5.2.3. Singular forcing terms

In seismology and acoustics it is common to use spatially singular forcing terms which can look like

$$\mathbf{f} = \mathbf{F}\delta(\mathbf{x})g(t), \quad (5.12)$$

where \mathbf{F} is a constant direction vector. A numeric method for (2.4a) needs to approximate the Dirac function correctly in order to achieve full convergence. Obviously we cannot expect convergence close to the source as the solution will be singular for two and three dimensional domains.

The analyzes in [13, 14] demonstrate that it is possible to derive regularized approximations of the Dirac function, which result in point wise convergence of the solution away from the sources. Based on these analyzes, we define one 2nd (δ_{h^2}) and one 4th (δ_{h^4}) order regularized approximations of the one dimensional Dirac function,

$$\delta_{h^2}(\tilde{x}) = \frac{1}{h} \begin{cases} 1 + \tilde{x}, & -h \leq \tilde{x} < 0, \\ 1 - \tilde{x}, & 0 \leq \tilde{x} < h, \\ 0, & \text{elsewhere,} \end{cases} \quad (5.13)$$

$$\delta_{h^4}(\tilde{x}) = \frac{1}{h} \begin{cases} 1 + \frac{11}{6}\tilde{x} + \frac{5}{8}\tilde{x}^2 + \frac{1}{6}\tilde{x}^3, & -2h \leq \tilde{x} < -h, \\ 1 + \frac{1}{2}\tilde{x} - \tilde{x}^2 - \frac{1}{2}\tilde{x}^3, & -h \leq \tilde{x} < 0, \\ 1 - \frac{1}{2}\tilde{x} - \tilde{x}^2 + \frac{1}{2}\tilde{x}^3, & 0 \leq \tilde{x} < h, \\ 1 - \frac{11}{6}\tilde{x} + \tilde{x}^2 - \frac{1}{6}\tilde{x}^3, & h \leq \tilde{x} < 2h, \\ 0, & \text{elsewhere,} \end{cases} \quad (5.14)$$

where in the above $\tilde{x} = x/h$. The two and three dimensional Dirac functions are then approximated as $\delta_{h^{2A}}(\tilde{x})\delta_{h^{2A}}(\tilde{y})$ and $\delta_{h^{2A}}(\tilde{x})\delta_{h^{2A}}(\tilde{y})\delta_{h^{2A}}(\tilde{z})$. The chosen time dependence was a smooth function given by

$$g(t) = \begin{cases} \exp\left(\frac{-1}{t(1-t)}\right), & 0 \leq t < 1, \\ 0, & \text{elsewhere,} \end{cases} \quad (5.15)$$

which is C^∞ . Using this forcing function we can compute the analytical solution by integrating the Green's function given in [15] in time. The integration was done using numerical quadrature routines from Matlab. Figures 11 and 12 shows examples of what the errors look like on a radius passing through the singular source at time $t = 0.8$ for different grid sizes h and the two approximations δ_{h^2} and δ_{h^4} . As can be seen the error is smooth and converges a small distance away from the source. However, using δ_{h^2} limits the convergence to 2nd order, while using δ_{h^4} gives the full 4th order convergence away from the singular source. When $t > 1$ the forcing goes to zero and the solution will be smooth everywhere. Table 3 shows the convergence behavior at time $t = 1.1$ for four different grids. Note that the full convergence is achieved even if the lower order δ_{h^2} is used as an approximation for the

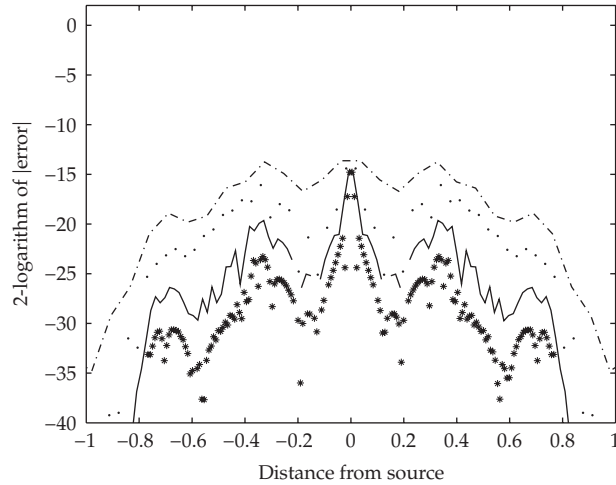


Figure 11: The 2-logarithm of the error along a line going through the source point for a point force located at $x = 0, y = 0$, and approximated in space by (5.14). Note that the error decays as $\mathcal{O}(h^4)$ away from the source, but not near it. The grid sizes were $h = 0.05$ (— · —), 0.025 (· · ·), 0.0125 (—), 0.006125 (*). The numerical quadrature had an absolute error of approximately $10^{-11} \approx 2^{-36}$, so the error cannot be resolved beneath that limit.

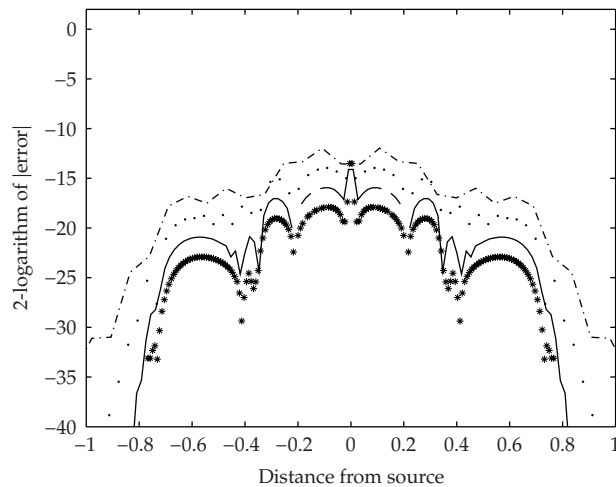


Figure 12: The 2-logarithm of the error along a line going through the source point for a point force located at $x = 0, y = 0$, and approximated in space by (5.13). Note that the error only decays as $\mathcal{O}(h^2)$ away from the source. The grid sizes were $h = 0.05$ (— · —), 0.025 (· · ·), 0.0125 (—), 0.006125 (*).

Dirac function. The convergence rate approaches 4 as we refine the grids, even though the solution was singular up to time $t = 1$.

Remark 5.6. For a two dimensional problem the 4th order explicit method (3.10) can be implemented using approximately 160 *floating point operations* (flops) per grid point. For example, the split method requires approximately 120 flops (first step) plus 2 times 68 flops (second and third step) for a total of 256 flops. This increase of about 60% in the number of flops is somewhat offset by the larger time steps allowed by the split method, especially for

Table 3: Errors in max-norm for decreasing h and analytical solution \mathbf{U}^{true} . Convergence rate approaches 4th order after the singular forcing term goes to zero.

h	$t = 1.1, e_h = \ \mathbf{U}^n - \mathbf{U}^{\text{true}}\ _\infty$	$\log_2(e_{2h}/e_h)$
0.05	1.1788e-04	—
0.025	1.4146e-05	3.0588
0.0125	1.3554e-06	3.3836
0.00625	1.0718e-07	3.6606
0.003125	7.1890e-09	3.8981

smooth material properties, making the two methods roughly comparable in computational cost.

5.3. Three-dimensional test example for the elastic wave propagation

The motivation to compute also the three dimensional elastic wave propagation arose from the need to understand the anisotropy of the different dimensions, see [2]. We apply the three-dimensional model (2.4a)–(2.4c) for our proposed splitting schemes.

5.3.1. The splitting scheme

In three dimensions a 4th order difference approximation of the right hand side operator becomes

$$\mathcal{M}_4 = \begin{pmatrix} (\lambda + 2\mu)\left(1 - \frac{h^2}{12}D^{x^2}\right)D^{x^2} + \mu\left(1 - \frac{h^2}{12}D^{y^2}D^{y^2} + 1 - \frac{h^2}{12}D^{z^2}\right)D^{z^2} \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{x^2}\right)D_0^x\left(1 - \frac{h^2}{6}D^{y^2}\right)D_0^y \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{x^2}\right)D_0^x\left(1 - \frac{h^2}{6}D^{z^2}\right)D_0^z \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{x^2}\right)D_0^x\left(1 - \frac{h^2}{6}D^{y^2}\right)D_0^y \\ (\lambda + 2\mu)\left(1 - \frac{h^2}{12}D^{y^2}\right)D^{y^2} + \mu\left(1 - \frac{h^2}{12}D^{x^2}\right)D^{x^2} + \mu\left(1 - \frac{h^2}{12}D^{z^2}\right)D^{z^2} \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{z^2}\right)D_0^z\left(1 - \frac{h^2}{6}D^{y^2}\right)D_0^y \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{x^2}\right)D_0^x\left(1 - \frac{h^2}{6}D^{z^2}\right)D_0^z \\ (\lambda + \mu)\left(1 - \frac{h^2}{6}D^{y^2}\right)D_0^y\left(1 - \frac{h^2}{6}D^{z^2}\right)D_0^z \\ (\lambda + 2\mu)\left(1 - \frac{h^2}{12}D^{z^2}D^{z^2} + \mu\left(1 - \frac{h^2}{12}D^{x^2}\right)D^{x^2} + 1 - \frac{h^2}{12}D^{y^2}\right)D^{y^2} \end{pmatrix}, \quad (5.16)$$

operating on grid functions $\mathbf{U}_{j,k,l}^n$ defined at grid points x_j, y_k, z_l, t_n similarly to the two dimensional case. We can split \mathcal{M}_4 into six parts; $\mathcal{M}_{xx}, \mathcal{M}_{yy}, \mathcal{M}_{zz}$ containing the three second order directional difference operators, and $\mathcal{M}_{xy}, \mathcal{M}_{yz}, \mathcal{M}_{xz}$ containing the mixed difference operators.

There are a number of different ways we could split this scheme, depending on how we treat the mixed derivative terms. We have chosen to implement the following split scheme in three dimensions:

$$\begin{aligned}
(1) \quad \rho \frac{\mathbf{U}_{j,k,l}^* - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}}{\Delta t^2} &= \mathcal{M}_4 \mathbf{U}_{j,k,l}^n + \theta \mathbf{f}_{j,k,l}^{n+1} + (1 - 2\theta) \mathbf{f}_{j,k,l}^n + \theta \mathbf{f}_{j,k,l}^{n-1} \\
(2) \quad \rho \frac{\mathbf{U}_{j,k,l}^{**} - \mathbf{U}_{j,k,l}^*}{\Delta t^2} &= \theta \mathcal{M}_{xx} (\mathbf{U}_{j,k,l}^{**} - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}) + \frac{\theta}{2} (\mathcal{M}_{xy} + \mathcal{M}_{xz}) (\mathbf{U}_{j,k,l}^* - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}) \\
(3) \quad \rho \frac{\mathbf{U}_{j,k,l}^{***} - \mathbf{U}_{j,k,l}^{**}}{\Delta t^2} &= \theta \mathcal{M}_{xx} (\mathbf{U}_{j,k,l}^{***} - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}) + \frac{\theta}{2} (\mathcal{M}_{xy} + \mathcal{M}_{yz}) (\mathbf{U}_{j,k,l}^{**} - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}) \\
(4) \quad \rho \frac{\mathbf{U}_{j,k,l}^{n+1} - \mathbf{U}_{j,k,l}^{***}}{\Delta t^2} &= \theta \mathcal{M}_{xx} (\mathbf{U}_{j,k,l}^{n+1} - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}) + \frac{\theta}{2} (\mathcal{M}_{xz} + \mathcal{M}_{yz}) (\mathbf{U}_{j,k,l}^{***} - 2\mathbf{U}_{j,k,l}^n + \mathbf{U}_{j,k,l}^{n-1}).
\end{aligned} \tag{5.17}$$

The properties such as splitting error, accuracy, stability, and so forth, for the three dimensional case are similar to the two dimensional case treated in the earlier sections.

5.3.2. Testing the three dimensional scheme

We have done some numerical experiments with the three dimensional scheme in order to test the convergence and stability. We used a forcing

$$\begin{aligned}
\mathbf{f} = & -(-1 + \lambda + 4\mu) \sin(t-x) \sin(y) \sin(z) - (\lambda + \mu) \cos(x) (2 \sin(t) \sin(y) \sin(z) + \cos(t) \sin(y+z)), \\
& -(-1 + \lambda + 4\mu) \sin(x) \sin(t-y) \sin(z) - (\lambda + \mu) \cos(y) (2 \sin(t) \sin(x) \sin(z) + \cos(t) \sin(x+z)), \\
& -(\lambda + \mu) \cos(t-y) \cos(z) \sin(x) \\
& - \sin(y) ((\lambda + \mu) \cos(t-x) \cos(z) + (-1 + \lambda + 4\mu) \sin(x) \sin(t-z)) \mathbf{T},
\end{aligned} \tag{5.18}$$

giving the analytical solution

$$\mathbf{U}^{\text{true}} = (\sin(x-t) \sin(y) \sin(z), \sin(y-t) \sin(x) \sin(z), \sin(z-t) \sin(x) \sin(y)) \mathbf{T}. \tag{5.19}$$

As earlier we tested this for a number of different grid sizes. Using the same two sets of material parameters as for the two dimensional case we ran up until $t = 2$ and checked the max error for all components of the solution. The results are given in Table 4. We have also tested the three dimensional scheme using singular forcing functions approximated using (5.13) and (5.14). The results are very similar to the two dimensional case and we have therefore omitted them here.

6. Conclusion

We have presented time splitting methods for the scalar and vectorial wave equation. The contributions of this article concerns the higher order splitting methods, based on LOD

Table 4: Errors in max-norm for decreasing h and smooth analytical solution \mathbf{U}^{true} . Convergence rate indicates 4th order convergence for the three dimensional split scheme.

h	$t = 2$		$e_h = \ \mathbf{U}^n - \mathbf{U}^{\text{true}}\ _\infty$	
	case 1	$\log_2(e_{2h}/e_h)$	case 2	$\log_2(e_{2h}/e_h)$
0.1	4.2986e-07	—	1.8542e-06	—
0.05	3.5215e-08	3.61	1.3605e-07	3.77
0.025	3.0489e-09	3.53	8.0969e-09	4.07
0.0125	2.0428e-10	3.90	4.7053e-10	4.10

method. We have designed with higher order spatial and time discretization methods the stable higher order splitting methods. The benefit of the splitting methods is due to the different scales and therefore the computational process in decoupling the stiff and the nonstiff operators into different equation is accelerated. The LOD method as a 4th-order method has the advantage of higher accuracy and can be used for such decoupling regards. For our realistic application in elastic wave propagation, the split scheme has been proven to work well in practice for different types of material properties It is comparable to the fully explicit 4th order scheme (3.10) in terms of computational cost, but should be easier to implement, as no difference approximations of higher order operators are needed.

In a next work we will discuss a model in seismology, which have to be more accurate in the boundary conditions. For such models we have to develop higher order stable splitting methods.

References

- [1] S. Kim and H. Lim, "High-order schemes for acoustic waveform simulation," *Applied Numerical Mathematics*, vol. 57, no. 4, pp. 402–414, 2007.
- [2] St. M. Day, J. Bielak, D. Dreger, et al., "Test of 3D elastodynamic codes: final report for lifelines project 1A01," Tech. Rep., Pacific Earthquake Engineering Research Center, Berkeley, Calif, USA, 2001.
- [3] Z. S. Alterman and A. Rotenberg, "Seismic waves in a quarter plane," *Bulletin of the Seismological Society of America*, vol. 59, no. 1, pp. 347–368, 1969.
- [4] A. Ben-Menahem and S. J. Singh, *Seismic Waves and Sources*, Dover, Mineola, NY, USA, 2000.
- [5] G. Fairweather and A. R. Mitchell, "A high accuracy alternating direction method for the wave equation," *Journal of the Institute of Mathematics and Its Applications*, vol. 1, no. 4, pp. 309–316, 1965.
- [6] G. C. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations*, Scientific Computation, Springer, Berlin, Germany, 2002.
- [7] M. Lees, "Alternating direction methods for hyperbolic differential equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 4, pp. 610–616, 1962.
- [8] R. M. Beam and R. F. Warming, "Alternating direction implicit methods for parabolic equations with a mixed derivative," *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 1, pp. 131–159, 1980.
- [9] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1995.
- [10] C. V. Pao, *Nonlinear Parabolic and Elliptic Equations*, Plenum Press, New York, NY, USA, 1992.
- [11] L. C. Evans, *Partial Differential Equations*, vol. 19 of *Graduate Studies in Mathematics*, American Mathematical Society, Providence, RI, USA, 1998.
- [12] G. I. Marchuk, "Some applications of splitting-up methods to the solution of problems in mathematical physics," *Aplikace Matematiky*, vol. 1, pp. 103–132, 1968.
- [13] J. Waldén, "On the approximation of singular source terms in differential equations," *Numerical Methods for Partial Differential Equations*, vol. 15, no. 4, pp. 503–520, 1999.

- [14] A.-K. Tornberg and B. Engquist, "Numerical approximations of singular source terms in differential equations," *Journal of Computational Physics*, vol. 200, no. 2, pp. 462–488, 2004.
- [15] G. Eason, J. Fulton, and I. N. Sneddon, "The generation of waves in an infinite elastic solid by variable body forces," *Philosophical Transactions of the Royal Society of London. Series A*, vol. 248, no. 955, pp. 575–607, 1956.