CHALMERS
FINITE ELEMENT CENTER

PREPRINT 2002–05

Adaptive hybrid FEM/FDM methods for inverse scattering problems

Larisa Beilina

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2002

CHALMERS FINITE ELEMENT CENTER

Preprint 2002–05

Adaptive hybrid FEM/FDM methods for inverse scattering problems

Larisa Beilina

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, March 2002

Adaptive hybrid FEM/FDM methods for inverse scattering problems
Larisa Beilina
NO 2002–05
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 2002

ADAPTIVE HYBRID FEM/FDM METHODS FOR INVERSE SCATTERING PROBLEMS

LARISA BEILINA

Abstract. We apply an adaptive hybrid FEM/FDM method for an inverse scattering problem for the time-dependent acoustic wave equation in $2D$ and $3D$, where we seek to reconstruct an unknown sound velocity $c(x)$ from measured wave-reflection data. Typically, this corresponds to identifying an unknown object (scatterer) in a surrounding homogeneous medium.

We use an optimal control approach where we seek a sound velocity $c(x)$ which minimizes the difference between computed and measured output data in a discrete L_2 norm. We solve the optimization problem by a quasi-Newton method where in each step we compute the gradient by solving a forward (direct) and an backward (adjoint) wave propagation problem.

To compute the backward and forward wave propagation problems we use an adaptive hybrid finite element/finite difference method, where we exploit the flexibility of mesh refinement and adaption of the finite element method in a domain covering the object, and the efficiency of a structured mesh finite difference method in the surrounding homogeneous domain. The hybrid scheme can be viewed as a finite element scheme on a partially unstructured mesh which gives a stable coupling of the two methods.

We use an adaptive mesh refinement algorithm to improve the accuracy of the reconstruction and speed up the convergence of the quasi-Newton method.

1. Introduction

This work is devoted to adaptive hybrid finite element/finite difference methods for an inverse scattering problem for the time-dependent acoustic wave equation of the form of a parameter identification problem, where one seeks to determine an unknown variable wave speed $c(x)$ from measured wave reflection data. Typical applications concern nondestructive testing of materials, shape reconstruction, ultrasound imaging, subsurface depth imaging of geological structures and seismic prospectation.

To solve the inverse problem we use an optimal control approach, where we seek to minimize a cost functional :

$$E(p) = \frac{1}{2} \parallel p - \tilde{p} \parallel^2, \tag{1.1}$$

depending on the state p, satisfying a differential equation of state

$$A(p, c) = f, \tag{1.2}$$

Date: 28th March 2002.

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

by varying the coefficient $c(x)$ representing the wave speed. Here, f is a given function, $\tilde{p}$ is observed data at a finite set of observation points and $\parallel . \parallel$ is a discrete L_2 norm.

The minimization problem is reformulated as the problem of finding a stationary point of a Lagrangian involving a forward wave equation (the state equation), a backward wave equation (the adjoint equation), and an equation expressing that the gradient with respect to the wave speed c vanishes. For efficient implementation of the backward and forward wave propagation we use a hybrid finite element/finite difference method, see [4]. We exploit the flexibility of mesh refinement and adaption of the finite element method in a domain including the object, and the efficiency of a structured mesh finite difference method in the surrounding homogeneous domain. The hybrid scheme can be viewed as a finite element scheme on a partially unstructured mesh which gives a stable coupling of the two methods.

The mesh adaptation is based on an a posteriori error estimate for the error in the Lagrangian involving the residuals of the state, adjoint state equation and the gradient with respect to c.

An outline of the work is following: in Section 2 we formulate the inverse scattering problem for the wave equation, in Section 3 we formulate the finite element method, in Section 4 we present a fully discrete version used in the computations. In Section 6 we prove a posteriori error estimate underlying the adaptivity, and in Section 9 we present computational results for reconstructions in 2 and 3 dimensions .

2. The inverse scattering problem

We consider the scalar wave equation modeling acoustic wave propagation in a bounded domain $\Omega \subset \mathbf{R}^d$, $d = 2, 3$, with boundary Γ:

$$\begin{aligned} \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \triangle p &= f, \qquad \text{in } \Omega \times (0,T), && (2.1)\\ p(\cdot,0) &= 0, \quad \frac{\partial p}{\partial t}(\cdot,0) = 0, \quad \text{in } \Omega, && (2.2)\\ p\big|_{\Gamma} &= 0, \quad \text{on } \Gamma \times (0,T), && (2.3) \end{aligned}$$

where $p(x,t)$ is the pressure, $c(x)$ is the wave speed depending on $x \in \Omega$, t is the time variable and T is a final time, and $f(x,t)$ is a given source function.

Our goal is to find the coefficient $c(x)$ which minimizes the quantity

$$E(p) = \frac{1}{2}\int_0^T \int_\Omega (p - \tilde{p})^2 \delta_{obs} \; dxdt, \tag{2.4}$$

where $\tilde{p}$ is observed data at a finite set of observation points x_{obs}, p satisfies (2.1) and thus depends on c, and $\delta_{obs} = \sum \delta(x_{obs})$ is a sum of delta-functions corresponding to the observation points.

To approach this minimization problem, we introduce the Lagrangian

$$L(\lambda, p, c) = E(p) + \int_0^T \int_\Omega \big(-\frac{1}{c^2}\frac{\partial \lambda}{\partial t}\frac{\partial p}{\partial t} + \nabla\lambda\nabla p - f\lambda\big) \; dxdt, \tag{2.5}$$

and search for a stationary point with respect to (λ, p, c) satisfying for all $(\bar\lambda, \bar p, \bar c)$

$$L'(\lambda, p, c)(\bar\lambda, \bar p, \bar c) = 0, \tag{2.6}$$

where L' is the gradient of L and we assume that $\lambda(\cdot, T) = \bar\lambda(\cdot, T) = 0$ and $p(\cdot, 0) = \bar p(\cdot, 0) = 0$, together with homogeneous Dirichlet boundary conditions.

The equation (2.6) expresses that for all $(\bar\lambda, \bar p, \bar c)$,

$$0 = \frac{\partial L}{\partial \lambda}(\lambda, p, c)(\bar\lambda) = \int_0^T \int_\Omega \big(- \frac{1}{c^2}\frac{\partial \bar\lambda}{\partial t}\frac{\partial p}{\partial t} + \nabla\bar\lambda\nabla p - f\bar\lambda \big) \ dxdt, \tag{2.7}$$

$$\begin{aligned} 0 = \frac{\partial L}{\partial p}(\lambda, p, c)(\bar p) \ &= \int_0^T \int_\Omega (p - \tilde p) \ \bar p \ \delta_{obs} \ \ dxdt + \\ & \int_0^T \int_\Omega \big(- \frac{1}{c^2}\frac{\partial \lambda}{\partial t}\frac{\partial \bar p}{\partial t} + \nabla\lambda\nabla\bar p \big) \ \ dxdt, \end{aligned} \tag{2.8}$$

$$0 = \frac{\partial L}{\partial c}(\lambda, p, c)(\bar c) = \frac{2}{c^3}\int_0^T \int_\Omega \frac{\partial \lambda(x,t)}{\partial t}\frac{\partial p(x,t)}{\partial t}\bar c \ \ dxdt, \ x \in \Omega. \tag{2.9}$$

The equation (2.7) is a weak form of the state equation (2.1 - 2.3), the equation (2.8) is a weak form of the adjoint state equation

$$\begin{aligned} \frac{1}{c^2}\frac{\partial^2 \lambda}{\partial t^2} - \triangle\lambda \ &= \ -(p - \tilde p)\delta_{obs}, \quad x \in \Omega, \ 0 < t < T, \\ \lambda(T) \ &= \ \frac{\partial \lambda(T)}{\partial t} = 0, \\ \lambda \ &= \ 0 \ \text{ on } \ \Gamma \times (0, T), \end{aligned} \tag{2.10}$$

and (2.9) expresses stationarity with respect to c.

To solve the minimization problem we shall use a discrete form of the following steepest descent or gradient method starting from an initial guess c^0 and computing a sequence c^n in the following steps:

1. Compute the solution p^n of the forward problem (2.1) with $c = c^n$.
2. Compute the solution λ^n of the adjoint problem (2.11).
3. Update the velocity according to

$$c^{n+1}(x) = c^n(x) - \alpha^n \frac{2}{(c^n(x))^3}\int_0^T \frac{\partial \lambda^n(x,t)}{\partial t}\frac{\partial p^n(x,t)}{\partial t}dt, \tag{2.11}$$

where the step length α^n is computed using the one-dimensional search algorithm given in [20].

More precisely, we will consider a quasi-Newton method with limited storage with the gradient method being a special case.

3. FINITE ELEMENT DISCRETIZATION.

We now formulate a finite element method for (2.6) 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 $(0,T)$ 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 of the Laplacian.

To formulate the finite element method for (2.6) we introduce the finite element spaces V_h, W_h^p and W_h^λ defined by :

$$\begin{aligned}
V_h &:= \{v \in L_2(\Omega) : v \in P_0(K), \forall K \in K_h\}, && (3.1)\\
W^p &:= \{p \in H^1(\Omega \times J) : p(\cdot,0) = 0, p|_\Gamma = 0\}, && (3.2)\\
W^\lambda &:= \{\lambda \in H^1(\Omega \times J) : \lambda(\cdot,T) = 0, \lambda|_\Gamma = 0\}, && (3.3)\\
W_h^p &:= \{v \in W^p : v|_{K\times J} \in P_1(K) \times P_1(J), \forall K \in K_h, \forall J \in J_k\}, && (3.4)\\
W_h^\lambda &:= \{v \in W^\lambda : v|_{K\times J} \in P_1(K) \times P_1(J), \forall K \in K_h, \forall J \in J_k\}, && (3.5)
\end{aligned}$$

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 $c_h \in V_h, \lambda_h \in W_h^\lambda, p_h \in W_h^p$, such that

$$L'(\lambda_h, p_h, c_h)(\bar{\lambda}, \bar{p}, \bar{c}) = 0 \quad \forall \bar{c} \in V_h, \bar{\lambda} \in W_h^\lambda, \bar{p} \in W_h^p. \tag{3.6}$$

4. FULLY DISCRETE SCHEME

Expanding p, λ and c in terms of the standard continuous piecewise linear functions $\varphi_i(x)$ in space and $\psi_i(t)$ in time and substituting this into (2.7 - 2.8), the following system of linear equations is obtained:

$$M(\mathbf{p}^{k+1} - 2\mathbf{p}^k + \mathbf{p}^{k-1}) = \tau^2 F^k - \tau^2 K(\frac{1}{6}\mathbf{p}^{k-1} + \frac{2}{3}\mathbf{p}^k + \frac{1}{6}\mathbf{p}^{k+1}), \tag{4.1}$$

$$M(\boldsymbol{\lambda}^{k+1} - 2\boldsymbol{\lambda}^k + \boldsymbol{\lambda}^{k-1}) = -\tau^2 S^k - \tau^2 K(\frac{1}{6}\boldsymbol{\lambda}^{k-1} + \frac{2}{3}\boldsymbol{\lambda}^k + \frac{1}{6}\boldsymbol{\lambda}^{k+1}), \tag{4.2}$$

with initial conditions :

$$p(0) = 0, \ \dot{p}(0) \approx 0, \tag{4.3}$$

$$\lambda(T) = 0, \ \dot{\lambda}(T) \approx 0. \tag{4.4}$$

Here, M is the mass matrix in space, K is the stiffness matrix, $k = 1, 2, 3\ldots$ denotes the time level, F^k, S^k are the load vectors, $\mathbf{p}$ is the unknown discrete field values of p, $\boldsymbol{\lambda}$ is the unknown discrete field values of λ and τ is the time step.

The explicit formulas for the entries in system (4.1 - 4.2) at the element level can be given as:

$$M^e_{i,j} = (\frac{1}{c^2}\varphi_i, \varphi_j)_e, \tag{4.5}$$

$$K^e_{i,j} = (\nabla\varphi_i, \nabla\varphi_j)_e, \tag{4.6}$$

$$F^e_{j,m} = (f, \varphi_j \psi_m)_{e\times J}, \tag{4.7}$$

$$S^e_{j,m} = (p - \tilde{p}, \varphi_j \psi_m)_{e\times J}, \tag{4.8}$$

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, F^e and S^e are the contributions from element e to the assembled source vectors F and vector of the right hand side of (2.11), correspondingly .

To obtain an explicit scheme we approximate M with the lumped mass matrix M^L, the diagonal approximation obtained by taking the row sum of M, see e.g. [13]. By multiplying (4.1) - (4.2) with $(M^L)^{-1}$ and replacing the terms $\frac{1}{6}\mathbf{p}^{k-1} + \frac{2}{3}\mathbf{p}^k + \frac{1}{6}\mathbf{p}^{k+1}$ and $\frac{1}{6}\boldsymbol{\lambda}^{k-1} + \frac{2}{3}\boldsymbol{\lambda}^k + \frac{1}{6}\boldsymbol{\lambda}^{k+1}$ by $\mathbf{p}^k$ and $\boldsymbol{\lambda}^k$, respectively, we obtain an efficient explicit formulation:

$$\mathbf{p}^{k+1} = \tau^2 (M^L)^{-1} F^k + 2\mathbf{p}^k - \tau^2 (M^L)^{-1} K \mathbf{p}^k - \mathbf{p}^{k-1}, \tag{4.9}$$

$$\boldsymbol{\lambda}^{k-1} = -\tau^2 (M^L)^{-1} S^k + 2\boldsymbol{\lambda}^k - \tau^2 (M^L)^{-1} K \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}. \tag{4.10}$$

The discrete version of (2.9) takes the form:

$$0 = \frac{\partial L}{\partial c}(\lambda_h, p_h, c_h)(\bar{c}) = \frac{2}{c_h^3} \int_0^T \int_\Omega \frac{\partial \lambda_h}{\partial t} \frac{\partial p_h}{\partial t} \bar{c} \ \ dxdt, \forall \bar{c} \in V_h. \tag{4.11}$$

5. Optimization by quasi-Newton

5.1. **quasi-Newton with limited storage.** To solve the discrete problem (3.6), we use a quasi-Newton method, where we compute a sequence $c_h^k, k = 0, 1, ...,$ of approximations of c_h with nodal values c_k given by

$$c_{k+1} = c_k - \alpha^k H^k g_k, \tag{5.1}$$

where g_k are the nodal values of

$$g^k(x) = \frac{2}{(c_k(x))^3} \int_0^T \frac{\partial \lambda_h^k(x,t)}{\partial t} \frac{\partial p_h^k(x,t)}{\partial t} dt, \tag{5.2}$$

where p_h^k and λ_h^k solve the discrete analogs of (2.1-2.3) and (2.11), the H^k are given by the usual BFGS update formula of the Hessian (see [19])

$$H^{k+1} = (I - \rho s_k y_k^T) H^k (I - \rho y_k s_k^T) + \rho s_k s_k^T, \tag{5.3}$$

where $\rho = 1/(y_k^T s_k)$ and

$$s_k = c_{k+1} - c_k, \tag{5.4}$$

$$y_k = g_{k+1} - g_k, \tag{5.5}$$

and the step length α^k is given by a one dimensional search algorithm.

We now describe a special BFGS with limited storage, where we only store a finite number m of corrections. For example, suppose that we have performed 3 iterations, allowing $m = 2$ corrections to be stored. Assuming $H_0 = I$ and writing

$$\begin{aligned} \rho_i &= 1/y_i^T s_i, \\ v_i &= (I - \rho_i y_i s_i^T), \end{aligned} \tag{5.6}$$

the usual BFGS update (5.3) gives

$$\begin{aligned} H_1 &= v_0^T H_0 v_0 + \rho_0 s_0 s_0^T, \\ H_2 &= v_1^T v_0^T H_0 v_0 v_1 + v_1^T \rho_0 s_0 s_0^T v_1 + \rho_1 s_1 s_1^T, \end{aligned}$$

while the special BFGS with $m = 2$ stored corrections is given by

$$\bar{H}_2 = v_1^T H_0 v_1 + \rho_1 s_1 s_1^T, \tag{5.7}$$

and thus

$$H_3 = v_2^T v_1^T H_0 v_1 v_2 + v_2^T \rho_1 s_1 s_1^T v_2 + \rho_2 s_2 s_2^T. \tag{5.8}$$

In the general case, if $k + 1 \le m$, we have the usual BFGS update

$$\begin{aligned} H_{k+1} &= v_k^T v_{k-1}^T ... v_0^T H_0 v_0 ... v_{k-1} v_k \\ &+ v_k^T ... v_1^T \rho_0 s_0 s_0^T v_1 ... v_k \\ &. \\ &. \\ &. \\ &+ v_k^T v_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T v_{k-1} v_k \\ &+ v_k^T \rho_{k-1} s_{k-1} s_{k-1}^T v_k \\ &+ \rho_k s_k s_k^T. \end{aligned} \tag{5.9}$$

If $k + 1 \ge m$ we have the special BFGS update

$$\begin{aligned} H_{k+1} &= v_k^T v_{k-1}^T ... v_{k-m+1}^T H_0 v_{k-m+1} ... v_{k-1} v_k \\ &+ v_k^T ... v_{k-m+2}^T \rho_{k-m+2} s_{k-m+1} s_{k-m+1}^T v_{k-m+2} ... v_k \\ &. \\ &. \\ &. \\ &+ v_k^T \rho_{k-1} s_{k-1} s_{k-1}^T v_k \\ &+ \rho_k s_k s_k^T. \end{aligned} \tag{5.10}$$

Note that instead of explicitely computing the Hessian H^k in (5.1), we compute the product product $H^{k+1} g^k$ from (5.3) to get:

$$((I - \rho s y^T) H^k (I - \rho y s^T) + \rho s s^T) g^k = ((I - \rho s y^T) H^k (g^k - \rho y s^T g^k) + \rho s s^T g^k, \tag{5.11}$$

involving only scalar products of a vectors and computing $H^k g^k$ similarly. The gradient method is a special case of the quasi-Newton method, with $m = 0$.

5.2. **The complexity of the quasi-Newton method.** In this section we analyze the complexity of the quasi-Newton method with limited storage. To compute the product $v_i g^k$ in BFGS-method:

$$v_i g^k = (I - \rho_i y_i s_i^T) g^k = g^k - \rho_i y_i s_i^T g^k, \tag{5.12}$$

we need

0. $2n - 1$ operations to compute $s_i^T g^k$,
1. n operations to multiply the obtained constant on step 0 by vector y^k,
2. n operations to compute product of the obtained vector in step 1 by const ρ^i,
3. $2n - 1 + n$ operations to compute constant ρ^i,
4. n operations to compute the difference $g^k - \rho_i y_i s_i^T g^k$.

It follows, that complete number of operations to compute $v_i g^k$ is $8n - 2$. The maximal number of such multiplications is equal to $2m$, where m is number of stored corrections. Taking sum in BFGS formula we have number of operations equal to $8n \cdot 2m + 8n \cdot 2(m - 1) + 8n \cdot 2(m-2) + ... + 8n \cdot 2 = 8n \cdot (2m + 2(m-1) + 2(m-2) + ... + 2) = 8nm \cdot (m+1)$. Then the total complexity of the quasi-Newton method is $8nm(m + 1)$, where n is the number of nodes in the grid and m is the number of stored corrections.

6. An a posteriori error estimate for the Lagrangian and adaptive algorithm

We shall now indicate a proof an a posteriori error estimate for the Lagrangian. By C we denote various constants of moderate size. We start by writing an equation for the error e in the Lagrangian as

$$\begin{aligned} e &= L(\lambda, p, c) - L(\lambda_h, p_h, c_h) \\ &= \frac{1}{2} L'(\lambda_h, p_h, c_h)((\lambda, p, c) - (\lambda_h, p_h, c_h)) + R \\ &= \frac{1}{2} L'(\lambda_h, p_h, c_h)(\lambda - \lambda_h, p - p_h, c - c_h) + R, \end{aligned} \tag{6.1}$$

where R denotes (a small) second order term. For full details of the arguments we refer to [3] and [10]. Using the Galerkin orthogonality (3.6) and the splitting $\lambda - \lambda_h = (\lambda - \lambda_h^I) + (\lambda_h^I - \lambda_h)$, $p - p_h = (p - p_h^I) + (p_h^I - p_h)$, $c - c_h = (c - c_h^I) + (c_h^I - c_h)$, where $(\lambda_h^I, p_h^I, c_h^I)$ denotes an interpolant of $(\lambda, p, c) \in W_h^\lambda \times W_h^p \times V_h$, and neglecting the term R, we get:

$$e \approx \frac{1}{2} L'(\lambda_h, p_h, c_h)(\lambda - \lambda_h^I, p - p_h^I, c - c_h^I) = \frac{1}{2}(I_1 + I_2 + I_3), \tag{6.2}$$

where

$$
\begin{aligned}
(6.3)\qquad I_1 &= \int_0^T \int_{\Omega} (-\frac{1}{c_h^2}\frac{\partial(\lambda-\lambda_h^I)}{\partial t}\frac{\partial p_h}{\partial t} + \nabla(\lambda-\lambda_h^I)\nabla p_h \\
&\quad -f(\lambda-\lambda_h^I))\ \ dxdt, \\
(6.4)\qquad I_2 &= \int_0^T \int_{\Omega} (p_h-\tilde{p})(p-p_h^I)\ \delta_{obs}\ \ dxdt \\
&+ \int_0^T \int_{\Omega} \left(-\frac{1}{c_h^2}\frac{\partial \lambda_h}{\partial t}\frac{\partial (p-p_h^I)}{\partial t} + \nabla\lambda_h \nabla(p-p_h^I) \right)\ \ dxdt, \\
(6.5)\qquad I_3 &= \frac{2}{c_h^3}\int_0^T \int_{\Omega} \frac{\partial\lambda_h(x,t)}{\partial t}\frac{\partial p_h(x,t)}{\partial t}(c-c_h^I)\ \ dxdt.
\end{aligned}
$$

To estimate (6.3) we integrate by parts in the first and second terms to get:

$$
\begin{aligned}
(6.6)\qquad |I_1| &= |\int_0^T \int_{\Omega} \left(\frac{1}{c_h^2}\frac{\partial^2 p_h}{\partial t^2}(\lambda-\lambda_h^I) - \triangle p_h(\lambda-\lambda_h^I) - f(\lambda-\lambda_h^I) \right)\ \ dxdt \\
&+ \sum_K \int_0^T \int_{\partial K} \frac{\partial p_h}{\partial n_K}(\lambda-\lambda_h^I)\ dsdt \\
&- \sum_k \int_{\Omega} \frac{1}{c_h^2}\Big[\frac{\partial p_h}{\partial t}(t_k)\Big](\lambda-\lambda_h^I)(t_k)\ dx|,
\end{aligned}
$$

where the terms $\frac{\partial p_h}{\partial n_K}$ and $\left[\frac{\partial p_h}{\partial t}\right]$ appear during the integration by parts and denote the derivative of p_h in the outward normal direction n_K of the boundary ∂K of element K, and the jump of the derivative of p_h in time, respectively. In the second term of the (6.6) we sum over the element boundaries, and each internal side $S \in S_h$ occurs twice. Denoting by $\partial_s p_h$ the derivative of a function p_h in one of the normal directions of each side S, we can write

$$
(6.7)\qquad \sum_K \int_{\partial K} \frac{\partial p_h}{\partial n_K}(\lambda-\lambda_h^I)\ \ ds = \sum_S \int_S \left[\partial_s p_h\right](\lambda-\lambda_h^I)\ \ ds,
$$

where $\left[\partial_s p_h\right]$ is jump in the derivative $\partial_s p_h$ computed from the two triangles sharing S. We distribute each jump equally to the two sharing triangles and return to a sum over elements edges ∂K :

$$
(6.8)\qquad \sum_S \int_S [\partial_s p_h]\cdot(\lambda-\lambda_h^I)\ \ ds = \sum_K \frac{1}{2} h_K^{-1} \int_{\partial K} \left[\partial_s p_h\right](\lambda-\lambda_h^I)\ h_K\ \ ds.
$$

We formally set $dx = h_K ds$ and replace the integrals over the element boundaries ∂K by integrals over the elements K, to get:

$$
(6.9)\qquad \left| \sum_K \frac{1}{2} h_K \int_{\partial K} \left[\partial_s p_h\right](\lambda-\lambda_h^I)\ h_K\ \ ds \right| \le C \max_{S\subset\partial K} h_K^{-1} \int_{\Omega} \left|\left[\partial_s p_h\right]\right| \left|(\lambda-\lambda_h^I)\right|\ \ dx,
$$

where $\left[\partial_s p_h\right]\big|_K = \max_{S\subset\partial K}\left[\partial_s p_h\right]\big|_S$.

In a similar way we can estimate the third term in (6.6):

$$\begin{aligned}
\left|\sum_k \int_\Omega \frac{1}{c_h^2}\left[\frac{\partial p_h}{\partial t}(t_k)\right](\lambda-\lambda_h^I)(t_k)\;dx\right| &\leq \sum_k \int_\Omega \frac{1}{c_h^2}\tau^{-1}\cdot\left|\left[\frac{\partial p_h}{\partial t}(t_k)\right]\right|\cdot\left|(\lambda-\lambda_h^I)(t_k)\right|\;\tau dx \\
&\leq C\sum_k \int_{J_k}\int_\Omega \frac{1}{c_h^2}\tau^{-1}\cdot\left|\left[\partial p_{h_{t_k}}\right]\right|\cdot\left|(\lambda-\lambda_h^I)\right|\;dxdt \\
&= C\int_0^T\int_\Omega \frac{1}{c_h^2}\tau^{-1}\cdot\left|\left[\partial p_{h_t}\right]\right|\cdot\left|(\lambda-\lambda_h^I)\right|\;dxdt,
\end{aligned}$$

where

$$[\partial p_{h_{t_k}}] = \max_k\left(\left[\frac{\partial p_h}{\partial t}(t_k)\right],\;\left[\frac{\partial p_h}{\partial t}(t_{k+1})\right]\right), \tag{6.10}$$

$$[\partial p_{h_t}] = [\partial p_{h_{t_k}}]\;\text{ on }\;J_k. \tag{6.11}$$

Substituting both above expressions for the second and third terms in (6.6), we get:

$$\begin{aligned}
|I_1| \;\leq\;& \left|\int_0^T\int_\Omega \big(\frac{1}{c_h^2}\frac{\partial^2 p_h}{\partial t^2}-\triangle p_h - f\big)(\lambda-\lambda_h^I)\;dxdt\right| \\
&+ C\int_0^T\int_\Omega \max_{S\subset\partial K} h_k^{-1}\cdot\left|\left[\partial_s p_h\right]\right|\cdot\left|(\lambda-\lambda_h^I)\right|\;dxdt \\
&+ \frac{C}{c_h^2}\int_0^T\int_\Omega \tau^{-1}\cdot\left|\left[\partial p_{h_t}\right]\right|\cdot\left|(\lambda-\lambda_h^I)\right|\;dxdt \\
\leq\;& C\int_0^T\int_\Omega \left|\frac{1}{c_h^2}\frac{\partial^2 p_h}{\partial t^2}-\triangle p_h - f\right|\cdot\left(\tau^2\left|\frac{\partial^2\lambda}{\partial t^2}\right| + h^2\left|D_x^2\lambda\right|\right)\;dxdt \\
&+ C\int_0^T\int_\Omega \max_{S\subset\partial K} h_k^{-1}\cdot\left|\left[\partial_s p_h\right]\right|\cdot\left(\tau^2\left|\frac{\partial^2\lambda}{\partial t^2}\right| + h^2\left|D_x^2\lambda\right|\right)\;dxdt \\
&+ \frac{C}{c_h^2}\int_0^T\int_\Omega \tau^{-1}\cdot\left|\left[\partial p_{h_t}\right]\right|\cdot\left(\tau^2\left|\frac{\partial^2\lambda}{\partial t^2}\right| + h^2\left|D_x^2\lambda\right|\right)\;dxdt,
\end{aligned} \tag{6.12}$$

where we used standard interpolation estimates for $\lambda-\lambda_h^I$, and C denotes interpolation constants. Next, the terms $\frac{\partial^2 p_h}{\partial t^2}$ and $\triangle p_h$ disappears in the first integral in (6.13) (p_h is continuous piecewise linear function). We estimate $\frac{\partial^2\lambda}{\partial t^2} \approx \frac{\left[\frac{\partial\lambda_h}{\partial t}\right]}{\tau}$ and $D_x^2\lambda \approx \frac{\left[\frac{\partial\lambda_h}{\partial n}\right]}{h}$ to get:

$$\begin{aligned}
\left|I_1\right| \leq \quad & C\int_0^T \int_\Omega \left|f\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial n}\right]}{h} \right| \right) \; dxdt \\
+ \; & C\int_0^T \int_\Omega \max_{S\subset\partial K} h_k^{-1}\left|\left[\partial_s p_h\right]\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial n}\right]}{h} \right| \right) \; dxdt \\
+ \; & \frac{C}{c_h^2}\int_0^T \int_\Omega \tau^{-1}\left|\left[\partial p_{ht}\right]\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial\lambda_h}{\partial n}\right]}{h} \right| \right) \; dxdt.
\end{aligned} \tag{6.13}$$

We estimate I_2 similarly:

$$\begin{aligned}
\left|I_2\right| \leq \; & \int_0^T \int_\Omega \left| \left(\frac{1}{c_h^2}\frac{\partial^2\lambda_h}{\partial t^2}(p - p_h^I) - \triangle\lambda_h(p - p_h^I) - (p_h - \tilde{p})(p - p_h^I) \right) \right| \; dxdt \\
+ \; & C \int_0^T \int_\Omega \max_{S\subset\partial K} \; h_k^{-1} \cdot \left|\left[\partial_s\lambda_h\right]\right| \cdot \left|(p - p_h^I)\right| \; dxdt \\
+ \; & \frac{C}{c_h^2}\int_0^T \int_\Omega \tau^{-1} \cdot \left|\left[\partial\lambda_{ht}\right]\right| \cdot \left|(p - p_h^I)\right| \; dxdt \\
\leq \; & C\int_0^T \int_\Omega \left| \left(\frac{1}{c_h^2}\frac{\partial^2\lambda_h}{\partial t^2} - \triangle\lambda_h - (p_h - \tilde{p}) \right) \right| \cdot \left|(p - p_h^I)\right| \; dxdt \\
+ \; & C\int_0^T \int_\Omega \max_{S\subset\partial K} h_k^{-1} \cdot \left|\left[\partial_s\lambda_h\right]\right| \cdot \left|(p - p_h^I)\right| \; dxdt \\
+ \; & \frac{C}{c_h^2}\int_0^T \int_\Omega \tau^{-1} \cdot \left|\left[\partial\lambda_{ht}\right]\right| \cdot \left|(p - p_h^I)\right| \; dxdt \\
\leq \; & C \int_0^T \int_\Omega \left| \left(\frac{1}{c_h^2}\frac{\partial^2\lambda_h}{\partial t^2} - \triangle\lambda_h - (p_h - \tilde{p}) \right) \right| \left(\tau^2 \left| \frac{\partial^2 p}{\partial t^2} \right| + h^2 \left| D_x^2 p \right| \right) \; dxdt \\
+ \; & C \int_0^T \int_\Omega \max_{S\subset\partial K} \; h_k^{-1} \cdot \left|\left[\partial_s\lambda_h\right]\right| \left(\tau^2 \left| \frac{\partial^2 p}{\partial t^2} \right| + h^2 \left| D_x^2 p \right| \right) \; dxdt \\
+ \; & \frac{C}{c_h^2}\int_0^T \int_\Omega \tau^{-1} \cdot \left|\left[\partial\lambda_{ht}\right]\right| \cdot \left(\tau^2 \left| \frac{\partial^2 p}{\partial t^2} \right| + h^2 \left| D_x^2 p \right| \right) \; dxdt \\
\leq \; & C\int_0^T \int_\Omega \left|(p_h - \tilde{p})\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial p_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial p_h}{\partial n}\right]}{h} \right| \right) \; dxdt \\
+ \; & C \int_0^T \int_\Omega \max_{S\subset\partial K} \; h_k^{-1}\left|\left[\partial_s\lambda_h\right]\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial p_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial p_h}{\partial n}\right]}{h} \right| \right) \; dxdt \\
+ \; & \frac{C}{c_h^2}\int_0^T \int_\Omega \tau^{-1} \cdot \left|\left[\partial\lambda_{ht}\right]\right| \cdot \left(\tau^2 \left| \frac{\left[\frac{\partial p_h}{\partial t}\right]}{\tau} \right| + h^2 \left| \frac{\left[\frac{\partial p_h}{\partial n}\right]}{h} \right| \right) \; dxdt.
\end{aligned} \tag{6.14}$$

To estimate I_3 we use a standard approximation estimate of the form $c - c_h^I \approx hD_x c$ to get:

$$
\begin{aligned}
(6.15)\qquad \big|I_3\big| &\leq \frac{2}{c_h^3}\int_0^T\int_\Omega \left|\frac{\partial \lambda_h(x,t)}{\partial t}\cdot\frac{\partial p_h(x,t)}{\partial t}\right|\cdot h\cdot \big|D_x c\big| \;\; dxdt \\
&\leq C\,\frac{2}{c_h^3}\int_0^T\int_\Omega \left|\frac{\partial \lambda_h(x,t)}{\partial t}\cdot\frac{\partial p_h(x,t)}{\partial t}\right|\cdot h\cdot \big|\frac{[c_h]}{h}\big| \;\; dxdt \\
&\leq C\,\frac{2}{c_h^3}\int_0^T\int_\Omega \left|\frac{\partial \lambda_h(x,t)}{\partial t}\cdot\frac{\partial p_h(x,t)}{\partial t}\right|\cdot \big|[c_h]\big| \;\; dxdt.
\end{aligned}
$$

Defining the residuals

$$
\begin{aligned}
R_{p_1} &= \big|f\big|,\;\; R_{p_2} = \frac{1}{2}\max_{S\subset\partial K} h_k^{-1}\big|\big[\partial_s p_h\big]\big|,\;\; R_{p_3} = \frac{1}{2\;c_h^2}\tau^{-1}\big|\big[\partial p_{ht}\big]\big|, \\
R_{\lambda_1} &= \frac{1}{2}\max_{S\subset\partial K} h_k^{-1}\big|\big[\partial_s \lambda_h\big]\big|,\;\; R_{\lambda_3} = \frac{1}{2\;c_h^2}\tau^{-1}\big|\big[\partial \lambda_{ht}\big]\big|, \\
R_c &= \frac{2}{{c_h}^3}\left|\frac{\partial \lambda_h}{\partial t}\right|\cdot\left|\frac{\partial p_h}{\partial t}\right|,
\end{aligned}
$$

and interpolation errors in the form

$$
\sigma_\lambda = C\tau\left|\left[\frac{\partial \lambda_h}{\partial t}\right]\right| + Ch\left|\left[\frac{\partial \lambda_h}{\partial n}\right]\right|, \tag{6.16}
$$

$$
\sigma_p = C\tau\left|\left[\frac{\partial p_h}{\partial t}\right]\right| + Ch\left|\left[\frac{\partial p_h}{\partial n}\right]\right|, \tag{6.17}
$$

$$
\sigma_c = C\big|[c_h]\big|, \tag{6.18}
$$

we obtain the following a posteriori estimate

$$
\begin{aligned}
(6.19)\;\; \big|e\big| \;\leq\;& \frac{1}{2}\Big(\int_0^T\int_\Omega R_{p_1}\sigma_\lambda \;\; dxdt + \int_0^T\int_\Omega R_{p_2}\sigma_\lambda \;\; dxdt + \int_0^T\int_\Omega R_{p_3}\sigma_\lambda \;\; dxdt + \\
&+ \int_0^T\int_\Omega R_{\lambda_1}\sigma_p \;\; dxdt + \int_0^T\int_\Omega R_{\lambda_2}\sigma_p \;\; dxdt + \int_0^T\int_\Omega R_{\lambda_3}\sigma_p \;\; dxdt \\
&+ \int_0^T\int_\Omega R_c\sigma_c \;\; dxdt\Big).
\end{aligned}
$$

In the computations below we use the following variant of the gradient method with adaptive mesh selection:

1. Choose an initial mesh K_h and an initial time partition J_k of the time interval $(0, T)$.
2. Compute the solution p on K_h and J_k of the forward problem (2.1) with $c = c^{(n)}$.
3. Compute the solution λ of the adjoint problem

$$
\frac{1}{c^2}\frac{\partial^2 \lambda}{\partial t^2} - \triangle\lambda = -(p - \tilde{p})\delta_{obs}, \quad x \in \Omega, 0 < t < T
$$

on K_h and J_k.

4. Update the velocity on K_h and J_k according to

$$c^{(n+1)}(x) = c^{(n)}(x) - \alpha^{(n)} \frac{2}{c^3} \int_0^T \frac{\partial \lambda(x,t)}{\partial t} \frac{\partial p(x,t)}{\partial t} dt.$$

   Make steps $1 - 4$ as long the gradient quickly decreases.

5. Refine all elements, where $R_c \sigma_c > tol$ and construct a new mesh K_h and a new time partition J_k. Here tol is a tolerance chosen by the user. Return to 1.

7. AN A PRIORI ERROR ESTIMATE FOR THE WAVE EQUATION.

In this section we prove an a priori error estimate for the wave equation approximation(4.9) in the simplified case with $c = 1$:

$$\frac{\partial^2 p}{\partial t^2} - \triangle p = f, \quad \text{in } \Omega \times (0,T), \tag{7.1}$$

$$p(\cdot,0) = 0, \quad \frac{\partial p}{\partial t}(\cdot,0) = 0, \quad \text{in } \Omega, \tag{7.2}$$

$$p\big|_\Gamma = 0, \quad \text{on } \Gamma \times (0,T). \tag{7.3}$$

We thus consider the corresponding discrete problem: Find $p_h^n \in W_h^p$, for $n = 1, ..., N$ such that

$$(\partial_t^2 p_h^n, v) + (\triangledown p_h^n, \triangledown v) = (f^n, v) \quad \forall v \in W_h^p, \tag{7.4}$$

where

$$\partial_t^2 p_h^n = \frac{p_h^{n+1} - 2p_h^n + p_h^{n-1}}{\tau^2}, \tag{7.5}$$

and $p_h^0 = 0, p_h^1 = 0$. For simplicity we assume that h is constant.

For $w \in H_0^1(\Omega)$ we define the elliptic projection $\pi w \in V_{h,1}$, where $V_{h,1} := \{v \in H_0^1 : v \in P_1(k), \forall K \in K_h\}$ by

$$(\triangledown \pi w, \triangledown v) = (\triangledown w, \triangledown v) \quad \forall v \in V_{h,1}. \tag{7.6}$$

We shall estimate the difference between the discrete solution $p_h^n \in W_h^p$ and the elliptic projection $\pi p^n \in W_h^p$, and we define

$$\Theta^n = p_h^n - \pi p_n. \tag{7.7}$$

Using the definition (7.6) and (7.1) - (7.3), we obtain:

$$\begin{aligned} (\partial_t^2(\pi p^n), v) + (\triangledown(\pi p^n), \triangledown v) &= (\partial_t^2(\pi p^n), v) + (\triangledown p^n, \triangledown v) \\ &= (\rho^n, v) + (f^n, v) \quad \forall v \in W_h^p, \end{aligned} \tag{7.8}$$

where $\rho^n = \partial_t^2(\pi p^n) - \frac{\partial^2 p}{\partial t^2}(t_n)$ and $p^n = p(., t_n)$.

Subtracting (7.8) from (7.4) and using (7.7), we get the following error equation:

$$(\partial_t^2 \Theta^n, v) + (\triangledown \Theta^n, \triangledown v) = -(\rho^n, v) \quad \forall v \in W_h^p, \tag{7.9}$$

where

$$||\rho^n|| \leq C(\tau^2||(D_t^4 p)^n|| + h^2||(D_x^2 D_t^2 p)^n||). \tag{7.10}$$

We now choose

$$v = \frac{1}{2}\Big(\frac{\Theta^{n+1}-\Theta^n}{\tau} + \frac{\Theta^n - \Theta^{n-1}}{\tau}\Big) \tag{7.11}$$

and use the fact that

$$\partial_t^2 \Theta^n = \frac{\Theta^{n+1} - 2\Theta^n + \Theta^{n-1}}{\tau^2} = \frac{1}{\tau}\left(\frac{\Theta^{n+1}-\Theta^n}{\tau} - \frac{\Theta^n - \Theta^{n-1}}{\tau}\right),$$

to get

$$\begin{aligned}\frac{1}{2\tau}\left(\frac{\Theta^{n+1}-\Theta^n}{\tau} - \frac{\Theta^n - \Theta^{n-1}}{\tau}, \frac{\Theta^{n+1}-\Theta^n}{\tau} + \frac{\Theta^n - \Theta^{n-1}}{\tau}\right) &+ \frac{1}{2\tau}(\bigtriangledown\Theta^n, \bigtriangledown\big(\Theta^{n+1} - \Theta^{n-1}\big)) \\ &= \frac{1}{2\tau}(\rho^n, \Theta^{n+1} - \Theta^{n-1}),\end{aligned}$$

which reduces to

$$\left(\frac{||\Theta^{n+1}-\Theta^n||^2}{\tau^2} - \frac{||\Theta^n - \Theta^{n-1}||^2}{\tau^2}\right) + (\bigtriangledown\Theta^n, \bigtriangledown\big(\Theta^{n+1} - \Theta^{n-1}\big)) = (\rho^n, \Theta^{n+1} - \Theta^{n-1}). \tag{7.13}$$

Summing over n in the first term of (7.13) we get :

$$\sum_{n=1}^{N-1} \frac{||\Theta^{n+1}-\Theta^n||^2}{\tau^2} - \frac{||\Theta^n - \Theta^{n-1}||^2}{\tau^2} = \frac{||\Theta^N - \Theta^{N-1}||^2}{\tau^2}, \tag{7.14}$$

and in the second term of (7.13) :

$$\sum_{n=1}^{N-1} (\bigtriangledown\Theta^n, \bigtriangledown\big(\Theta^{n+1} - \Theta^{n-1}\big)) = (\bigtriangledown\Theta^N, \bigtriangledown\Theta^{N-1}), \tag{7.15}$$

and thus we have:

$$\frac{||\Theta^N - \Theta^{N-1}||^2}{\tau^2} + (\bigtriangledown\Theta^N, \bigtriangledown\Theta^{N-1}) = \sum_{n=1}^{N-1} (\rho^n, \Theta^{n+1} - \Theta^{n-1}). \tag{7.16}$$

Using an inverse estimate and assuming $\frac{c\tau}{h} \leq 1$ we have

$$\begin{aligned}(\bigtriangledown\Theta^N, \bigtriangledown\Theta^{N-1}) &= (\bigtriangledown\Theta^N, \bigtriangledown\Theta^N) - (\bigtriangledown(\Theta^N - \Theta^{N-1}), \bigtriangledown\Theta^N) \\ &\geq ||\bigtriangledown\Theta^N||^2 - \tau||\frac{\bigtriangledown(\Theta^N - \Theta^{N-1})}{\tau}|| \cdot ||\bigtriangledown\Theta^N|| \\ &\geq ||\bigtriangledown\Theta^N||^2 - \frac{c\tau}{h}||\frac{(\Theta^N - \Theta^{N-1})}{\tau}|| \cdot ||\bigtriangledown\Theta^N|| \\ &\geq \frac{1}{2}||\bigtriangledown\Theta^N||^2 - \frac{1}{2}||\frac{(\Theta^N - \Theta^{N-1})}{\tau}||^2.\end{aligned}$$

We conclude that

$$\frac{||\Theta^N - \Theta^{N-1}||^2}{\tau^2} + ||\bigtriangledown \Theta^N||^2 \leq 2\tau \sum_{n=1}^{N-1} \left(\rho^n, \frac{\Theta^{n+1} - \Theta^{n-1}}{2\tau} \right). \tag{7.17}$$

Using the definition of ρ and (7.10), we get:

$$\sum_{n=1}^{N-1} (\rho^n, \frac{\Theta^{n+1} - \Theta^{n-1}}{2\tau}) \approx \sum_{n=1}^{N-1} (\tau^2 ||(D_t^4 p)^n|| + h^2 ||(D_x^2 D_t^2 p)^n||) \partial_t \Theta. \tag{7.18}$$

We substitute this expression into (7.17) to get:

$$||\partial_t \Theta||^2 + ||\bigtriangledown \Theta^N||^2 \leq 2\tau \left(\tau^2 \int_0^T ||(D_t^4 p)^n|| dt + h^2 \int_0^T ||(D_x^2 D_t^2 p)^n|| dt \right) \cdot \max_n ||\partial_t \Theta^n||,$$

which gives the following a priori estimate for Θ:

$$\max_n \left(||\frac{\partial \Theta^n}{\partial t}|| + ||\bigtriangledown \Theta^n||^2 \right) \leq C(\tau^2 + h^2). \tag{7.20}$$

Using the fact that

$$\max_n (||\partial_t (p - \pi p)^n|| + ||\bigtriangledown (p - \pi p)^n||) \leq C_1 (\tau + h), \tag{7.21}$$

we get finally

$$\max_n (||\partial_t (p - p_h)^n|| + ||\bigtriangledown (p - p_h)^n||) \leq C_1 (\tau + h), \tag{7.22}$$

Integrating (7.22) we can also obtain

$$\max_n (||(p - p_h)^n|| + ||\bigtriangledown (p - p_h)^n||) \leq C(\tau + h). \tag{7.23}$$

8. IMPLEMENTATION ISSUES

We have chosen C++ as the implementation language. It allows us to implement the problem and the algorithms on a high level of abstraction without much loss of efficiency. We have implemented important notions such as grid, boundary, operator, and grid function as C++ classes. The software package PETSc [2] is used for matrix vector computations. See [4] for more information about implementation of the hybrid method.

The run program to implement reconstruction algorithm for solving inverse problem is written in perl and has following form:

```
$FILE = $ARGV[0];
$data =  $ARGV[1];
$rc=0;
        # return code of step4 controls the loop
        # 0 means continue, 1 -- the fixpoint is reached, 0xFF -internal error
$iter = 0;

$|=1;
$code1 = 0;
```

```
$code2 = 0;

# compute exact solution of the problem
 `step1_3d_4w $FILE $data`;

while ($rc==0)
{
    $iter++;

    $it = $iter;

    print"\n\n################ iteration $it ######\n";
# compute solution of the forward problem

     `step2_4waves $FILE $data`;
# compute solution of the adjoint problem
    print `step3_3d_4w adjref2_ubot.dat $data`;

    $code2 = 0;
# alfa-optimization algorithm
while ( $code2 == 0)
{
    $code1 = 0;

    while ($code1 == 0 )
    {

         print `step4_alfa_3d $FILE $data`;
        `alfa_4w $FILE alfa2.m $data`;  # solve problem with parameter alfa2

        `alfa_4w  $FILE  alfa.m $data`;  # solve problem with parameter alfa

         print `find_alfa2_3d $FILE $data`;  # returns code1
        $code1 = $? >> 8;
    }


    `alfa_4w $FILE alfa1.m $data`;  # solve problem with parameter alfa1

    print  `find_alfa1_3d $FILE $data`;  # returns code2
    $code2 = $? >> 8;
    print"\n\n  after find_alfa1_3d : code2 is $code2.\n";


}
```

```
 print"\n\n===== works step4 ---> compute new velocity \n ";
  print ‘step4_3d $FILE $data ‘;

‘cp New_Vel.m  ref2_4w_$it.m‘;

‘cp vel.inp   ref2_4w_$it.inp‘;

print"return code = $rc \n";
}

print  $rc==1 ? "fixpoint reached after $iter iterations \n" :
    "there was an internal error\n";
```

9. Numerical examples

In this section we present computational results for our adaptive method for two and three dimensional inverse scattering problems.

In all the examples we solve the model problem (2.1) in the domain $\Omega = [0,1]^d,\ d = 2,3$, with a combination of Dirichlet and absorbing boundary conditions, and with the initial conditions $u = \frac{\partial u}{\partial t} = 0$. The domain Ω is decomposed into two domains Ω_{FEM} and Ω_{FDM} in such a way that their meshes overlap in two layers of nodes, see [4]. In Ω_{FDM} domain we effectively use a FDM with a combination of Dirichlet and absorbing boundary conditions. The space mesh for finite element method in in Ω_{FEM} consists of triangles, and in the three dimensional examples of tetrahedra. For the reconstruction we use plane waves initialized at a boundary of Ω. A plane wave, moving in the positive x_1 direction, has the form

$$u(x,t) = \begin{cases} 0.1 \sin\left(k\ (t - x_1) - \pi/2\right) + 0.1 & \text{if } 0 \le t - x_1 \le \frac{2\pi}{k}, \\ 0 & \text{else,} \end{cases}$$

and is initialized for $x_1 = 0$ by using a Dirichlet condition.

We choose $k = 25, 100, 200$, corresponding to increasingly sharp waves. In one experiment we use one plane wave moving from left to right side. In other experiments we use several plane waves, moving from left to right, from right to left or from top to bottom or bottom to top.

For all computational tests we choose a time step to respect the CFL criterion:

$$\tau \le \frac{h}{ac} \ , \tag{9.1}$$

where h is the minimal local mesh size of the elements, and a is a constant.

We choose the observation points outside in Ω_{FDM} close to Ω_{FEM} giving somewhat better reconstruction.

First, we present a model applications of reconstructing one or two elliptic scatterers in two dimensions. Next, we present three dimensional reconstructions of a one or two scatterers. In all the examples we apply the adaptive mesh refinement algorithm described in Section 6.

9.1. **Two dimensional numerical examples.** In this section we describe the experiments with one and two plane waves and one or two elliptical scatterers. Appropriate initial values of the parameters in both experiments in the one dimensional optimization algorithm are $\alpha = 0.001$ and $\beta = 0.001$. In the experiments with one plane wave, the wave is initialized at the left boundary $x_1 = 0$ of Ω and goes through the Ω_{FEM} to the right boundary $x_1 = 1$ during the time interval $[0, t],\ t < T$. We use absorbing boundary conditions on the other boundaries. As the observation points we choose eleven nodes near the right boundary at $x = 0.88$.

We obtain the data for the reconstruction by solving the direct problem with $c = 1.3,\ 1.5$ in the scaterrer, depending on the experiment, and $c = 1$ outside. The solution is presented in Fig. 6-a. The observation time is 0.5 and the time step is 0.002.

In Fig.(6-c - 6-d) we present reconstructions of a single elliptic scatterer. We start running the reconstruction algorithm with $c = 1.0$ in the whole domain and continue until the stopping condition $||\frac{\partial J}{\partial c}||_{L_2} < 0.001$ is achieved.

To improve the reconstruction and achieve better convergence we use the adaptive algorithm, described in Section 6.

The mesh refinement algorithm is as follows(we call two elements neighbours if they have a common edge):

1. Mark all elements to be refined with code 1, neighbors of these elements with a common edge - mark with code 2, and neighbors of the elements with code 2 - mark with code 3.
2. Check condition: if neighbors of the element with code 2 have code 2 - mark both elements with code 3.
3. If the element with code 2 has two neighbour elements with code 1, we mark these elements with code 1 and go to the step 1.
4. Refine elements with code 1 into 4 triangles, using the middle points of the edges.
5. Refine elements with code 2 into 3 triangles. When refining by three triangles, we take the largest side of the triangle and find middle point of this side.
6. Triangulate elements with code 3 into 2 triangles in the following way : make connections between added new nodes and nodes in the elements from the coarse grid.

To illustrate the effect of the adaptively refined meshes we present reconstructions of one and two ellipses. The exact scatterer is shown in Fig. 2-a. In Fig. 3 we present a sequence of adaptively refined meshes for reconstruction this scatterer.

We make experiments with two plane waves: the one plane wave is initialized at the left boundary $x_1 = 0$ of Ω and goes through the Ω_{FEM} to the right boundary $x_1 = 1$, and a second plane wave is initialized at the right boundary $x_1 = 1$ of Ω_{FDM} and goes through the Ω_{FEM} to the left boundary $x_1 = 0$ during the time interval $[0, t],\ t < T$. We use absorbing boundary conditions on the other boundaries.

The observation points are located at both the left and right sides of the object, as described in Section 9, using a total of 22 observation points. The maximal computational time is 1.0 and the number of the time steps is 500 for the exact type of material equal to 1.3, and 1000 for the exact type of material equal to 2.0 inside the scatterer. The computations have been performed on three different locally adaptive refined meshes. See Fig. 10 for the exact type of material equal to 2.0, and Fig. 11-13 for the exact type of material equal to 1.3. In Fig. 5-a we present an exact solution of the problem (2.1) at the time 0.18, and at Fig. 5 - b,c,d we present the solution of the forward problem for different meshes at the same moment of time.

Another example is a reconstruction of two ellipses with the exact type of material equal to 1.3 inside the scatterers. The computer simulations are presented in Fig. 14 – 15. The exact scatterers are shown in Fig. 2-b .

The Tables 1 and 2 show that we can gain a significant reduction of the number of iterations steps in the reconstruction algorithm by using appropriately refined meshes.

9.2. **Three dimensional numerical examples.** In this section we describe the experiments to reconstruct three dimensional objects. We use the adaptive algorithm for computation of the parameter c described in Section 6. We perform experiments with $2, 4$ and 6 plane waves. Appropriate initial values of the parameters in both experiments in the one dimensional optimization algorithm are $\alpha = 0.01$ and $\beta = 0.01$. The adaptive method is as follows:

(1) Choose an initial mesh K_h and an initial time partition J_k of the time interval $[0, T]$.
(2) Compute the solution p on K_h and J_k of the forward problem (2.1) with $c = c^{(n)}$.
(3) Compute the solution λ of the adjoint problem

$$\frac{1}{c^2}\frac{\partial^2 \lambda}{\partial t^2} - \triangle \lambda = -(p - \tilde{p})\delta_{obs}, \quad x \in \Omega, 0 < t < T, \tag{9.2}$$

on K_h and J_k.

(4) Update the velocity on K_h and J_k according to

$$c^{(n+1)}(x) = c^{(n)}(x) - \alpha^{(n)} \frac{2}{{c^{(n)}}^3} \int_0^T \frac{\partial \lambda(x,t)}{\partial t} \frac{\partial p(x,t)}{\partial t} dt. \tag{9.3}$$

(5) Compute a posteriori error estimate for c defined in (6.19).
(6) Refine all tetrahedra, where $R_c \sigma_c > \varepsilon$, ε is a desired tolerance. Each tetrahedron can be divided into 2, to 8 new tetrahedra. We present different ways of the refinement, of the one tetrahedra in Fig. 4.
(7) Construct a new mesh K_h: we connect a refined grid with the previous one, to maintain the consistence of the grid.

9.2.1. *Example 1.* In this test we use four plane waves from the left,right,top and bot boundaries. To get data for the reconstruction we solve the wave equation with four plane waves of the form (9) with $k = 100$. The time interval is $[0, 0.4]$ and is divided into four phases, 0.1 each: first, one plane wave starts at the left boundary $x_1 = 0$ of Ω and goes through the Ω_{FEM} to the right boundary $x_1 = 1$, then at time 0.1 a second wave starts at the right boundary $x_1 = 1$ of Ω and goes to the left, and the third and fourth plane waves begin at time 0.2 and 0.3 at the top/bottom boundaries of Ω_{FDM}, respectively, and goes to the bottom/top boundaries. We use absorbing boundary conditions on the other boundaries. The observation points are placed around the object.

We obtain the data for the reconstruction by solving the direct problem (2.1)-(2.3) with $c = 2$ in the scatterer and $c = 1$ outside.

In Fig. 16 we present the computed exact solution of the problem (2.1) – (2.3) inside Ω_{FEM} with four plane waves and absorbing boundary conditions at all the boundaries. The computer simulations of the reconstructed scatterer on the adaptive refined meshes are shown in Fig. 20. First, we compute parameter c using the reconstruction algorithm on the coarse grid. Then we refine the coarse grid by using the estimate (6.19), and interpolate the previously computed parameter c into a new refined mesh. This value is used as starting value for computations on the new mesh. Next, we perform all the steps of the reconstruction algorithm and repeat the previously described procedure of the interpolation of the computed parameter c into the refined mesh, until the desired tolerance is achieved.

9.2.2. *Example2. Reconstruction using quasi-Newton method.* We recall that we update the coefficient c by

$$c^{k+1} = c^k + \alpha^k H^k(m) g^k, \tag{9.4}$$

where $H^k(m)$ is a quasi-Newton matrix and m is the number of the stored correction pairs (s_k, y_k), defined by

$$s_k = c_{k+1} - c_k, \tag{9.5}$$

$$y_k = g_{k+1} - g_k. \tag{9.6}$$

Here, c_k and g_k are the stored vectors of the computed parameters and the gradients at the iteration k.

Instead of forming the Hessian $H(m)$, we can store scalars ρ_i and vectors $(s_i, y_i), i = k-1, ..., k-m$, which determine the matrices V_i, defined as (5.6). The advantage of the recursive formula (5.3) is that we can compute the product $H^k(m) \cdot g^k$ without matrix vector multiplications. In the numerical examples we choose $m = 0, 5, 7$, which corresponds to the number of stored vector pairs (s_k, y_k) equal to $1, 6, 8$. Note, that when $m = 0$ we have the usual steepest descent method. To perform quasi Newton computations, we define the first m approximations of the Hessian $H^k(m)$ using BFGS update formula, until the storage is full. Then we delete the oldest correction pair from the set (s_k, y_k) and add a new one . Then the new Hessian approximation is defined again by (5.3), using the newly added set pair. This process is repeated during all iterations of the optimization algorithm.

We performed different experiments with $m = 0, 5, 7$ on adaptively refined meshes. First we present reconstruction of a cube inside Ω_{FEM}. The exact scatterer is shown in Fig. 2-e. The computational mesh inside Ω_{FEM} is unstructured and was generated using a quality tetrahedral mesh generator, which can be obtained from http://www.weboo.com/sh. The meshes for computing the data are presented in Fig. 1. In the Fig. 28-a we present computed L2-norms $\frac{\partial L}{\partial c}$ of the gradient for different adaptively refined meshes. Here, the number of stored corrections is $m = 5$. The number of the nodes and elements in the adaptively refined meshes are presented in the Table 5. We restart the quasi-Newton method on a new mesh interpolating the computed approximation on the previous mesh into the new mesh.

Second example is a reconstruction of star-shaped scatterer presented in Fig. 2-d. In Fig. 28 - b we present computed L2-norms of the gradient for different adaptive refined meshes. The number of stored corrections is $m = 5$. The number of the nodes and elements in the adaptively refined meshes are presented in the Table 4. In the Table 4 we present the effectiveness of the quasi-Newton iteration, compared with a steepest descent method. We note that increasing the storage beyond 5 corrections will not have a very big effect: the time, which is required to compute the Hessian, will increase with no increase in accuracy.

The reconstructed objects in both examples can be viewed in Fig. 21 and Fig. 25.

9.2.3. *Example 3. Reconstruction with reflected waves.* In order to get a better reconstruction of the object we tested a focusing technique, letting the incoming wave be equal to a reflected non-plane measured wave. It seems to be helpful to make tests with reflected waves from the boundaries of Ω_{FDM}. We hope that with such kind of tests we can get more information about the nature of the object.

Tests with non-plane incoming waves from the left and the right side of Ω_{FDM}.

We modulate problem in the following steps:

0. A plane wave (9) is initialized at the left boundary of Ω_{FDM} during the time interval $[0, \frac{2\pi}{k}]$. We use absorbing boundary conditions on the other boundaries.
1. Values at the observation points of the incoming wave are registered during the time interval $[0, T]$.
2. Values of the incoming wave are registered at time T at the right boundary of Ω_{FDM}.
3. A non-plane wave is initialized at the right boundary of Ω_{FDM} during the time interval $[T, T + \frac{2\pi}{k}]$, using the values of 2.
4. Values at the observation points are registered during the time interval $[T, 2T]$ for the new incoming wave.
5. Values of the incoming wave are registered at time $2T$ at the left boundary of Ω_{FDM}.
6. A new incoming non-plane wave is initialized at the left boundary of Ω_{FDM} during time interval $[2T, 2T + \frac{2\pi}{k}]$, using the values of 5.
7. Values at the observation points are registered during the time interval $[2T, 3T]$ for incoming wave.
8. Values of the incoming wave are registered at time $3T$ at the right boundary of Ω_{FDM}.

Steps $3 - 8$ can be performed many times. We have tested with waves being reflected 2 and 4 times. The observation points are placed in Ω_{FDM} at the left and right side of Ω_{FEM}. In our examples we have used 76 observation points, 38 at the left and 38 at the right side of Ω_{FEM}. To illustrate the strategy with reflected waves, we try to reconstruct the object given in Fig. 23-a. The reconstructed object is presented in Fig. 23-b using a $5-6$ times adaptively refined mesh. In the optimization algorithm we have used quasi-Newton method with $m = 0, 5, 11$. We present the computed L_2-norm of the gradient for different adaptive refined meshes in Fig. 27-23-c,d. We see, that the best reconstruction is obtained with a 4 times refined mesh. If we refine 5 or 6 times, the L_2-norm of the computed gradient increases. This is because we use a $5-6$ times refinement to compute the data. We see a significantly better convergence, when we use 5 or more stored corrections in the quasi-Newton method than in the obvious steepest descent method, what corresponds to $m = 0$.

Tests with reflected wave from the left side of Ω_{FDM}.

We perform a similar experiments with reflected waves. The observation points are placed at the left boundary of Ω_{FDM}. We modulate the problem in following steps:

0. A plane wave (9) is initialized at the left boundary of Ω_{FDM} during the time interval $[0, \frac{2\pi}{k}]$. We use absorbing boundary conditions on all boundaries.
1. Values at the observation points are registered during the time interval $[0, T]$ for the reflected wave.
2. Values of the reflected wave are registered at time T at the left boundary of Ω_{FDM}.
3. A new incoming non-plane wave is initialized at the left boundary of Ω_{FDM} during the time interval $[T, T + \frac{2\pi}{k}]$.
4. Values at the observation points are registered during the time interval $[T, 2T]$ for incoming wave.
5. Values of the reflected wave are registered at time 2T at the left boundary of Ω_{FDM}.

Steps $3-5$ can be repeated. We have tested with 2 and 3 times reflected waves. The observation points are placed as in the previous example. To test this model, we reconstructed the object, given in Fig. 23-a. We performed computations of the inverse problem on a 5 times adaptive refined meshes. In the optimization algorithm we have used quasi-Newton method with different numbers of the stored corrections equal to $m = 0, 5, 11$. We present the computed L_2-norm of the gradient for different adaptive refined meshes in Fig. 29. As we see from the computed norm, these model gives worse result than the previous model with reflected waves from the left and the right side of Ω_{FDM}.

9.3. **Performance comparisons.** We demonstrate the performance comparison of the adaptive mesh refinement procedure on the computer implementation of the reconstruction algorithm. The benchmarks were run on Intel 600 Mhz processor with 192800Mb memory, 512992Mb in swap, 274280Mb used and 234780MB free memory. Table 1 presents the rate of the convergence to reconstruct an ellipse. Inside an ellipse the wave distribution speed parameter $c = 1.3$ and in the rest of the domain $c = 1.0$. In the Table 2is presented the rate of the convergence to reconstruct two ellipses. Computational grids in these tables are generated by the adaptive procedure. All tests have been performed for the time interval $[0, 1]$ with time step satisfying the CFL-criterion.

In the Table 3 we present performance of the computing for the forward and adjoint solutions of the two dimensional reconstruction in the terms of the memory and cpu time. The forward and adjoint solutions have been computed for a time interval $[0, 1]$ with the

Type of the mesh	No. of nod.	Rate of the convergence, n.r it.	eps
coarse	624	85	0.001
1 time ref.	719	43	0.001
2 times ref.	852	16	0.001

Table 1. 2D reconstruction. Rate of convergence to reconstruct one ellipse.

Type of the mesh, 2D	No. of nodes	Rate of the convergence, n.r it.	eps
coarse	1039	120	0.001
1 time ref.	1322	49	0.001
2 time ref.	1506	35	0.001

Table 2. Rate of convergence to reconstruct two ellipses.

No. of nodes	No. of elements	Forward problem		Adjoint problem	
		mem	time, sec	mem	time,sec
466	834	19	5	36.0	6
624	1150	20.5	5.5	38.3	7
719	1340	21.1	7	39.3	9
852	1606	22	7.2	40.2	11
1160	2222	24	8	43.6	13
1296	2494	25	10	44.9	15

Table 3. Performance for the Forward and the Adjoint problems

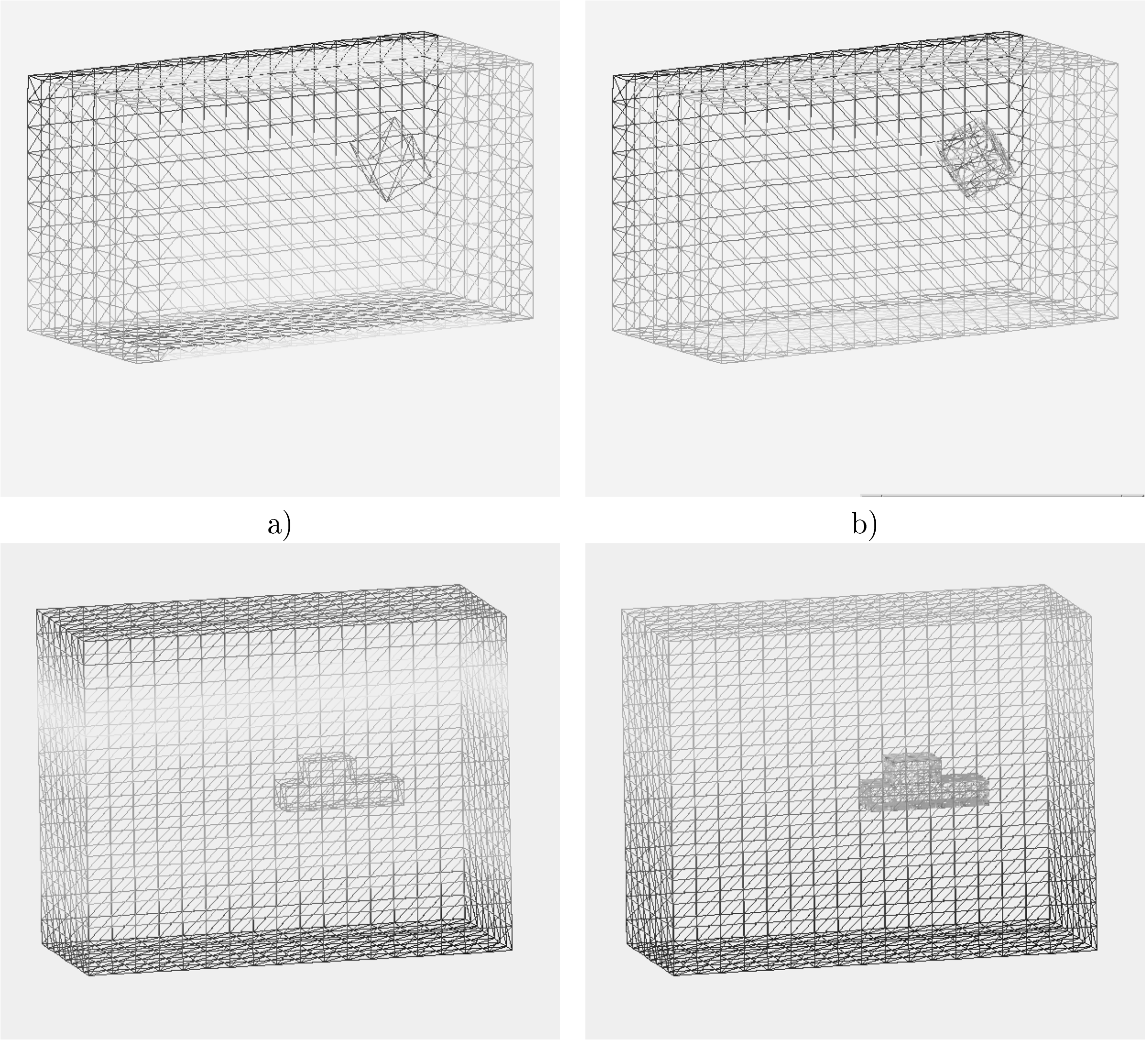
time step computed satisfying the CFL-criterion. We choose time step $\tau = 0.002$ in all the tests for all grids to compare memory consumption and time performance for different grids. In the Table 4 we present performance of the computing for the forward and adjoint solutions of the three dimensional reconstruction in the terms of the memory and cpu time. The forward and adjoint solutions have been computed for the time interval $[0, 4]$ with time step $\tau = 0.0002$. The tests have been performed with 6 plane waves, going from the 6 sides of the outer cubic domain.

No. of nodes	No. of elements	Forward problem		Adjoint problem	
		mem	time, min	mem	time,min
466	1908	15	0.23	50.7	1.05
683	2852	17	1.02	51.2	1.56
1097	5410	18	1.34	51.8	2.28
1396	7248	19	2.02	52.3	3.02
2041	10006	20.5	2.30	54	3.28
2565	12096	21	2.41	55	3.31
2812	13578	22	2.2	57	3.49
3376	16008	25	3.54	58	4.50
4342	21854	28	4.50	60	6.36
7270	39602	30	5.76	70	8.38

Table 4. Performance for the Forward and the Adjoint problems in the three dimensional reconstruction of the star shaped object.

No. of nodes	No. of elements	Forward problem		Adjoint problem	
		mem	time, min	mem	time,min
1430	5981	17.1	1.34	50	2.02
1559	6621	18.8	1.40	58	2.30
1813	8097	24.6	1.80	65	3.15
2117	9789	25	2.2	68	3.40

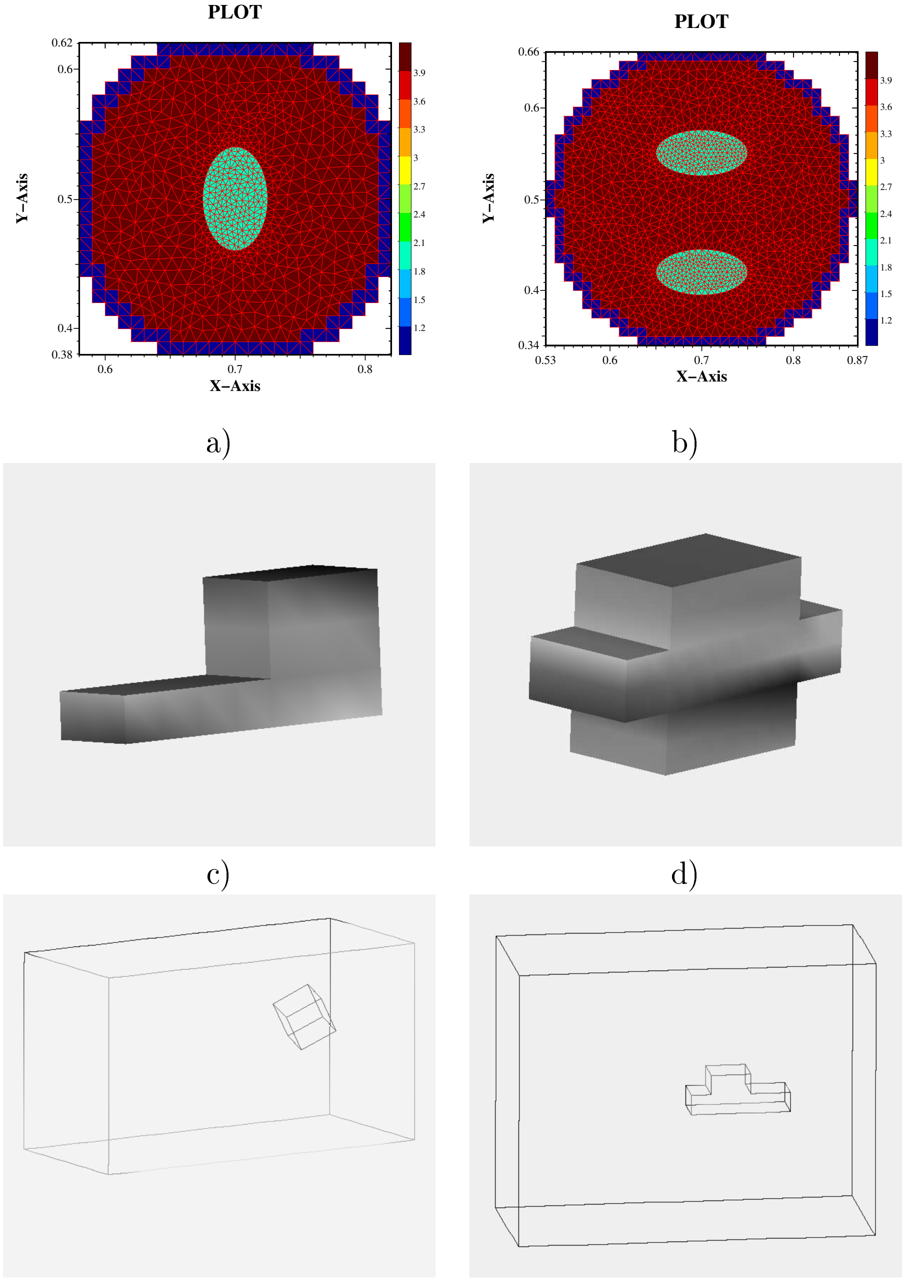
Table 5. Performance for the Forward and the Adjoint problems in the three dimensional reconstruction of the cubic object.



a) b)

c) d)

Figure 1. a) Coarse mesh constructed using a quality tetrahedral mesh generator, see http://www.weboo.com/sh. b) Refined mesh for computations of the exact solution of the wave equation. c) Coarse mesh. d) Refined mesh for computations of the exact solution of the wave equation.

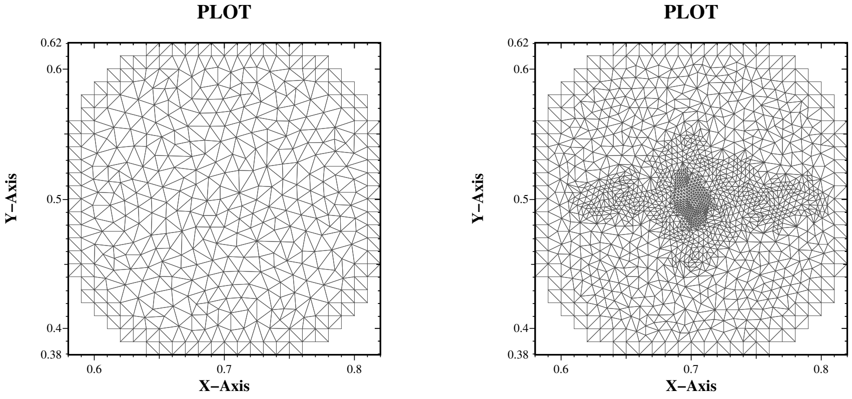


a) b)

c) d)

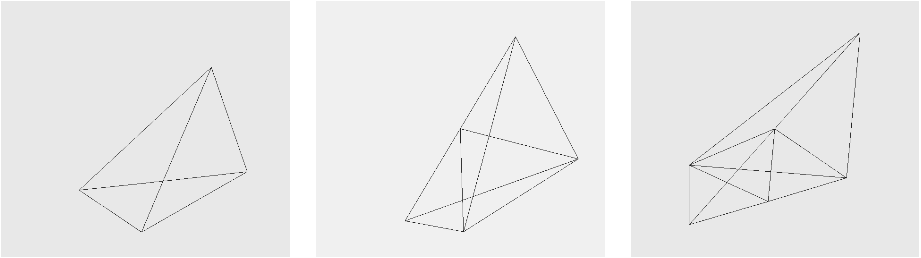
e) f)

FIGURE 2. Exact scatterers in two and three dimensions.



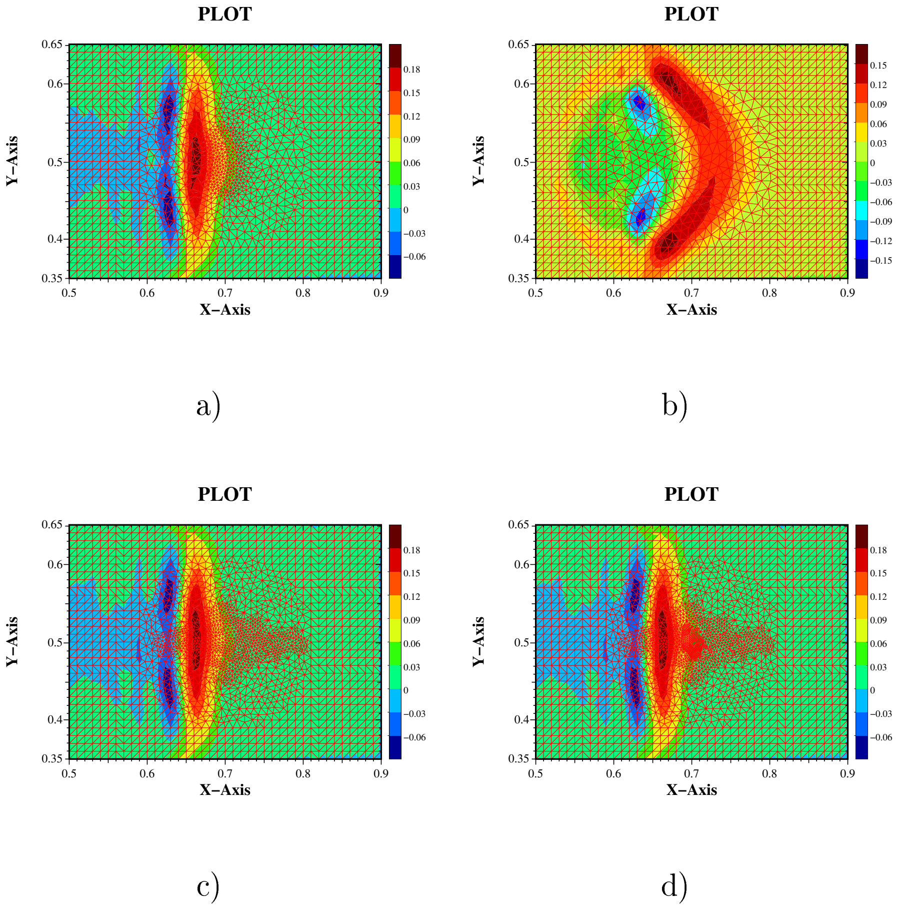
a) b)

FIGURE 3. Sequence of adaptively refined meshes for one elliptic scatterer. The exact scatterer is presented in Figure 2-a.



a) b) c)

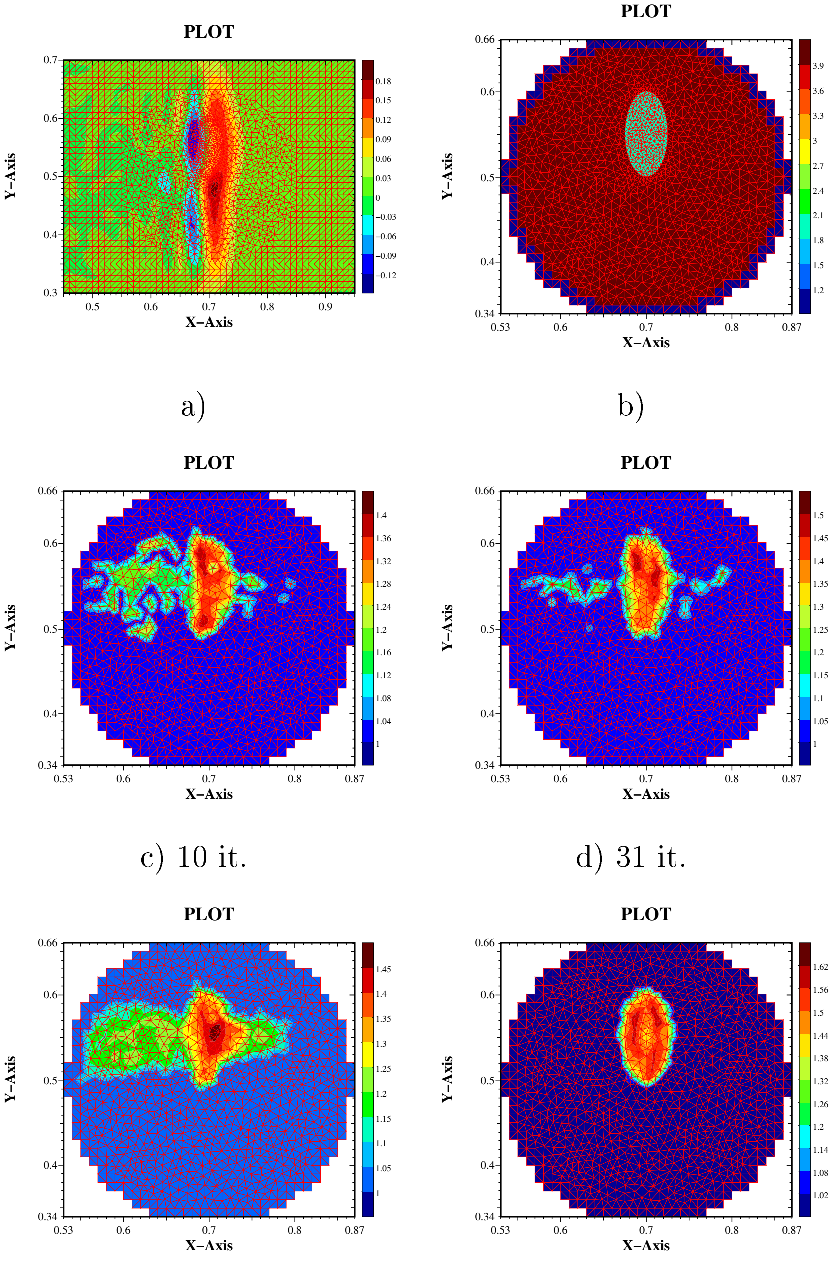
FIGURE 4. (a) tetrahedron, (b) refined tetrahedron into 2 tetrahedra, (c) refined tetrahedron into 3 tetrahedra.



a) b)

c) d)

FIGURE 5. The exact solution of the wave equation with one elliptic scatterer with $c = 2$ is presented in a), while b)-c) give computed approximated solution on different adaptively refined meshes at the time moment 0.18.

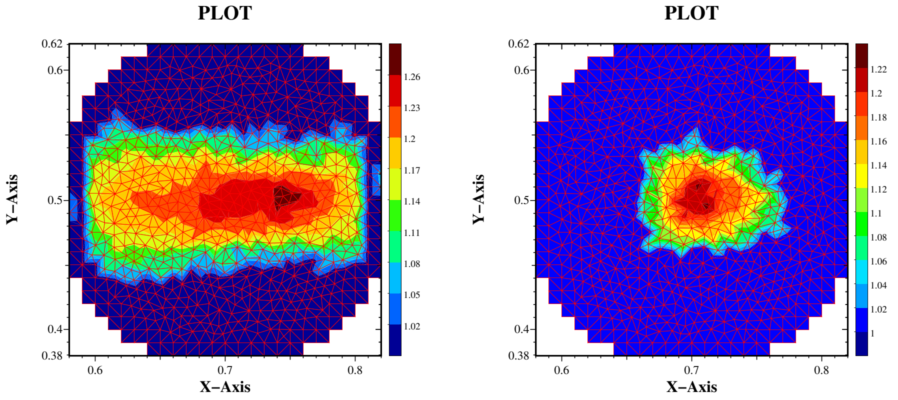


a) b)

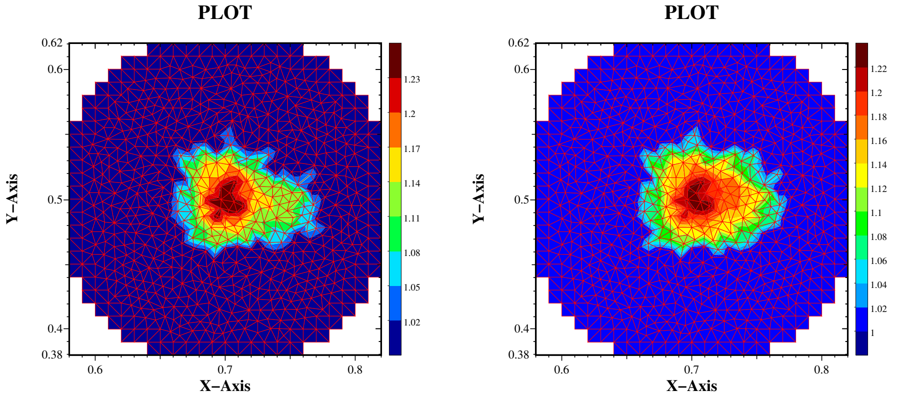
c) 10 it. d) 31 it.

e) 15 it. f) 65 it.

FIGURE 6. Reconstruction of one ellipse: a) exact solution with one plane wave with $c = 1.5$ in the scatterer and with absorbing b.c. at all the boundaries, b) single elliptic scatterer; c) - f) reconstructions of a single elliptic scatterer after $10, 15, 31$ and 65 iterations.

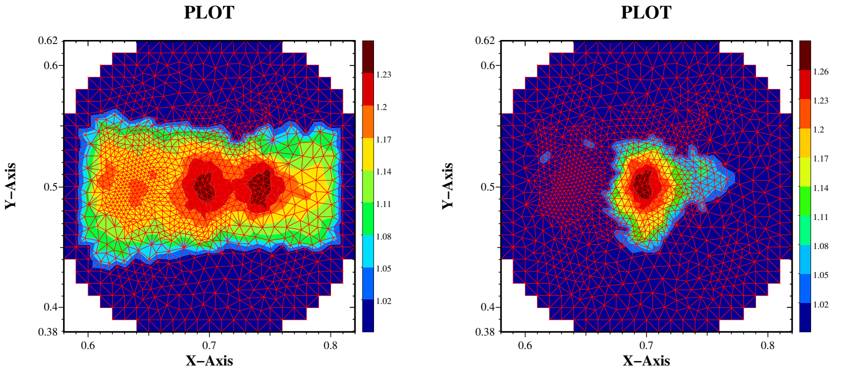


1-th it. 30-th it.

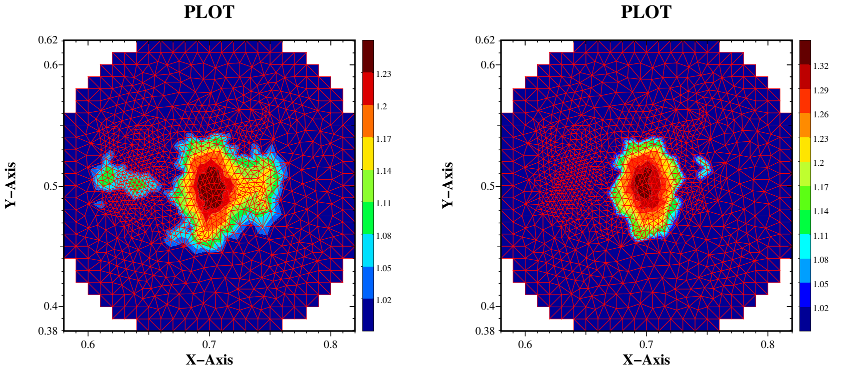


35-th it. 47-th it.

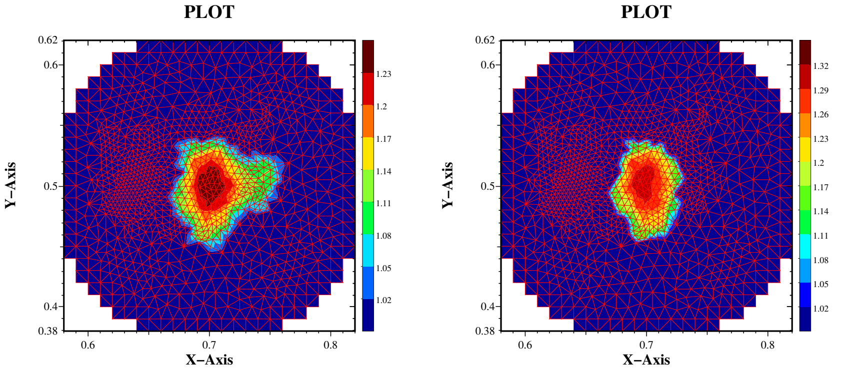
Figure 7. Reconstruction of one ellipse. The mesh consist of 656 nodes.



1-th it. 30-th it.

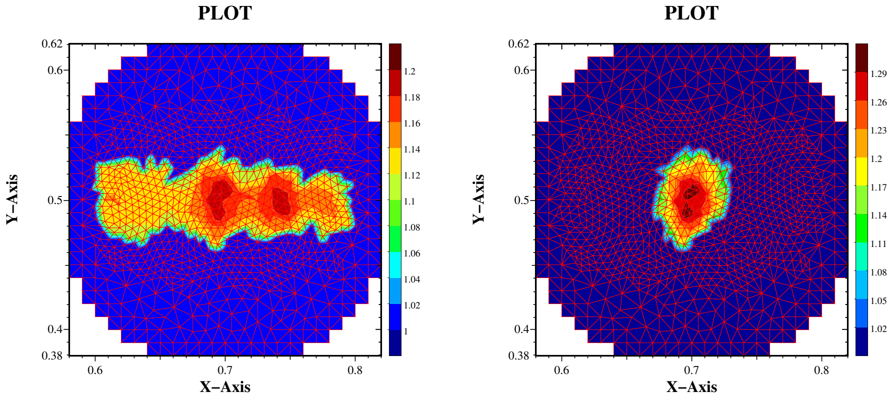


10-th it. 40-th it.

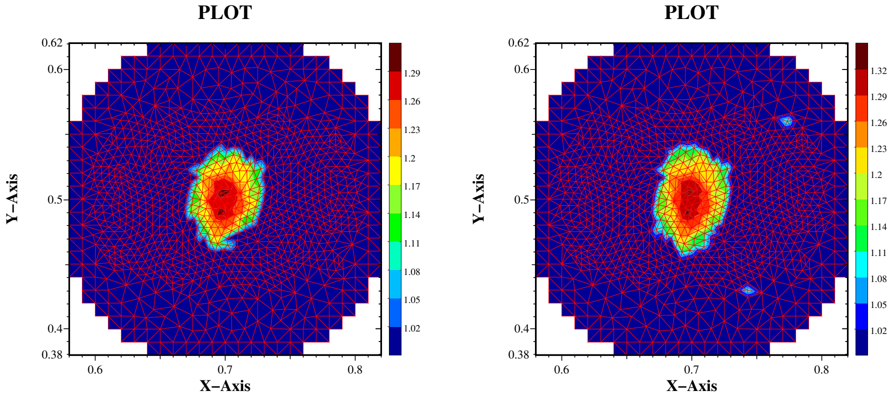


20-th it. 50-th it.

FIGURE 8. Reconstruction of one ellipse. The mesh consist of 1160 nodes.

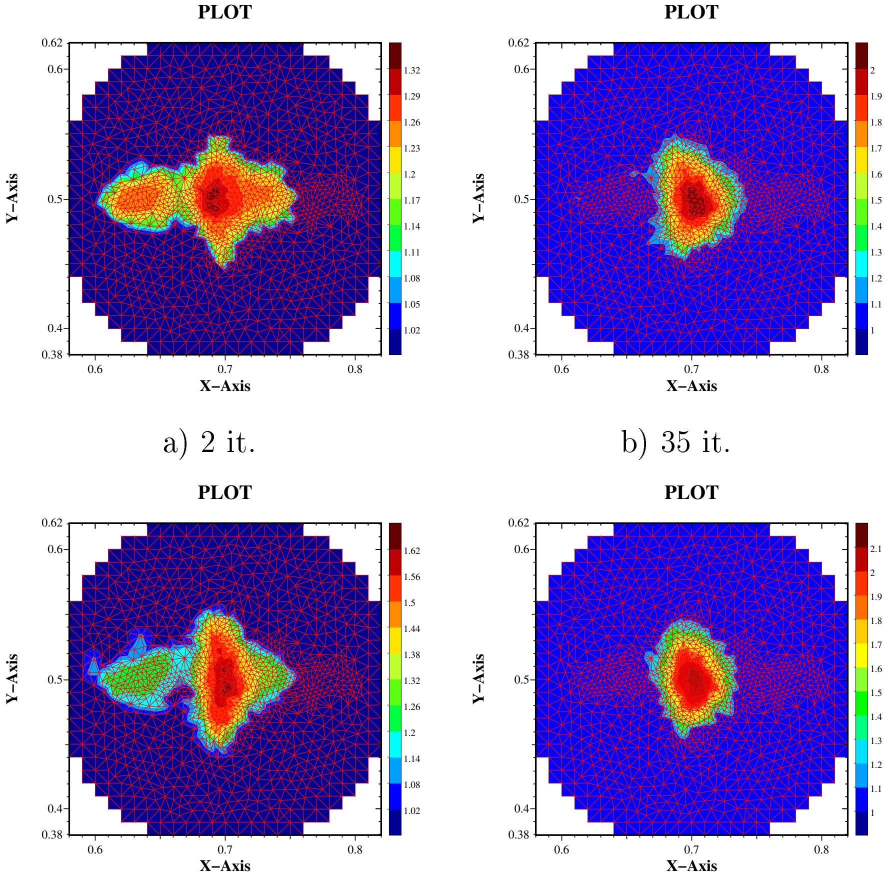


1-th it. 18-th it.



15-th it. 21-th it.

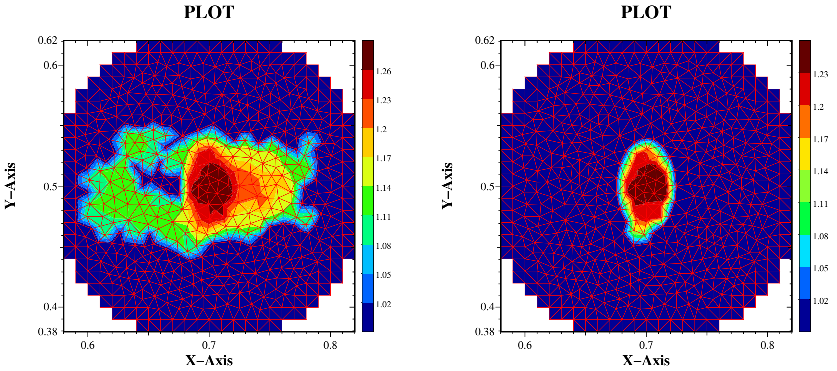
FIGURE 9. Reconstruction of one ellipse. The mesh consist of 1296 nodes.



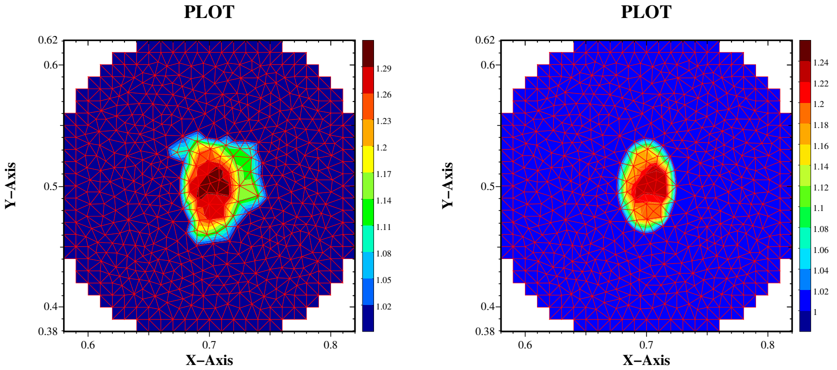
a) 2 it. b) 35 it.

c) 2 it. d) 11 it.

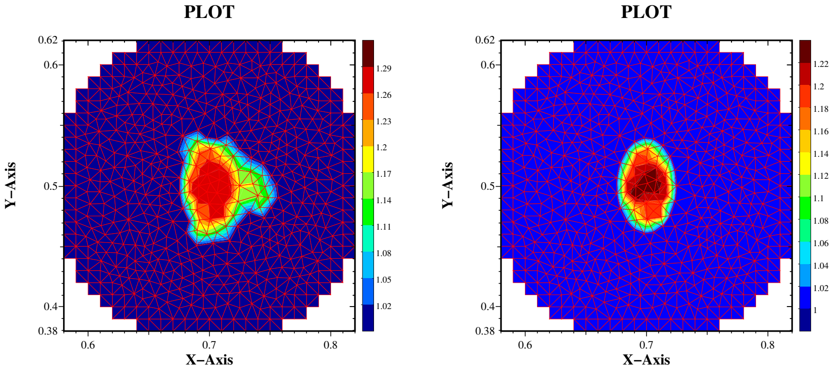
FIGURE 10. Reconstruction of a single elliptic scatterer with c=1.5 in the scatterer.



5-th it. 36-th it.

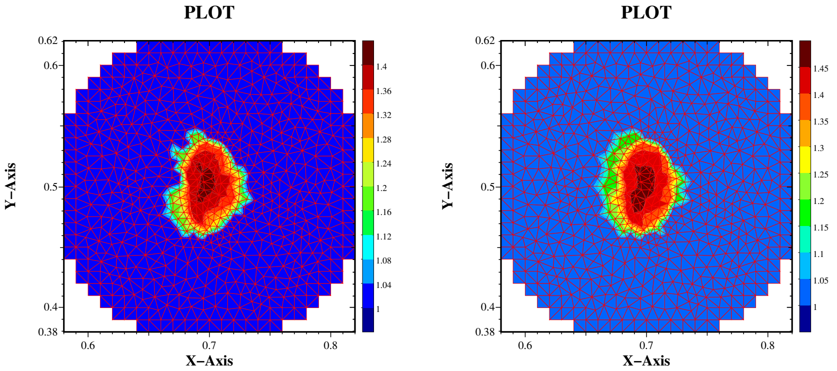


15-th it. 81-th it.

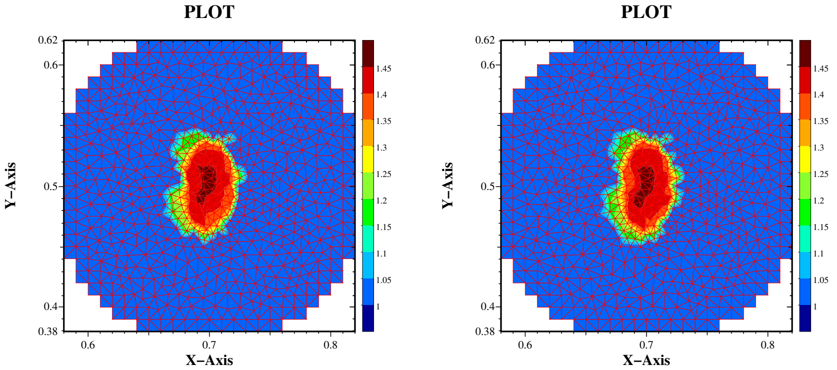


25-th it. 83-th it.

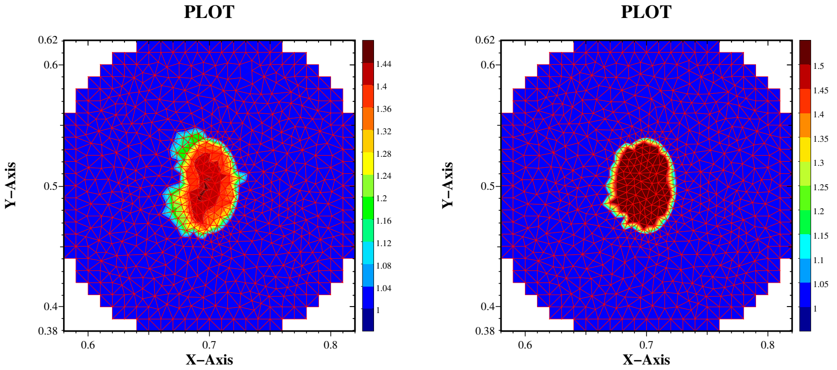
FIGURE 11. Reconstruction of a single elliptic scatterer on coarse mesh which consists of 624 nodes.



7-th it. 20-th it.

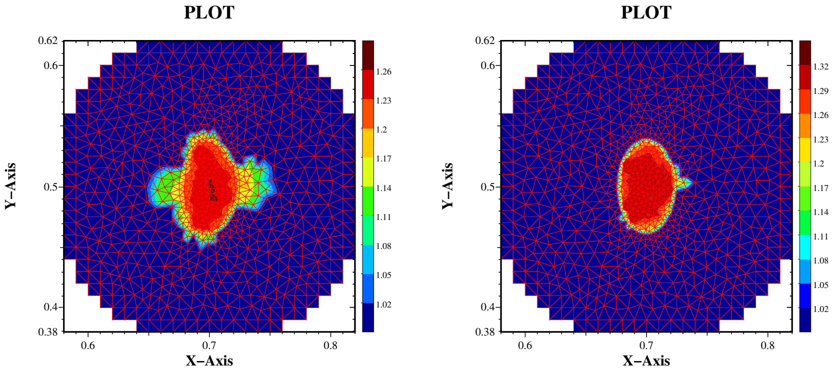


10-th it. 25-th it.

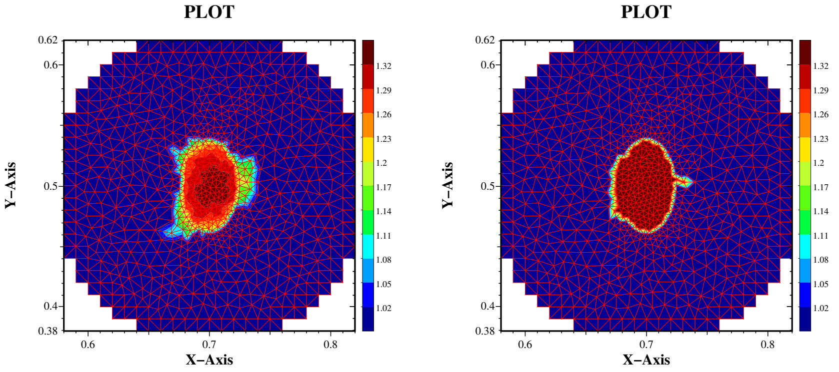


15-th it. 43-th it.

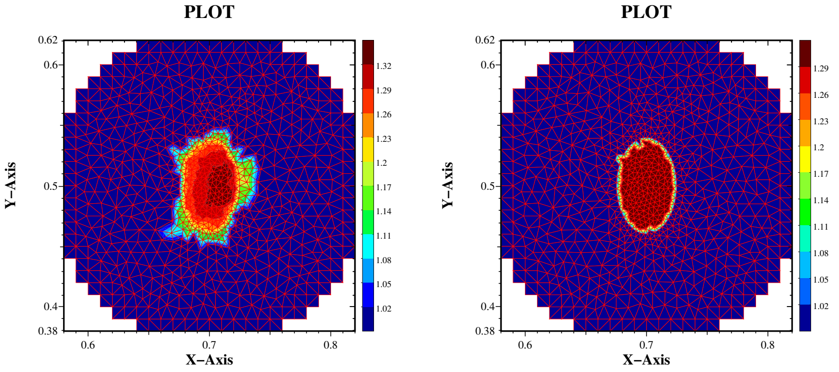
FIGURE 12. Reconstruction of a single elliptic scatterer on a mesh with 719 nodes.



5-th it. 13-th it.

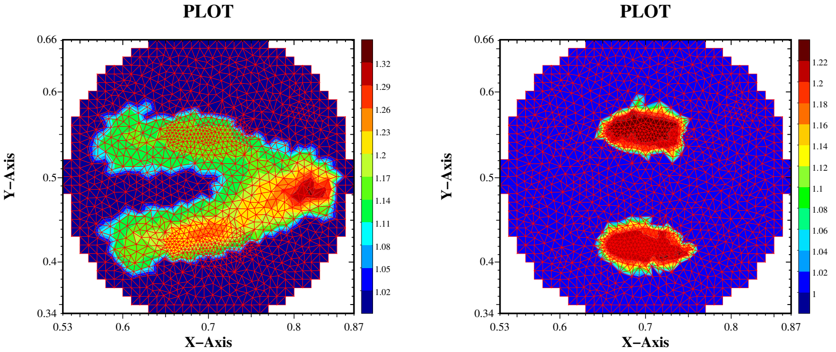


10-th it. 15-th it.

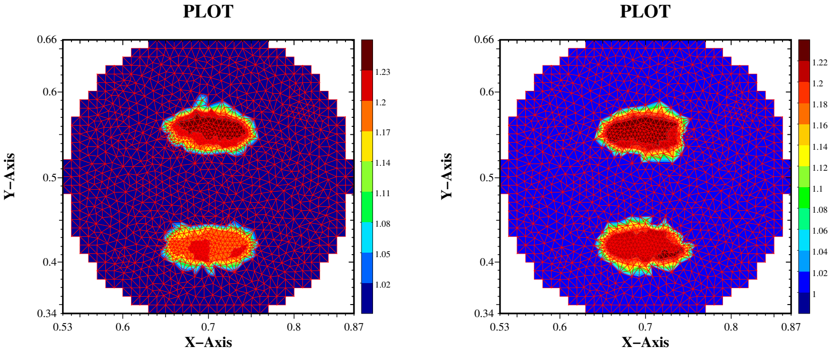


11-th it. 16-th it.

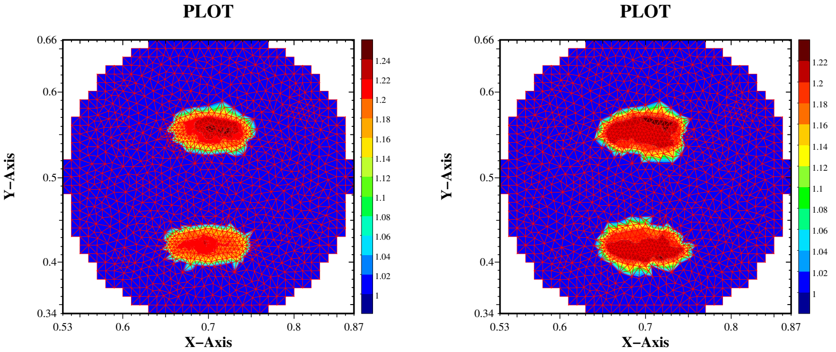
Figure 13. Reconstruction of a single elliptic scatterer on a mesh with 852 nodes.



1-th it. 42-th it.

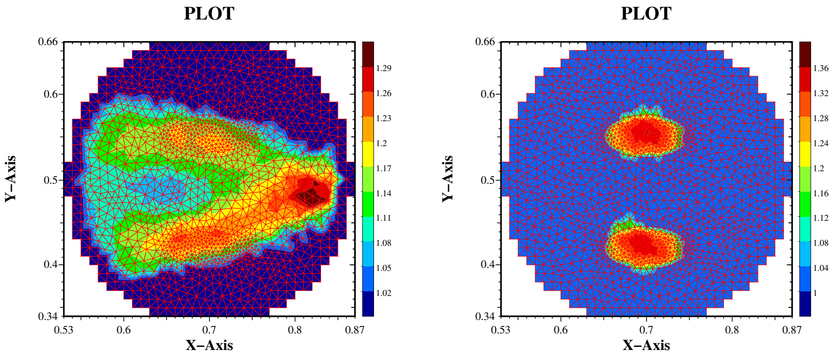


34-th it. 47-th it.

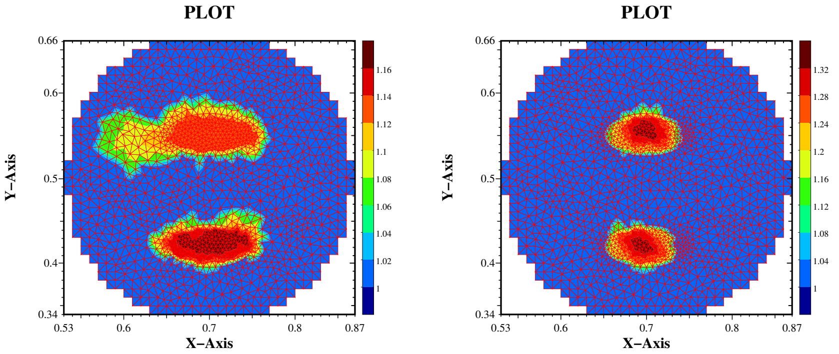


36-th it. 49-th it.

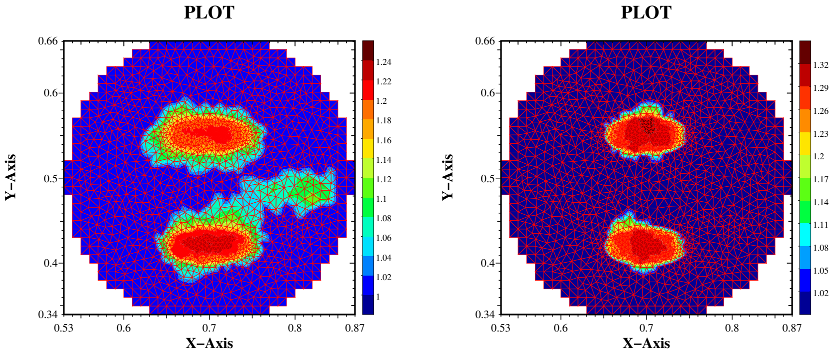
FIGURE 14. Reconstruction of two ellipses on a mesh with 1322 nodes.



1-th it. 30-th it.

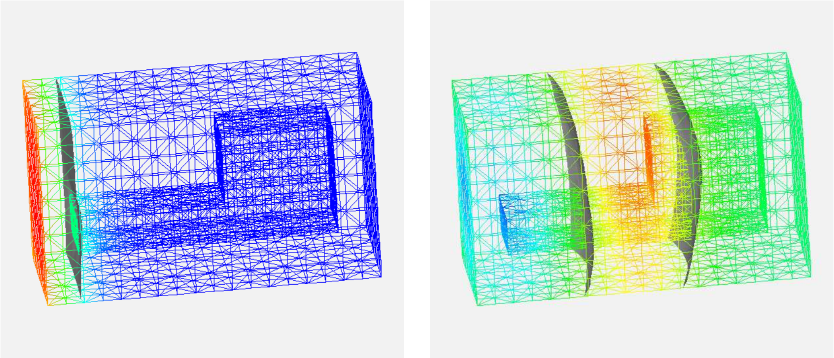


10-th it. 34-th it.

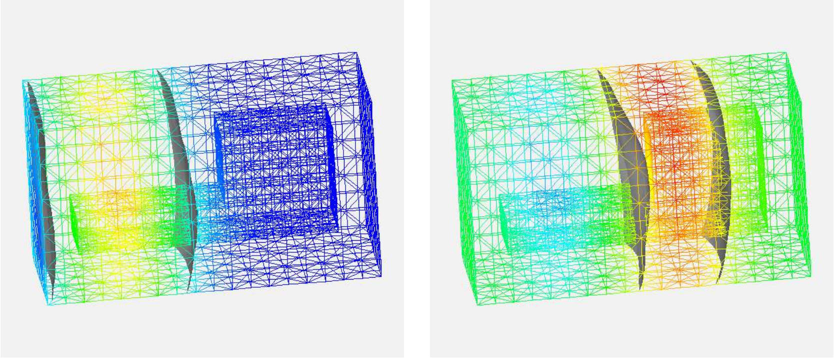


13-th it. 35-th it.

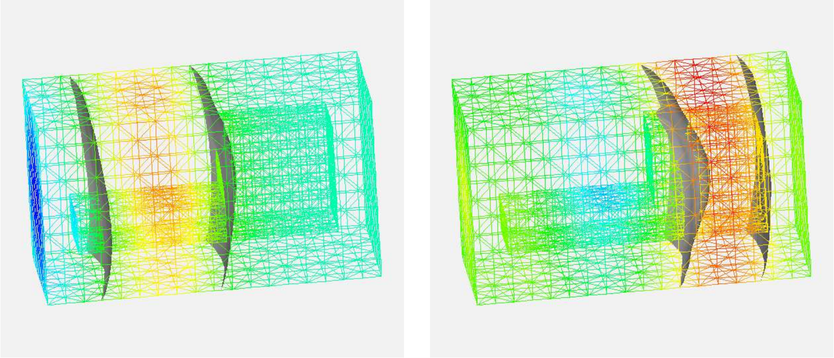
FIGURE 15. Reconstruction of two ellipses on a mesh with 1506 nodes.



time is 0.05 time is 0.1

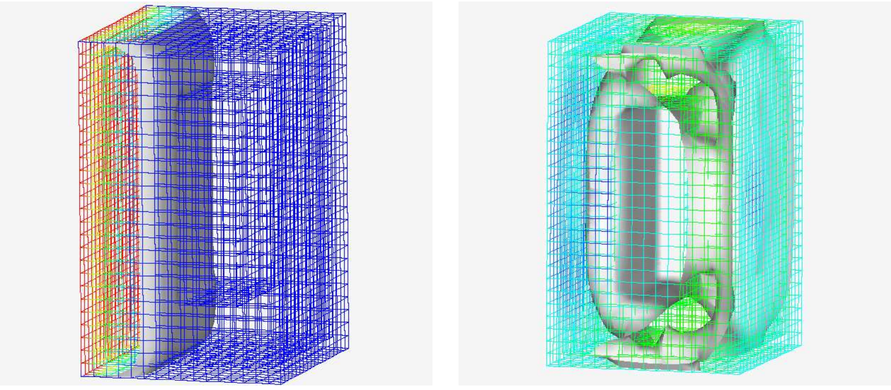


time is 0.08 time is 0.11

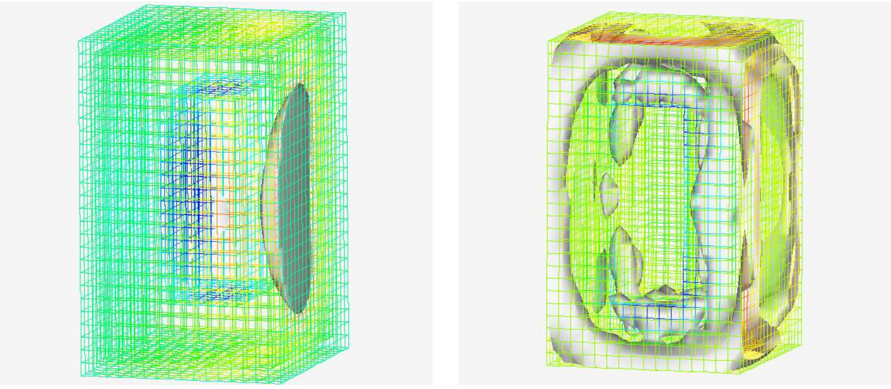


time is 0.09 time is 0.12

Figure 16. Exact solution of the problem (2.1) - (2.3) with one scatterer inside of Ω_{FEM} with c=2 in the scatterer. At the boundary Ω_{FDM} we apply absorbing boundary conditions. We present also the location of the isosurface with value 0.04 at the different time moments, when the plane wave goes from the left to the right side.

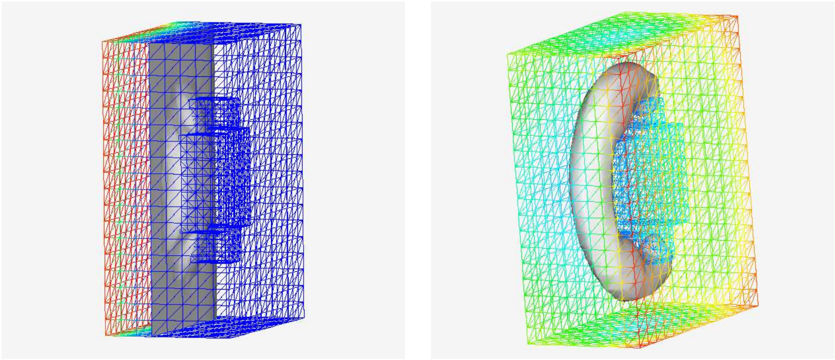


time is 0.045 time is 0.18

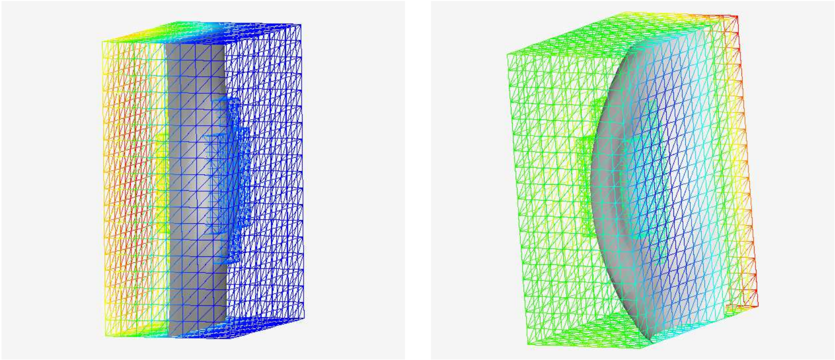


time is 0.105 time is 0.3

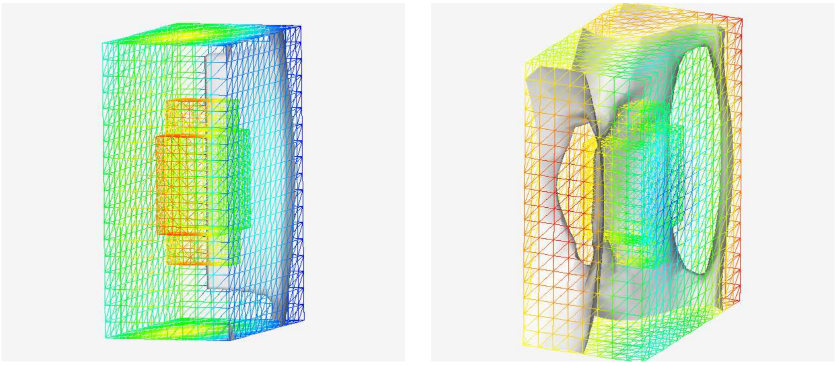
Figure 17. Exact solution of the problem (2.1) - (2.3) with one star-shape scatterer inside of Ω_{FDM} with c=2 in the scatterer. Here we present the solution in Ω_{FDM}. At the boundary Ω_{FDM} we apply absorbing boundary conditions. We present also the location of different isosurfaces.



time is 0.045 time is 0.18

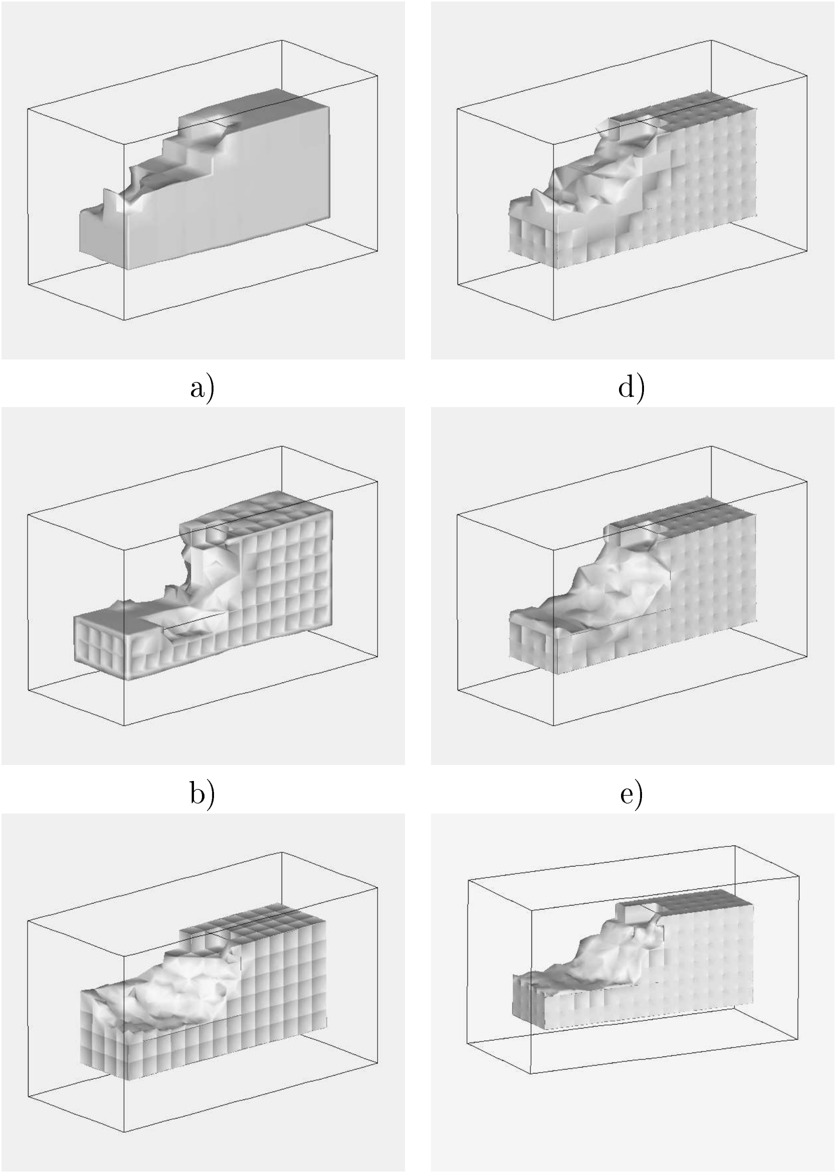


time is 0.105 time is 0.3



time is 0.105 time is 0.3

FIGURE 18. Exact solution of the problem (2.1) - (2.3) with one star-shape scatterer inside of Ω_{FEM} with c=2 in the scatterer. Here we present the solution in Ω_{FEM}. At the boundary Ω_{FDM} we apply absorbing boundary conditions. We present also the location of different isosurfaces.



a) d)

b) e)

c) f)

FIGURE 19. Reconstruction of the one scatterer in Fig. 2-c on adaptively refined meshes: a) on a coarse mesh after 32 iterations, b) on a one time refined mesh after 9 iterations, c) on a two times refined mesh after 5 iterations d),e) on a three times refined mesh after one and three iterations, f) on a four times refined mesh after 5 iterations of the steepest descent algorithm.

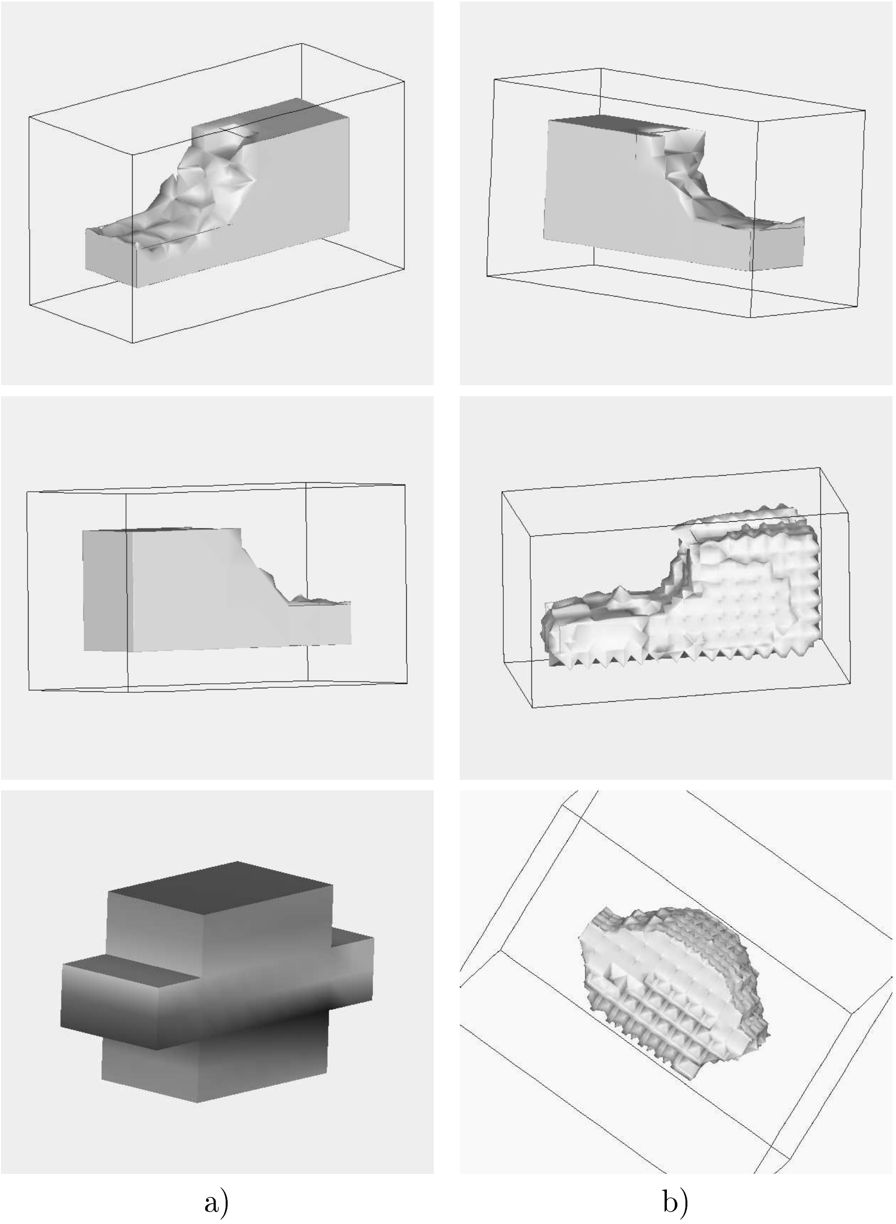
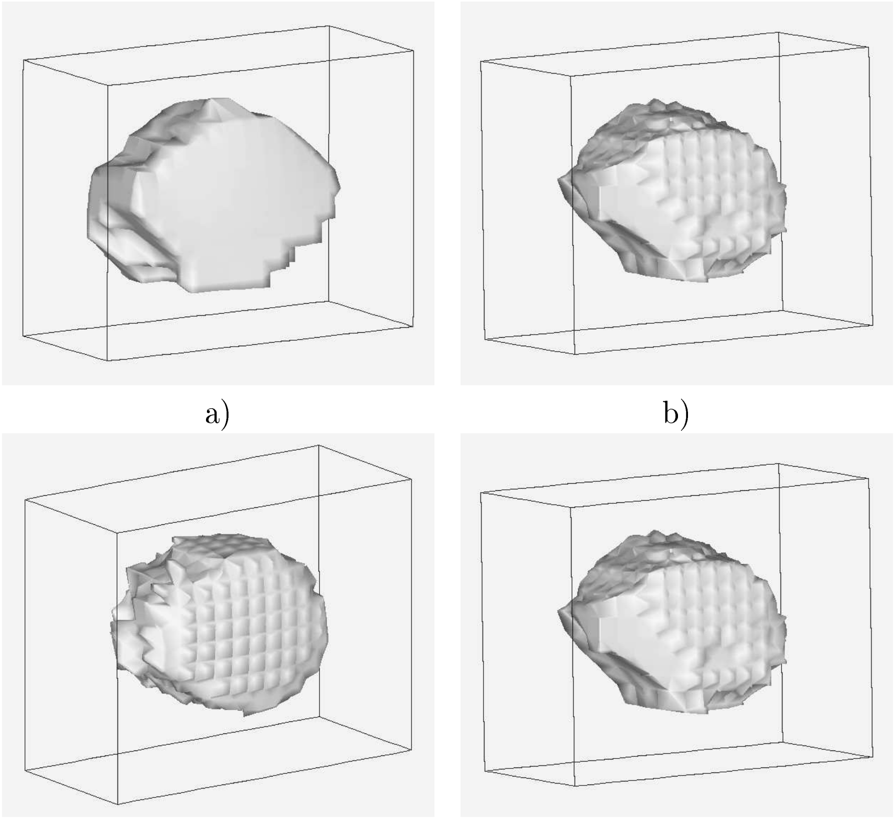


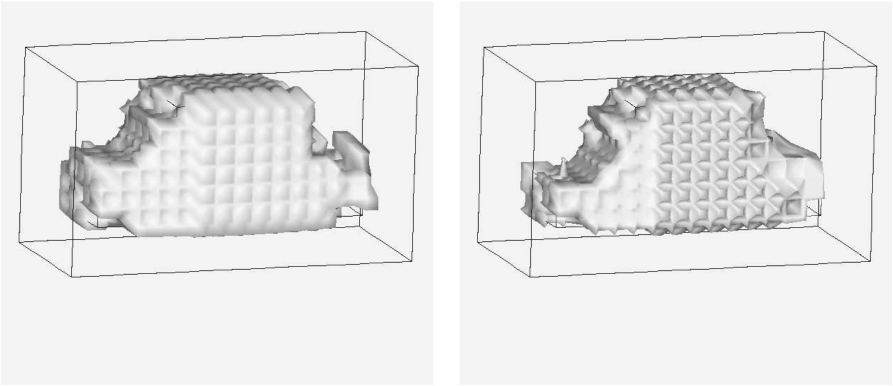
FIGURE 20. Reconstruction of the one scatterer on a five times adaptively refined mesh from different views. The value of the showed isosurface is 1.98. In a) we present another scatterer and in b) its reconstruction on a four times refined mesh.



a) b)

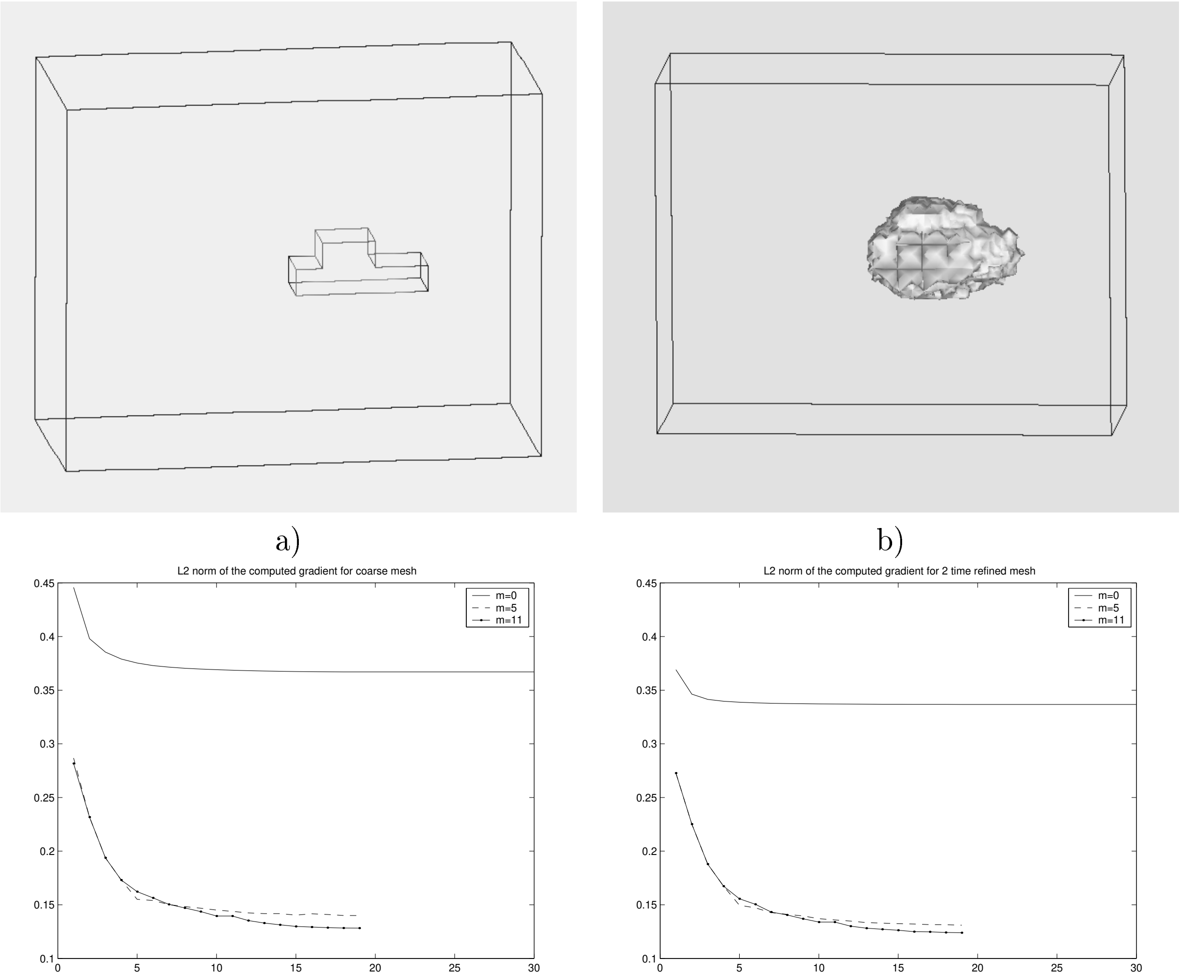
c) d)

Figure 21. Reconstruction of the second in Fig. 20 scatterer on adaptively refined meshes: a) on a coarse mesh after 6 quasi-Newton iterations, b) on a 6 times refined mesh after 4 quasi-Newton iterations, c) on a 5 times refined mesh after 4 quasi-Newton iterations, d) on a 7 times refined mesh after 5 quasi-Newton iterations.



a) b)

Figure 22. Reconstruction of the scatterer from Fig. 2-f on adaptively refined meshes: a) on a 2 times refined mesh after 40 steepest descent method iterations, b) on a 3 times refined mesh after 7 steepest descent iterations.



a) b)

c) d)

FIGURE 23. a),b) Exact and reconstructed scatterer on a five times refined mesh; c)d) L_2 norm of the computed gradient on the coarse, 2 and 4 times refined meshes for different number of the stored corrections.

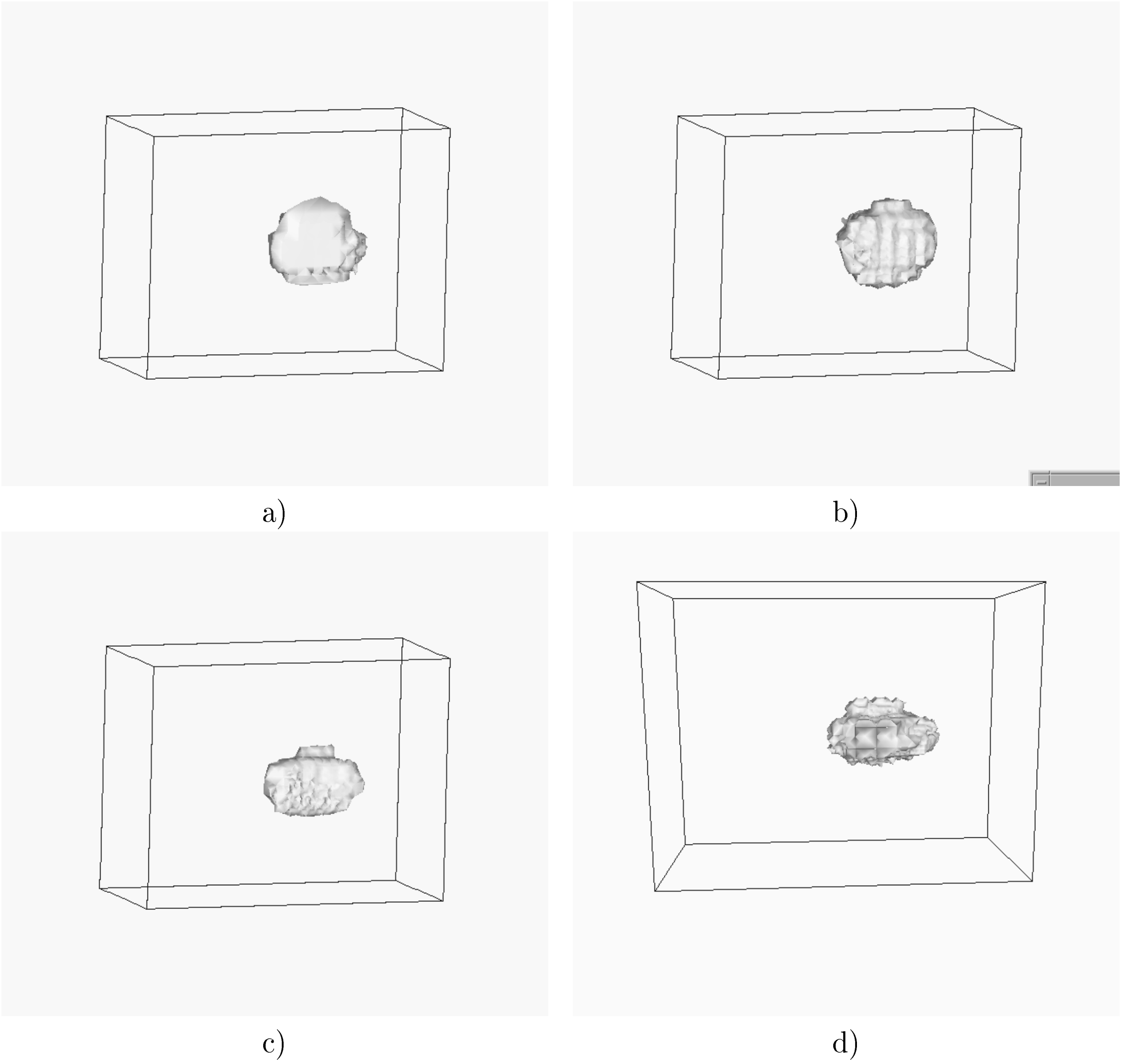
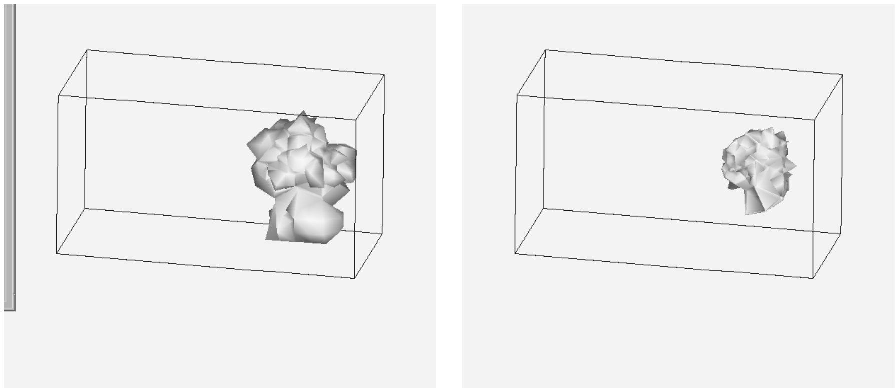
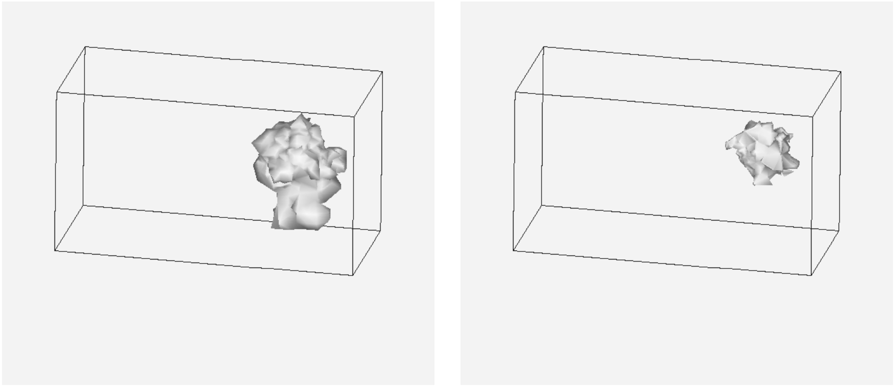


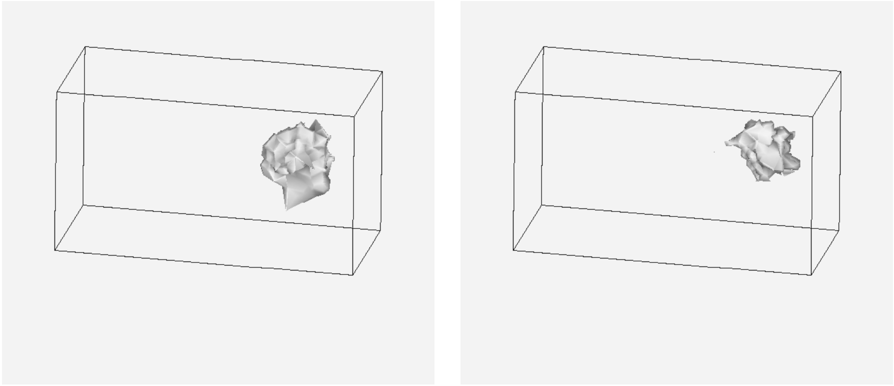
FIGURE 24. Reconstruction of the scatterer in Fig. 23 a) on adaptively refined meshes: a) on a coarse mesh after 4 quasi-Newton iterations (m=7), b), c) on a 5 and 6 times refined mesh after 4 quasi-Newton iterations(m=7), d) after 60 quasi-Newton iterations.



a)after 2 QN it. b) after 5 QN it.



c) after 3 QN it. d) after 6 QN it.



e) after 4 QN it. f) after 7 QN it.

Figure 25. Reconstruction of a cubic scatterer on the 3 times adaptive refined meshes using the quasi-Newton method (QN).

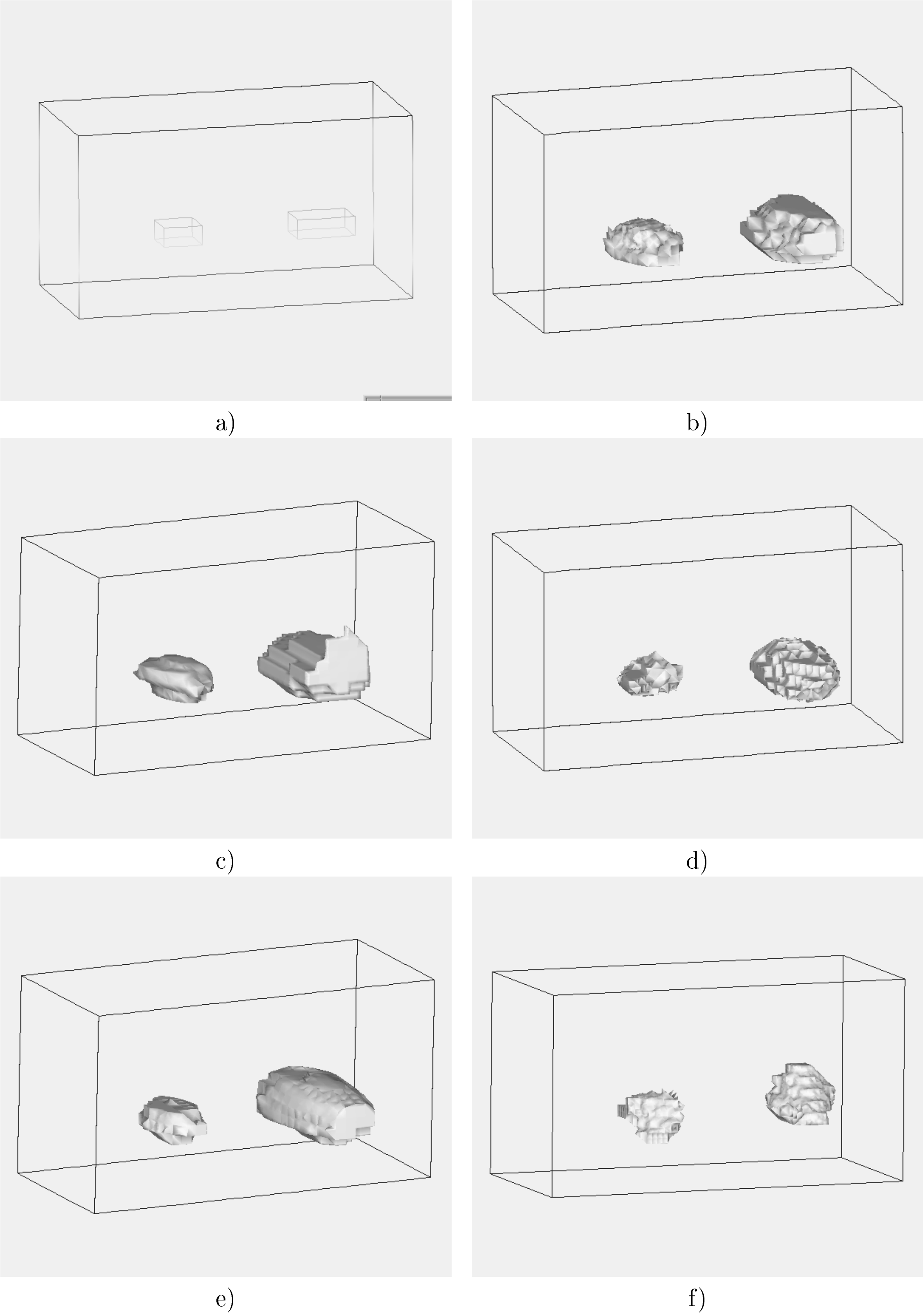
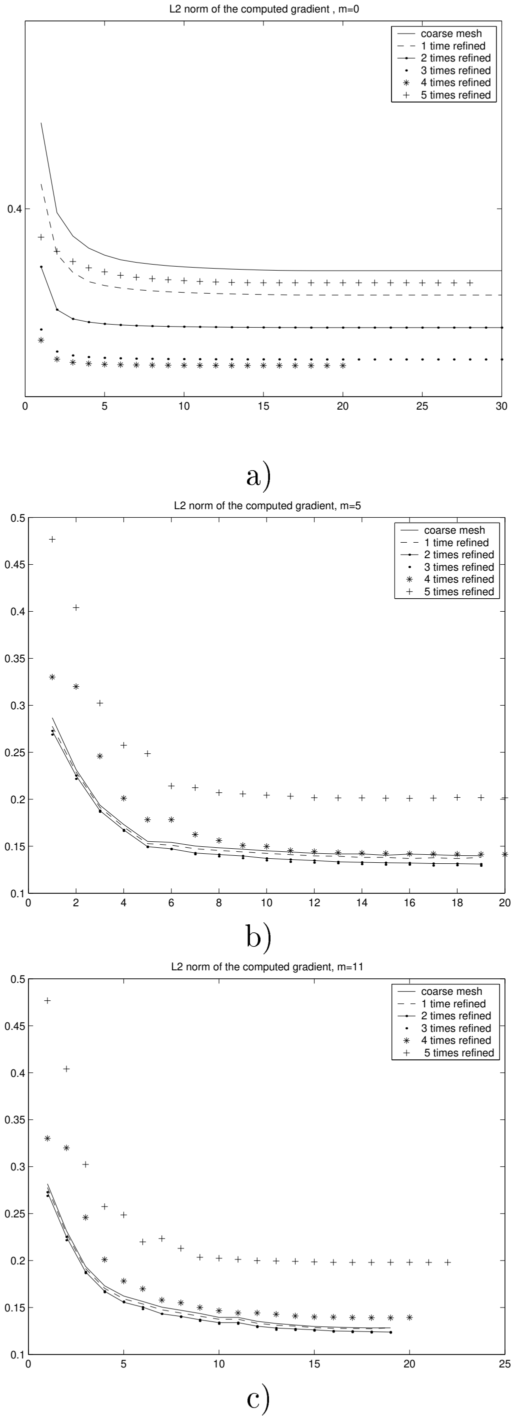


FIGURE 26. Exact and reconstructed scatterers on adaptively refined meshes. We performed computations of the inverse problem on a 4 times adaptive refined meshes. In the optimization algorithm we have used quasi-Newton method with the number of the stored corrections equal to $m = 5$.



L2 norm of the computed gradient , m=0

a)

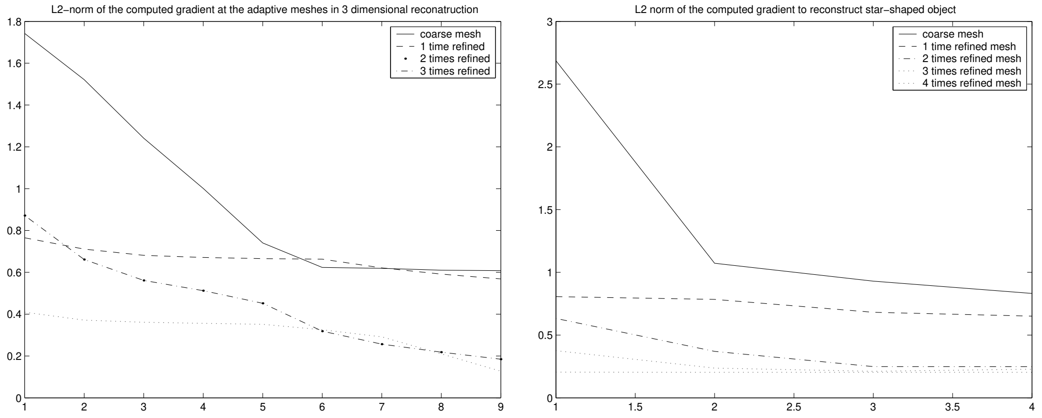
L2 norm of the computed gradient, m=5

b)

L2 norm of the computed gradient, m=11

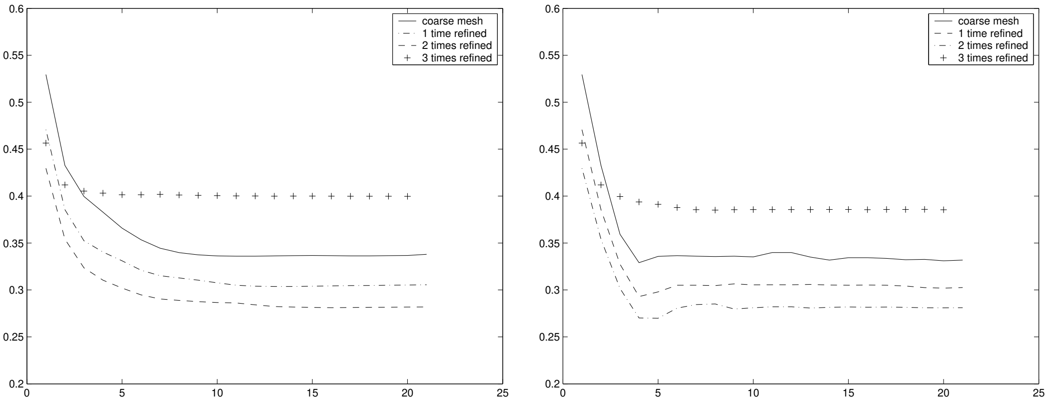
c)

FIGURE 27. L_2 norm of the computed gradient for number of the stored corrections $m = 0, 5$ and 11 on adaptively refined meshes for the scatterer in Fig. 23.



a) b)

FIGURE 28. a) L2-norm of the computed gradient in the quasi-Newton algorithm for different adaptive refined meshes for reconstruction of a cubic scatterer. The number of stored corrections is $m = 5$. b) L2-norm of the computed gradient in the quasi-Newton algorithm for different adaptively refined meshes for reconstruction star-shaped scatterer. The number of stored corrections is $m = 5$.



a) b)

FIGURE 29. L2-norm of the computed gradient in the quasi-Newton algorithm for different adaptively refined meshes for reconstruction of the scatterer in Fig. 23 . Tests performed for reflected waves registered on the same side of the scatterer as the incoming wave with: a) number of stored corrections is $m = 0$, b) number of stored corrections is $m = 11$.

References

[1] E. Acklam, A. Jacobsen and H.P. Langtangen. Optimizing C++ code for explicit finite difference schemes. Oslo Scientific Computing Archive, Report 1998-4.

[2] S. Balay, W. Gropp, L-C. McInnes, B. Smith. PETSc user manual, http://www.mcs.anl.gov/petsc

[3] R. Becker. Adaptive finite elements for optimal control problems. Habilitationsschrift, Heidelberg (2001).

[4] L. Beilina, K. Samuelsson, K. Åhlander. A hybrid method for the wave equation. Technical report, Chalmers Finite Element Center (2001).

[5] J. Cea . A numerical method for the computation of an optimal domain. Lecture notes in computer sciences, Vol. 41,.Springer, New York , (1976).

[6] C. Douglas, J. Hu, M. Kowarschik, U. Rüde and C. Weiss. Cache optimization for structured and unstructured grid multigrid. ETNA volume 10, pp.21-40, (2000)

[7] F. Edelvik, U. Andersson, and Gunnar Ledfelt. Hybrid finite volume - finite difference solver for the Maxwell equations. In AP2000 Millennium Conference on Antennas & Propagation, Davos, Switzerland, (2000).

[8] F. Edelvik and G. Ledfelt. Explicit hybrid time domain solver for the Maxwell equations in 3D. J. Sci. Comput., 2000. to appear.

[9] B. Engquist, A. Majda. Absorbing boundary conditions for the numerical simulation of waves, Math. Comp., Volume 31, number 139, p.629–651, (1977).

[10] K. Eriksson, D. Estep and C. Johnson. Computational Differential Equations. Studentlitteratur, Lund, (1996).

[11] R. Fletcher . Practical methods of optimization. John Wiley and Sons, Ltd, (1986).

[12] P.C. Hansen. Rank-deficient and discrete ill-posed problems. SIAM, Philadelphia, (1998).

[13] T. J. R. Hughes. The Finite Element Method. Prentice Hall, (1987).

[14] J.L. Lions. Optimal control of systems governed by partial differentional equations. Springer-Verlag, New York, (1971).

[15] F.Murat J.Simon. Studies on optimal shape design problems. Lecture Notes in Computer Science 41. Springer-Verlag, Berlin (1976).

[16] F. Natterer, F. Wubbeling . Mathematical Methods in Image reconstruction. SIAM 2001.

[17] F. Natterer. An initial value approach to the inverse Helmholtz problem at fixed frequency. Proceeding of the Conference in Oberwolfach, Springer 1997.

[18] F. Natterer. A finite difference method for the inverse scattering problem at fixed frequency (with F.Wubbeling). Proceedings of the Lapland conference on Inverse Problems, June 1992.

[19] J. Nocedal. Updating quasi-Newton matrices with limited storage. Mathematical of Comp., V. 35,N. 151,pp.773-782.

[20] O. Pironneau. Optimal shape design for elliptic systems. Springer Verlag, (1984).

[21] E. Polak. Computatioanl methods in optimization. Academic, New York, (1971).

Appendix A. Optimization algorithms

A.1. **The steepest descent with optimal step size.** Let $E(z)$ is functional to be minimized, then method of steepest descent is following:

(1) Choose z^o, M, ε. For $m = 0, ..., M$ do
(2) Compute $h^m = - \bigtriangledown E(z^m)$.
(3) Compute $\rho^m = \arg\ \min\ E(z^m + \rho h^m)\ \ \forall \rho \geq 0$.
(4) Set $z^{m+1} = z^m + \rho^m h^m$.
(5) If $\| \bigtriangledown E(z^m)\| \leq \varepsilon$, stop.

A.2. **Conjugate gradient.** Algorithm:

(1) Choose z^o, M, ε. Set $h^{-1} = 0, g^{-1} = - \bigtriangledown E(z^0)$. For $m = 0, ..., M$ do
(2)

$$g^m = - \bigtriangledown E(z^m), \tag{A.1}$$

$$\gamma^m = \frac{< g^m - g^{m-1}, g^m >}{||g^{m-1}||^2}, \tag{A.2}$$

$$h^m = g^m + \gamma^m h^{m-1}. \tag{A.3}$$

(3) Compute $\rho = \arg\ \min_{\rho > 0} E(z^m + \rho h^m)$.
(4) Set $z^{m+1} = z^m + \rho h^m$.
(5) If $||g^m|| < \varepsilon$, stop.

A.3. **Newtons method for the wave equation.** We here present Newtons method for the problem (2.1)-(2.3). Let us write this problem in the operator form

$$A(u) = f, \tag{A.4}$$

where $u = (p, \lambda, c)$, $A(u)$ is the differentional operator

$$A(u) = \left\{ \begin{array}{l} \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \triangle p \\ \frac{1}{c^2}\frac{\partial^2 \lambda}{\partial t^2} - \triangle \lambda - p \\ \int_0^T - \frac{2}{c^3}\frac{\partial p}{\partial t}\frac{\partial \lambda}{\partial t} dt \end{array} \right.$$

and $f = (f, -\tilde{p}, 0)$.

Newtons method takes the form:

$$u^{n+1} = u^n - \alpha A'(u^n)^{-1} A(u^n), \tag{A.5}$$

where

$$A'(u) = \left(\begin{array}{ccc} \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \triangle & 0 & -\frac{2}{c^3}\frac{\partial^2 p}{\partial t^2} \\ -I & \frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \triangle & -\frac{2}{c^3}\frac{\partial^2 \lambda}{\partial t^2} \\ -\int_0^T \frac{2}{c^3}\frac{\partial \lambda}{\partial t}\frac{\partial}{\partial t} dt & -\int_0^T \frac{2}{c^3}\frac{\partial p}{\partial t}\frac{\partial}{\partial t} dt & \int_0^T \frac{6}{c^4}\frac{\partial p}{\partial t}\frac{\partial \lambda}{\partial t} dt. \end{array} \right)$$

and involves the solution of the equation $A'(u^n)\ du = A(u^n)$.

A.4. **One-dimensional minimization.** Let

$$E(\rho) = E(z^m + \rho h^m) - E(z^m) \tag{A.6}$$

and

$$\frac{dE}{d\rho}(0) =< \bigtriangledown E(z^m), h^m >= -\| \bigtriangledown E(z^m)\|^2. \tag{A.7}$$

The one-dimensional minimization algorithm is following:

0. Choose ρ and set $d = \rho > 0$.
1. Set $\rho_2 = \rho + d, \rho_1 = \rho - d$.
2. If $E(\rho) > E(\rho_2)$, then set $\rho = \rho + d$ and go to step 1. If $E(\rho_1) < E(\rho)$, then set $\rho = \rho - d/2, d = d/2$ and go to step 1; else go to step 3.
3. Compute the minimum of the parabola running through the three points $\rho_1, E(\rho_1), \rho, E(\rho), \rho_2, E(\rho_2)$, i.e., set

$$\rho = \rho + \frac{1}{2}(\rho - \rho_1)\frac{E(\rho_1) - E(\rho_2)}{E(\rho_2) + E(\rho_1) - 2E(\rho)}. \tag{A.8}$$

4. If more precision is required go back to step 1, $d = d/2$.

Chalmers Finite Element Center Preprints

2000–01	*Adaptive finite element methods for the unsteady Maxwell's equations* Johan Hoffman
2000–02	*A multi-adaptive ODE-solver* Anders Logg
2000–03	*Multi-adaptive error control for ODEs* Anders Logg
2000–04	*Dynamic computational subgrid modeling* (Licentiate Thesis) Johan Hoffman
2000–05	*Least-squares finite element methods for electromagnetic applications* (Licentiate Thesis) Rickard Bergström
2000–06	*Discontinuous galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method* Peter Hansbo and Mats G. Larson
2000–07	*A discountinuous Galerkin method for the plate equation* Peter Hansbo and Mats G. Larson
2000–08	*Conservation properties for the continuous and discontinuous Galerkin methods* Mats G. Larson and A. Jonas Niklasson
2000–09	*Discontinuous Galerkin and the Crouzeix-Raviart element: application to elasticity* Peter Hansbo and Mats G. Larson
2000–10	*Pointwise a posteriori error analysis for an adaptive penalty finite element method for the obstacle problem* Donald A. French, Stig Larson and Ricardo H. Nochetto
2000–11	*Global and localised a posteriori error analysis in the maximum norm for finite element approximations of a convection-diffusion Problem* Mats Boman
2000–12	*A posteriori error analysis in the maximum norm for a penalty finite element method for the time-dependent obstacle problem* Mats Boman
2000–13	*A posteriori error analysis in the maximum norm for finite element approximations of a time-dependent convection-diffusion problem* Mats Boman
2001–01	*A simple nonconforming bilinear element for the elasticity problem* Peter Hansbo and Mats G. Larson
2001–02	*The $\mathcal{LL}^*$ finite element method and multigrid for the magnetostatic problem* Rickard Bergström, Mats G. Larson, and Klas Samuelsson
2001–03	*The Fokker-Planck operator as an asymptotic limit in anisotropic media* Mohammad Asadzadeh
2001–04	*A posteriori error estimation of functionals in elliptic problems: experiments* Mats G. Larson and A. Jonas Niklasson
2001–05	*A note on energy conservation for Hamiltonian systems using continuous time finite elements* Peter Hansbo

2001–06	*Stationary level set method for modelling sharp interfaces in groundwater flow* Nahidh Sharif and Nils-Erik Wiberg
2001–07	*Integration methods for the calculation of the magnetostatic field due to coils* Marzia Fontana
2001–08	*Adaptive finite element computation of 3D magnetostatic problems in potential formulation* Marzia Fontana
2001–09	*Multi-adaptive galerkin methods for ODEs I: theory & algorithms* Anders Logg
2001–10	*Multi-adaptive galerkin methods for ODEs II: applications* Anders Logg
2001–11	*Energy norm a posteriori error estimation for discontinuous Galerkin methods* Roland Becker, Peter Hansbo, and Mats G. Larson
2001–12	*Analysis of a family of discontinuous Galerkin methods for elliptic problems: the one dimensional case* Mats G. Larson and A. Jonas Niklasson
2001–13	*Analysis of a nonsymmetric discontinuous Galerkin method for elliptic problems: stability and energy error estimates* Mats G. Larson and A. Jonas Niklasson
2001–14	*A hybrid method for the wave equation* Larisa Beilina, Klas Samuelsson and Krister Åhlander
2001–15	*A finite element method for domain decomposition with non-matching grids* Roland Becker, Peter Hansbo and Rolf Stenberg
2001–16	*Application of stable FEM-FDTD hybrid to scattering problems* Thomas Rylander and Anders Bondeson
2001–17	*Eddy current computations using adaptive grids and edge elements* Y. Q. Liu, A. Bondeson, R. Bergström, C. Johnson, M. G. Larson, and K. Samuelsson
2001–18	*Adaptive finite element methods for incompressible fluid flow* Johan Hoffman and Claes Johnson
2001–19	*Dynamic subgrid modeling for time dependent convection–diffusion–reaction equations with fractal solutions* Johan Hoffman
2001–20	*Topics in adaptive computational methods for differential equations* Claes Johnson, Johan Hoffman and Anders Logg
2001–21	*An unfitted finite element method for elliptic interface problems* Anita Hansbo and Peter Hansbo
2001–22	*A P^2–continuous, P^1–discontinuous finite element method for the Mindlin-Reissner plate model* Peter Hansbo and Mats G. Larson
2002–01	*Approximation of time derivatives for parabolic equations in Banach space: constant time steps* Yubin Yan
2002–02	*Approximation of time derivatives for parabolic equations in Banach space: variable time steps* Yubin Yan

2002–03	*Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations* Thomas Rylander and Anders Bondeson
2002–04	*A computational study of transition to turbulence in shear flow* Johan Hoffman and Claes Johnson
2002–05	*Adaptive hybrid FEM/FDM methods for inverse scattering problems* Larisa Beilina

These preprints can be obtained from

`www.phi.chalmers.se/preprints`