Author(s): Mönkölä, Sanna; Heikkola, Erkki; Pennanen, Anssi; Rossi, Tuomo

Title: Time-harmonic elasticity with controllability and higher-order discretization methods

Year: 2008

Version:

Please cite the original version:

Mönkölä, S., Heikkola, E., Pennanen, A., & Rossi, T. (2008). Time-harmonic elasticity with controllability and higher-order discretization methods. Journal of Computational Physics, 227(11), 5513-5534. https://doi.org/10.1016/j.jcp.2008.01.054

Time-harmonic elasticity with controllability and higher-order discretization methods

Sanna Mönkölä [a,*] Erkki Heikkola [b] Anssi Pennanen [a]
Tuomo Rossi [a]

[a] *Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland*

[b] *Numerola Oy, P.O. Box 126, FI-40101 Jyväskylä, Finland*

Abstract

The time-harmonic solution of the linear elastic wave equation is needed for a variety of applications. The typical procedure for solving the time-harmonic elastic wave equation leads to difficulties solving large-scale indefinite linear systems. To avoid these difficulties, we consider the original time dependent equation with a method based on an exact controllability formulation. The main idea of this approach is to find initial conditions such that after one time-period, the solution and its time derivative coincide with the initial conditions.

The wave equation is discretized in the space domain with spectral elements. The degrees of freedom associated with the basis functions are situated at the Gauss-Lobatto quadrature points of the elements, and the Gauss-Lobatto quadrature rule is used so that the mass matrix becomes diagonal. This method is combined with the second-order central finite difference or the fourth-order Runge-Kutta time discretization. As a consequence of these choices, only matrix-vector products are needed in time dependent simulation. This makes the controllability method computationally efficient.

Key words: Exact controllability, Navier equation, Spectral element method, Mass lumping, Fourth-order Runge-Kutta
PACS: 46.40

[*] Corresponding author.
Email address: sanna.monkola@jyu.fi (Sanna Mönkölä).

Article published in Journal of Computational Physics 227 (11) (2008) 5513–5534

1 Introduction

The linear theory of elasticity [1] models mechanical properties of solids assuming small deformations. This theory describes several phenomena, including seismic waves in the earth and the passing of ultrasonic waves through materials in order to detect flaws. Efficient solution methods, such as domain decomposition [2,3,4,5], fictitious domain [6], and multigrid [7,8], have been developed for solving the elasticity problem. These methods are typically used when the solution is based directly on the complex-valued time-harmonic equations and low-order finite elements (see e.g. [9,10,11]).

To obtain accurate results, the discretization mesh needs to be adjusted to the wavelength. This leads to a large-scale indefinite linear system for which it is difficult to develop efficient iterative solution methods. Furthermore, the error increases as the wavenumber increases, which makes finding an accurate solution even more challenging. Thus, several modifications of the classical finite element method (FEM) have been used to maintain the accuracy of the solution. For instance, these are methods such as ultra weak variational formulation (UWVF) [12,13], Galerkin generalized least-squares [14], discontinuous enrichment [15], discontinuous Galerkin [16,17], and spectral collocation [18].

Our objective is to solve the problem in a manner that does not require solution of an indefinite linear system. For this purpose we use a controllability algorithm [19,20,21,22]. The main idea of the algorithm is to return to the time dependent wave equation and find initial conditions such that after one time-period the solution and its time derivative coincide with the initial conditions. This is why the controllability problem is reformulated as a least-squares optimization problem. This problem is solved with a preconditioned conjugate gradient algorithm.

We use the spectral element method (SEM) [23,24] for spatial discretization. It provides for a convenient treatment of complex geometries and varying material properties. The basis functions are higher-order Lagrange interpolation polynomials, and the nodes of these functions are placed at the Gauss-Lobatto collocation points. The integrals in the weak form of the equation are evaluated with the corresponding Gauss-Lobatto quadrature formulas. As a consequence of this choice, spectral element discretization leads to diagonal mass matrices significantly improving the computational efficiency of the explicit time-integration used. Moreover, when using higher-order elements, same accuracy is reached with fewer degrees of freedom than when using lower-order finite elements.

The paper is organized as follows. The statement of the problem and some preliminaries are presented in Section 2. We give the exact controllability formulation in Section 3 and discretization schemes in Section 4. We discretize the elastic wave equation in space domain with spectral elements in Section 4.1. Time discretization is accomplished using central finite differences in Section 4.2.1 and the fourth-order Runge-Kutta scheme in Section 4.2.2. In Section 5, we present the control problem and a preconditioned conjugate gradient algorithm that is related to the one developed in [25] for the linear acoustic wave equation. In Sections 5.1 and 5.2, we compute the gradient of the functional, an essential point of the method, using the adjoint state technique. The algebraic multigrid method [26,27] is used for preconditioning the conjugate gradient algorithm in Section 5.3. Numerical experiments concerning the propagation of time-harmonic waves show the efficiency of the algorithm in Section 6.

2 The time-harmonic elastic wave equation

In an elastic, homogeneous, and isotropic body $\Omega \subset \mathbb{R}^2$ with density ρ, the propagation of time-harmonic waves with angular frequency ω is governed by the Navier equation

$$-\omega^2 \rho \mathbf{u} - \nabla \cdot \sigma(\mathbf{u}) = 0 \quad \text{in } \Omega, \tag{1}$$

where $\mathbf{u}$ denotes the displacement field $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^T$, which depends on the spatial variable $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$. The strains are related to the displacements by the linearized strain tensor ϵ, which is defined by

$$\epsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right). \tag{2}$$

The stress tensor $\sigma(\mathbf{u})$ is then expressed as

$$\sigma(\mathbf{u}) = \rho\left(c_p^2 - 2c_s^2\right)(\nabla \cdot \mathbf{u}) I + 2\rho c_s^2 \epsilon(\mathbf{u}). \tag{3}$$

Coefficients c_p and c_s represent the speed of the pressure waves (P-waves) and the speed of the shear waves (S-waves), respectively. The P-waves have a compressional motion, while the motion of the S-waves is perpendicular to the direction of wave propagation [28].

The boundary $\partial\Omega$ surrounding the domain Ω is divided into two distinct parts

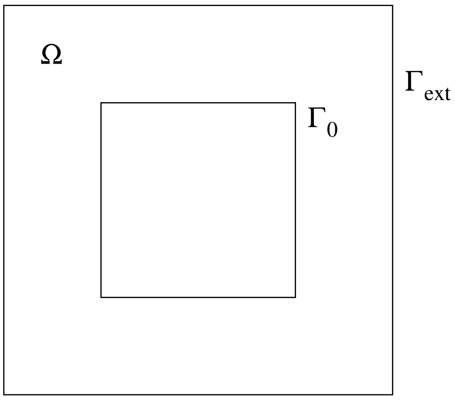


Figure 1. Domain Ω, and the two parts of the boundary $\partial\Omega = \Gamma_0 \cup \Gamma_{\text{ext}}$ of the domain Ω.

(see Figure 1). The boundary Γ_0 is assumed to be rigid,

$$\mathbf{u} = 0 \quad \text{on } \Gamma_0, \tag{4}$$

whereas on the artificial boundary Γ_{ext} we impose the absorbing boundary condition

$$i\omega\rho\mathbf{B}\mathbf{u} + \sigma(\mathbf{u})\mathbf{n} = \mathbf{g}_{\text{ext}}. \tag{5}$$

Here $\mathbf{B}$ is a symmetric positive definite 2×2-matrix [29,30] given by

$$\mathbf{B} = \begin{pmatrix} c_p n_1^2 + c_s n_2^2 & (c_p - c_s)n_1 n_2 \\ (c_p - c_s)n_1 n_2 & c_s n_1^2 + c_p n_2^2 \end{pmatrix}, \tag{6}$$

where $\mathbf{n} = (n_1, n_2)^T$ is the outward pointing normal vector on Γ_{ext}.

3 The exact controllability problem

Solving the time-harmonic equation given by (1), (4), and (5) is equivalent to finding a time-periodic solution for the corresponding time dependent wave equation

$$\rho\frac{\partial^2 \mathbf{U}}{\partial t^2} - \nabla \cdot \sigma(\mathbf{U}) = 0, \qquad \text{in } Q = \Omega \times (0, T), \tag{7}$$

$$\mathbf{U} = 0, \qquad \text{on } \gamma_0 = \Gamma_0 \times (0, T), \tag{8}$$

$$\rho\mathbf{B}\frac{\partial \mathbf{U}}{\partial t} + \sigma(\mathbf{U})\mathbf{n} = \mathbf{G}_{\text{ext}}, \qquad \text{on } \gamma_{\text{ext}} = \Gamma_{\text{ext}} \times (0, T), \tag{9}$$

where $\mathbf{U} = (U_1, U_2)^T$ and $\mathbf{G}_{\text{ext}} = (G_{\text{ext}1}, G_{\text{ext}2})^T$. In addition to the system (7)-(9), we take into account the initial conditions

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{e}_0, \quad \frac{\partial \mathbf{U}(\mathbf{x}, 0)}{\partial t} = \mathbf{e}_1. \tag{10}$$

The time-period corresponding to the angular frequency ω is given by $T = \frac{2\pi}{\omega}$, and the T-periodic solution can be achieved by controlling the initial conditions such that the terminal conditions are equal to the initial conditions (10) at the end of the computation.

For the weak formulation of the problem (7)-(10), we introduce the function space

$$\mathbf{W} = \{\mathbf{y} \in H^1(\Omega) \times H^1(\Omega) \text{ such that } \mathbf{y} = 0 \text{ on } \Gamma_0\}. \tag{11}$$

By multiplying the equation (7) with any test function $\mathbf{v}$ in the space $\mathbf{W}$, using Green's formula, and substituting the boundary conditions, we get the following weak formulation: Find $\mathbf{U}$ satisfying $\mathbf{U}(t) \in \mathbf{W}$ for any $t \in [0, T]$ and

$$\int\limits_{\Omega} \rho \frac{\partial^2 \mathbf{U}}{\partial t^2} \cdot \mathbf{v}\, dx + \int\limits_{\Omega} \sigma(\mathbf{U}) : \epsilon(\mathbf{v})\, dx + \int\limits_{\Gamma_{\text{ext}}} \rho \mathbf{B} \frac{\partial \mathbf{U}}{\partial t} \cdot \mathbf{v}\, ds = \int\limits_{\Gamma_{\text{ext}}} \mathbf{G}_{\text{ext}} \cdot \mathbf{v}\, ds \tag{12}$$

for any $\mathbf{v} \in \mathbf{W}$ and $t \in [0, T]$.

Introducing the Hilbert space $\mathbf{Z}$ for the initial conditions $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1)^T \in \mathbf{Z}$ by

$$\mathbf{Z} = \mathbf{W} \times L^2(\Omega), \tag{13}$$

we formulate the exact controllability problem as follows: Find initial conditions $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1)^T$ such that the weak formulation (12) holds with the terminal conditions

$$\mathbf{U}(\mathbf{x}, T) = \mathbf{e}_0, \quad \frac{\partial \mathbf{U}(\mathbf{x}, T)}{\partial t} = \mathbf{e}_1. \tag{14}$$

4 Discretization

Discretization methods play a large role in the efficiency of the controllability method. The key factor in developing efficient solution methods is the use of high-order approximations without computationally demanding matrix inversions. We attempt to meet these requirements by using the spectral element [23] method for space discretization.

As, for instance, in [31], we locate the degrees of freedom corresponding to the basis functions at the Gauss-Lobatto integration points of the elements. With the Gauss-Lobatto integration rule, this makes the mass matrices diagonal without reducing the order of accuracy. Thus, the inversion of the mass matrix is a trivial and computationally efficient operation.

Since we have returned to the time dependent wave equation, also time discretization is needed. For time discretization we compare the central finite difference (CD) scheme with the fourth-order accurate Runge-Kutta (RK) method. With respect to the time step Δt, the CD method is second-order accurate, while the RK method is fourth-order accurate. Both methods lead to an explicit time-stepping scheme, and only matrix-vector products are needed in time dependent simulation. These properties are essential for computational efficiency. The drawback is that the schemes need to satisfy the stability condition, which limits the length of the time step. In addition, the computational effort of the RK method is approximately four times that of the central finite difference scheme at each time step.

4.1 Spatial discretization

The physical domain Ω is divided into N_e quadrilateral elements Ω_i, $i = 1, \ldots, N_e$, such that $\Omega = \bigcup_{i=1}^{N_e} \Omega_i$. For the discrete formulation, we define the reference element $\Omega_{\text{ref}} = [0,1]^2$ and affine mappings $\mathcal{G}_i : \Omega_{\text{ref}} \rightarrow \Omega_i$ such that $\mathcal{G}_i(\Omega_{\text{ref}}) = \Omega_i$. We define the finite dimensional space

$$\mathbf{W}_h^r = \{\mathbf{y} = (y_1, y_2) \in \mathbf{W} \text{ such that } y_k|_{\Omega_i} \circ \mathcal{G}_i \in Q^r, k = 1, 2\}, \qquad (15)$$

where Q^r is the set of polynomials of order r in each variable in space. The order $r = 1$ corresponds to bilinear finite elements.

Denoting by $\mathbf{U}(t)$ the global vector containing the nodal values of the displacement $\mathbf{U}(\mathbf{x}, t)$ at time t, we write the semi-discrete equation in the form

$$\mathcal{M}\frac{\partial^2\mathbf{U}(t)}{\partial t^2}+\mathcal{S}\frac{\partial\mathbf{U}(t)}{\partial t}+\mathcal{K}\mathbf{U}(t)=\mathcal{F}, \tag{16}$$

where $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{K}$ are 2×2 block matrices and $\mathcal{F}$ is a block vector as follows:

$$\mathcal{M}=\begin{pmatrix}\mathcal{M}_{11} & 0\\ 0 & \mathcal{M}_{22}\end{pmatrix},\quad \mathcal{S}=\begin{pmatrix}\mathcal{S}_{11} & \mathcal{S}_{12}\\ \mathcal{S}_{21} & \mathcal{S}_{22}\end{pmatrix},\quad \mathcal{K}=\begin{pmatrix}\mathcal{K}_{11} & \mathcal{K}_{12}\\ \mathcal{K}_{21} & \mathcal{K}_{22}\end{pmatrix},\quad \mathcal{F}=\begin{pmatrix}\mathcal{F}_1\\ \mathcal{F}_2\end{pmatrix}.$$

The components of these block forms are

$$\begin{aligned}
(\mathcal{M}_{11})_{ij} &= \int\limits_{\Omega} \rho\varphi_j\varphi_i\,dx,\\
(\mathcal{M}_{22})_{ij} &= \int\limits_{\Omega} \rho\varphi_j\varphi_i\,dx,\\
(\mathcal{S}_{11})_{ij} &= \int\limits_{\Gamma_{ext}} \rho\left(c_p n_1^2+c_s n_2^2\right)\varphi_j\varphi_i\,ds,\\
(\mathcal{S}_{12})_{ij} &= \int\limits_{\Gamma_{ext}} \rho(c_p-c_s)n_1 n_2\varphi_j\varphi_i\,ds,\\
(\mathcal{S}_{21})_{ij} &= \int\limits_{\Gamma_{ext}} \rho(c_p-c_s)n_1 n_2\varphi_j\varphi_i\,ds,\\
(\mathcal{S}_{22})_{ij} &= \int\limits_{\Gamma_{ext}} \rho\left(c_p n_2^2+c_s n_1^2\right)\varphi_j\varphi_i\,ds,\\
(\mathcal{K}_{11})_{ij} &= \int\limits_{\Omega}\left(\rho\left(c_p^2-2c_s^2\right)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_1}+2\rho c_s^2\left(\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_1}+\frac{1}{2}\frac{\partial\varphi_j}{\partial x_2}\frac{\partial\varphi_i}{\partial x_2}\right)\right)dx,\\
(\mathcal{K}_{12})_{ij} &= \int\limits_{\Omega}\left(\rho\left(c_p^2-2c_s^2\right)\frac{\partial\varphi_j}{\partial x_2}\frac{\partial\varphi_i}{\partial x_1}+\rho c_s^2\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}\right)dx,\\
(\mathcal{K}_{21})_{ij} &= \int\limits_{\Omega}\left(\rho\left(c_p^2-2c_s^2\right)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}+\rho c_s^2\frac{\partial\varphi_j}{\partial x_2}\frac{\partial\varphi_i}{\partial x_1}\right)dx,\\
(\mathcal{K}_{22})_{ij} &= \int\limits_{\Omega}\left(\rho\left(c_p^2-2c_s^2\right)\frac{\partial\varphi_j}{\partial x_2}\frac{\partial\varphi_i}{\partial x_2}+2\rho c_s^2\left(\frac{1}{2}\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_1}+\frac{\partial\varphi_j}{\partial x_2}\frac{\partial\varphi_i}{\partial x_2}\right)\right)dx,\\
(\mathcal{F}_1)_i &= \int\limits_{\Gamma_{ext}} \mathbf{G}_{ext1}\varphi_i\,ds,\\
(\mathcal{F}_2)_i &= \int\limits_{\Gamma_{ext}} \mathbf{G}_{ext2}\varphi_i\,ds,
\end{aligned}$$

where $i,j=1,\ldots,\hat{N}$. By $\hat{N}$ we denote the total number of Gauss-Lobatto points in the space discretization, which is the number of degrees of freedom (DOF) in each space variable.

4.2 Time discretization

The time discretization of the semi-discrete equation is performed with the central finite differences (CD) in Section 4.2.1 and with the fourth-order Runge-Kutta (RK) method in Section 4.2.2. Since the mass matrix $\mathcal{M}$ is diagonal, explicit time-stepping with central finite differences or the Runge-Kutta scheme requires only matrix-vector multiplications. After dividing the time interval $[0, T]$ into N time steps, each of size $\Delta t = T/N$, applying the appropriate time discretization into the semidiscretized form (16), and taking into account the initial conditions (10), we obtain matrix form of the fully discrete state equation.

4.2.1 Central finite difference method

The spectral element approximation in space is combined with the standard second-order central finite difference scheme in time by replacing the time derivatives in the semidiscretized form (16) at time $i\Delta t$ by the following approximations

$$\frac{\partial^2 \mathbf{U}(i\Delta t)}{\partial t^2} \approx \frac{\mathbf{U}^{i+1} - 2\mathbf{U}^i + \mathbf{U}^{i-1}}{\Delta t^2}, \quad \frac{\partial \mathbf{U}(i\Delta t)}{\partial t} \approx \frac{\mathbf{U}^{i+1} - \mathbf{U}^{i-1}}{2\Delta t}, \quad i = 0, \dots, N, \tag{17}$$

where $\mathbf{U}^i$ is the vector $\mathbf{U}$ at time $i\Delta t$. Taking into account the initial conditions (10), we obtain the fully discrete state equation, which can be represented in the matrix form

$$s^{\mathrm{CD}}(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e})) = \begin{pmatrix} \mathcal{I} & & & & & \\ \mathcal{C}_0 & \mathcal{M} & & & & \\ \mathcal{B} & \mathcal{C} & \mathcal{D} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mathcal{B} & \mathcal{C} & \mathcal{D} & \\ & & & \mathcal{B} & \mathcal{C} & \mathcal{D} \end{pmatrix} \begin{pmatrix} \mathbf{U}^0 \\ \mathbf{U}^1 \\ \vdots \\ \vdots \\ \mathbf{U}^N \\ \mathbf{U}^{N+1} \end{pmatrix} - \begin{pmatrix} \mathcal{I} & 0 \\ 0 & \Delta t\, \mathcal{B} \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \end{pmatrix} - \Delta t^2 \begin{pmatrix} 0 \\ \frac{1}{2}\mathcal{F}^0 \\ \mathcal{F}^1 \\ \vdots \\ \vdots \\ \mathcal{F}^N \end{pmatrix} = 0, \tag{18}$$

where $\hat{\mathbf{y}} = (\mathbf{U}^0, \mathbf{U}^1, \dots, \mathbf{U}^N, \mathbf{U}^{N+1})^T$ contains the vectors $\mathbf{U}^i$, the initial condition is $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1)^T$, and $\mathcal{F}^i$ is the vector $\mathcal{F}$ at time $t = i\Delta t$. The matrix blocks $\mathcal{C}_0$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are given by the formulas

$$\mathcal{C}_0 = \frac{\Delta t^2}{2}\mathcal{K} - \mathcal{M}, \tag{19}$$

$$\mathcal{D} = \mathcal{M} + \frac{\Delta t}{2}\mathcal{S}, \tag{20}$$

$$\mathcal{C} = \Delta t^2\mathcal{K} - 2\mathcal{M}, \tag{21}$$

$$\mathcal{B} = \mathcal{M} - \frac{\Delta t}{2}\mathcal{S}, \tag{22}$$

while $\mathcal{I}$ is the identity matrix. The form (18) is further used to derive the adjoint state equation in Section 5.

4.2.2 Fourth-order Runge-Kutta method

The state equation (16) can be presented as a system of differential equations

$$\frac{\partial \mathbf{y}}{\partial t} = f(t, \mathbf{y}(t)), \tag{23}$$

where $\mathbf{y} = (\mathbf{U}, \mathbf{V})^T$ is a vector of time-stepping variables $\mathbf{U}$ and $\mathbf{V} = \frac{\partial \mathbf{U}}{\partial t}$, and the function $f(t, y(t)) = (f_1(t, \mathbf{U}, \mathbf{V}), f_2(t, \mathbf{U}, \mathbf{V}))^T$ has components

$$f_1(t, \mathbf{U}, \mathbf{V}) = \mathbf{V}, \tag{24}$$

$$f_2(t, \mathbf{U}, \mathbf{V}) = -\mathcal{M}^{-1}\left(\mathcal{S}\mathbf{V} + \mathcal{K}\mathbf{U} - \mathcal{F}\right). \tag{25}$$

In the fourth-order Runge-Kutta method, the solution $\mathbf{y}$ at the ith time step can be presented as

$$\mathbf{y}^i = \mathbf{y}^{i-1} + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \tag{26}$$

where $\mathbf{y}^i = \left(\mathbf{U}^i, \frac{\partial \mathbf{U}^i}{\partial t}\right)^T$ contains the displacement vector $\mathbf{U}^i$ and its derivative $\mathbf{V}^i = \frac{\partial \mathbf{U}^i}{\partial t}$ at time $t = i\Delta t$, $i = 1, \ldots, N$. The initial condition is given by $\mathbf{y}^0 = \mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1)^T$, and $k_j = (k_{j1}, k_{j2})^T, j = 1, 2, 3, 4$, are the gradient estimates as follows:

$$\begin{pmatrix} k_{11} \\ k_{12} \end{pmatrix} = \begin{pmatrix} \Delta t f_1(i\Delta t, \mathbf{U}^i, \mathbf{V}^i) \\ \Delta t f_2(i\Delta t, \mathbf{U}^i, \mathbf{V}^i) \end{pmatrix}, \tag{27}$$

$$\begin{pmatrix} k_{21} \\ k_{22} \end{pmatrix} = \begin{pmatrix} \Delta t f_1(i\Delta t + \frac{\Delta t}{2}, \mathbf{U}^i + \frac{k_{11}}{2}, \mathbf{V}^i + \frac{k_{12}}{2}) \\ \Delta t f_2(i\Delta t + \frac{\Delta t}{2}, \mathbf{U}^i + \frac{k_{11}}{2}, \mathbf{V}^i + \frac{k_{12}}{2}) \end{pmatrix}, \tag{28}$$

$$\begin{pmatrix} k_{31} \\ k_{32} \end{pmatrix} = \begin{pmatrix} \Delta t f_1(i\Delta t + \frac{\Delta t}{2}, \mathbf{U}^i + \frac{k_{21}}{2} \mathbf{V}^i + \frac{k_{22}}{2}) \\ \Delta t f_2(i\Delta t + \frac{\Delta t}{2}, \mathbf{U}^i + \frac{k_{21}}{2} \mathbf{V}^i + \frac{k_{22}}{2}) \end{pmatrix}, \tag{29}$$

$$\begin{pmatrix} k_{41} \\ k_{42} \end{pmatrix} = \begin{pmatrix} \Delta t f_1(i\Delta t + \Delta t, \mathbf{U}^i + k_{31}, \mathbf{V}^i + k_{32}) \\ \Delta t f_2(i\Delta t + \Delta t, \mathbf{U}^i + k_{31}, \mathbf{V}^i + k_{32}) \end{pmatrix}. \tag{30}$$

In other words, in order to get the gradient estimates (27)-(30), the function f is evaluated at each time step four times by using the formulas (24)-(25) and then the successive approximation of $\mathbf{y}$ is calculated by the formula (26). To make the application of the adjoint equation technique in Section 5 more convenient, we present the fully discrete state equation in the case of the Runge-Kutta time discretization as

$$s^{\mathrm{RK}}(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e})) = \begin{pmatrix} \mathcal{I} & & & & \\ \mathcal{N} & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & \mathcal{N} & \mathcal{I} & \\ & & & \mathcal{N} & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{y}^0 \\ \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^{N-1} \\ \mathbf{y}^N \end{pmatrix} - \begin{pmatrix} \mathcal{I} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \mathbf{e} - \begin{pmatrix} 0 \\ \hat{\mathcal{F}}^1 \\ \vdots \\ \hat{\mathcal{F}}^{N-1} \\ \hat{\mathcal{F}}^N \end{pmatrix} = 0, \tag{31}$$

where $\hat{\mathbf{y}} = (\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^{N-1}, \mathbf{y}^N)^T$ includes the vectors $\mathbf{y}^i = \left(\mathbf{U}^i, \frac{\partial \mathbf{U}^i}{\partial t}\right)^T$, $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1)^T$ contains the initial values, and the matrix $\mathcal{N}$ and the vector $\hat{\mathcal{F}}^i$ are defined by

$$\mathcal{N} = -\begin{pmatrix} \hat{\mathcal{C}} \\ 2\hat{\mathcal{C}} \\ 2\hat{\mathcal{C}} \\ \hat{\mathcal{C}} \end{pmatrix}^T \begin{pmatrix} \mathcal{I} & & & \\ \hat{\mathcal{B}} & \mathcal{I} & & \\ & \hat{\mathcal{B}} & \mathcal{I} & \\ & & 2\hat{\mathcal{B}} & \mathcal{I} \end{pmatrix}^{-1} \begin{pmatrix} 2\hat{\mathcal{B}} \\ 2\hat{\mathcal{B}} \\ 2\hat{\mathcal{B}} \\ 2\hat{\mathcal{B}} \end{pmatrix} - \mathcal{I}, \tag{32}$$

$$\hat{\mathcal{F}}^i = -\begin{pmatrix} \hat{\mathcal{C}} \\ 2\hat{\mathcal{C}} \\ 2\hat{\mathcal{C}} \\ \hat{\mathcal{C}} \end{pmatrix}^T \begin{pmatrix} \mathcal{I} & & & \\ \hat{\mathcal{B}} & \mathcal{I} & & \\ & \hat{\mathcal{B}} & \mathcal{I} & \\ & & 2\hat{\mathcal{B}} & \mathcal{I} \end{pmatrix}^{-1} \begin{pmatrix} \hat{\mathcal{D}}^{i-1} \\ \hat{\mathcal{D}}^{i-\frac{1}{2}} \\ \hat{\mathcal{D}}^{i-\frac{1}{2}} \\ \hat{\mathcal{D}}^{i} \end{pmatrix}. \tag{33}$$

The matrix blocks $\hat{\mathcal{C}}$ and $\hat{\mathcal{B}}$ and the vector blocks $\hat{\mathcal{D}}^i$ are given by the formulas

$$\hat{\mathcal{C}} = \begin{pmatrix} -\frac{1}{6}\mathcal{I} & 0 \\ 0 & -\frac{1}{6}\mathcal{I} \end{pmatrix},$$

$$\hat{\mathcal{B}} = \begin{pmatrix} 0 & -\frac{\Delta t}{2}\mathcal{I} \\ \frac{\Delta t}{2}\mathcal{M}^{-1}\mathcal{K} & \frac{\Delta t}{2}\mathcal{M}^{-1}\mathcal{S} \end{pmatrix},$$

$$\hat{\mathcal{D}}^i = \begin{pmatrix} \Delta t \mathcal{M}^{-1}\mathcal{F}^i \\ 0 \end{pmatrix}.$$

The block-matrix form (31) of the fully discrete state equation with the RK time-stepping is analogous to the state equation (18).

5 Control problem

In order to solve the exact controllability problem, we use the least-squares formulation

$$\min_{\mathbf{e}\in\mathbf{Z}} J(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e})), \tag{34}$$

where $\hat{\mathbf{y}}(\mathbf{e})$ solves equations (7)-(10) and

$$J(\mathbf{e},\hat{\mathbf{y}}(\mathbf{e})) = \frac{1}{2}\left(\mathbf{y}^N - \mathbf{e}\right)^T \begin{pmatrix} \mathcal{K} & 0 \\ 0 & \mathcal{M} \end{pmatrix} \left(\mathbf{y}^N - \mathbf{e}\right) \tag{35}$$

is the discretized objective function. Solving the minimization problem (34) is equivalent to finding initial conditions such that the gradient of the objective function (35) is zero. This can be done by a conjugate gradient algorithm. In order to implement the algorithm, we have to compute the derivative of the objective function (35).

The state equations (18) and (31) can be represented in the generic form $s(\mathbf{e},\hat{\mathbf{y}}(\mathbf{e})) = 0$, and by the adjoint equation technique we see that

$$\frac{dJ(\mathbf{e},\hat{\mathbf{y}}(\mathbf{e}))}{d\mathbf{e}_k} = \frac{\partial J(\mathbf{e},\hat{\mathbf{y}})}{\partial \mathbf{e}_k} - \hat{\mathbf{z}}^T \frac{\partial s(\mathbf{e},\hat{\mathbf{y}})}{\partial \mathbf{e}_k}, \quad k = 0, 1, \tag{36}$$

where $\hat{\mathbf{z}}$ contains the vectors of the adjoint state variable at time $i\Delta t$. The vector $\hat{\mathbf{z}}$ is the solution of the adjoint equation

$$\left(\frac{\partial s(\mathbf{e},\hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}}\right)^T \hat{\mathbf{z}} = \left(\frac{\partial J(\mathbf{e},\hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}}\right)^T. \tag{37}$$

By $s_0(\mathbf{e},\hat{\mathbf{y}}(\mathbf{e})) = 0$ we denote the state equation ((18) or (31)) in the special case with $\mathcal{F}^i = 0$ for all i.

5.1 The adjoint equation with the central finite difference method

In the matrix form corresponding to (18), the adjoint state equation is given by

$$\begin{pmatrix} \mathcal{I} & \mathcal{C}_0 & \mathcal{B} & & & \\ & \mathcal{M} & \mathcal{C} & \mathcal{B} & & \\ & & \mathcal{D} & \ddots & \ddots & \\ & & & \ddots & \ddots & \mathcal{B} \\ & & & & \mathcal{D} & \mathcal{C} \\ & & & & & \mathcal{D} \end{pmatrix} \begin{pmatrix} \mathbf{P}^0 \\ \mathbf{P}^1 \\ \vdots \\ \vdots \\ \mathbf{P}^N \\ \mathbf{P}^{N+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{\partial J}{\partial \mathbf{U}^{N-1}} \\ \frac{\partial J}{\partial \mathbf{U}^{N}} \\ \frac{\partial J}{\partial \mathbf{U}^{N+1}} \end{pmatrix}, \tag{38}$$

where

$$\frac{\partial J}{\partial \mathbf{U}^{N-1}} = \frac{1}{2\Delta t}\mathcal{M}\left(\mathbf{e}_1 - \frac{\partial \mathbf{U}^N}{\partial t}\right), \tag{39}$$

$$\frac{\partial J}{\partial \mathbf{U}^{N+1}} = \frac{1}{2\Delta t}\mathcal{M}\left(\frac{\partial \mathbf{U}^N}{\partial t} - \mathbf{e}_1\right), \tag{40}$$

$$\frac{\partial J}{\partial \mathbf{U}^{N}} = \mathcal{K}\left(\mathbf{U}^N - \mathbf{e}_0\right). \tag{41}$$

The gradient components are then the following:

$$\frac{dJ(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e}))}{d\mathbf{e}_0} = \mathcal{K}(\mathbf{e}_0 - \mathbf{U}^N) + \mathbf{P}^0, \tag{42}$$

$$\frac{dJ(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e}))}{d\mathbf{e}_1} = \mathcal{M}\Big(\mathbf{e}_1 - \frac{\partial \mathbf{U}^N}{\partial t}\Big) + \Delta t \mathcal{B}\mathbf{P}^1. \tag{43}$$

5.2 The adjoint equation with the fourth-order Runge-Kutta method

The adjoint equation corresponding to the state equation (31) is

$$\begin{pmatrix} \mathcal{I} & \mathcal{N}^T & & & \\ & \mathcal{I} & \mathcal{N}^T & & \\ & & \ddots & \ddots & \\ & & & \mathcal{I} & \mathcal{N}^T \\ & & & & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{z}^0 \\ \mathbf{z}^1 \\ \vdots \\ \mathbf{z}^{N-1} \\ \mathbf{z}^N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \frac{\partial J}{\partial \mathbf{y}^N} \end{pmatrix}, \tag{44}$$

where $\mathbf{z}^i = (\mathbf{P}^i, \frac{\partial \mathbf{P}^i}{\partial t})^T$ contains the solution of the adjoint equation and its time derivative at $t = i\Delta t$, $i = N-1, \dots, 0$, and in addition we have

$$\frac{\partial J}{\partial \mathbf{y}^N} = \begin{pmatrix} \mathcal{K}(\mathbf{U}^N - \mathbf{e}_0) \\ \mathcal{M}(\frac{\partial \mathbf{U}^N}{\partial t} - \mathbf{e}_1) \end{pmatrix}.$$

In the case of the fourth-order Runge-Kutta scheme, the gradient components are

$$\frac{dJ(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e}))}{d\mathbf{e}_0} = \mathcal{K}(\mathbf{e}_0 - \mathbf{U}^N) + \mathbf{P}^0, \tag{45}$$

$$\frac{dJ(\mathbf{e}, \hat{\mathbf{y}}(\mathbf{e}))}{d\mathbf{e}_1} = \mathcal{M}\left(\mathbf{e}_1 - \frac{\partial \mathbf{U}^N}{\partial t}\right) + \frac{\partial \mathbf{P}^0}{\partial t}. \tag{46}$$

5.3 Preconditioned conjugate gradient method

We solve the least-squares problem with the following conjugate gradient (CG) algorithm:

Algorithm 1 *Preconditioned CG algorithm*

```
Compute the initial value $\mathbf{e}^0 = (\mathbf{e}_0^0, \mathbf{e}_1^0)^T$.
Solve the state equation $s(\mathbf{e}^0, \hat{\mathbf{y}}(\mathbf{e}^0)) = 0$.
Solve the adjoint state equation $\left(\frac{\partial s(\mathbf{e}^0, \hat{\mathbf{y}}(\mathbf{e}^0))}{\partial \hat{\mathbf{y}}(\mathbf{e}^0)}\right)^T \hat{\mathbf{z}} = \left(\frac{\partial J(\mathbf{e}^0, \hat{\mathbf{y}}(\mathbf{e}^0))}{\partial \hat{\mathbf{y}}(\mathbf{e}^0)}\right)^T$ .
Compute the gradient $\mathbf{g} = (g_0, g_1)^T$ by the formulas (42)-(43) or (45)-(46).
Solve linear system with the preconditioner $\mathcal{L}\mathbf{w} = -\mathbf{g}$.
Set $c_0 = -(\mathbf{w}, \mathbf{g})$, $c = c_0$ and $i = 1$.
Repeat until $\sqrt{\frac{c}{c_0}} < \varepsilon$
    Solve the state equation $s_0(\mathbf{w}, \hat{\mathbf{y}}(\mathbf{w})) = 0$.
    Solve the adjoint state equation $\left(\frac{\partial s(\mathbf{w}, \hat{\mathbf{y}}(\mathbf{w}))}{\partial \hat{\mathbf{y}}(\mathbf{w})}\right)^T \hat{\mathbf{z}} = \left(\frac{\partial J(\mathbf{w}, \hat{\mathbf{y}}(\mathbf{w}))}{\partial \hat{\mathbf{y}}(\mathbf{w})}\right)^T$ .
    Compute the gradient update $\mathbf{v} = (v_0, v_1)^T$ by the formulas (42)-(43) or (45)-(46) .
    Compute $\eta = \frac{c}{(\mathbf{w}, \mathbf{v})}$.
    $\mathbf{e}^i = \mathbf{e}^{i-1} + \eta \mathbf{w}$.
    $\mathbf{g} = \mathbf{g} + \eta \mathbf{v}$.
    Solve linear system with the preconditioner $\mathcal{L}\mathbf{v} = -\mathbf{g}$.
    $\gamma = \frac{1}{c}$, $c = -(\mathbf{v}, \mathbf{g})$, $\gamma = c\gamma$.
    $\mathbf{w} = \mathbf{v} + \gamma \mathbf{w}$, $i = i + 1$.
```

Smooth initial approximations $\mathbf{e}^0 = (\mathbf{e}_0^0, \mathbf{e}_1^0)^T$ for the algorithm are computed with a transition procedure, which is presented in [32]. Values of the control variables $\mathbf{e}$ at the ith iteration are denoted by $\mathbf{e}_0^i$ and $\mathbf{e}_1^i$.

Each conjugate gradient iteration step requires computation of the gradient of the least-squares functional, ∇J, which involves the solution of the state equation (18) or (31) and the corresponding adjoint equation (38) or (44), the solution of a linear system with the preconditioner, and some other matrix-vector operations.

The solution of a linear system with the block-diagonal preconditioner, which we have chosen to be

$$\mathcal{L} = \begin{pmatrix} \mathcal{K} & 0 \\ 0 & \mathcal{M} \end{pmatrix}, \tag{47}$$

requires the solution of systems with the stiffness matrix $\mathcal{K}$ and the diagonal mass matrix $\mathcal{M}$. Efficient solution of linear systems with the matrix $\mathcal{K}$ is critical for the overall efficiency of the control method. At this stage, we use a modification of Kickinger's [33] algebraic multigrid (AMG) introduced in [26]. As the name of the method indicates, a number of different grid levels are used on the domain, ranging between fine and coarse levels. A sequence of linear problems

$$\mathcal{K}_l \tilde{\mathbf{w}}_l = \tilde{\mathbf{g}}_l \tag{48}$$

is generated, corresponding to grid levels $l = 0, \ldots, \tilde{k}$, where $\tilde{k}$ represents the coarsest level. Each AMG iteration starts with the finest level matrix $\mathcal{K}_0$, right hand side vetor $\tilde{\mathbf{g}}_0$, and an approximation $\tilde{\mathbf{w}}_0$. For a particular level l, the residual is given by $\tilde{r}_l = \tilde{\mathbf{g}}_l - \mathcal{K}_l \tilde{\mathbf{w}}_l$. This is used as the basis of a correction equation $\tilde{\mathbf{w}}_l = \tilde{\mathbf{w}}_l + \tilde{e}_l$. The error $\tilde{e}_l$ is related to the residual by $\mathcal{K}_l \tilde{e}_l = \tilde{r}_l$. Unlike the classical geometric multigrid methods [7,8], in the AMG the actual coarsenig of the given mesh is not needed for finding coarser grid levels.

The coarsening, i.e., selection of the unknowns for coarser levels, is based on the graph of the stiffness matrix, rather than on the actual values stored in the stiffness matrix. This approach ensures fast computation of coarser level components. The coarsening process operates in a geometric fashion by sequentially choosing a coarse node and eliminating the neighboring nodes of the graph. In selecting the unknowns for coarser levels, the primary criterion is to take the node with minimum degree when eliminations have taken into account. The secondary criterion is to follow the original node numbering.

The use of the AMG methods for spectral elements has recently been studied in [34]. The number of connections between unknowns of the problem increases when higher-order elements are used. In this case, the coarsening strategy described above leads to unacceptably coarse systems and the convergence factor of the AMG degrades rapidly as the order of the approximation polynomials increases. We overcome this problem by employing a graph constructed so that unknowns are connected to each other as if low-order finite elements were used in the discretization process. Only the unknowns corresponding to the nearest neighbouring Gauss-Lobatto points are connected to each other. Additionally, in vector valued problems it is necessary to prevent mixture of various types of unknowns also on coarser levels. This is achieved by giving the method an initial graph where the sets of graph nodes corresponding to different types of unknowns are not interconnected.

The grid transfer operators are the restriction operator $\widetilde{\mathbf{R}}$ and the prolongation operator $\widetilde{\mathbf{P}}$. The matrices $\mathcal{K}_l$, which are used at multigrid levels $l = 0, \ldots, \tilde{k}$, are set as an initialization step of the AMG algorithm. For this purpose we need the restriction operator $\widetilde{\mathbf{R}}_l^{l+1}$ from the fine level l to the coarse level $(l+1)$

$$\widetilde{\mathbf{R}}_l^{l+1} = \begin{pmatrix} \mathbf{R}_l^{l+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_l^{l+1} \end{pmatrix}, \tag{49}$$

where the components of the restriction matrices $\mathbf{R}_l^{l+1}$ are

$$\left(\mathbf{R}_l^{l+1}\right)_{ij} = \begin{cases} 1 & \text{for a fine grid point } j \text{ which is a coarse grid point } i, \\ \frac{1}{k} & \text{for a fine grid point } j \text{ which is a neighbor of coarse grid} \\ & \text{point } i \text{ and has } k \text{ neighboring coarse grid points,} \\ 0 & \text{otherwise.} \end{cases} \tag{50}$$

When the fine level matrix $\mathcal{K}_l$ is known, the coarse grid operator is given by the Galerkin formula $\mathcal{K}_{l+1} = \widetilde{\mathbf{R}}_l^{l+1}\mathcal{K}_l(\widetilde{\mathbf{R}}_l^{l+1})^T$. The prolongation operator $\widetilde{\mathbf{P}}_{l+1}^l$ from the coarse level $(l+1)$ to the fine level l is chosen to be the transpose of the restriction,

$$\widetilde{\mathbf{P}}_{l+1}^l = (\widetilde{\mathbf{R}}_l^{l+1})^T.$$

As a smoother of the AMG we have used successive over-relaxation (SOR), with over-relaxation parameter 1.2, unless other mentioned. One iteration of the SOR is used for pre- and post-smoothing. Additionally, in the beginning of every multigrid iteration, four iterations of the SOR are used to smooth the solution initially. The so called W-cycle [7,35] is utilized as a multigrid iteration until the residual norm of the solution is smaller than 10^{-6}.

When solving the state equation (18) or (31), $\mathcal{M}^{-1}$ is the only matrix inversion which is involved in time-stepping. Since the matrix $\mathcal{M}$ is diagonal, it is inverted simply by inverting each of its diagonal elements. This requires only $2\hat{N}$ floating point operations, which is the number of degrees of freedom in the space discretization. The operation count of a matrix-vector product with any one of the matrices $\mathcal{M}$, $\mathcal{M}^{-1}$, $\mathcal{S}$, or $\mathcal{S}^{-1}$ (or some linear combination of these) is of order $O(\hat{N})$. In the matrix-vector multiplication involving the sparse stiffness matrix $\mathcal{K}$, only non-zero matrix entries are multiplied, which requires on the order of $r^2\hat{N}$ operations. Besides these, some additions and multiplications are needed at each time step. Thus, solving the state equation needs $O(r^2\hat{N})$ floating point operations at each time step in the CD and the RK time-steppings. From this, we can conclude that the computational demand for computing the solution for the state equation with N time steps is $O(Nr^2\hat{N})$. The number of computational operations needed for solving the adjoint state equation is obviously of the same order as that needed for solving the state equation. On the whole, the computational cost for one iteration of the CG algorithm is of order $O(Nr^2\hat{N})$. Assuming that the number of time steps N is fixed, the number of iterations is approximately constant, and the element order r has small integer values, the computational demand for the overall CG algorithm is $O(\hat{N})$.

6 Numerical experiments

We consider several time-harmonic problems including propagation and scattering of the linear elastic wave equation in domain Ω. The domain Ω is defined such that the boundary surrounding it, Γ_{ext}, coincides with the perimeter of the rectangle $[0,0] \times [4,4]$, where coordinates are given in kilometers. Wave propagation as discussed in Section 6.3 is simulated in the rectangle $[0,0] \times [4,4]$, whereas in the experimental results reported in Sections 6.1 and 6.2, we have set a rigid square obstacle, having side length 2 km and boundary Γ_0, in the center of the domain. In these experiments, polygonal geometries are used to eliminate the error in approximating the geometry. The propagation direction is chosen to be $\vec{\omega} = (\omega_1, \omega_2) = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\omega$, and for angular frequency we mainly use the value $\omega = 2\pi$ Hz. The source function on the absorbing boundary is $\mathbf{G}_{\text{ext}} = \rho\mathbf{B}\frac{\partial \mathbf{U}_{\text{inc}}}{\partial t} + \sigma(\mathbf{U}_{\text{inc}})\mathbf{n}$, where the incident plane wave is

$$\mathbf{U}_{\text{inc}} = \begin{pmatrix} \omega_1 \cos(\omega t - \frac{\omega}{c_p}\mathbf{x}\cdot\vec{\omega}) + \omega_2 \cos(\omega t - \frac{\omega}{c_s}\mathbf{x}\cdot\vec{\omega}) \\ \omega_2 \cos(\omega t - \frac{\omega}{c_p}\mathbf{x}\cdot\vec{\omega}) - \omega_1 \cos(\omega t - \frac{\omega}{c_s}\mathbf{x}\cdot\vec{\omega}) \end{pmatrix}. \tag{51}$$

All the computations have been carried out on an AMD Opteron 885 at 2.6 GHz.

6.1 Accuracy and efficiency

Since we are particularly interested in the accuracy of the spatial discretization, we have eliminated the error due to the absorbing boundary condition by using a modified problem,

$$\begin{aligned}
\rho\frac{\partial^2\hat{\mathbf{U}}}{\partial t^2} - \nabla\cdot\sigma(\hat{\mathbf{U}}) &= -\rho\frac{\partial^2\mathbf{g}}{\partial t^2} + \nabla\cdot\sigma(\mathbf{g}), && \text{in } Q = \Omega\times[0,T],\\
\hat{\mathbf{U}} &= 0, && \text{on } \gamma_0 = \Gamma_0\times[0,T],\\
\rho\mathbf{B}\frac{\partial\hat{\mathbf{U}}}{\partial t} + \sigma(\hat{\mathbf{U}})\mathbf{n} &= \mathbf{G}_{\text{ext}} && \text{on } \gamma_{\text{ext}} = \Gamma_{\text{ext}}\times[0,T],\\
\hat{\mathbf{U}}(\mathbf{x},0) &= \mathbf{U}_{\text{inc}}(\mathbf{x},0) - \mathbf{g}(\mathbf{x},0), && \text{in } \Omega,\\
\frac{\partial\hat{\mathbf{U}}(\mathbf{x},0)}{\partial t} &= \frac{\partial\mathbf{U}_{\text{inc}}(\mathbf{x},0)}{\partial t} - \frac{\partial\mathbf{g}(\mathbf{x},0)}{\partial t}, && \text{in } \Omega,
\end{aligned}$$

$$\mathbf{g}|_{\Gamma_0} = \mathbf{U}_{\text{inc}}, \quad \mathbf{g}|_{\Gamma_{\text{ext}}} = 0, \quad \frac{\partial\mathbf{g}}{\partial\mathbf{n}}|_{\Gamma_{\text{ext}}} = 0,$$

the solution of which is known to be $\hat{\mathbf{U}} = \mathbf{U}_{\text{inc}} - \mathbf{g}$. The accuracy and computational efficiency of the method is considered with this modified problem in an isotropic homogeneous elastic medium with $c_p = 2$ km/s, $c_s = 1$ km/s, and $\rho = 2700$ kg/m^3.

6.1.1 Approximation error with constant angular frequency

In the first experiment we have used a constant spatial discretization such that the ratio between the order of elements r and the mesh stepsize h is $r/h \approx 20$ km^{-1}. We have also compared the CD time discretization with the RK time discretization for element orders $r = 1,\dots,5$. In each case, the number of time steps needed for stability is tested by using $10i$ time steps per time-period, for $i = 1,2,3,\dots,$ until a stable solution is achieved. Stability conditions corresponding to the largest stable time step are given in Table 1. According to our numerical tests, these values are the same with both the CD and the RK time-stepping. Moreover, Δt satifies the well known CFL condition

Table 1
Stability conditions.

	r	1	2	3	4	5
Number of time steps	CD	50	60	80	90	100
	RK	50	60	80	90	100
$\Delta t/h$	CD	0.40	0.17	0.09	0.06	0.04
	RK	0.40	0.17	0.09	0.06	0.04

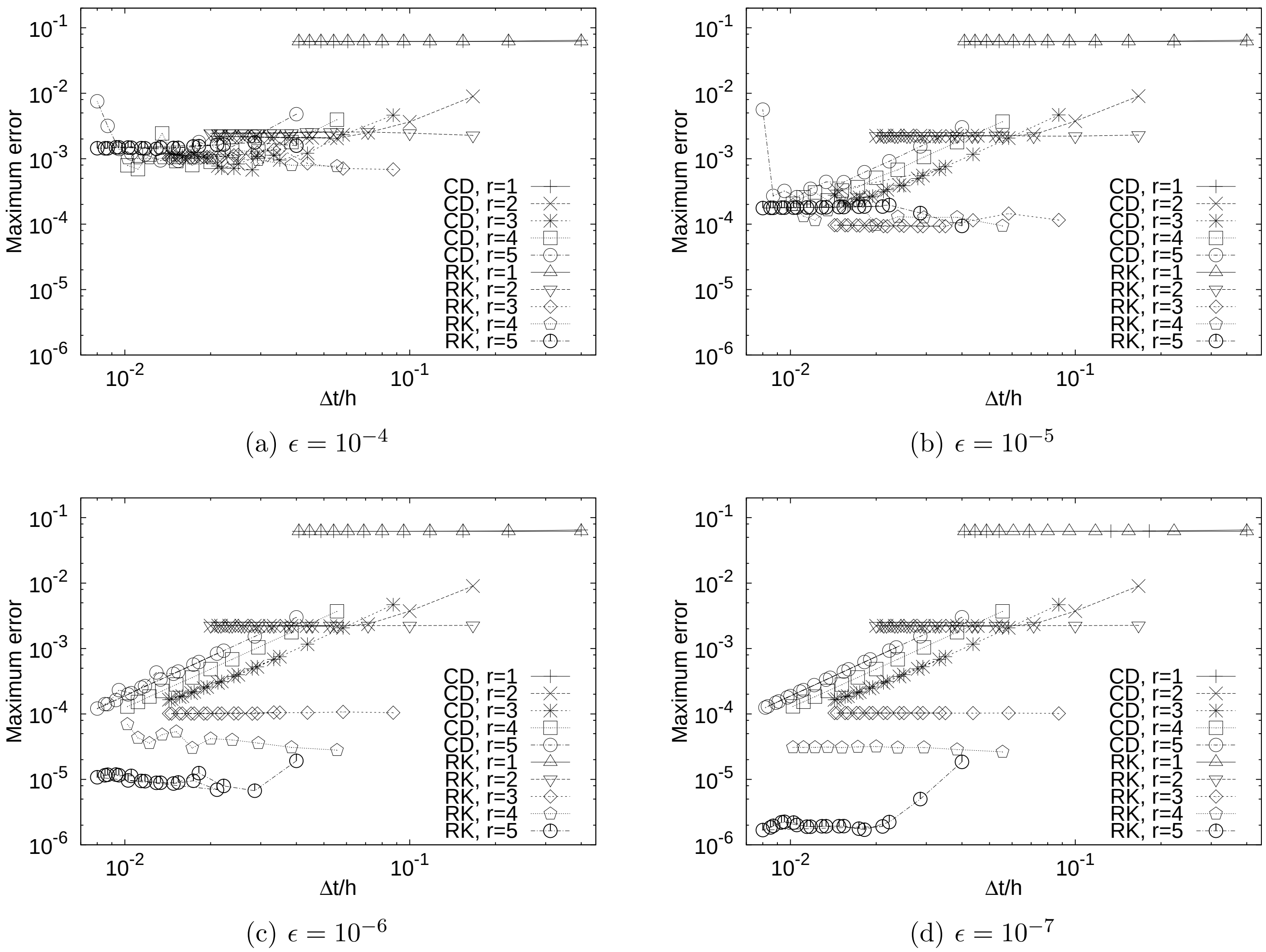
$$\frac{\Delta t}{h} = \frac{\alpha_r}{c_p\sqrt{2}}, \tag{52}$$

where α_r is a stability constant for element order r [23].

We started computations with the largest stable time step, and then repeatedly added 40 to the number of time steps $N = T/\Delta t$, until N was larger than 500. In this way, we achieved numerical results for a variety of time step lengths. Errors between the analytical solution and the experimental result are computed as L^∞-norms. Accuracy of the numerical solution is shown in Figure 2 as a function of the ratio between the time step Δt and the mesh step size h for both the CD and the RK time-steppings with five element orders r and four different stopping criteria ϵ.

When the first-order SEM (i.e., $r = 1$ which corresponds to classical bilinear finite elements) is used, spatial error dominates for each stable length of the time step with both time discretization schemes. This is seen as horizontal lines describing the error level of the particular spatial discretization in Figures 2(a)-2(d). Since the lowest order space discretization gives such a poor accuracy, the controllability method is not useful in practice unless higher-order elements are used. As the order of the approximation in space increases, the solution becomes more accurate until the effect of the stopping criterion or the error of time or space discretization becomes dominant. Figure 2(a) shows that solutions with small time step and $r > 1$ are limited by the stopping criterion. When the stopping criterion is tightened, more accurate solutions are reached with higher-order elements (see Figures 2(a)-2(d)). In Figure 2(d), the stopping criterion affects the accuracy only when the RK time discretization is used with $r = 5$. This is seen as a trifling oscillation in the left part of the error curve. The residual stayed slightly over 10^{-7} for some values of Δt in the algorithm with CD time discretization and $r = 1$. This is why there are some blank spaces in the corresponding curve in Figure 2(d).

When the stopping criterion ϵ is tightened, the number of CG iterations needed to attain the stopping criterion grows. This implies a larger computational effort, as seen in Figures 3(a)-3(b). On each curve, presenting the computational



(a) $\epsilon = 10^{-4}$

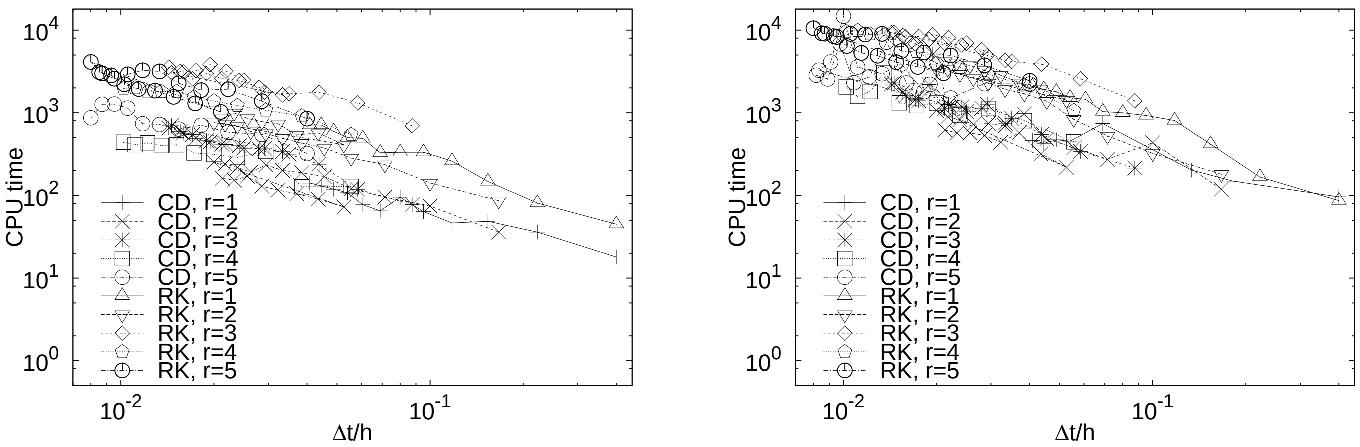
(b) $\epsilon = 10^{-5}$

(c) $\epsilon = 10^{-6}$

(d) $\epsilon = 10^{-7}$

Figure 2. Side by side comparisons of the maximum errors obtained in the case of the CD and the RK time-stepping with four different stopping criteria ϵ.

cost with respect to $\Delta t/h$, the number of degrees of freedom and r are constants. The linear dependence between CPU time and $\Delta t/h$ shows that the order of the number of iterations remains constant when time step refinement is done. Although the RK time discretization consumes more CPU time than the CD time discretization for a particular time step, the method with the RK time discretization is the more efficient one with higher-order elements (see Figure 4). The secret behind this behaviour, which is not in line with the proportion of arithmetic operations needed at each CG iteration, lies in the number of CG iterations. Since the algorithm with the RK time discretization solves the problem at each iteration more accurately than the algorithm with the CD time discretization, the RK version needs a smaller number of iterations. This phenomenon is emphasized when the stopping criterion is tightened. In the case of the RK time discretization with $r = 4$ and $\epsilon \leq 10^{-5}$



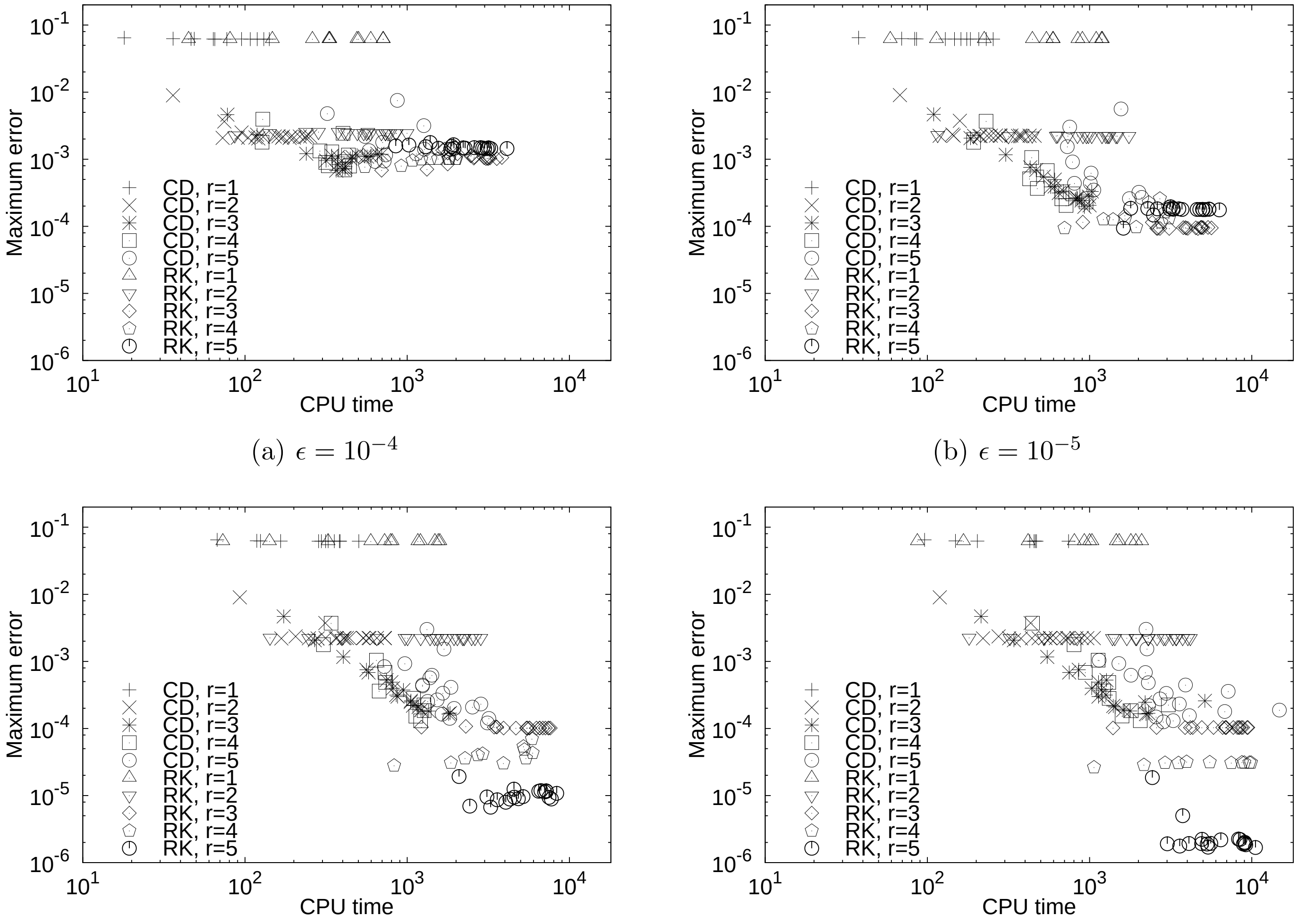
(a) $\epsilon = 10^{-4}$

(b) $\epsilon = 10^{-7}$

Figure 3. Side by side comparisons of the computational efforts (CPU time in seconds) obtained in the case of the CD and the RK time-stepping with two different stopping criteria ϵ.

the error of the stopping criterion has little or no influence on the solution. Moreover, the error of time discretization has no effect at this point, which makes possible to use the largest stable time step accurately. These properties induce the remarkable efficiency for $r = 4$ with the RK time discretization and the largest stable time step, as seen in Figures 4(b)-4(d).

A large time step allows us to compute the solution utilizing only small amount of CPU time, but it may involve an error which deteriorates the accuracy of the method. As a result, we tested which magnitude of the time step leads to prescribed accuracy with a relatively low number of time steps, i.e., with a comparatively low computational cost. The comparison between the two time discretization schemes shows that with higher-order elements, the RK time discretization gives a more accurate solution with a larger time step than the CD time discretization. The fourth-order accurate RK time discretization does not limit the accuracy of the solution at all with any stable time step and any $r < 5$. With the second-order CD time discretization, the error level of the space discretization is achieved with $r > 1$ only by choosing an appropriately small time step. Since we are interested in particular in the accuracy of spatial discretization, the smallest number of time steps which eliminate the error of temporal discretization are determined from Figure 2(d) and shown in Table 2. These are also the values which we will use for the later tests in this article. The proportion of CPU time required for different parts of the CG algorithm, with number of time steps shown in Table 2, is seen in Figure 5. It is noteworthy that with the CD time discretization the number of time steps needed to attain the given accuracy is at least ten times that of the RK time discretization for $r \geq 3$. This confirms the better efficiency of the RK time discretization with higher-order elements.



(a) $\epsilon = 10^{-4}$

(b) $\epsilon = 10^{-5}$

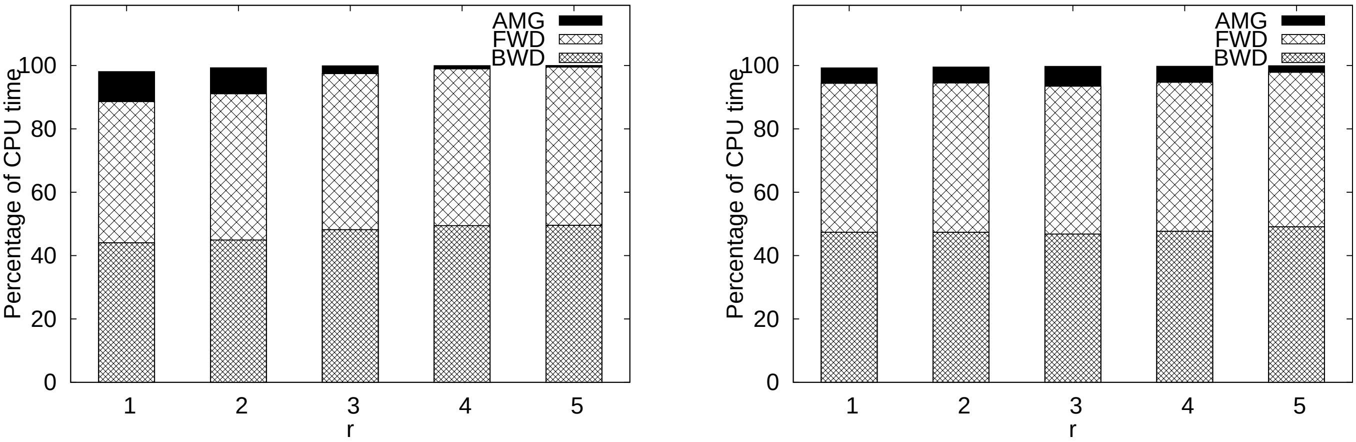
(c) $\epsilon = 10^{-6}$

(d) $\epsilon = 10^{-7}$

Figure 4. Side by side comparisons of the maximum errors with respect to CPU time (in seconds) obtained in the case of the CD and the RK time-stepping with four different stopping criteria ϵ.

Table 2
Number of time steps needed to attain the error level of spatial discretization for different spectral orders. For the CD time discretization with spectral order $r = 3$ or higher, the length of time step that eliminates the error of temporal discretization is determined by extrapolating the curves in Figure 2(d) to smaller time steps.

	r	1	2	3	4	5
Number of time steps	CD	50	140	800	1300	3600
	RK	50	60	80	90	210



(a) CD

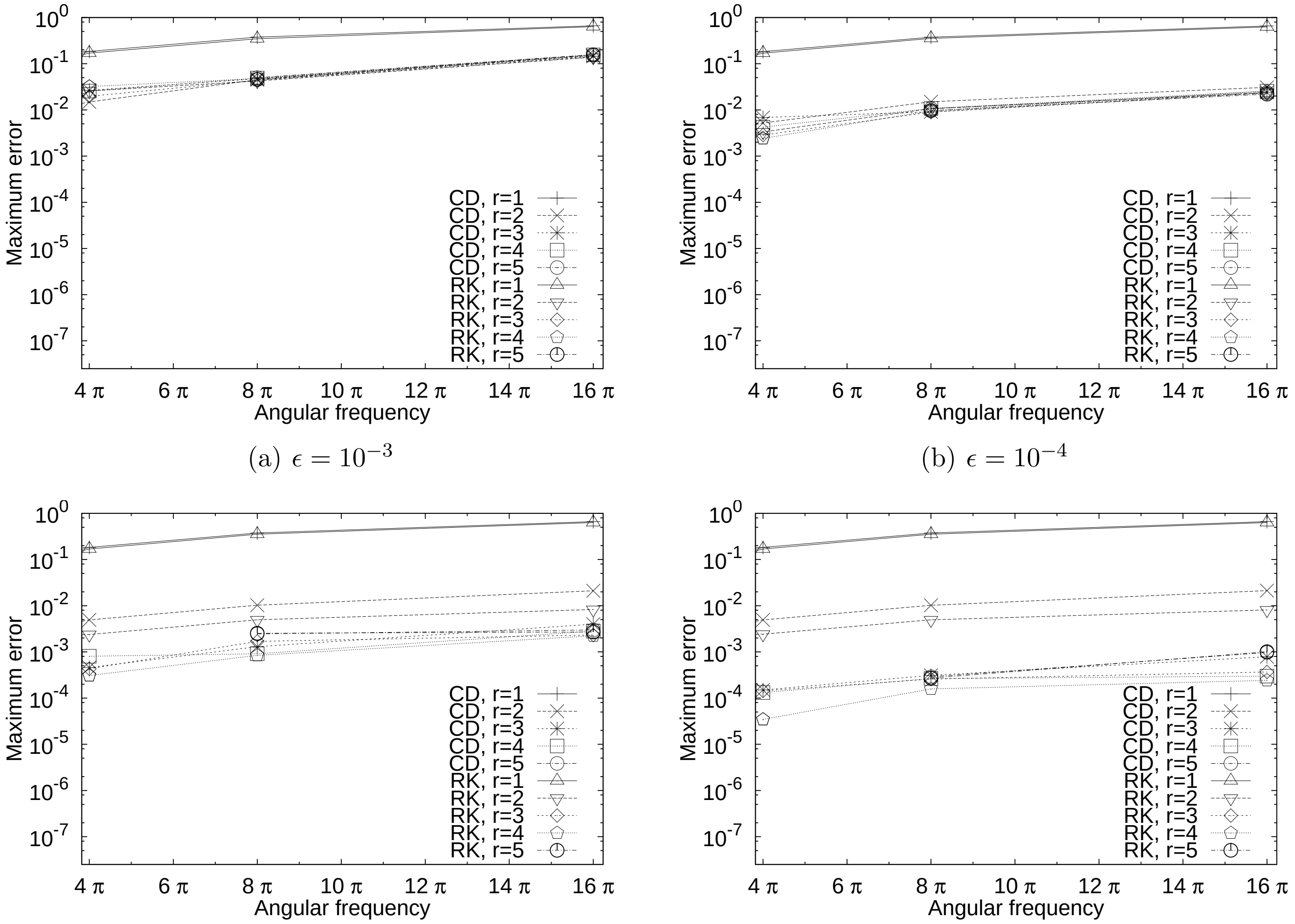
(b) RK

Figure 5. Side by side comparisons of the proportions of CPU time (in percent) required by AMG cycles in the case of the CD and the RK time-steppings with $\epsilon = 10^{-7}$.

6.1.2 Numerical dispersion with increasing angular frequency

Even though we have eliminated the main error sources in Section 6.1.1, numerical dispersion deteriorates the accuracy of solutions with small waves. The computed wavenumber differs from the wavenumber of the exact solution, and with high angular frequencies this part of approximation error becomes dominant. This is also referred to as the pollution effect. To show that using higher-order elements alleviates this inaccuracy, we performed another set of experiments, varying both the angular frequency and the resolution of the mesh so that $\omega h = r\pi/10$ km/s. Since the efficiency of the method is not significantly better with $\epsilon = 10^{-7}$ than with $\epsilon = 10^{-6}$, and the method with the CD time discretization had problems converging to the stopping criterion $\epsilon = 10^{-7}$ with certain lengths of the time step, we hereafter concentrate on the case with $\epsilon = 10^{-6}$. In this case, the stopping criterion does not significantly limit the accuracy of the numerical solutions computed with the CG algorithm.

The accuracy of the solution with respect to the angular frequency is presented in Figure 6. As the wavenumber grows, the error increases for all orders of the elements. In the case of the classical finite element discretization, i.e., $r = 1$, the error becomes very large as the wavenumber increases. This happens even if ωh is kept constant. With higher-order elements, the pollution effect is not eliminated, but the accuracy is significantly better, even for high angular frequencies. As seen in Figure 6, when the error of time discretization is eliminated, the same level of accuracy is attained with the CD and the RK time discretizations. This is also true for high wavenumbers. With the RK time discretization, this level of accuracy is achieved with lower computational cost than with the CD time discretization (see Figure 7). From Figures 6 and 7, we



(a) $\epsilon = 10^{-3}$

(b) $\epsilon = 10^{-4}$

(c) $\epsilon = 10^{-5}$

(d) $\epsilon = 10^{-6}$

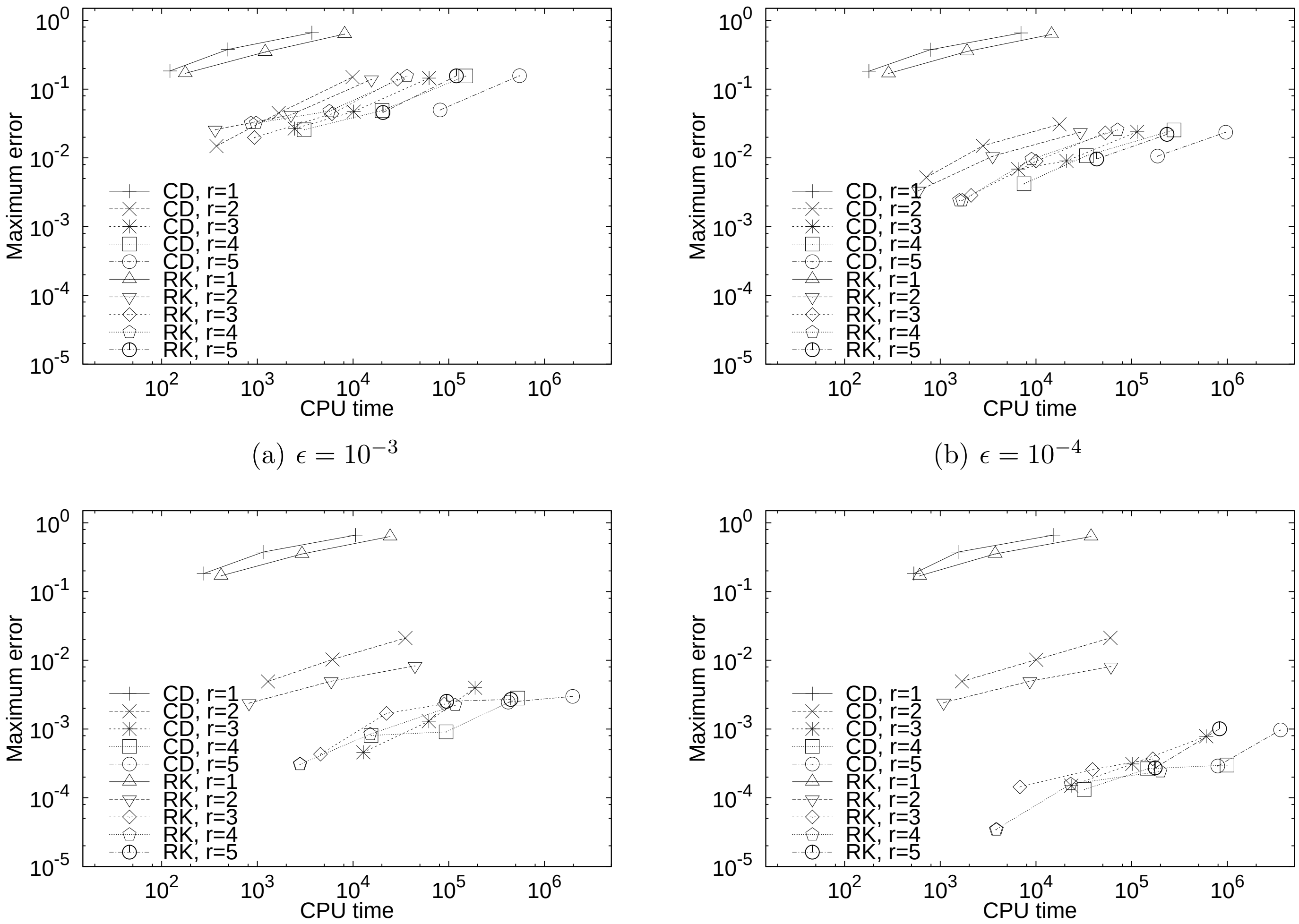
Figure 6. Side by side comparisons of the maximum errors obtained in the case of the CD and the RK time-stepping with four different stopping criteria ϵ.

further notice that CPU time for the algorithm grows with the wavenumber. The reason for this is the increase in number of CG iterations.

6.2 Elastic scattering by a rigid obstacle

We now discuss elastic scattering by a rigid square obstacle of side length 2 km in an isotropic homogeneous elastic medium. The material parameters are the same as in the previous test case, and the absorbing boundary Γ_{ext} is located around the obstacle at a distance of 1 km, as in Figure 1.

To take closer look at the iteration numbers, the computational effort, and the accomplishment of the preconditioner, we tested these properties with respect



(a) $\epsilon = 10^{-3}$

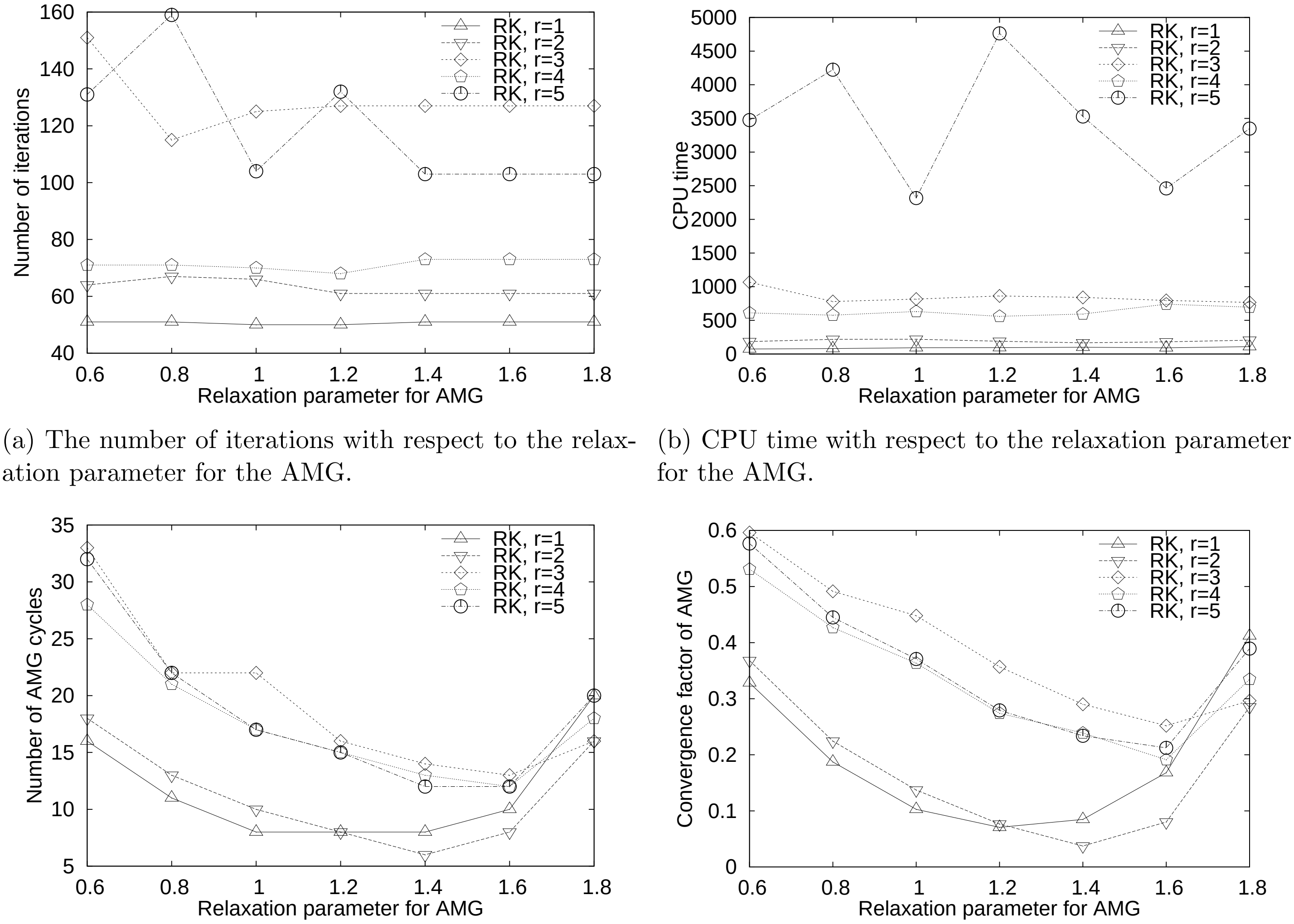
(b) $\epsilon = 10^{-4}$

(c) $\epsilon = 10^{-5}$

(d) $\epsilon = 10^{-6}$

Figure 7. Side by side comparisons of the maximum errors with respect to CPU time in the case of the CD and the RK time-stepping with four different stopping criteria ϵ.

to the relaxation parameter for the AMG (see Figure 8). The AMG solver performs best when the value of the relaxation parameter is between 1.2 and 1.6, depending on the order of the elements (see Figures 8(c) and 8(d)). From Figure 8(b), we see that the computational effort is considerably larger for $r = 5$ than for the other element orders. This is because a very fine time step is needed with $r = 5$ to eliminate the temporal error. The number of iterations varies between different element orders more than expected (see Figure 8(a)). The reason for this is probably the tightness of the stopping criterion ϵ. Also, our previous experiments show that when a very tight stopping criterion is used, the convergence rate may be lower, and the residual may converge to a level that is higher than the stopping criterion. That is why the number of iterations is almost constant for all element orders when $\epsilon \gtrsim 10^{-4}$, but differs



(a) The number of iterations with respect to the relaxation parameter for the AMG.

(b) CPU time with respect to the relaxation parameter for the AMG.

(c) Number of AMG cycles in the first CG iteration with respect to the relaxation parameter for the AMG.

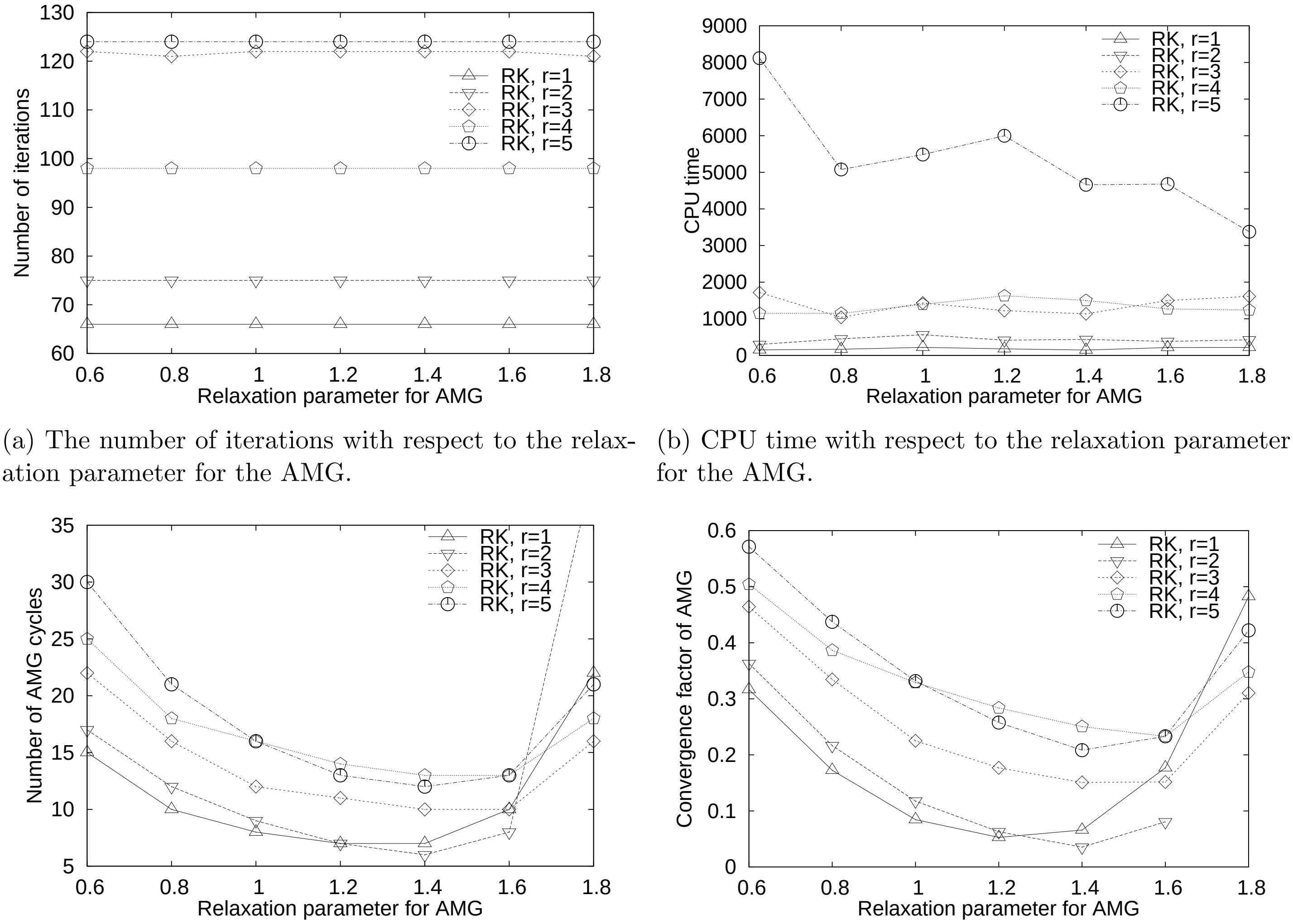
(d) Convergence factor in the first CG iteration with respect to the relaxation parameter for the AMG.

Figure 8. Elastic scattering by a square obstacle in the case of the RK time-stepping with $\omega = 2\pi$ Hz and $\epsilon = 10^{-6}$ in homogeneous domain, where $c_p = 2$ km/s, $c_s = 1$ km/s, and $\rho = 2700$ kg/m^3.

to a certain extent between the element orders for more strict stopping criteria.

6.3 Wave propagation in homogeneous and heterogeneous material

In this section, we consider the wave propagation in homogeneous and piecewise heterogeneous media without rigid obstacles. In this case, the boundary Γ_0 is not involved. Moreover, to avoid problems arising from the singularity of the stiffness matrix, we have used $\mathcal{K} + \frac{\mathcal{M}}{10^6}$ instead of $\mathcal{K}$ in preconditioning. In the homogeneous case, the domain is a square $\Omega = [0, 4] \times [0, 4]$



(a) The number of iterations with respect to the relaxation parameter for the AMG.

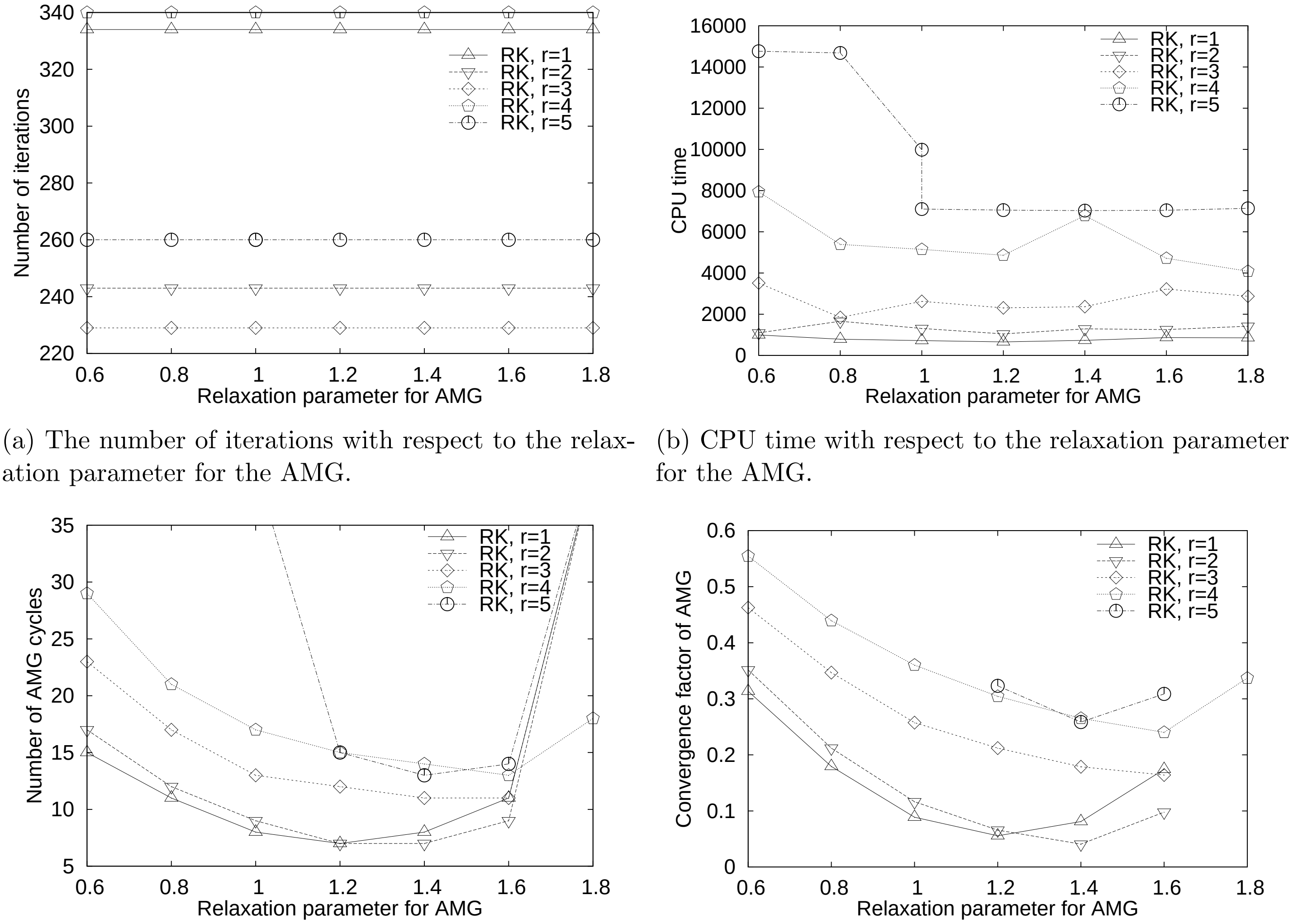
(b) CPU time with respect to the relaxation parameter for the AMG.

(c) Number of AMG cycles in the first CG iteration with respect to the relaxation parameter for the AMG.

(d) Convergence factor in the first CG iteration with respect to the relaxation parameter for the AMG.

Figure 9. Wave propagation in the case of the RK time-stepping with $\omega = 2\pi$ Hz and $\epsilon = 10^{-6}$ in homogeneous domain, where $c_p = 2$ km/s, $c_s = 1$ km/s, and $\rho = 2700$ kg/m^3 in $\Omega = [0,4] \times [0,4]$.

surrounded by the boundary Γ_{ext}, and is provided with material parameters $c_p = 2$ km/s, $c_s = 1$ km/s, and $\rho = 2700$ kg/m^3. The piecewise heterogeneous material is constructed by fixing the above described material parameters in $\Omega \setminus ([1,3] \times [1,3])$ and the material parameters $c_p = 0.6$ km/s, $c_s = 0.4$ km/s, and $\rho = 7200$ kg/m^3 in the square $[1,3] \times [1,3]$. The results with homogeneous and heterogeneous material are shown in Figures 9 and 10, respectively. The AMG solver performs best when the value of the relaxation parameter is between 1.2 and 1.6, depending on the order of the elements (see Figures 9(c), 9(d), 10(c), and 10(d)). The number of iterations is not dependent on the value of the relaxation parameter (see Figures 9(a) and 10(a)), but too



(a) The number of iterations with respect to the relaxation parameter for the AMG.

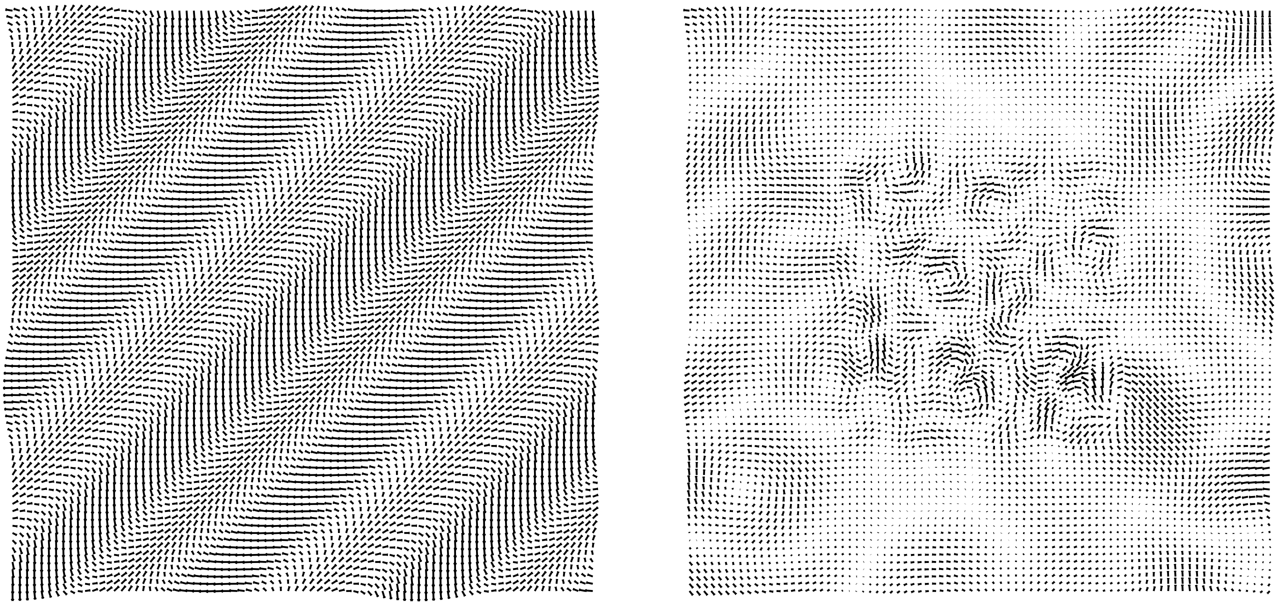
(b) CPU time with respect to the relaxation parameter for the AMG.

(c) Number of AMG cycles in the first CG iteration with respect to the relaxation parameter for the AMG.

(d) Convergence factor in the first CG iteration with respect to the relaxation parameter for the AMG.

Figure 10. Wave propagation in the case of the RK time-stepping with $\omega = 2\pi$ Hz and $\epsilon = 10^{-6}$ in heterogeneous domain, where $c_p = 600$ m/s, $c_s = 400$ m/s, and $\rho = 7200$ kg/m^3 in $[1,3] \times [1,3]$ and $c_p = 2$ km/s, $c_s = 1$ km/s, and $\rho = 2700$ kg/m^3 in $\Omega \setminus ([1,3] \times [1,3])$.

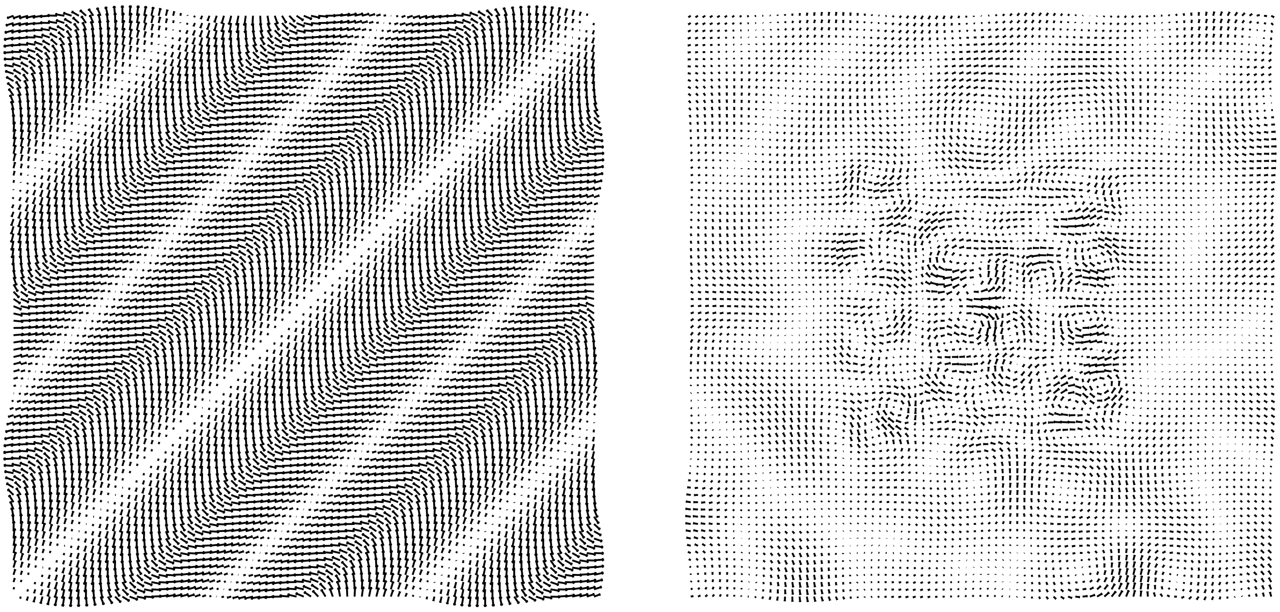
small of a value of the relaxation parameter causes an increase in the number of AMG cycles and decrease in the convergence factor of the AMG. This induces remarkably large CPU time requirements, which are seen in the left part of Figures 9(b) and 10(b). Since smaller wave lengths are involved in the center of the domain in the heterogeneous case than in the homogeneous case, a larger number of iterations and a larger amount of CPU time is needed for computations in the heterogeneous domain (see the displacement and velocity fields of the solutions with $r = 3$ in Figures 11 and 12). The convergence histories of the simulations in homogeneous and heterogeneous media, in which



(a) Homogeneous domain.

(b) Heterogeneous domain.

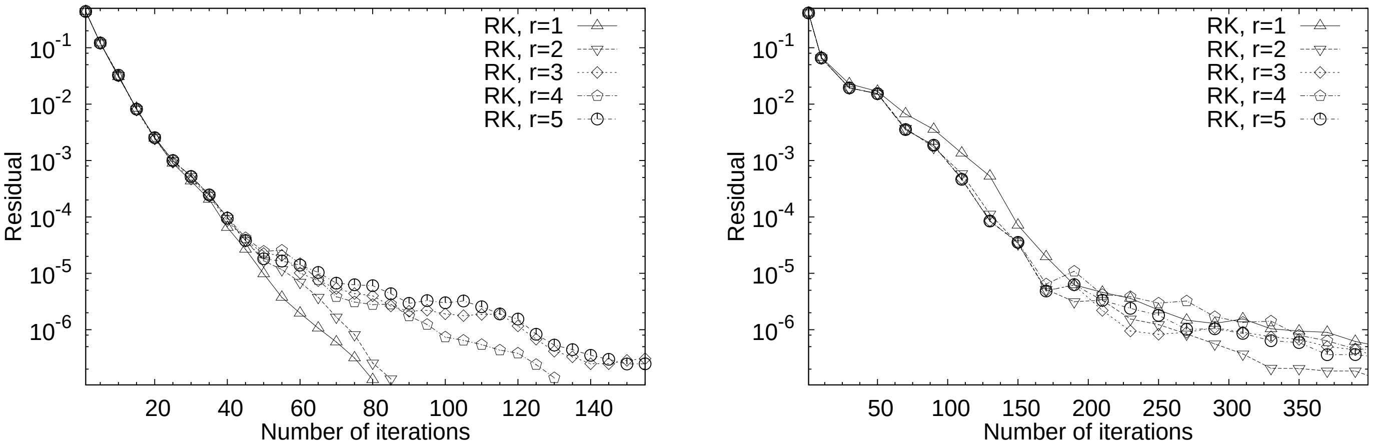
Figure 11. Displacement vector field for the real part of the solution of elastic scattering in homogeneous and heterogeneous domains.



(a) Homogeneous domain.

(b) Heterogeneous domain.

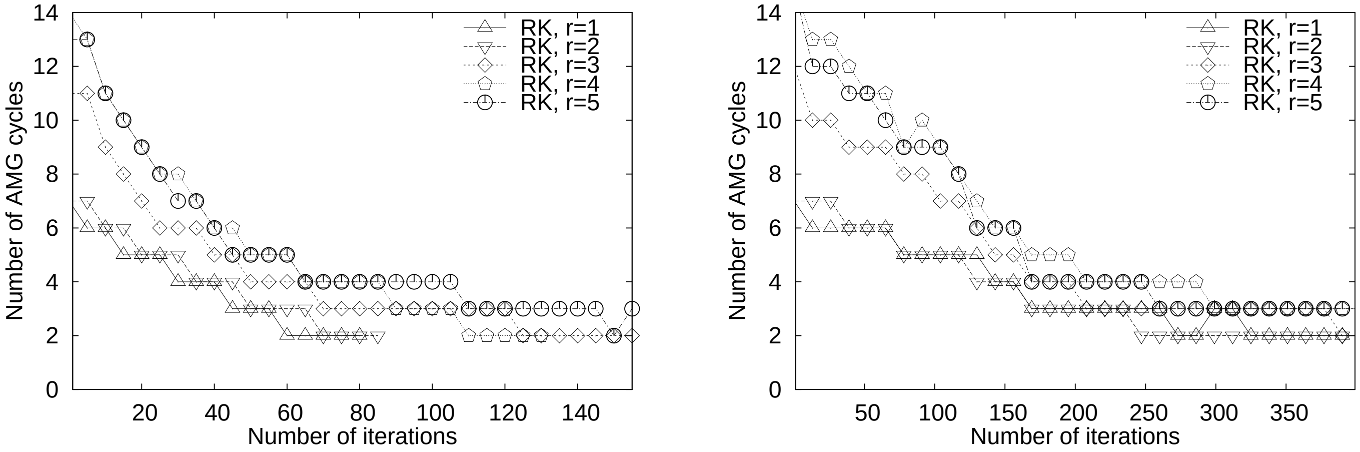
Figure 12. Velocity vector field for the real part of the solution of elastic scattering in homogeneous and heterogeneous domains.



(a) Homogeneous domain.

(b) Heterogeneous domain.

Figure 13. Convergence histories of elastic scattering in homogeneous and heterogeneous domains.



(a) Homogeneous domain.

(b) Heterogeneous domain.

Figure 14. Number of AMG cycles with respect to the number of CG iterations in homogeneous and heterogeneous domains.

the relaxation parameter 1.2 is used for the AMG, are shown in Figure 13. In both of these cases, where residuals are plotted with respect to the number of iterations, the convergence rate is equal for all element orders during the first iterations. As the residual becomes smaller, the convergence rate might become lower for some element orders. Still, the highest element order does not necessarily mean the poorest convergence rate. Insufficient preconditioning is the most feasible explanation for this observation. As the residual becomes smaller, the values in the right hand side vector **g**, which appears in preconditioning (see Algorithm 1), become closer to zero. With small residuals, the stopping criterion of the AMG solver is fullfilled with a small number of AMG cycles (or without any AMG cycles) as is seen in Figure 14.

7 Conclusions

To make good use of higher-order elements, the time discretization should also be done with a higher-order scheme. As a rule of thumb, we can say that the efficiency of the overall method suffers from the time discretization error if the order of the element is greater than the order of the time discretization method. The second-order CD time discretization method is efficient with $r = 1$, and the insufficiency of computational capacity might be a good reason to use it for $r = 2$ as well. When high accuracy is needed, it is best to use the RK time discretization method and small ϵ. The largest stable time step can be used when $r \leq 4$, but for elements of order $r \geq 5$, smaller time steps are recommended to guarantee high accuracy. According to our computations, the most efficient solution strategy with this controllability algorithm is a combination of the RK time discretization, the largest stable time step and the stopping criterion of $\epsilon = 10^{-6}$ with $r = 4$.

Acknowledgements

The authors thank Dr. Janne Martikainen, Dr. Timo Männikkö, and Dr. Jari Toivanen for useful comments and discussions.

References

[1] L. D. Landau, E. M. Lifshitz, Theory of Elasticity, Pergamon Press, Oxford, 1975.

[2] L. S. Bennethum, X. Feng, A domain decomposition method for solving a Helmholtz-like problem in elasticity based on the Wilson nonconforming element, RAIRO Modélisation Mathématique et Analyse Numérique 31 (1997) 1–25.

[3] C. Farhat, J. Li, An iterative domain decomposition method for the solution of a class of indefinite problems in computational structural dynamics, Applied Numerical Mathematics 54 (2005) 150–166.

[4] S. Collino, F. Ghanemi, P. Joly, Domain decomposition method for harmonic wave propagation: a general presentation, Computer Methods in Applied Mechanics and Engineering 184 (2-4) (2000) 171–211.

[5] E. Faccioli, F. Maggio, A. Quarteroni, A. Tagliani, Spectral domain decomposition methods for the solution of acoustic and elastic wave propagation, Geophysics 61 (1996) 1160–1174.

[6] E. Bécache, P. Joly, C. Tsogka, Application of the fictitious domain method to 2D linear elastodynamic problems, Journal of Computational Acoustics 9 (3) (2001) 1175–1202.

[7] W. Hackbusch, Multigrid Methods and Applications, Springer-Verlag, Berlin, Germany, 1985.

[8] U. Trottenberg, C. W. Oosterlee, A. Schüller, Multigrid, Academic Press, London, 2001.

[9] D. S. Bindel, S. Govindjee, Elastic PMLs for resonator anchor loss simulation, International Journal for Numerical Methods in Engineering 64 (6) (2005) 789–818.

[10] G. K. Gächter, M. J. Grote, Dirichlet-to-Neumann map for three-dimensional elastic waves, Wave Motion 37 (3) (2003) 293–311.

[11] I. Harari, U. Albocher, Studies of FE/PML for exterior problems of time-harmonic elastic waves, Computer Methods in Applied Mechanics and Engineering 195 (29-32) (2006) 3854–3879.

[12] O. Cessenat, B. Després, Using plane waves as base functions for solving time harmonic equations with the ultra weak variational formulation, Journal of Computational Acoustics 11 (2) (2003) 227–238.

[13] T. Huttunen, P. Monk, F. Collino, J. Kaipio, The ultra-weak variational formulation for elastic wave problems, SIAM Journal on Scientific Computing 25 (5) (2004) 1717–1742.

[14] I. Harari, S. Haham, Improved finite element methods for elastic waves, Computer Methods in Applied Mechanics and Engineering 166 (1-2) (1998) 143–164.

[15] L. Zhang, R. Tezaur, C. Farhat, The discontinuous enrichment method for elastic wave propagation in the medium-frequency regime, International Journal for Numerical Methods in Engineering 66 (13) (2006) 2086–2114.

[16] M. Käser, M. Dumbser, An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes I: The two-dimensional isotropic case with external source terms, Geophysical Journal International 166 (2) (2006) 855–877.

[17] B. Cockburn, G. E. Karniadakis, C.-W. Shu (Eds.), Discontinuous Galerkin Methods: Theory, Computation and Applications, Springer-Verlag, Berlin, 2000.

[18] A. Cividini, A. Quarteroni, E. Zampieri, Numerical solution of linear elastic problems by spectral collocation methods, Computer Methods in Applied Mechanics and Engineering 104 (1) (1993) 49–76.

[19] R. Glowinski, J. L. Lions, Exact and approximate controllability for distributed parameter systems (II), Acta Numerica (1995) 159–333.

[20] M. O. Bristeau, R. Glowinski, J. Périaux, Controllability methods for the computation of time-periodic solutions; application to scattering, Journal of Computational Physics 147 (2) (1998) 265–292.

[21] E. Heikkola, S. Mönkölä, A. Pennanen, T. Rossi, Controllability method for acoustic scattering with spectral elements, Journal of Computational and Applied Mathematics 204 (2) (2007) 344–355.

[22] E. Heikkola, S. Mönkölä, A. Pennanen, T. Rossi, Controllability method for the Helmholtz equation with higher-order discretizations, Journal of Computational Physics 225 (2) (2007) 1553–1576.

[23] G. Cohen, Higher-Order Numerical Methods for Transient Wave Equations, Springer-Verlag, Berlin, 2001.

[24] D. Komatitsch, J. P. Vilotte, R. Vai, J. M. Castillo-Covarrubias, F. J. Sánchez-Sesma, The Spectral Element method for elastic wave equations: application to 2D and 3D seismic problems, International Journal for Numerical Methods in Engineering 45 (1999) 1139–1164.

[25] M. O. Bristeau, R. Glowinski, J. Périaux, Using exact controllability to solve the Helmholtz equation at high wave numbers, in: R. Kleinman, T. Angell, D. Colton, F. Santosa, I. Stakgold (Eds.), Mathematical and Numerical Aspects of Wave Propagation, SIAM, Philadelphia, Pennsylvania, 1993, pp. 113–127.

[26] J. Martikainen, A. Pennanen, T. Rossi, Application of an algebraic multigrid method to incompressible flow problems, Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, B 2/2006, Department of Mathematical Information Technology, University of Jyväskylä (2006).

[27] T. Airaksinen, E. Heikkola, A. Pennanen, J. Toivanen, An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation, Journal of Computational Physics 226 (2007) 1196–1210.

[28] C. P. A. Wapenaar, A. J. Berkhout, Elastic Wave Field Extrapolation: Redatuming of Single and Multi-component Seismic Data, Elsevier, Amsterdam, 1989.

[29] B. Engquist, A. Majda, Radiation boundary conditions for acoustic and elastic wave calculations, Communications on Pure and Applied Mathematics 32 (1979) 313–357.

[30] A. Quarteroni, T. Tagliani, E. Zampieri, Generalized Galerkin approximations of elastic waves with absorbing boundary conditions, Computer Methods in Applied Mechanics and Engineering 163 (1998) 323–341.

[31] D. Komatitsch, R. Martin, J. Tromp, M. A. Taylor, B. A. Wingate, Wave propagation in 2-D elastic media using a spectral element method with triangles and quadrangles, Journal of Computational Acoustics 9 (2) (2001) 703–718.

[32] G. Mur, The finite-element modeling of three-dimensional electromagnetic fields using edge and nodal elements, IEEE Transactions on Antennas and Propagation 41 (7) (1993) 948–953.

[33] F. Kickinger, Algebraic multigrid for discrete elliptic second-order problems, in: Multigrid Methods V (Stuttgart, 1996), Springer-Verlag, Berlin, 1998, pp. 157–172.

[34] J. J. Heys, T. A. Manteuffel, S. F. McCormick, L. N. Olson, Algebraic multigrid for higher-order finite elements, Journal of Computational Physics 204 (2) (2005) 520–532.

[35] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition, SIAM, Philadelphia, 2003.