CHALMERS
FINITE ELEMENT CENTER

PREPRINT 2003–01

A hybrid method for elastic waves

Larisa Beilina

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2003

CHALMERS FINITE ELEMENT CENTER

Preprint 2003–01

A hybrid method for elastic waves

Larisa Beilina

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, February 2003

A hybrid method for elastic waves
Larisa Beilina
NO 2003–01
ISSN 1404–4382

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000
Fax: +46 (0)31 772 3595
`www.phi.chalmers.se`

Printed in Sweden
Chalmers University of Technology
Göteborg, Sweden 2003

A HYBRID METHOD FOR ELASTIC WAVES

LARISA BEILINA

Abstract. We study a hybrid finite element/finite difference simulation of the time-dependent elastic wave equation in two and three dimensions. The method is hybrid in the sense that different numerical methods, finite elements and finite differences, are used in different subdomains. The goal of this approach is to combine the flexibility of finite elements and the efficiency of finite differences.

An explicit hybrid method for the elastic wave equation is presented where the finite difference schemes and finite element schemes are used explicitly, applying finite differences on the structured subdomains and finite elements on the unstructured domains.

The hybrid approach is illustrated by the numerical simulations of the elastic wave equation in isotropic case in two and three dimensions with absorbing boundary conditions.

Comparison of the efficiency of the different approaches is a very important aspect of this study. In our test cases, the hybrid approach is about 11 times faster in three dimensional computations than the corresponding highly optimized finite element method. We conclude that the hybrid approach may be an important tool to reduce the execution time and memory requirements for large scale computations.

1. Introduction

In this work we extend our previous study [5] of hybrid finite element/finite difference methods for the simulation of transient acoustic waves, to waves in elastic media. The hybrid FEM/FDM combines the efficiency of FDM on structured grids with the flexibility of FEM on unstructured grids. Our aim is to develop a fast time-domain solver for the time-dependent wave equation (acoustic, elastic, electromagnetic) which allows efficient simulation of wave propagation in complex three dimensional geometry.

With such a solver we may approach a large variety of inverse problems occuring in seismic, non-destructive testing and medical imaging. We have shown part of this potential in [6] with a study of inverse scattering for the acoustic wave equation, and we plan a similar application of the elastic wave solver presented in this note.

In order to evaluate the hybrid approach, the elastic wave equation is simulated in two and three dimensions. The computational domains for this kind of problems often exhibit large regions where the geometry is simple, and small regions where the geometry is complex. In this paper, we use the explicit methods to discretize the time-dependent elastic wave equation in isotropic case. Absorbing boundary conditions are applied at the boundaries of the computational domain.

Date: 3rd February 2003.

Larisa Beilina, Department of Mathematics, Chalmers University of Technology, S–412 96 Göteborg, Sweden, *email*: larisa@math.chalmers.se.

The implementation of the hybrid method is important point of this research. A set of C++ classes is developed in order to handle space discretizations that consist of both structured, Cartesian, grids, to be used with finite differences, and unstructured grids, for usage with finite elements. The techniques of communication between subdomains are also presented.

We present the performance results to compare hybrid approach with a pure finite element method, which show that the hybrid method is faster, in particular for problems where the memory demands are high.

We conclude that our hybrid approach is especially useful for large problems where the computational domain consists of big regions, where Cartesian grids can be used, together with relatively small regions, where the geometry is more complex and unstructured grids have to be used.

The outline of the work is following: in Section 2 we present the mathematical model of the time-dependent wave equation in elastodynamics, in Section 3 we formulate the hybrid FEM/FDM method, in Section 4 we present a fully discrete version used in the computations, in Section 5 we present the numerical simulation of the elastic wave equation in two and three dimensions together with performance comparison.

2. The mathematical model

2.1. **The wave equation in elastodynamic.** Wave propagation in a non-homogeneous anisotropic elastic medium occupying a bounded domain $\Omega \subset \mathbf{R}^d,\ d = 2, 3$, with boundary Γ, is described by the linear wave equation:

$$\rho\, \frac{\partial^2 v}{\partial t^2} - \bigtriangledown \cdot \tau = f, \qquad \text{in } \ \Omega \times (0, T), \tag{2.1}$$

$$\tau = C\ \epsilon, \tag{2.2}$$

$$v = v_0,\ \frac{\partial v}{\partial t} = 0 \tag{2.3}$$

where $v(x, t) \subset \mathbf{R}^d$, is the displacement, τ is the stress tensor, $\rho(x)$ is the density of the material depending on $x \in \Omega$, t is the time variable, T is a final time, and $f(x, t) \subset \mathbf{R}^d$, is a given source function. Further, ϵ is the strain tensor with components

$$\epsilon_{ij} = \epsilon_{ij}(v) = \frac{1}{2}\Big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\Big) \tag{2.4}$$

coupled to τ by Hooke's law

$$\tau_{ij} = \sum_{k=1}^{d}\sum_{l=1}^{d} C_{ijkl}\ \epsilon_{kl}, \tag{2.5}$$

where C is a cyclic symmetric tensor, satisfying

$$C_{ijkl} = C_{klij} = C_{jikl}. \tag{2.6}$$

If the constants $C_{ijkl}(x)$ do not depend on x, the material of the body is said to be homogeneous. If the constants $C_{ijkl}(x)$ do not depend on the choice of the coordinate system, the material of the body is said to be isotropic at the point x. Otherwise, the material is anisotropic at the point x.

In the isotropic case C can be written as

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu(\delta_{ij} \delta_{kl} + \delta_{il} \delta_{jk}), \tag{2.7}$$

where δ_{ij} is Kronecker symbol, in which case (2.5) takes the form of Hooke's law

$$\tau_{ij} = \lambda \delta_{ij} \sum_{k=1}^{d} \epsilon_{kk} + 2\mu \epsilon_{ij}, \tag{2.8}$$

where λ and μ are the Lame's coefficients, depending on x, given by

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \tag{2.9}$$

where E is the modulus of elasticity (Young modulus) and ν is the Poisson's ratio of the elastic material. We have that

$$\lambda > 0, \mu > 0 \iff E > 0, 0 < \nu < 1/2. \tag{2.10}$$

Eliminating the strain tensor using Hooke's law we can verify the elastic wave equation in terms of v only. In the isotropic case with $d = 3$ (2.1) then takes the following form:

$$\begin{aligned}
\rho \frac{\partial^2 v_1}{\partial t^2} &- \frac{\partial}{\partial x_1}\big((\lambda + 2\mu)\frac{\partial v_1}{\partial x_1} + \lambda \frac{\partial v_2}{\partial x_2} + \lambda \frac{\partial v_3}{\partial x_3}\big) \\
&- \frac{\partial}{\partial x_2}\big(\mu\big(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1}\big)\big) \\
&- \frac{\partial}{\partial x_3}\big(\mu\big(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1}\big)\big) = f_1, \\
\rho \frac{\partial^2 v_2}{\partial t^2} &- \frac{\partial}{\partial x_2}\big((\lambda + 2\mu)\frac{\partial v_2}{\partial x_2} + \lambda \frac{\partial v_1}{\partial x_1} + \lambda \frac{\partial v_3}{\partial x_3}\big) \\
&- \frac{\partial}{\partial x_1}\big(\mu\big(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1}\big)\big) \\
&- \frac{\partial}{\partial x_3}\big(\mu\big(\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2}\big)\big) = f_2, \\
\rho \frac{\partial^2 v_3}{\partial t^2} &- \frac{\partial}{\partial x_3}\big((\lambda + 2\mu)\frac{\partial v_3}{\partial x_3} + \lambda \frac{\partial v_2}{\partial x_2} + \lambda \frac{\partial v_1}{\partial x_1}\big) \\
&- \frac{\partial}{\partial x_2}\big(\mu\big(\frac{\partial v_3}{\partial x_2} + \frac{\partial v_2}{\partial x_3}\big)\big) \\
&- \frac{\partial}{\partial x_1}\big(\mu\big(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1}\big)\big) = f_3,
\end{aligned}$$

or in more compact form

$$\rho \frac{\partial^2 v}{\partial t^2} - \nabla \cdot (\mu \nabla v) - \nabla((\lambda + \mu)\nabla \cdot v) = f. \tag{2.11}$$

Inserting a Helmholtz decomposition

$$v = \nabla \phi + \nabla \times \psi \tag{2.12}$$

with a scalar potential ϕ and a vector potential ψ into (2.11), we get

$$\rho \frac{\partial^2}{\partial t^2}(\nabla \phi + \nabla \times \psi) = \mu \triangle (\nabla \phi + \nabla \times \psi) + (\lambda + \mu)\nabla(\nabla \cdot (\nabla \phi + \nabla \times \psi)), \tag{2.13}$$

which using that

$$\begin{aligned} \nabla \cdot (\nabla \phi) &= \triangle \phi, \\ \nabla \cdot (\nabla \times \psi) &= 0, \end{aligned}$$

reduces to

$$\nabla \big(\rho \frac{\partial^2 \phi}{\partial t^2} - (\lambda + 2\mu)\triangle \phi\big) + \nabla \times \big(\rho \frac{\partial^2 \psi}{\partial t^2} - \mu \triangle \psi\big) = 0. \tag{2.14}$$

We conclude that if the potentials ϕ and ψ satisfy the wave equations

$$\rho \frac{\partial^2 \phi}{\partial t^2} - (\lambda + 2\mu)\triangle \phi = 0, \tag{2.15}$$

$$\rho \frac{\partial^2 \psi}{\partial t^2} - \mu \triangle \psi = 0, \tag{2.16}$$

then $v = \nabla \phi + \nabla \times \psi$ satisfies (2.11). We note that $v = \nabla \phi$ corresponds to a pressure wave with speed

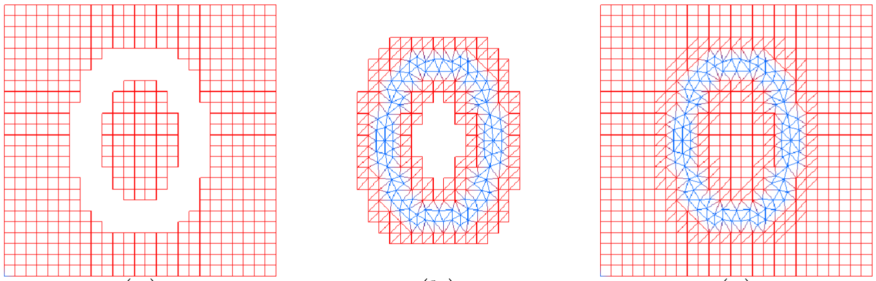
$$V_p = \Big(\frac{\lambda + 2\mu}{\rho}\Big)^{1/2},$$

and $v = \nabla \times \psi$ to a shear wave with speed

$$V_s = \Big(\frac{\mu}{\rho}\Big)^{1/2}.$$

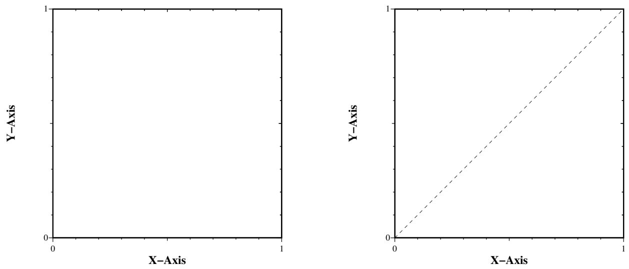
In the pressure wave displacement is parallel to the direction of wave propagation, and the shear wave is orthogonal to the direction of propagation.

2.2. **Communication between FEM/FDM.** In many real-life applications of the time-dependent elastic wave equation ranging from subsurface and underwater imaging to medical and industrial diagnostics, only a small part of the computational domain Ω is complex, where unstructured discretization can be used, whereas quite large regions of the computational domain are sufficiently discretized with simple, Cartesian grids. Consider a domain consisting of two regions, Ω_{FEM} and Ω_{FDM}. In the relatively small Ω_{FEM} subdomain, we assume that an unstructured discretization is appropriate. In the Ω_{FDM} domain, we assume that a structured, Cartesian, grid is suitable. Fig. 1 illustrates the domain decomposition in two dimensions. Our three-dimensional geometries are built up similarly. Fig. 2 shows that the FEM grid is generated such that the thin overlapping domain consists of

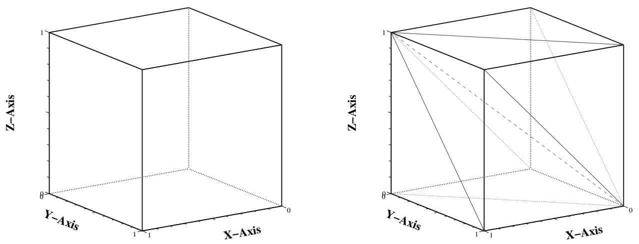


(a) (b) (c)

FIGURE 1. Domain decomposition. The hybrid mesh (c) is a combination of the structured mesh (a) and the unstructured mesh (b) with a thin overlapping of structured elements. The unstructured grid is constructed so that the grid contains edges approximating an ellipse.



(a) A quadrilateral, split into two triangles.

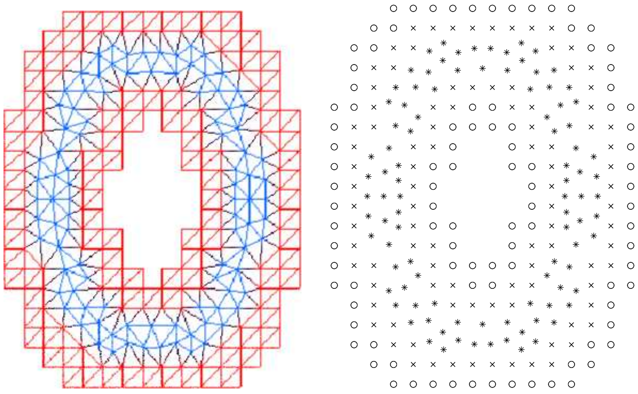


(b) A hexahedron, split into six tetrahedra.

FIGURE 2. In the overlapping domain the finite element grid is created by splitting the structured cells into simplexes as depicted in (a) and (b) for 2D and 3D, respectively.

simplexes obtained by splitting the structured cells. In the interior part of the FEM grid the discretization can be only unstructured.

In most of our test cases, we have used the absorbing boundary condition, taken from [10].



(a) (b)

FIGURE 3. Coupling of FEM and FDM. The nodes of the unstructured FEM grid of (a) is shown in (b), where rings and crosses are nodes which are shared between the FEM and FDM grids. The remaining nodes are marked with stars. The ring nodes are interior to the FDM grid, while the nodes crosses are interior to the FEM grid. At each time iteration, the FDM solution values at ring nodes are copied to the corresponding FEM solution values. At the same time at cross nodes the FEM solution values are copied to the FDM solution values.

2.3. **The numerical method.** In this section we will formulate the hybrid method, which uses a hybrid discretization of the computational domain as described in the previous section.

We observe that the interior nodes of the computational domain belong to either of the following sets:

- ω_o: : Nodes interior to Ω_{FDM} and lying on the boundary of Ω_{FEM},
- $\omega_\times$: : Nodes interior to Ω_{FEM} and lying on the boundary of Ω_{FDM},
- ω_*: : Nodes interior to Ω_{FEM} and not contained in Ω_{FDM},
- ω_D: : Nodes interior to Ω_{FDM} and not contained in Ω_{FEM}.

Fig. 3 illustrates a two-dimensional domain where some nodes are located on the ellipse, which requires an unstructured discretization. The exterior and the interior of the ellipse may use a structured discretization. Nodes belonging to Ω_{FDM} are not shown.

In our algorithm, we store the nodes belonging to ω_o and $\Omega_\times$ twice, both as nodes belonging to Ω_{FEM} and Ω_{FDM}. The main loop of the simulation for explicit time stepping scheme can be formulated as follows:

For every time step we

(1) Update the solution in the interior of Ω_{FDM}, i.e. at nodes ω_D and ω_o using FDM.
(2) Update the solution in the interior of Ω_{FEM}, i.e. at nodes ω_* and $\omega_\times$ using FEM.

(3) Copy values at nodes $\omega_\times$ from Ω_{FEM} to Ω_{FDM}.
(4) Copy values at nodes ω_o from Ω_{FDM} to Ω_{FEM}.

3. Finite element discretization.

We now formulate a finite element method for (2.1) based on using continuous piecewise linear functions in space and time. We discretize $\Omega \times [0,T]$ in the usual way denoting by $K_h = \{K\}$ a partition of the domain Ω into elements K (triangles in $\mathbf{R}^2$ and tetrahedra in $\mathbf{R}^3$ with $h = h(x)$ being a mesh function representing the local diameter of the elements), and we let $J_k = \{J\}$ be a partition of the time interval $I = [0,T]$ into time intervals $J = (t_{k-1}, t_k]$ of uniform length $\tau = t_k - t_{k-1}$. In fully discrete form the resulting method corresponds to a centered finite difference approximation for the second order time derivative and a usual finite element approximation in space.

To formulate the finite element method for (2.1) we introduce the finite element spaces W_h^v defined by :

$$
\begin{aligned}
W^v &:= \{v \in \left[H^1(\Omega \times I)\right]^3 : v(\cdot,0) = 0, v|_\Gamma = 0\}, &(3.1)\\
W_h^v &:= \{v \in W^v : v|_{K\times J} \in [P_1(K) \times P_1(J)]^3, \forall K \in K_h, \forall J \in J_k\}, &(3.2)
\end{aligned}
$$

where $P_1(K)$ and $P_1(J)$ are the set of linear functions on K and J, respectively.

The finite element method now reads: Find $v_h \in W_h^v$, such that

$$
\begin{aligned}
\left(\rho\frac{v_h^{k+1} - 2v_h^k + v_h^{k-1}}{\tau^2}, \bar{v}\right) &+ \left(\mu \nabla v_h^k, \nabla \bar{v}\right) \\
&+ \left((\lambda+\mu)\nabla\cdot v_h, \nabla\cdot\bar{v}\right) = \left(f^k, \bar{v}\right) \quad \forall \bar{v} \in W_h^v, \\
&v_h(0) = \dot{v}_h(0) = 0.
\end{aligned} \tag{3.3}
$$

4. Fully discrete scheme

Expanding v in terms of the standard continuous piecewise linear functions $\varphi_i(x)$ in space and $\psi_i(t)$ in time and substituting this into (3.3), the following system of linear equations is obtained:

$$
\begin{aligned}
M(\mathbf{v}^{k+1} - 2\mathbf{v}^k + \mathbf{v}^{k-1}) &= \frac{\tau^2}{\rho}F^k - \frac{\tau^2}{\rho}\mu K(\frac{1}{6}\mathbf{v}^{k-1} + \frac{2}{3}\mathbf{v}^k + \frac{1}{6}\mathbf{v}^{k+1}) \\
&- \frac{\tau^2}{\rho}(\lambda+\mu)D\mathbf{v}^k,
\end{aligned} \tag{4.1}
$$

with initial conditions $v^0 = v^1 = 0$.

Here, M is the mass matrix in space, K is the stiffness matrix, D is the divergence matrice, $k = 1, 2, 3\ldots$ denotes the time level, F^k is the load vector, $\mathbf{v} = (\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3})$ is the unknown discrete field values of $v = (v_1, v_2, v_3)$, and τ is the time step.

The explicit formulas for the entries in (4.1) at the element level can be given as:

$$M^e_{i,j} = (\varphi_i, \varphi_j)_e, \tag{4.2}$$

$$K^e_{i,j} = (\nabla\varphi_i, \nabla\varphi_j)_e, \tag{4.3}$$

$$D^e_{i,j} = (\nabla\cdot\varphi_i, \nabla\cdot\varphi_j)_e, \tag{4.4}$$

$$F^e_{j,m} = (f, \varphi_j \psi_m)_{e\times J}, \tag{4.5}$$

where $(.,.)_e$ denotes the $L_2(e)$ scalar product. The matrix M_e is the contribution from element e to the global assembled matrix in space M, K^e is the contribution from element e to the global assembled matrix K, D^e is the contribution from element e to the global assembled matrix D, F^e is the contribution from element e to the assembled source vector F.

To obtain an explicit scheme we approximate M with the lumped mass matrix M^L, given by

$$M^L_{i,j} = \begin{cases} \sum_n M_{i,n} & , \quad i = j, \\ 0 & , \quad i \neq j, \end{cases} \tag{4.6}$$

where the diagonal approximation is obtained by taking the row sum of M, see e.g. [12].

By multiplying (4.1) with $(M^L)^{-1}$ and replacing the terms $\frac{1}{6}\mathbf{v}^{k-1} + \frac{2}{3}\mathbf{v}^k + \frac{1}{6}\mathbf{v}^{k+1}$ by $\mathbf{v}^k$, we obtain an efficient explicit formulation

$$\begin{aligned} \mathbf{v}^{k+1} &= \frac{\tau^2}{\rho}(M^L)^{-1}F^k + 2\mathbf{v}^k - \frac{\tau^2}{\rho}\mu(M^L)^{-1}K\mathbf{v}^k \\ &- \frac{\tau^2}{\rho}(\lambda+\mu)(M^L)^{-1}D\mathbf{v}^k - \mathbf{v}^{k-1}. \end{aligned} \tag{4.7}$$

4.1. **Finite difference formulation.** In this section we present the finite difference discretization for elastic wave equation in three dimensions.

Using centered finite difference approximation for the second order time derivative and a following approximation of the cross derivative (see [7]) appearing in the scheme

$$\frac{\partial^2 w}{\partial x \partial y} \approx \frac{w^k_{i+1,j+1,l} - w^k_{i-1,j+1,l} - w^k_{i+1,j-1,l} + w^k_{i-1,j-1,l}}{4dxdy}, \tag{4.8}$$

we get following finite difference discretization for elastic wave equation (2.1) in three dimensions

$$\begin{aligned} v^{k+1}_{1_{i,j,l}} &\approx \frac{\tau^2}{\rho}(f^k_{i,j,l} + \mu\Delta v^k_{1_{i,j,l}}) + 2v^k_{1_{i,j,l}} - v^{k-1}_{1_{i,j,l}} \\ &+ (\lambda+\mu)\frac{v^k_{1_{i+1,j,l}} - 2v^k_{1_{i,j,l}} + v^k_{1_{i-1,j,l}}}{dx^2} \\ &+ (\lambda+\mu)\frac{v^k_{2_{i+1,j+1,l}} - v^k_{2_{i-1,j+1,l}} - v^k_{2_{i+1,j-1,l}} + v^k_{2_{i-1,j-1,l}}}{4dxdy} \\ &+ (\lambda+\mu)\frac{v^k_{3_{i+1,j,l+1}} - v^k_{3_{i-1,j,l+1}} - v^k_{3_{i+1,j,l-1}} + v^k_{3_{i-1,j,l-1}}}{4dxdz}, \end{aligned} \tag{4.9}$$

$$
\begin{aligned}
v_{2_{i,j,l}}^{k+1} \approx\ & \frac{\tau^2}{\rho}(f_{i,j,l}^k + \mu \Delta v_{2_{i,j,l}}^k) + 2v_{2_{i,j,l}}^k - v_{2_{i,j,l}}^{k-1} \\
+\ & (\lambda+\mu)\frac{v_{2_{i,j+1,l}}^k - 2v_{2_{i,j,l}}^k + v_{2_{i,j-1,l}}^k}{dy^2} \\
+\ & (\lambda+\mu)\frac{v_{1_{i+1,j+1,l}}^k - v_{1_{i-1,j+1,l}}^k - v_{1_{i+1,j-1,l}}^k + v_{1_{i-1,j-1,l}}^k}{4dxdy} \\
+\ & (\lambda+\mu)\frac{v_{3_{i,j+1,l+1}}^k - v_{3_{i,j-1,l+1}}^k - v_{3_{i,j+1,l-1}}^k + v_{3_{i,j-1,l-1}}^k}{4dydz},
\end{aligned} \tag{4.10}
$$

$$
\begin{aligned}
v_{3_{i,j,l}}^{k+1} \approx\ & \frac{\tau^2}{\rho}(f_{i,j,l}^k + \mu \Delta v_{3_{i,j,l}}^k) + 2v_{3_{i,j,l}}^k - v_{3_{i,j,l}}^{k-1} \\
+\ & (\lambda+\mu)\frac{v_{3_{i,j,l+1}}^k - 2v_{3_{i,j,l}}^k + v_{3_{i,j,l-1}}^k}{dz^2} \\
+\ & (\lambda+\mu)\frac{v_{1_{i+1,j,l+1}}^k - v_{1_{i-1,j,l+1}}^k - v_{1_{i+1,j,l-1}}^k + v_{1_{i-1,j,l-1}}^k}{4dxdz} \\
+\ & (\lambda+\mu)\frac{v_{2_{i,j+1,l+1}}^k - v_{2_{i,j-1,l+1}}^k - v_{2_{i,j+1,l-1}}^k + v_{2_{i,j-1,l-1}}^k}{4dydz},
\end{aligned} \tag{4.11}
$$

where $v_{s_{i,j,l}}^k,\ s=1,2,3$ is the solution on time iteration k at point (i,j,l), $f_{i,j,l}^k$ is the source function, τ is the time step, and $\Delta v_{s_{i,j,l}}^k,\ s=1,2,3$ is the discrete Laplacian :

$$
\begin{aligned}
\Delta v_{s_{i,j,l}}^k =\ & \frac{v_{s_{i+1,j,l}}^k - 2v_{s_{i,j,l}}^k + v_{s_{i-1,j,l}}^k}{dx^2} + \frac{v_{s_{i,j+1,l}}^k - 2v_{s_{i,j,l}}^k + v_{s_{i,j-1,l}}^k}{dy^2} + \\
& \frac{v_{s_{i,j,l+1}}^k - 2v_{s_{i,j,l}}^k + v_{s_{i,j,l-1}}^k}{dz^2},
\end{aligned} \tag{4.12}
$$

where dx, dy, and dz are the steps of the discrete finite difference meshes in the directions x, y, z, respectively.

4.2. **Dispersion relation.** In this section we present the dispersion relations for the time-dependent elastic wave equation. These relations allows as to get the information on the stability and accuracy properties of the numerical models.

Searching for a plane wave solution of the homogeneous two-dimensional elastic equation in the form

$$v_1 = v_{1_0} e^{i(\omega t + k_1 x + k_2 y)}, \tag{4.13}$$

$$v_2 = v_{2_0} e^{i(\omega t + k_1 x + k_2 y)} \tag{4.14}$$

we get two dispersion relations by identifying eigenvalues of the problem

$$AX = \rho \sin^2 \frac{\omega \tau}{2} X, \tag{4.15}$$

where $X = (v_{1_0}, v_{2_0})$ and

$$A = \frac{\tau^2}{h^2}\begin{pmatrix} (\lambda+2\mu)\sin^2\frac{k_1h}{2} + \mu\sin^2\frac{k_2h}{2} & (\lambda+\mu)\sin\frac{k_1h}{2}\sin\frac{k_2h}{2}\cos\frac{k_1h}{2}\cos\frac{k_2h}{2} \\ (\lambda+\mu)\sin\frac{k_1h}{2}\sin\frac{k_2h}{2}\cos\frac{k_1h}{2}\cos\frac{k_2h}{2} & (\lambda+2\mu)\sin^2\frac{k_2h}{2} + \mu\sin^2\frac{k_1h}{2}, \end{pmatrix}$$

which gives

$$\begin{aligned} \rho\sin^2\frac{\omega\tau}{2} &= \frac{\tau^2}{h^2}\Big((\lambda+2\mu)(\sin^2\frac{k_1h}{2} + \sin^2\frac{k_2h}{2}) \\ &- (\lambda+\mu)\sin^2\frac{k_1h}{2}\sin^2\frac{k_2h}{2}\Big), \end{aligned} \tag{4.16}$$

$$\begin{aligned} \rho\sin^2\frac{\omega\tau}{2} &= \frac{\tau^2}{h^2}\Big(\mu(\sin^2\frac{k_1h}{2} + \sin^2\frac{k_2h}{2}) \\ &+ (\lambda+\mu)\sin^2\frac{k_1h}{2}\sin^2\frac{k_2h}{2}\Big). \end{aligned} \tag{4.17}$$

4.3. **Stability criterion.** To determine the time step restriction for stability of our explicit method, we assume $\sin\frac{\omega t}{2} \leq 1$ and $k_1h = k_2h = k_3h = \pi$ corresponding to the highest spatial frequency resolved by the grid, and deduce from (4.16 - 4.17) the CFL condition:

$$\tau \leq h\sqrt{\frac{\rho}{\lambda+3\mu}}. \tag{4.18}$$

4.4. **Absorbing boundary conditions.** We have also simulated a variation of the problem (2.1) applying absorbing boundary conditions at the boundary Ω_{FDM}. It means, that these boundary conditions approximate the solution on the boundaries. We use the following boundary condition taken from [10] :

$$\frac{\partial}{\partial t}u - \frac{\partial}{\partial x}u\Big|_{x=0} = 0. \tag{4.19}$$

We are using forward finite difference approximation in the middle point of the condition (4.19), which gives a numerical approximation of a higher order than the ordinary (backward or forward) approximation. For example, for the left boundary of the Ω_{FDM} we obtain:

$$\begin{aligned} &\frac{u_{i,j,l}^{k+1} - u_{i,j,l}^{k}}{dt} + \frac{u_{i+1,j,l}^{k+1} - u_{i+1,j,l}^{k}}{dt} - \\ &\frac{u_{i+1,j,l}^{k} - u_{i,j,l}^{k}}{dx} - \frac{u_{i+1,j,l}^{k+1} - u_{i,j,l}^{k+1}}{dx} = 0, \end{aligned} \tag{4.20}$$

which can be transformed to

$$\begin{aligned} u_{i,j,l}^{k+1} = u_{i+1,j,l}^{k} \ &+ \ u_{i,j,l}^{k}\frac{dx-dt}{dx+dt} - \\ &u_{i+1,j,l}^{k+1}\frac{dx-dt}{dx+dt}. \end{aligned} \tag{4.21}$$

For other boundaries of the Ω_{FDM} we find analogous boundary conditions.

4.5. **Neumann boundary condition.** The problem (2.1) have also been simulated applying Neumann boundary conditions at the boundary Ω_{FDM}.

If the surface forces $T_i \in C(\Gamma), i = 1, 2, 3$ are prescribed on the boundary Γ this leads to the condition

$$\tau_{ij} n_j = T_i, \;\; x \in \Gamma \tag{4.22}$$

where τ_{ij} are stresses (2.5) and n_j are the components of the unit outer normal to Γ. Using the generilized Hooke's law for isotropic material, we get

$$\lambda \delta_{ij} n_i \sum_{k=1}^{d} \epsilon_{kk} + 2\mu \epsilon_{ij} n_j = T_i. \tag{4.23}$$

In this case we seek a solution $u_i \in C^2(\Omega), i = 1, 2, 3,$ which satisfies (2.1) and (4.23). The explicit expressions for the computations of the surface forces $T_i \in C(\Gamma), i = 1, 2, 3$ at the boundary Γ in the three dimensional case take the following form:

$$\begin{aligned}
T_1 &= (\lambda + 2\mu) \frac{\partial v_1}{\partial x_1} n_1 + \lambda (\frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3}) n_1 + \mu (\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1}) n_2 + \mu (\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1}) n_3, \\
T_2 &= (\lambda + 2\mu) \frac{\partial v_2}{\partial x_2} n_2 + \lambda (\frac{\partial v_1}{\partial x_1} + \frac{\partial v_3}{\partial x_3}) n_2 + \mu (\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2}) n_3 + \mu (\frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2}) n_1, \\
T_3 &= (\lambda + 2\mu) \frac{\partial v_3}{\partial x_3} n_3 + \lambda (\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2}) n_3 + \mu (\frac{\partial v_3}{\partial x_2} + \frac{\partial v_2}{\partial x_3}) n_2 + \mu (\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1}) n_1.
\end{aligned}$$

5. Implementational issues

To implement the hybrid FEM/FDM method we use C++. This language allows us to implement the problem and the algorithms on a high level of abstraction without much loss of efficiency. Important notions such as grid, boundary, elastic equation operator, input/output of results are implemented as C++ classes. The new class `ElasticEqEpOpt` was written for implementation of the hybrid method for elastic wave equation. This class combines previously written classes for implementation of FDM, FEM for scalar wave equation (see [5] for more details) and also include new classes for FEM/FDM method to implement elastic wave equation in two and three dimensions. The software package PETSc [3] is used for matrix vector computations.

We present the use of the hybrid method on problem (2.1). For flexible implementation the class `ElasticEqEpOpt` have different constructors which are configured either to use FDM in the whole region (using a structured grid), or to use FEM in the whole region (using a hybrid or a structured grid), or to use only the hybrid method. In a simplified form, the main program looks as following (Compare with the algorithm in Section 2.3):

```
ElasticEqOpOpt(gg,                       # constructor with meshes
       sdg,                   # for FEM and FDM, space
       nsd,                   # dimensions, bool
       EXCHANGE,              # variables, values
```

```
       USE_FEM,                  # of the parameters
       USE_FDM,
       USE_RHS,
       USE_DIRICHLET_FEM,
       USE_DIRICHLET_FDM,
       USE_ABSORB,
       PRINT_FILES,
       nrSTEPS,
       lambda,mu,
       maxtime,rhs,
       type_of_material,
       velocity,
       guess_velocity);
   dt = p.InitTime();                     # time initialization
   notimesteps = int(maxtime/dt);
    if (EXCHANGE)                         # for hybrid method,
      {                                   # initialization FDM, FEM,
p.InitFDM();            # and EXCHANGE in the common nodes
p.InitFEMforElasticMat(type_of_material,
       velocity,
       mu,
       lambda);
p.InitExchangeCommon();
      }
    if (USE_FDM)
p.InitFDM();                             # initialization FDM
    if (USE_FEM)
      p.InitFEMforElasticMat(type_of_material,  # initialization FEM
     velocity,
     mu,
     lambda);
    MV_Vector<real> thetimesteps(notimesteps);
    for( k = 0; k < notimesteps; k++)          # main loop
      {
      if (EXCHANGE)                            # for hybrid method
{
  p. ElasticFDM( k);                     # perform one time step
                                                # using FDM
  p.DifMatElastiqEqSolver(t,k,           # perform one time step
         type_of_material,     # using FEM
  velocity);
  p.ApplyExchangeCommon();               #  perform exchanges
  p.ApplySwapElasticFEM();               # swap solution vectors
  p.ApplySwapElasticFDM();
}
      if (USE_FDM)
```

```
{
  p.ElasticFDM( k);
  p.ApplySwapElasticFDM();
}
      if (USE_FEM)
{
  p.DifMatElastiqEqSolver(t,k,
  type_of_material,
  velocity);
  p.ApplySwapElasticFEM();
}
       thetimesteps(k)=t;
       t+=dt;
     }
```

The main work is done for each time step in `DifMatElastiqEqSolver` and `elasticFDM` where the source function f is evaluated and matrix vector multiplications are performed.

5.1. **Notes on optimization.** In this section are described some of the optimization steps for evaluation of the hybrid method. They are similar to the ones, used for the scalar wave equation, see [5] .

In (4.7) we can write $2\mathbf{v_s}^k - \frac{\tau^2}{\rho}\mu(M^L)^{-1}K\mathbf{v_s}^k = A\mathbf{v_s}^k,\ s = 1,2,3,$ where $A = 2I - \frac{\tau^2}{\rho}\mu(M^L)^{-1}K$ is a sparse matrix, and I the identity matrix. If the time step τ is constant, A is independent of time and can therefore be computed in the initialization step. We can write the term for the divergence matrix in a similar way $\frac{\tau^2}{\rho}(\lambda+\mu)(M^L)^{-1}D\mathbf{v_s}^k = DV\mathbf{v_s}^k$, where $\frac{\tau^2}{\rho}(\lambda+\mu)(M^L)^{-1}D = DV$ and DV is following block-matrix in R^3:

$$DV = \begin{pmatrix} DV_{11} & DV_{12} & DV_{13} \\ DV_{21} & DV_{22} & DV_{23} \\ DV_{31} & DV_{32} & DV_{33} \end{pmatrix}$$

Here, $DV_{ij},\ i,j = 1,2,3$ are the matrices appearing in the assembling of the divergence matrix. Again, for constant time step matrix DV is independent on time and can be computed in the initialization step. A similar optimization is used for FDM.

Regarding the computation of the load vector F^k in (4.7), a standard assembly computation using Gaussian quadrature is expensive.

To compute the load vector F^k in (4.7) the nodal quadratures are used. Then the load vector will be $F^k = (M_L)^{-1}\mathbf{f}^k$, where the vector $\mathbf{f}^k$ is the source function evaluated at the nodes of the grid at time t_k. Alternatively, the source function can be approximated by the finite element function obtained by interpolating the source function at the nodes. In this case, exact quadrature yields that $F^k = M\mathbf{f}^k$.

size of the mesh elements, h	number of nodes in Ω_{FEM}	number of nodes in Ω	number of nodes in the overlapping layers	spatial dimension
0.0025	6561	160801	3192	2
0.005	1681	40401	1592	2
0.01	441	10201	152	2
0.02	121	2601	72	2
0.01	9261	1030301	4348	3
0.02	1331	132651	988	3
0.04	216	17576	208	3

TABLE 1. Meshes for the performance test.

Thus, we get the following expressions for FEM computations of the solution $\mathbf{v_s}^{k+1},\ s = 1,2,3$

$$\mathbf{v_1}^{k+1} = \frac{\tau^2}{\rho}\mathbf{f}^k + A\mathbf{v_1}^k + DV_{11}\mathbf{v_1}^k + DV_{12}\mathbf{v_2}^k + DV_{13}\mathbf{v_3}^k - \mathbf{v_1}^{k-1}, \tag{5.1}$$

$$\mathbf{v_2}^{k+1} = \frac{\tau^2}{\rho}\mathbf{f}^k + A\mathbf{v_2}^k + DV_{21}\mathbf{v_1}^k + DV_{22}\mathbf{v_2}^k + DV_{23}\mathbf{v_3}^k - \mathbf{v_2}^{k-1}, \tag{5.2}$$

$$\mathbf{v_3}^{k+1} = \frac{\tau^2}{\rho}\mathbf{f}^k + A\mathbf{v_3}^k + DV_{31}\mathbf{v_1}^k + DV_{32}\mathbf{v_2}^k + DV_{33}\mathbf{v_3}^k - \mathbf{v_3}^{k-1}, \tag{5.3}$$

As mentioned, the matrices A and $DV = DV_{ij},\ i,j = 1,2,3$ are sparse. Using the same motivation, as in [5], we replace the sparse matrices A and DV with matrices where the zero components are eliminated. Then a matrix vector product with a new reduced matrix has a correspondingly smaller cost. The size of the reduced matrix A is $5/7$ times the size of the original matrix in R^2 and $7/15$ in R^3. The correspondings factors for the divergence matrices DV_{11} and DV_{22} are $3/7$ in R^2. The factors for matrices $DV_{11}, DV_{22}, DV_{33}$ are $3/15$ in R^3. For $DV_{ij},\ i,j = 1,2,3, i \neq j$ the factors are 1. These factors appear during assembling the matrices A and DV. The numbers of zero and nonzero elements for matrices $A,\ DV$ are confirmed in the numerical tests, see Table 2 for two dimensional computations and Table 3 for three dimensional computations. The computational grids for these tests are presented in Table 1 .

The reader is referred to [5] for other optimizations in the C++ code. Compare also with [2, 4] which are dealing with optimizing C++ code and studying the effect of the cache sizes of the computer, respectively.

6. NUMERICAL EXAMPLES

In this section we present the use of the hybrid method on several examples. In the three dimensional examples we simulate model problem (2.1) in the domain $\Omega = [0,5.0]\times[0,2.5]\times[0,2.5]$, with absorbing boundary conditions, and with initial conditions $v_i = \frac{\partial v_i}{\partial t} = 0,\ i = 1,2,3$. In two dimensional examples the computational domain is $\Omega = [0,1.0]^2$. The domain

number of nodes in Ω_{FEM}	number of nonzero elements, A	number of zero elements, A	number of nonzero elements, DV_{11}, DV_{22}	number of zero elements, DV_{11}, DV_{22}
6561	32481	12800	19521	25760
1681	8241	3200	4961	6480
441	2121	800	1281	1640
121	561	200	341	420

TABLE 2. Number of zero and nonzero elements in two dimensional computations of the sparse matrices $A,\ DV_{11},\ DV_{22}$

number of nodes in Ω_{FEM}	number of nonzero elements, A	number of zero elements, A	number of nonzero elements, $DV_{ij},\ i=j$	number of zero elements, $DV_{ij},\ i=j$
1030301	7150901	8060000	3070501	12140400
132651	8241	3200	4961	6480
17576	2121	800	1281	1640

TABLE 3. Number of zero and nonzero elements in three dimensional computations of the sparse matrices $A,\ DV_{ij},\ i=j$

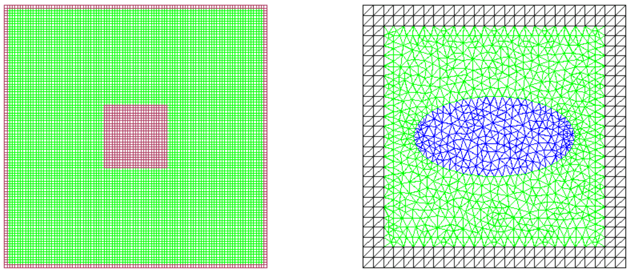
h	Hybrid	FDM	FEM
0.01	38.9	25.2	460
0.02	4.8	2.6	57
0.04	0.6	0.3	7.2

TABLE 4. Computational time in seconds to perform one step in elastic equation solver. For computations are used three dimensional meshes presented in Table 1.

Ω is decomposed into the two domains Ω_{FEM} and Ω_{FDM} with two overlapping layers of nodes. The inner domain in three dimensional tests is $\Omega_{FEM} = [0.3, 4.7] \times [0.3, 2.3] \times [0.3, 2.3]$ and in two dimensional tests $\Omega_{FEM} = [0.4, 0.6]^2$. In the Ω_{FDM} we apply FDM with absorbing boundary conditions. The space mesh in two dimensional examples in the Ω_{FEM} is unstructured and consists of triangles. In three dimensions we use tetrahedra for the unstructured grid. We present some examples with spherical pulses, generated at the different points in Ω_{FEM} or in Ω_{FDM}, which are given by the source functions

$$f_1(x, x_0) = \begin{cases} 10^3 \sin^2 \pi t & \text{if } 0 \le t \le 0.1 \text{ and } |x - x_0| < r, \\ 0 & \text{otherwise;} \end{cases} \tag{6.1}$$

$$f_2(x, x_0) = \begin{cases} -\pi^2 \sin \pi t & \text{if } 0 \le t \le 0.1 \text{ and } |x - x_0| < r, \\ 0 & \text{otherwise;} \end{cases} \tag{6.2}$$



a) b)

FIGURE 4. Computational grids for the two-dimensional numerical tests. In the graph a) we present mesh for FDM computations , and in the graph b) for FEM computations. The computational mesh for hybrid method is a combination of the structured mesh a) and the unstructured mesh b) with a thin overlapping of structured elements.

First, we present a few model applications in two and three dimensions. Next, we present performance comparisons of the different methods and efficiency results in terms of cpu time per node and time iteration.

6.1. **Two-dimensional examples.** In the first example we solve the problem (2.1) with absorbing boundary conditions on the Ω_{FDM} boundary. We have chosen the explicit scheme (4.7) to implement the FEM and the explicit scheme (4.9 – 4.11) to implement FDM. Computational tests were performed during the time interval $[0, 1.0]$ and with source function f_1, which was initialized at the point $(0.5, 0.5)$. The time step is $\tau \approx 0.001$ and satisfied to criterion (4.18) with the density $\rho = 1.0$ and Lame coefficients $\lambda = \mu = 0.5$. We plot the u_2 component of the hybrid method solution in Fig. 8, and $u = \sqrt{{u_1}^2 + {u_2}^2}$ in Fig. 9.

In Fig. 21 we show the solutions of the two dimensional elastic wave equation with absorbing boundary conditions at the one point with coordinates $(0.5, 0.5)$. In the graph Fig. 21-a we present FDM and hybrid solutions on the mesh with element size $h = 0.0025$, in the graphs Fig. 21-b and Fig. 21-c we present FDM and Hybrid solutions, respectively, on the meshes with element sizes $h = 0.0025, 0.05, 0.1$.

In the second example we present hybrid method for the two-dimensional elastic wave equation with absorbing boundary conditions in the inhomogeneous Ω_{FDM}. The domain Ω composed of different material types having different density ρ. We have chosen a source function located at the center of the unstructured domain, presented in Fig. 4-b. The coefficient is taken as $\rho = 0.5$ inside the elliptical domain (see Fig. 4-b) and $\rho = 1.0$ outside it. In Fig. 10-a we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FDM}, in Fig. 10-b we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FEM}, in Fig. 10-c,d are presented u_1 and u_2 components of the hybrid solution.

In the third example we present a plane wave propagation. The plane wave, initialized on the left boundary of the Ω_{FDM} and moving in the positive x direction, has the form

h	Hybrid	FDM	FEM	FEM/Hybrid	FEM/FDM
0.0025	1.19402e-06	1.1534e-06	2.62187e-06	2.1958	2.2732
0.005	1.19304e-06	1.19242e-06	3.28700e-06	2.7551	2.7566
0.01	1.37241e-06	1.11754e-06	2.9507e-06	2.1500	2.6404
0.02	1.23030e-06	1.16505e-06	3.46020e-06	2.8125	2.9700

TABLE 5. Performance for the two dimensional elastic wave equation.

h	Hybrid	FDM	FEM	FEM/Hybrid	FEM/FDM
0.01	3.85363e-06	2.22983e-06	4.41040e-05	11.448	19.7791
0.02	4.45080e-06	2.06256e-06	4.50294e-05	11.1171	21.8318
0.04	4.08512e-06	2.12790e-06	4.43559e-05	10.8579	20.8449

TABLE 6. Performance for three dimensioanl elastic wave equation.

$u = (u_1, 0, 0)$, where

$$u_1(x,y,z,t) \mid_{x=0} = 0.1 \sin\left(25 \left(t - \frac{2\pi}{25}\right) - \pi/2\right) + 0.1, \; 0 \leq t \leq \frac{2\pi}{25}. \tag{6.3}$$

We make tests using the hybrid method with the time step $\tau \approx 0.001$. The time-dependent computational solution in different time moments for hybrid method is presented in Fig. 5, the same solution but only in Ω_{FDM} is presented in Fig. 6 and solution only in Ω_{FEM} is presented in Fig. 7.

6.2. **Three-dimensional examples.** We demonstrate the use of the hybrid method on the domain $\Omega = [0, 5.0] \times [0, 2.5] \times [0, 2.5]$, with absorbing boundary conditions. For the numerical simulations we have chosen a finite element mesh with element size 0.04 and perform computations in the time interval $[0, 1.6]$. We compute with time step $\tau \approx 0.00266$, with the density $\rho = 1.0$ and Lame coefficients $\lambda = \mu = 0.5$. In the first example we are solving the model problem (2.1) with absorbing boundary conditions and with the one source function f_1 initialized at the center of Ω_{FEM}. We show the time-dependent hybrid method solution in Fig. 14.

The second example is solution of model problem (2.1) with absorbing boundary conditions and with two source functions, initialized in the Ω_{FDM}. The hybrid solution of this problem we present in Fig. 15, Fig. 16, Fig. 17.

The third example is the same as second one, but now with elements, which have different values of the density in the Ω_{FEM}. We present contour fill for the scalar result of the common solution $u = \sqrt{{u_1}^2 + {u_2}^2 + {u_3}^2}$ in Fig. 20. The values of the Lame coefficients are $\lambda, \mu = 0.5$, density $\rho = 2.0$ on the elements forming cone, and $\rho = 1.0$ in the rest of the domain.

In Figure 19-a we show the time-dependent hybrid and FDM plane waves solutions at the one point with coordinates $(0.4, 0.4, 0.4)$. The computational domain for these tests is cube $\Omega = [0,1] \times [0,1] \times [0,1]$ with element size $h = 0.02$. We perform computations during the time $[0, 1.0]$ with the time step $\tau = 0.001$, density $\rho = 1.0$ and Lame coefficients

$\lambda = \mu = 0.5$. The plane wave, initialized on the left boundary of the Ω_{FDM} and moving in the positive x direction, has the form $u = (u_1, 0, 0)$, where

$$u_1(x,y,z,t)\mid_{x=0} = 0.1\sin\left(25\left(t - \frac{2\pi}{25}\right) - \pi/2\right) + 0.1,\ \ 0 \le t \le \frac{2\pi}{25}. \tag{6.4}$$

In Figure 19-b), c), d) we show the time-dependent hybrid and FDM solutions at the one point with coordinates $(0.4, 0.4, 0.4)$. For this example we solved the elastic equation with absorbing boundary conditions and one source function, initialized at the point $(0.5, 0.7, 0.7)$ in Ω_{FDM}. The Tables 7 - 8 show the Hybrid and pure FDM solutions for previous example at the different time moments. We see, that Hybrid and FDM solutions slightly differ because the FDM and FEM stencils on a structured mesh are not completely identical.

6.3. **Performance comparisons.** To show the performance of the different methods we computed time-dependent elastic wave equation on structured grids, presented in Table 1, measuring the cpu time per node and per iteration.

The benchmarks were run on SUN workstation with 2069376K total memory, 1417844K used and 649208K free memory.

Table 5 and Table 6 present efficiency results, in terms of cpu time per node and iteration. The fractions FEM/Hybrid and FEM/FDM are also presented in the tables. We note that, for two dimensions, the fraction FEM/Hybrid ≈ 2.5 and the fraction FEM/FDM ≈ 2.7. In our three-dimensional tests, the corresponding fractions have increased. Here, the fraction FEM/Hybrid ≈ 11, and the fraction FEM/FDM is around 20.

The tables show that the fractions increase with the size of the grid. This can be explained by cache effects, since the required memory of the FEM sparse matrix is much larger than the corresponding FDM difference molecule. Another effect of importance is that the nodes at the boundary is making up a smaller part of the total number of nodes. For the hybrid method, the relative cost associated with computing the solution in the overlap region with both methods and exchanging solution values, decreases as the grid sizes increase, compare with Table 1.

We refer to [5] for remarks on the performance comparisons and memory consumption of hybrid method.

time it.	Hybrid, u_1	Hybrid, u_2	Hybrid, u_3	Hybrid,u
4	0.000001	0.000004	0.000004	0.000007
5	0.000053	0.000210	0.000209	0.000334
6	0.000852	0.002816	0.002806	0.004359
7	0.004986	0.015861	0.015863	0.024075
8	0.015755	0.049427	0.049462	0.073775
9	0.030699	0.093020	0.092995	0.137026
10	0.051720	0.132620	0.132203	0.196366
11	0.082044	0.169849	0.170238	0.255907
12	0.109924	0.203353	0.203432	0.309652
13	0.136333	0.232136	0.232058	0.356931
14	0.163883	0.256261	0.256631	0.399301
15	0.179513	0.282381	0.282537	0.439262

TABLE 7. The time-dependent hybrid solution at the one point with coordinates $(0.4, 0.4, 0.4)$.

time it.	FDM, u_1	FDM, u_2	FDM , u_3	FDM, u
4	0.000001	0.000005	0.000005	0.000008
5	0.000063	0.000218	0.000218	0.000348
6	0.000859	0.002808	0.002808	0.004354
7	0.004859	0.015759	0.015759	0.023899
8	0.015490	0.049374	0.049374	0.073645
9	0.031605	0.094499	0.094499	0.139392
10	0.053441	0.133898	0.133898	0.198797
11	0.083259	0.172020	0.172020	0.258965
12	0.111539	0.204361	0.204361	0.311519
13	0.137541	0.233072	0.233072	0.358595
14	0.165049	0.257830	0.257830	0.401586
15	0.180538	0.282534	0.282534	0.439811

TABLE 8. The time-dependent FDM solution at the one point with coordinates $(0.4, 0.4, 0.4)$.

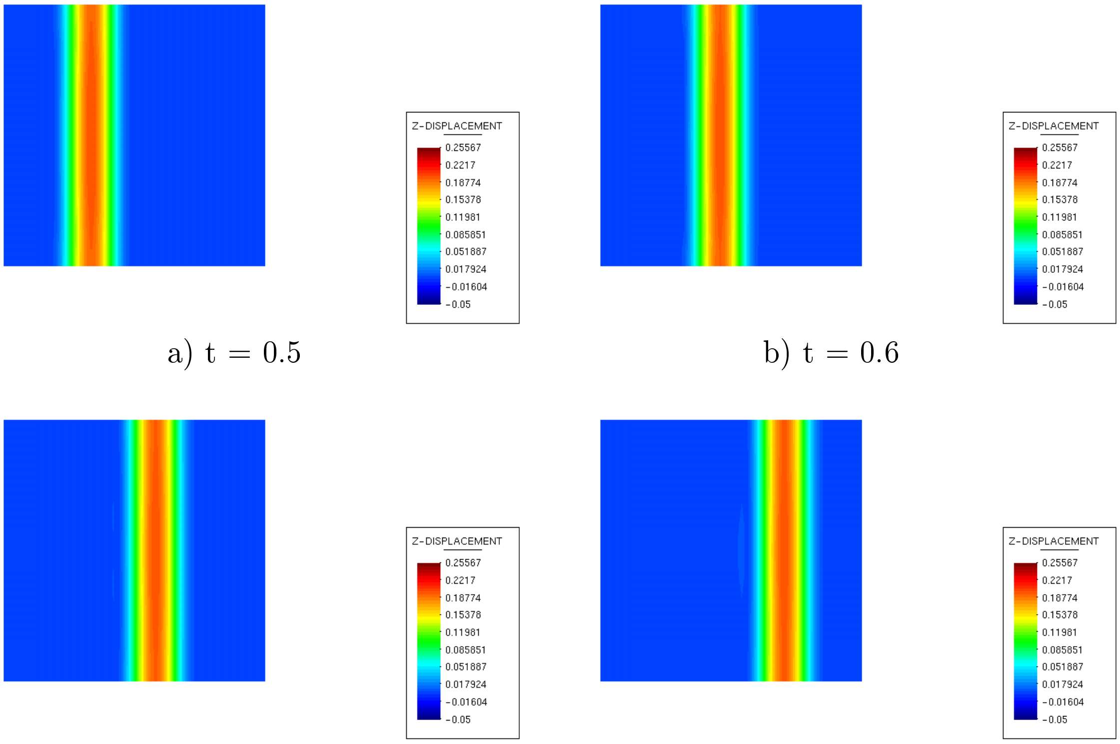
7. Conclusions

The explicit hybrid methods for the time-dependent elastic wave equation is presented. The efficiency of FDM is combined with the flexibility of FEM.

Object-oriented C++ classes are developed as independent modules for easy application of FEM, FDM and hybrid methods for elastic wave equation.

Numerical examples are presented, illustrating the capabilities of our approach in two and three dimensions.

Hybrid approach compared with pure FEM is studied. The test examples for structured grids indicates that the hybrid method is 2–11 faster, than a highly optimized pure finite element version.

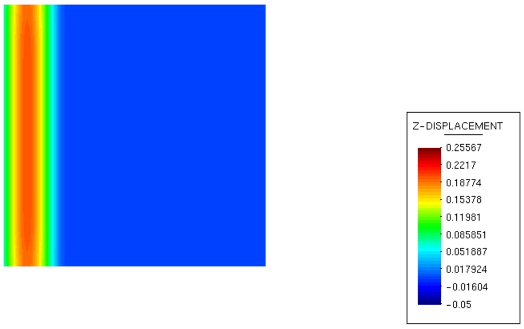


a) t = 0.5 b) t = 0.6

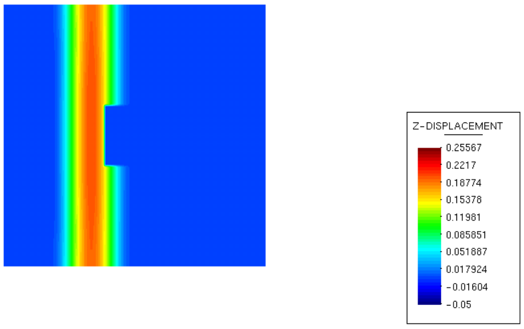
c) t = 0.7 d) t = 0.8

FIGURE 5. Hybrid method for the two-dimensional elastic wave equation with a plane wave. We define a plane wave on the left boundary of the outer domain.

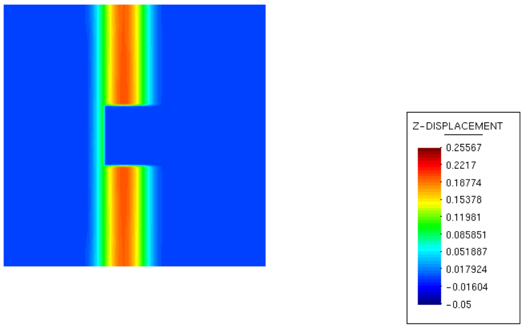
We conclude that the hybrid approach is advantageous, in particular for big problems where large parts of the computational domain may be discretized by uniform grids, while unstructured grids are more suitable in small regions of the domain.



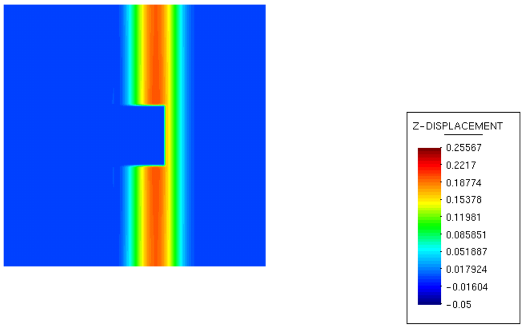
a) t = 0.3



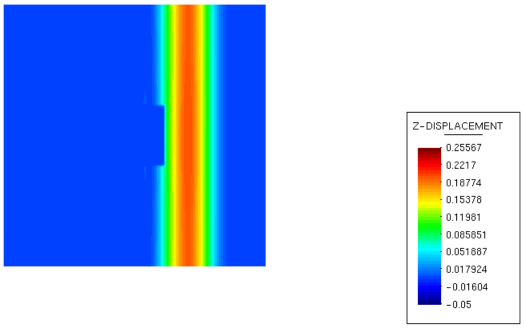
b) t = 0.5



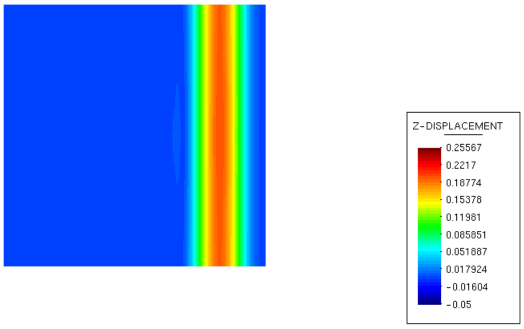
c) t = 0.6



d) t = 0.7

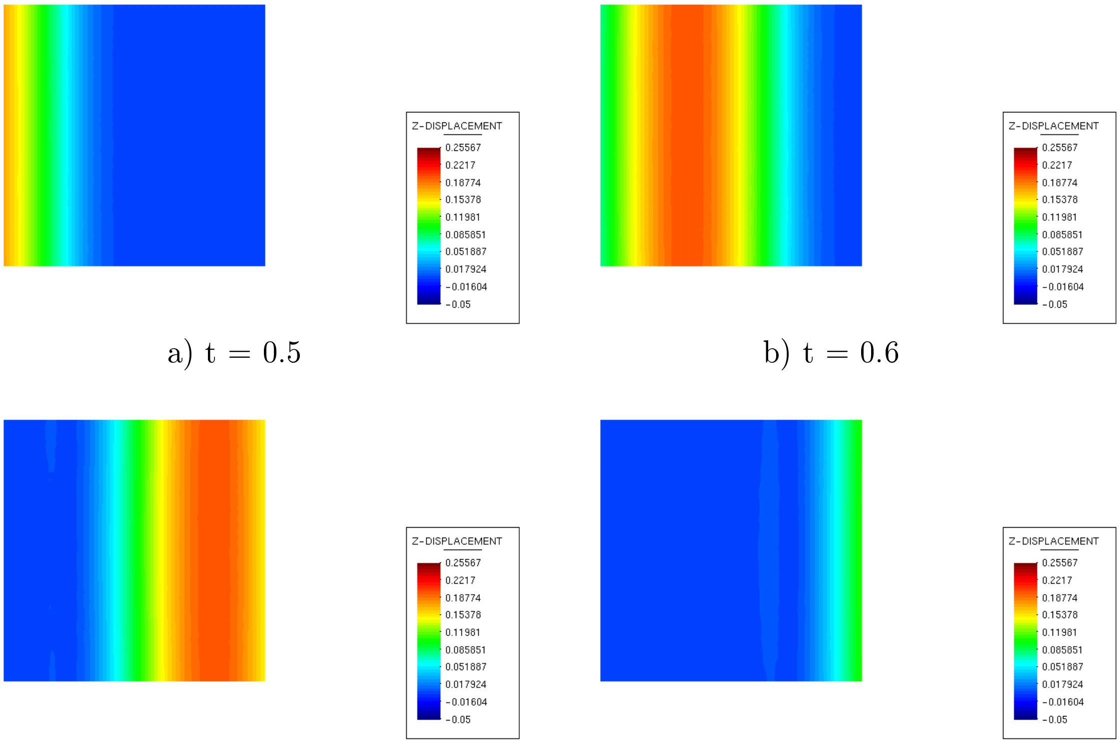


c) t = 0.8



d) t = 0.9

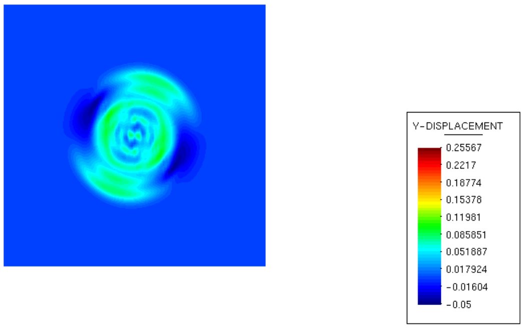
FIGURE 6. Hybrid method for the two-dimensional elastic wave equation with a plane wave. We present only FDM solution, which is a part of solution , presented in Fig. 5. The grid for the computations is presented in Fig. 4-a.



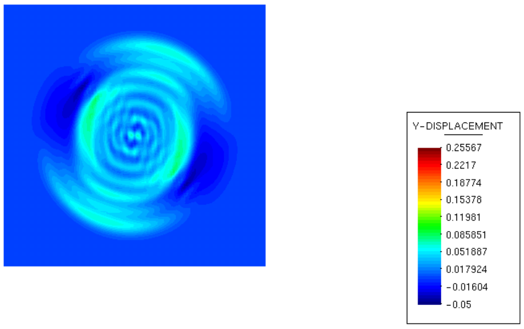
a) t = 0.5 b) t = 0.6

c) t = 0.7 d) t = 0.8

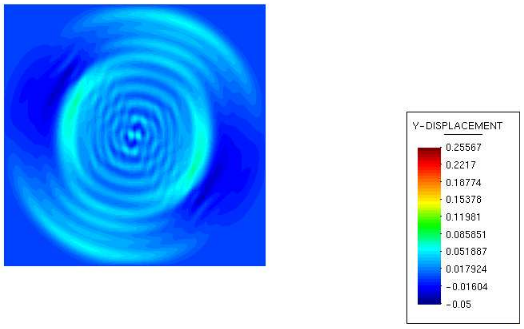
FIGURE 7. Hybrid method for the two-dimensional elastic wave equation with a plane wave. We present only FEM solution, which is a part of solution, presented in Fig. 5. The grid for computations is presented in Fig. 4-b.



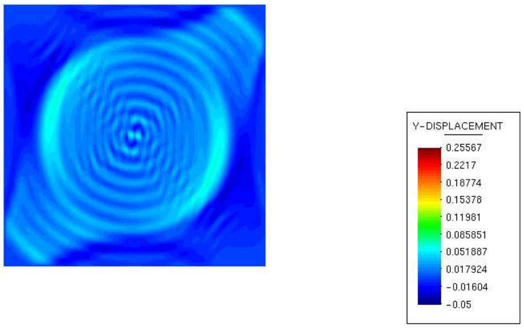
a) t = 0.5



b) t = 0.7



c) t = 0.9



d) t = 1.0

FIGURE 8. Hybrid method for the two-dimensional elastic wave equation with Dirichlet boundary conditions, only u_2-component. We choose a source function located at the center of the unstructured domain, presented in Fig. 4-b.

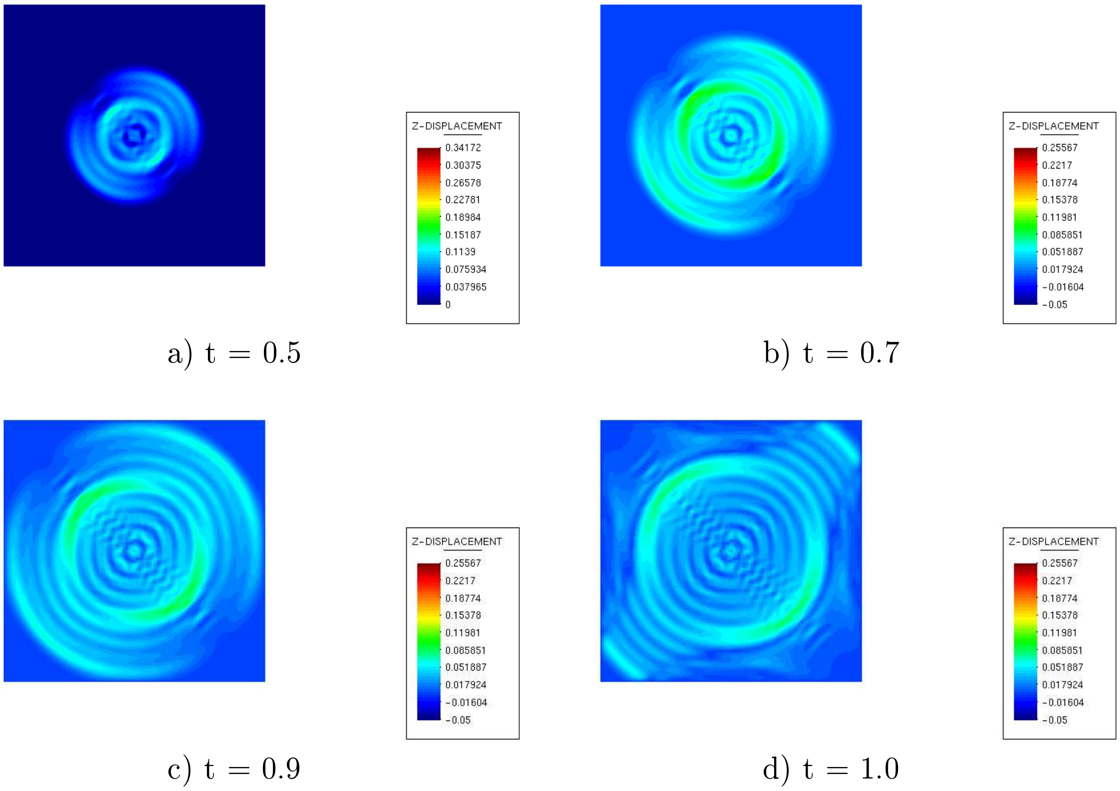
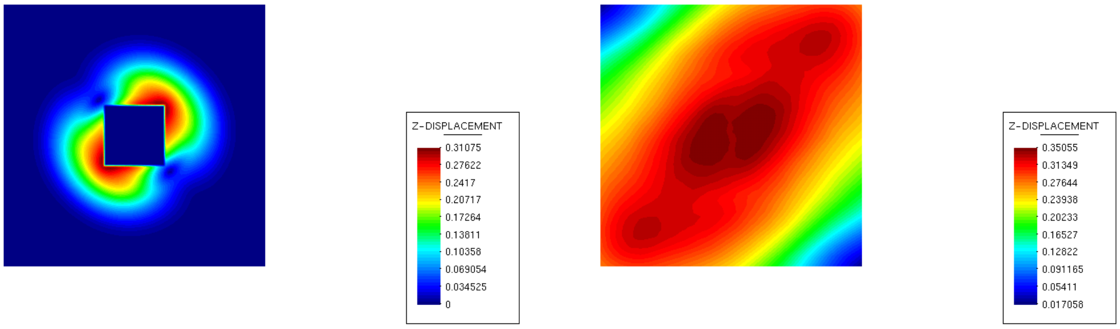
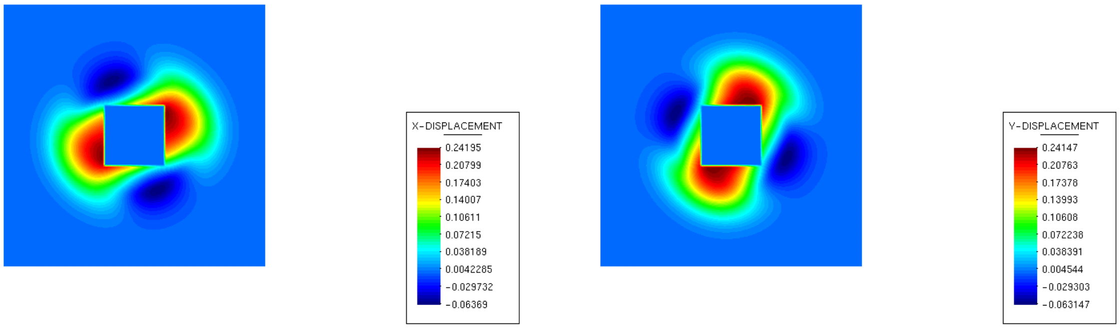


FIGURE 9. Hybrid method for the two-dimensional elastic wave equation with Dirichlet boundary conditions, common solution $u = \sqrt{{u_1}^2 + {u_2}^2}$. We choose a source function located at the center of the unstructured domain, presented in Fig. 4-b.



a) t = 0.3, Ω_{FDM} b) t = 0.3, Ω_{FEM}



c) u_1 in Ω_{FDM} d) u_2 in Ω_{FDM}

FIGURE 10. Hybrid method for the two-dimensional elastic wave equation with absorbing boundary conditions in the inhomogeneous Ω composed of different material types having different density ρ. We choose a source function located at the center of the unstructured domain, presented in Fig. 4-b. The coefficient is taken as $\rho = 0.5$ inside the elliptical domain (see Fig. 4-b) and $\rho = 1.0$ outside it. In Fig. 10-a we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FDM} , in Fig. 10-b we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FEM}, in Fig. 10-c,d are presented u_1 and u_2 components of the hybrid solution.

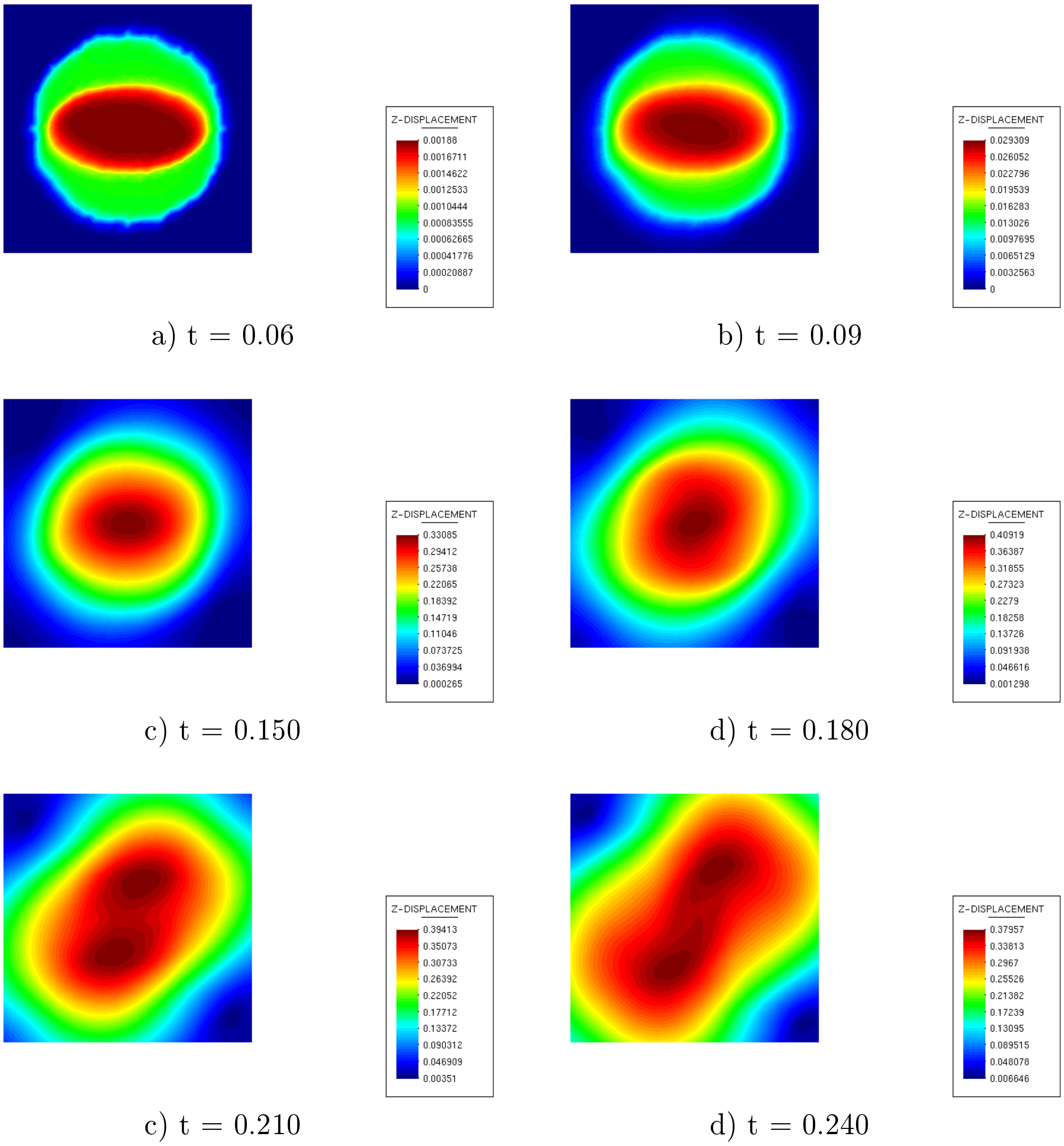
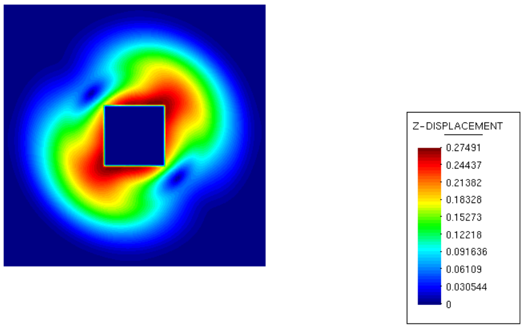
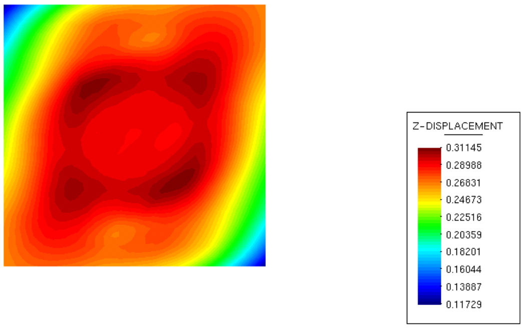


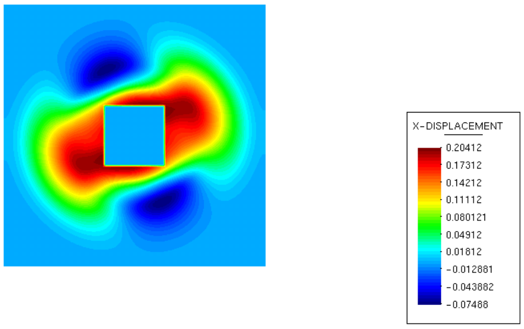
FIGURE 11. We present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FEM} at the different time moments.



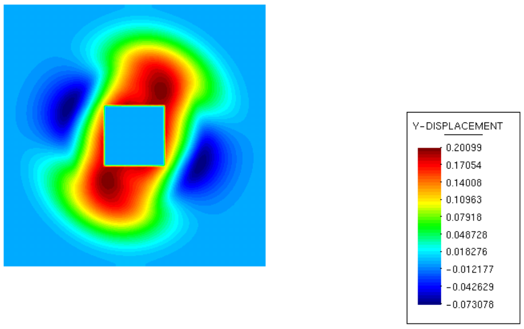
a) t = 0.360, Ω_{FDM}



b) t = 0.360, Ω_{FEM}

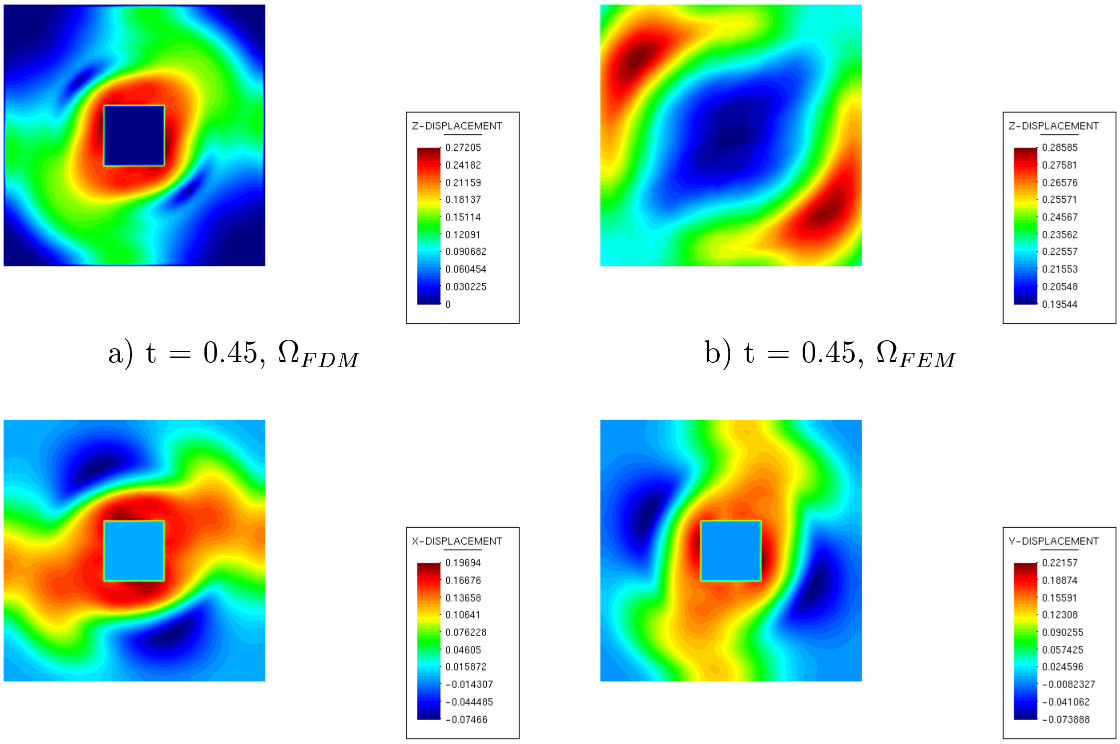


c)u_1 in Ω_{FDM}



d) u_2 in Ω_{FDM}

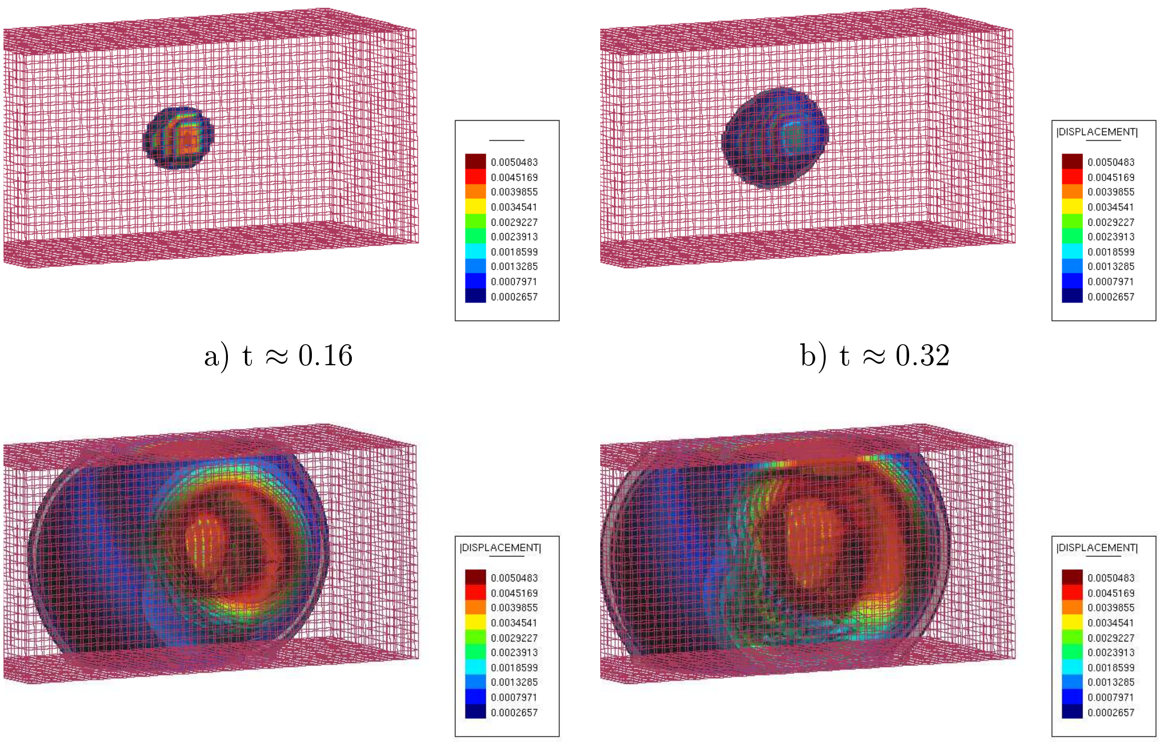
FIGURE 12. Hybrid method for the two-dimensional elastic wave equation with absorbing boundary conditions in the inhomogeneous Ω composed of different material types having different density ρ. We choose a source function located at the center of the unstructured domain, presented in Fig. 4-b. The coefficient is taken as $\rho = 0.5$ inside the elliptical domain (see Fig. 4-b) and $\rho = 1.0$ outside it. In Fig. 12-a we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FDM} , in Fig. 12-b we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FEM}, in Fig. 12-c,d are presented u_1 and u_2 components of the hybrid solution.



a) t $= 0.45$, Ω_{FDM} b) t $= 0.45$, Ω_{FEM}

c) u_1 in Ω_{FDM} d) u_2 in Ω_{FDM}

Figure 13. Hybrid method for the two-dimensional elastic wave equation with absorbing boundary conditions in the inhomogeneous Ω composed of different material types having different density ρ. We choose a source function located at the center of the unstructured domain, presented in Fig. 4-b. The coefficient is taken as $\rho = 0.5$ inside the elliptical domain (see Fig. 4-b) and $\rho = 1.0$ outside it. In Fig. 13-a we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FDM} , in Fig. 13-b we present $u = \sqrt{{u_1}^2 + {u_2}^2}$ in the Ω_{FEM}, in Fig. 13-c,d are presented u_1 and u_2 components of the hybrid solution.



a) t ≈ 0.16 b) t ≈ 0.32

c) t ≈ 1.28 d) t ≈ 1.6

FIGURE 14. Hybrid method for the three-dimensional elastic wave equation with absorbing boundary conditions at all the boundaries and one source function located in the center of the inner domain . We present contour fill for $u = \sqrt{{u_1}^2 + {u_2}^2 + {u_3}^2}$. The values of the Lame coefficients $\lambda, \mu = 0.5$ and density $\rho = 1.0$. The effect of the absorbing boundary conditions is presented in the graphs (c)-(d).