A globally convergent numerical method and the adaptivity technique for a hyperbolic coefficient inverse problem. Part II: numerical studies.

Larisa Beilina

Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University, Gothenburg, SE-421196, Sweden

E-mail: `larisa.beilina@chalmers.se`

Michael V. Klibanov

Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

E-mail: `mklibanv@uncc.edu`

Abstract. In this second part of the paper numerical experiments are presented. They show that a globally convergent numerical method provides a good starting point for the finite element adaptive method (adaptivity). This leads to a natural two-stage numerical procedure, which synthesizes both these methods.

1. Introduction

In this second part of the paper numerical experiments are presented. They demonstrate the performance of the globally convergent numerical method in combination with the Finite Element Adaptive technique.

This is the second part of our work, see [4] for the first part, where analytical results for our numerical method are presented. In this part we present numerical experiments referring for some analytical details to the first part [4]. In our numerical experiments we image a medium with small inclusions in it, although we do not assume a priori knowledge of such a structure. We refer to [1] and references cited there for another approach to imaging of small inclusions. There are also some other numerical methods for multidimensional CIPs, which do not use a good first guess for the solution. While the current paper works with a single measurement event, they work for some CIPs with the data resulting from multiple measurements [7, 8, 9, 12, 13, 14]. These publications were discussed in [2].

We now pose forward and inverse problems, for the convenience of the reader, see details in [4]. As the forward problem, we consider the Cauchy problem for a hyperbolic PDE,

$$c\left(x\right)u_{tt}=\Delta u\text{ in }\mathbb{R}^{3}\times\left(0,\infty\right), \tag{1}$$

$$u\left(x,0\right)=0,u_{t}\left(x,0\right)=\delta\left(x-x_{0}\right). \tag{2}$$

Since equation (1) governs a wide range of applications, including e.g., propagation of acoustic and electromagnetic waves, then the same is true for the CIP we consider. Let d_1 and d_2 be two positive constants and $\Omega\subset\mathbb{R}^3$ be a convex bounded domain with the boundary $\partial\Omega\in C^3$. We assume that the coefficient $c\left(x\right)$ of equation (1) is such that

$$c\left(x\right)\in\left[d_{1},2d_{2}\right],d_{1}<d_{2},c\left(x\right)=2d_{1}\text{ for }x\in\mathbb{R}^{3}\diagdown\Omega, \tag{3}$$

$$c\left(x\right)\in C^{2}\left(\mathbb{R}^{3}\right). \tag{4}$$

We consider the following

Inverse Problem. Suppose that the coefficient $c\left(x\right)$ satisfies (3) and (4), where the positive numbers d_1 and d_2 are given. Assume that the function $c\left(x\right)$ is unknown in the domain Ω. Determine the function $c\left(x\right)$ for $x\in\Omega$, assuming that the following function $g\left(x,t\right)$ is known for a single source position $x_0\notin\overline{\Omega}$

$$u\left(x,t\right)=g\left(x,t\right),\forall\left(x,t\right)\in\partial\Omega\times\left(0,\infty\right). \tag{5}$$

The paper is organized as follows. In section 2 we formulate an adaptive algorithm which uses a modified framework for the adaptivity technique presented in the first part [4]. In section 3 numerical experiments are presented. A preprint with this publication is available online [5].

2. The adaptive algorithm

In this section we present our adaptive algorithm, which is derived from Theorems 3.1-3.4 and Remark 3.1 of [4]. Ignoring third lines in estimates (33), (34) and (43), (48) (Remark 3.2) of [4], we conclude from (24) and (25) of [4] that we should find an approximate solution of the following problem on each mesh

$$\gamma \left(c_h - c_0 \right) - \int\limits_0^T \left[\partial_t u_h \left(c_h \right) \cdot \partial_t \varphi_h \left(c_h \right) \right] (x,t) \, dt = 0, c_h \left(x \right) \in V_h. \tag{6}$$

We solve the problem (6) via the quasi-Newton method given in [15]. On the first step of the adaptivity we take the same mesh as one which was used for the globally convergent method. The first guess $c_0 := c_{glob} \left(x \right)$ is also taken the one, which was obtained on the globally convergent stage.

For each mesh we compute iteratively the sequence $\left\{ c_h^m \right\}$, $m = 1, ...$ of approximations of c_h as

$$c_h^{m+1}(x) = c_h^m(x) - \alpha H^m g^m(x), \tag{7}$$

where α is the step length computed via the line-search algorithm [16]. Here, H^m is an approximate inverse of the Hessian of the Lagrangian. The approximate inverse of the Hessian is computed by the usual BFGS update formula:

$$\begin{aligned} H^{m+1} &= (I - d^m s^m y^{mT}) H^m (I - d^m y^m s^{mT}) + \rho s^m s^{mT}, m = 1, ..., \\ y^m &= g^{m+1} - g^m, d^m = 1/(y^{mT} s^m), m = 1, ... \end{aligned} \tag{8}$$

In (8) corrections s^m are defined as $s^m = c_h^{m+1} - c_h^m$. In our computations we have used a special BFGS update formula with limited storage for the Hessian [15] where we store a finite number $m_1 = m - 1$ of corrections for the computed gradients and parameters in (8). If $m = 0$, then the quasi-Newton method is the usual gradient method with $H^0 = I$. Let $u_h \left(c_h^m \right) \in W_h^u$ and $\varphi_h \left(c_h^m \right) \in W_h^u$ be FEM solutions of state (9) and adjoint (20) problems of [4], and $c_h^1 := c_0$. The gradient $g^m(x)$ in (7) is computed by (see (6) in[4])

$$g^m(x) = \gamma(c_h^m - c_0) - \int_0^T \left[\partial_t u_h\left(c_h^m\right) \cdot \partial_t \varphi_h\left(c_h^m\right)\right](x,t)\, dt. \qquad (9)$$

Although above a posteriori error estimates are approximate ones, our computational experience shows that they are sufficient. Thus, by Remark 3.1 of [4], we use the following adaptivity algorithm in our computations:

Step 0. Choose an initial mesh K_h and an initial time partition J_0 of the time interval $(0, T]$. Start with the initial approximation $c_0 = c_{glob}$, which was computed in the globally convergent algorithm, and compute the sequence of c_h^m in the following steps:

Step 1. Compute solutions $u_h\left(c_h^m\right) \in W_h^u$ and $\varphi_h\left(c_h^m\right) \in W_h^u$ of state (9) and adjoint (20) problems of [4], respectively on K_h and J_k.

Step 2. Update the coefficient $c := c_h^{m+1}$ on K_h and J_k using (7).

Step 3. Stop computing c_h on the above quasi-Newton method if either the norm of the gradient g^m of the Lagrangian with respect to the coefficient in (9) is $||g^m||_{L_2(\Omega)} < \theta$ or norms $||g^m||_{L_2(\Omega)}$ are stabilized. Otherwise set $m := m + 1$ and go to step 1. Here, θ is the tolerance in quasi-Newton updates. In our computations we took $\theta = 10^{-5}$.

Step 4. Compute the function $B_h(x)$ in (31) of [4]. Refine the mesh at all points where

$$B_h(x) > \beta \max_{\overline{\Omega}} B_h(x), \qquad (10)$$

where the tolerance number $\beta \in (0, 1)$ is chosen by the user, see section 3 for details.

Step 5. Construct a new mesh K_h and a new time partition J_k. On J_k the new time step τ should be chosen in such a way that the CFL condition is satisfied. Interpolate the initial approximation $c_0 = c_{glob}$, which was computed in the globally convergent algorithm, from the previous mesh to the new mesh. Return to step 1 and perform all the steps of the optimization algorithm on the new mesh.

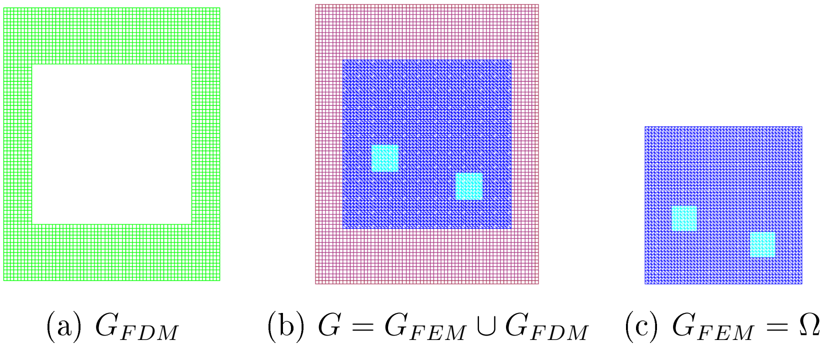

(a) G_{FDM} (b) $G = G_{FEM} \cup G_{FDM}$ (c) $G_{FEM} = \Omega$

Figure 1. The hybrid mesh (b) is a combinations of a structured mesh (a), where FDM is applied, and a mesh (c), where we use FEM, with a thin overlapping of structured elements. The solution of the inverse problem is computed in the square Ω and $c(x) = 1$ for $x \in G \diagdown \Omega$.

3. Numerical Studies

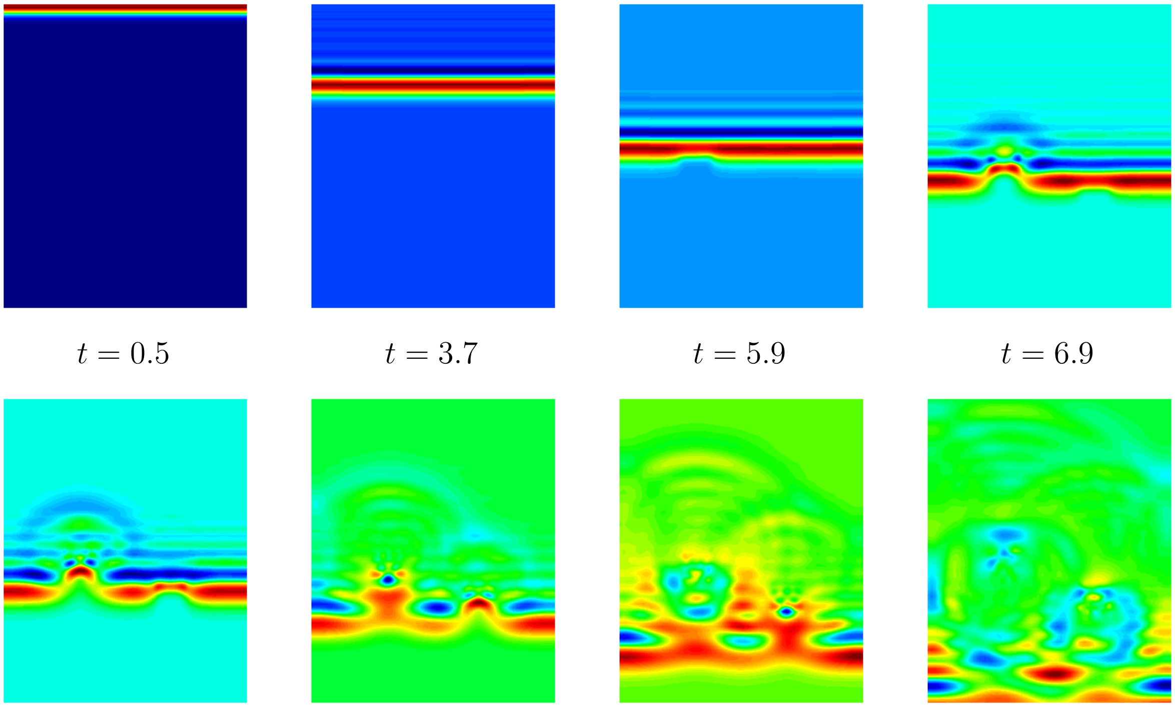
3.1. Computations of the forward problem

In this paper we work with the computationally simulated data. That is, the data are generated by computing the forward problem (12) with the given function $c(x)$. To solve the forward problem, we use the hybrid FEM/FDM method described in [6]. The computational domain in all our tests $G = G_{FEM} \cup G_{FDM}$ is set as $G = [-4.0, 4.0] \times [-5.0, 5.0]$. This domain is split into a finite element domain $G_{FEM} := \Omega = [-3.0, 3.0] \times [-3.0, 3.0]$ and a surrounding domain G_{FDM} with a structured mesh, see Figure 1. The space mesh in Ω consists of triangles and in G_{FDM} - of squares with the mesh size $\tilde{h} = 0.125$ in the overlapping regions. At the top and bottom boundaries of G we use first-order absorbing boundary conditions [10] which are exact in this particular case since the plane wave is initialized in normal direction into G in all our tests. At the lateral boundaries, mirror boundary conditions allow us to assume an infinite space domain in the lateral direction.

The forward problem is computed in the domain $G \subset \mathbb{R}^2$ (Figure 1). The coefficient $c(x)$ is unknown only in the domain $\Omega \subset G$ and

$$c(x) = 1 \text{ in } G \diagdown \Omega. \tag{11}$$

The trace of the solution of the forward problem is recorded at the boundary

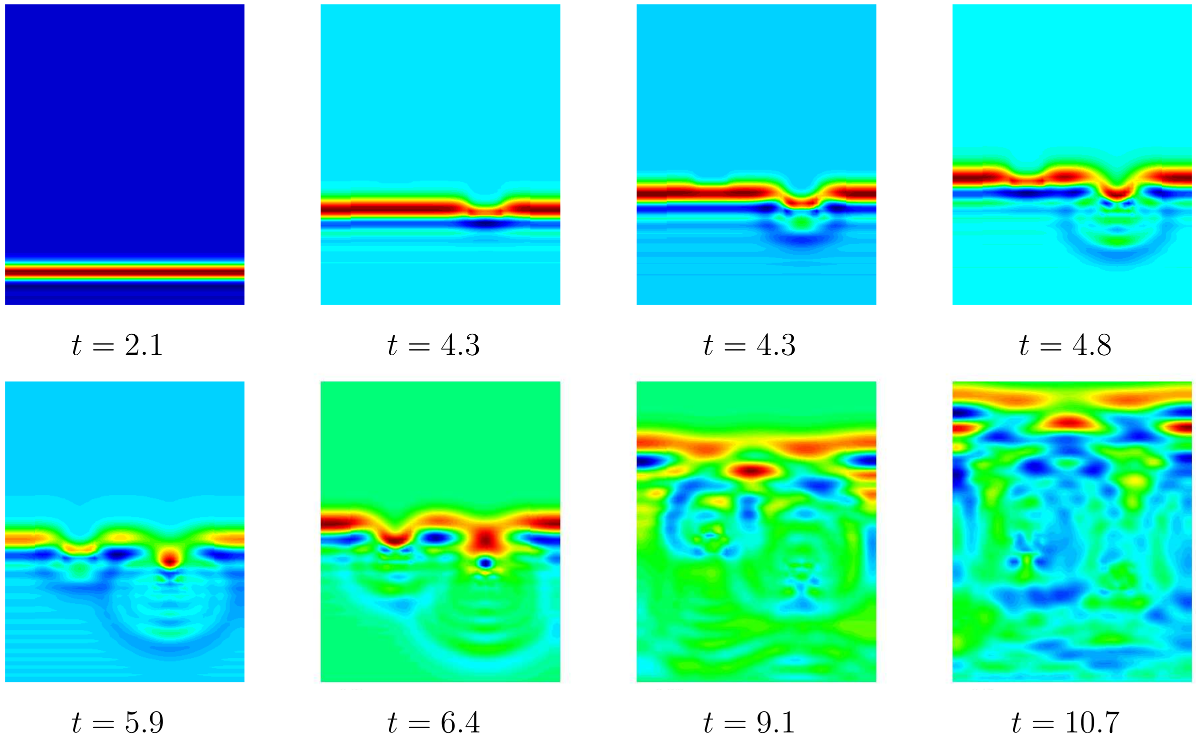


$t = 0.5$ $t = 3.7$ $t = 5.9$ $t = 6.9$

$t = 7.5$ $t = 8.5$ $t = 9.6$ $t = 11.2$

Figure 2. Test 1: Isosurfaces of the simulated exact solution to the forward problem (12) at different times with a plane wave initialized at the top boundary.

$\partial\Omega$. Next, the coefficient $c(x)$ is "forgotten", and our goal is to reconstruct this coefficient for $x \in \Omega$ from the data $\varphi\left(x,s\right)$. The boundary of the domain G is $\partial G = \partial G_1 \cup \partial G_2 \cup \partial G_3$. Here, ∂G_1 and ∂G_2 are respectively top and bottom sides of the largest domain of Figure 1 and ∂G_3 is the union of left and right sides of this



$t = 2.1$ $t = 4.3$ $t = 4.3$ $t = 4.8$

$t = 5.9$ $t = 6.4$ $t = 9.1$ $t = 10.7$

Figure 3. Test 2: Isosurfaces of the simulated exact solution to the forward problem (12) with a plane wave initialized at the bottom boundary.

domain. In our first test the forward problem is

$$\begin{aligned} c\,(x)\, u_{tt} - \triangle u &= 0, \quad \text{in } G \times (0,T), \\ u(x,0) = 0,\ u_t(x,0) &= 0,\ \text{in } G, \\ \partial_n u\big|_{\partial G_1} &= f\,(t)\,,\ \text{on } \partial G_1 \times (0,t_1], \\ \partial_n u\big|_{\partial G_1} &= -\partial_t u,\ \text{on } \partial G_1 \times (t_1,T), \\ \partial_n u\big|_{\partial G_2} &= -\partial_t u,\ \text{on } \partial G_2 \times (0,T), \\ \partial_n u\big|_{\partial G_3} &= 0,\ \text{on } \partial G_3 \times (0,T), \end{aligned} \tag{12}$$

where T is the final time and $f(t)$ is the plane wave defined as

$$f(t) = \frac{(\sin{(\overline{s}t{-}\pi/2)} + 1)}{10},\ 0 \le t \le t_1 := \frac{2\pi}{\overline{s}}, T = 17.8t_1.$$

Thus, the plane wave is initialized at the top boundary ∂G_1 and propagates into G for $t \in (0, t_1]$. First order absorbing boundary conditions [10] are used on $\partial G_1 \times (t_1, T]$ and $\partial G_2 \times (0, T]$, and the Neumann boundary condition is used on the bottom boundary ∂G_3. In the second test we consider the case when the plane wave is initialized at the bottom boundary and use the Neumann boundary condition at the top boundary. In the integral (7) of [4] of the Laplace transform we integrate for $t \in (0, T)$.

We now list main discrepancies between our theory and computations. Such discrepancies quite often occur in computations of ill-posed problems and seem to be inevitable. It is well known that computational results are usually less pessimistic than the theory. At the same time, theory usually provides a good guidance for computations. The first discrepancy is that we use the initializing plane wave in (12) instead of the point source in (2). This is because the point source was used only to better justify the certain asymptotic behaviour, see [2] for our numerical verification of this behaviour for the case of the above plane wave. The second discrepancy is that domains G and Ω depicted on Figure 1 have piecewise smooth rather than smooth boundaries. In principle, this might lead to singularities in solutions of both the hyperbolic equation (12) and elliptic equations for functions q_n in the above globally convergent numerical method. However, we have not observed such singularities in our computations. The third discrepancy is that due to some conveniences of our numerical implementation, we use piecewise constant functions c_h in our computations rather than those satisfying (10) in [4]. The next discrepancy is that we do not use the function $\zeta_{\varepsilon_1}(t)$ in (8) and (20) of [4] because we have observed in computations of the forward problem (12) that $(u\mid_{S_T} - g)(x, T) \approx 0$, which eliminates the need to use this function. Finally a discrepancy regarding smoothness requirements was outlined in subsection 3.1 of [4]. Regardless on these discrepancies, we have observed a good correspondence between our analytical and numerical results, see below.

3.2. Results of reconstruction by the globally convergent algorithm

We have performed numerical experiments to reconstruct the medium, which is homogeneous with $c(x) = 1$ except of two small squares, where $c(x) = 4$, see Figure 1-c). However, we have not assumed *a priori* knowledge of neither the structure of this medium nor of the background constant $c(x) = 1$ outside of those two small squares, although, following the Tikhonov concept (as mentioned in section 2), we have assumed the knowledge of the constant $d_1 = 1/2$, see (3) and (11). Because of this, the starting value for the tail $V_{1,1}\left(x,\overline{s}\right)$ was computed via solving the forward problem (12) for $c \equiv 1$. Let $w_{c\equiv 1}\left(x,\overline{s}\right)$ be the corresponding function $w\left(x,s\right)$ at $s=\overline{s}$. Then, we took $V_{1,1}\left(x,\overline{s}\right) = \overline{s}^{-2}\ln w_{c\equiv 1}\left(x,\overline{s}\right)$, see [2] for the details.

It was found in [2] that the interval $[\underline{s},\overline{s}] = [6.7, 7.45]$ is the optimal one for domains G, Ω specified above, and so we have used it in our numerical studies. We have chosen the step size with respect to the pseudo frequency $h = 0.05$. Hence, $N = 15$ in our case. We have chosen two sequences of regularization parameters $\lambda := \lambda_n$ and $\varepsilon = \varepsilon_n$ for $n = 1, ..., \overline{N}$, which are the same as ones in [2]. So, values of these parameters as well as the value of the regularization coefficient γ in the adaptivity were:

$$\lambda_n = 20, n = 1,2; \lambda_n = 200, n = 3,4,5; \lambda_n = 2000, n \geq 6;$$
$$\varepsilon_n = 0, n = 1,2; \varepsilon_n = 0.001, n = 3,4,5; \varepsilon_n = 0.01, n = 6,7; \varepsilon_n = 0.1 \text{ for } n \geq 8; \gamma = 0.01.$$

Once the function q_n is calculated, we update the function $c := c_n$, see subsection 7.3 of [2] for some numerical details. The resulting computed function is $c\left(x\right) := c_{\overline{N}}(x)$. Comparing with [2], in the current work we choose a completely different stopping rule. In calculating iterations with respect to the nonlinear term (section 4 and Theorem 5.1), we consider norms F_n^k,

$$F_n^k = ||q_{n,1}^k|_{\partial\Omega} - \overline{\psi}_n||_{L_2(\partial\Omega)}. \tag{13}$$

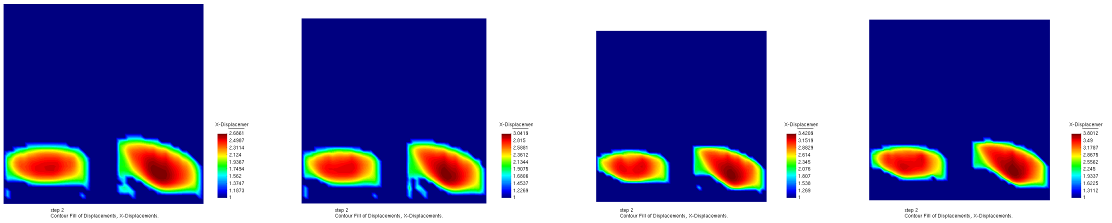
We stop our iterations with respect to nonlinear terms when

$$\text{either } F_n^k \geq F_n^{k-1} \text{ or } F_n^k \leq \nu,$$

where $\nu = 0.001$ is a small tolerance number of our choice. In other words, we stop iterations, when either norms F_n^k start to grow or are too small. Next, we iterate with respect to the tails (section 4) and use the same stopping criterion. Namely, we stop our iterations with respect to tails when either

$$\text{either } F_{n,i} \geq F_{n,i-1} \text{ or } F_{n,i} \leq \nu, \tag{14}$$

where $F_{n,i} = ||q_{n,i}|_{\partial\Omega} - \overline{\psi}_n||_{L_2(\partial\Omega)}$. So, following section 4, the number i, on which these iterations are stopped, is denoted as $i := m_n$. Once the criterion (13)-(14) is satisfied,



a) $c_{9,2}$ b) $c_{10,2}$ c) $c_{11,2}$ d) $c_{12,2}$

Figure 4. Test 1.1: spatial distribution of c_h after computing $q_{n,k}$; $n = 9, 10, 11, 12$, where n is number of the computed function q.

we take the last computed tail V_{n,m_n}, set $V_{n+1,1} := V_{n,m_n}$ and run computations again for q_{n+1}. Hence, the number m_n of iterations with respect to tails is chosen automatically "inside" of each iteration for q_n, which means that m_n, the number of iterations with respect to tails varies with n. So, new criteria (13), (14) means a more flexible stopping rule in the globally convergent algorithm compared with [2], since in [2] numbers m_n where not chosen automatically.

In all our tests we have introduced the multiplicative random noise in the boundary data, g_σ, by adding relative error to computed data g using the following expression

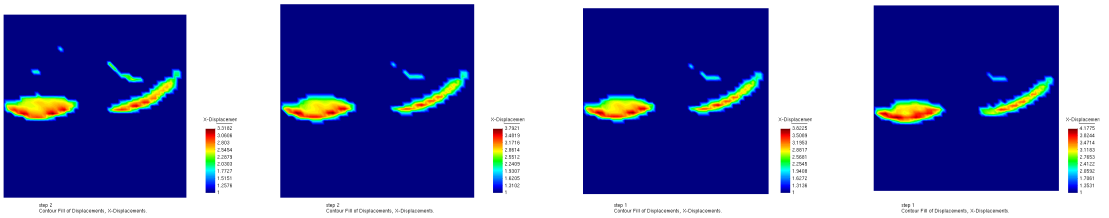
$$g_\sigma\left(x^i, t^j\right) = g\left(x^i, t^j\right)\left[1 + \frac{\alpha_j (g_{max} - g_{min})\sigma}{100}\right].$$

Here, $g\left(x^i, t^j\right) = u\left(x^i, t^j\right), x^i \in \partial\Omega$ is a mesh point at the boundary $\partial\Omega, t^j \in (0, T)$ is a mesh point in time, α_j is a random number in the interval $[-1; 1]$, g_{max} and g_{min} are maximal and minimal values of the computed data g, respectively, and $\sigma = 5\%$ is the noise level. Computations were performed on 16 parallel processors in NOTUR 2 production system at NTNU, Trondheim, Norway (67 IBM p575+ 16-way nodes, 1.9GHz dual-core CPU, 2464 GB memory).

Test 1.1

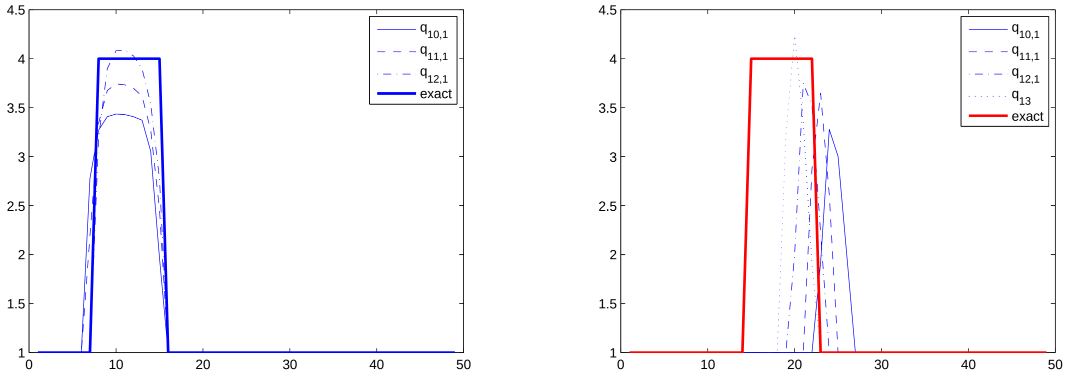
We test our numerical method on the reconstruction of the structure given on Figure 1-c). The plane wave f is initialized at the top boundary ∂G_1 of the computational domain G, propagates during the time period $(0, t_1]$ into G, is absorbed at the bottom boundary ∂G_2 for all times $t \in (0, T)$ and it is also absorbed at the top boundary ∂G_1 for times $t \in (t_1, T)$, see Figures 2.

One can see from Figure 4 that the location of the right small square is



a) $c_{10,2}$ b) $c_{11,1}$ c) $c_{12,1}$ d) $c_{13,1}$

Figure 5. Test 2.1: spatial distribution of c_h after computing $q_{n,k}; n = 10, 11, 12, 13$ where n is number of the computed function q.

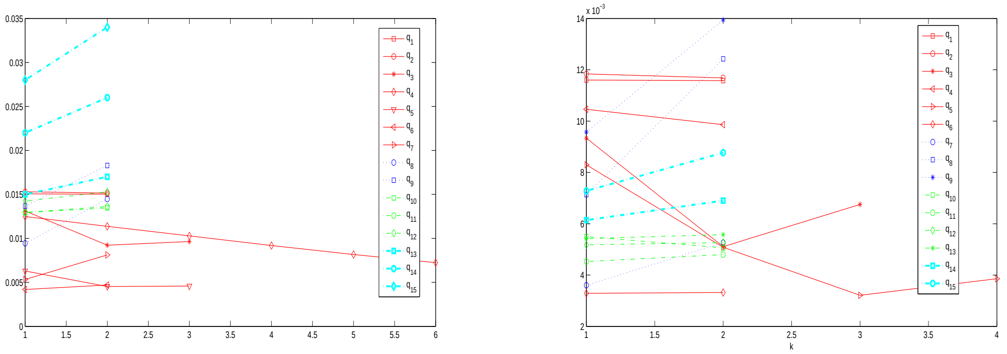


a) Test 1.1 b) Test 2.1

Figure 6. The one-dimensional cross-sections of the image of the function $c_{n,k}$ computed for corresponding functions $q_{n,1}$. On a) for Test 1.1 along the vertical line passing through the middle of the right small square; and on b) for Test 2.1 along the vertical line passing through the middle of the left small square.

imaged well. It follows from 4 -c) that the imaged contrast in this square is $3.8 : 1 = \max c_{11,2} : 1$ at $n := \overline{N} = 11$ (see below for this choice of $\overline{N}$). Thus, we have obtained the 5% error $(0.2/4)$ in the imaged contrast, which is the same as the error in the input data. As to the left small square, we got the same $3.8 : 1$ contrast in it for $c_{11,2}$. However, the location of the left square is shifted downwards, and both imaged squares are on about the same horizontal level. Values of the function $c(x) = 1$ outside of these squares are imaged accurately.

Figure 7-a) shows computed L_2-norms $F_{n,i}$. Using this figure, we analyze results of the reconstruction. One can see on Figure 7-a) that the number m_n of iterations



a) Test 1.1 b) Test 2.1

Figure 7. Computed L_2-norms of the $F_{n,i} = ||q_{n,i}\mid_{\partial\Omega} - \overline{\psi}_n||_{L_2(\partial\Omega)}$.

with respect to tails indeed varies with n, since m_n is chosen automatically now, using the criterion (13)-(14). We observe that the norms $F_{n,i}$ decrease until computing the function q_7. Next, they slightly grow, decay from $F_{9,2}$ to $F_{10,2}$ and then these norms stabilize on $n = 11, 12$. For $n = 13, 14, 15$ norms $F_{n,2}$ grow steeply. Thus, we conclude, that $\overline{N} = 11$ and we take $c_{11,2}$ as our final reconstruction result on the globally convergent stage.

Test 2.1

We now test our globally convergent method on the same structure of Figure 1-c). However, the difference with the previous test is that we now use the plane wave, which is initialized at the bottom boundary of computational domain G, see Fig.2. Figure 5 displays isosurfaces of resulting images of functions $c_{n,k}, n = 10, 11, 12, 13$. Figure 6-b) displays the one-dimensional cross-sections of computed images of functions $c_{n,k}$ superimposed with the correct one along the vertical line passing through the middle of the left small square. One can see from Figure 5 that the $3.8 : 1 = \max c_{11,1}(x) : 1$ contrast for $n := \overline{N} = 11$ (see below for this choice of $\overline{N}$) in the left square is imaged again with 5% error $(0.2/4)$ which is the same as the noise level in the data. As to the right small square, we got the same $3.8 : 1$ contrast as in the left square. However, the location of the right small square is shifted upwards.

Using Figure 7-b), which shows computed L_2-norms $F_{n,i}$, we analyze results of the reconstruction. We observe that computed norms $F_{n,i}$ decrease with n until computing the function q_7, i.e.., until $n = 7$ and these numbers grow with the increase of $n = 8, 9$. Next, we observe a steep decrease at $n = 10$ and a stabilization for $n = 11, 12$. For $n = 13, 14, 15$ norms $F_{n,i}$ grow steeply. Thus, we conclude, that $\overline{N} = 12$ and we take $c_{12,1}$ as our final reconstruction result on the globally convergent stage.

We observe, that in both Tests 1.1 and 2.1 the location of the small quare, which is located closer to the side from which the plane wave is launched, is imaged better than the location of the second small square. At the same time the inclusion/background contrast is imaged well in both small squares, so as the value of the coefficient $c(x) = 1$ outside of (imaged) small squares. Thus, to enhance images of locations, we are prompted to use the adaptivity technique.

3.3. Synthesis of the globally convergent algorithm with the adaptivity

The goal of two tests of this subsection is to demonstrate the performance of the synthesis of our globally convergence algorithm with the adaptivity technique. We take the starting point for the adaptivity the image obtained by the globally convergent method. Below "Test 1.2" (respectively "Test 2.2") means that we take the image obtained in the above Test 1.1 (respectively in Test 1.2), i.e. the function $c_{11,1}(x) := c_{glob}(x)$, as the starting point for our finite element adaptive algorithm. The boundary data $g = u\mid_{\partial\Omega}$ in both tests 1.2 and 2.2 are the same as ones in Test 1.1 and 2.1 respectively. In Tests 1.2 and 2.2 Γ denotes the side of the square Ω, opposite to the side from which the plane wave is launched and $\Gamma_T = \Gamma \times (0, T)$. In some sense the side Γ_T is the most sensitive one to the presence of those two small squares.

The adaptivity algorithm was described in section 2. Now the question is on how to choose the tolerance number β in (10). The choice of β depends on the behaviour of the computed value of $\max_{\overline{\Omega}} B_h(x)$ in right hand side of (10). If we choose β too small (for example, $\beta = 0$), then we will refine mesh in almost the entire domain Ω, since, realistically, after the optimization procedure $B_h(x)$ will be non- zero at almost all mesh points. Unlike this, our goal is to construct a new mesh with a few nodes as possible, while still getting a good enhancement of the solution obtained on the globally convergent stage of our two-stage numerical procedure. On the other hand, the parameter β can not be taken too close to 1 also, since in this case the automatic adaptive algorithm will come up with a too narrow region, where the mesh should be refined. Thus, the choice of β depends on concrete values of the function

opt.it.	4608 elements	5340 elements	6356 elements	10058 elements	14586 elements
1	0.0992683	0.097325	0.0961796	0.0866793	0.0880115
2	0.0988798	0.097322	0.096723	0.0868341	0.0880866
3	0.0959911	0.096723			0.0876543
4		0.096658			

Table 1. Test 1.2: $||u\ |_{\Gamma_T} - g||_{L_2(\Gamma_T)}$ on adaptively refined meshes. The number of stored corrections in the quasi-Newton method is $m = 15$. Computations was performed with the noise level $\sigma = 5\%$ and with the regularization parameter $\gamma = 0.01$.

$B_h(x)$ and should be chosen in numerical experiments. In (10) we take $\beta = 0.1$ on the coarse mesh, $\beta = 0.2$ on the one, two and three refined meshes, and $\beta = 0.6$ for all next refinements of the initial mesh.

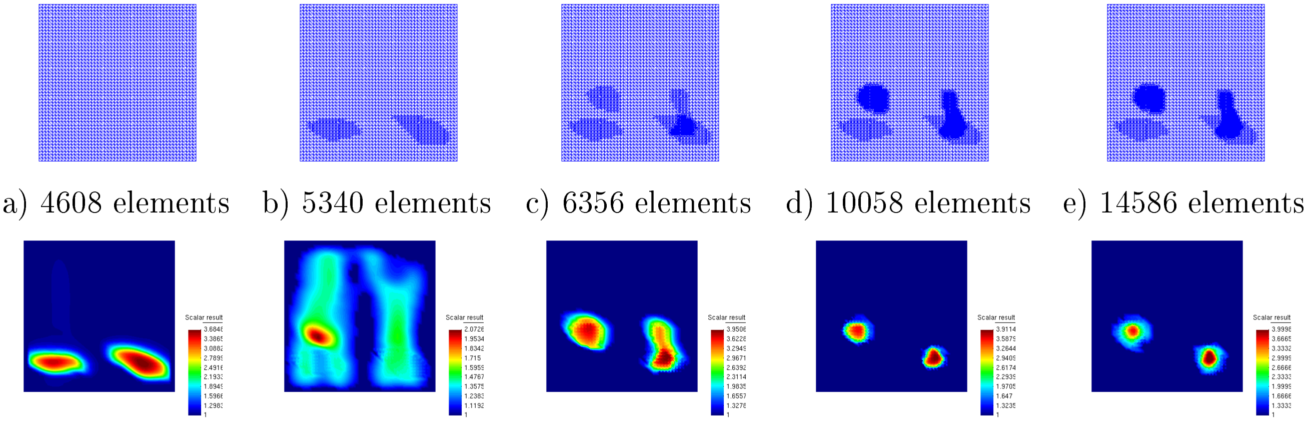
On all refined meshes we have used a cut-off parameter C_{cut} for the reconstructed coefficient c_h such that

$$c_h(x) = \begin{cases} c_h(x), & \text{if } |c_h(x) - c_{glob}(x)| \geq C_{cut} \\ c_{glob}(x), & \text{elsewhere.} \end{cases}$$

We choose $C_{cut} = 0$ for $m < 3$, $C_{cut} = 0.3$ for $m \geq 3$ in all tests, where m is the number of iterations in quasi-Newton method. Hence, the cut-off parameter ensures that we do not go too far from c_{glob}.

In the adaptive algorithm we can use box constrains for the reconstructed coefficient. We obtain these constraints using the solution obtained in the globally convergent part. Namely, in Tests 1.2 and 2.2 minimal and maximal values of the target coefficient in box constraints are taken using results of Tests 1.1 and 2.1. So, when conducting Tests 1.1 and 2.1, we have used only the knowledge of the number $d_1 = 0.5$ in (3). Now, since we know that the solution obtained on the first stage is a good approximation for the correct solution (Theorem 5.1 of [2]) and the maximal value of the computed coefficient is 3.8, we set $d_2 = 2$ in (3). Thus, in tests 1.2 and 2.2 we enforce that the coefficient $c(x)$ belongs to the set of admissible parameters, $c(x) \in C_M = \{c \in C(\overline{\Omega})|1 \leq c(x) \leq 4\}$.

Test 1.2.



a) 4608 elements b) 5340 elements c) 6356 elements d) 10058 elements e) 14586 elements

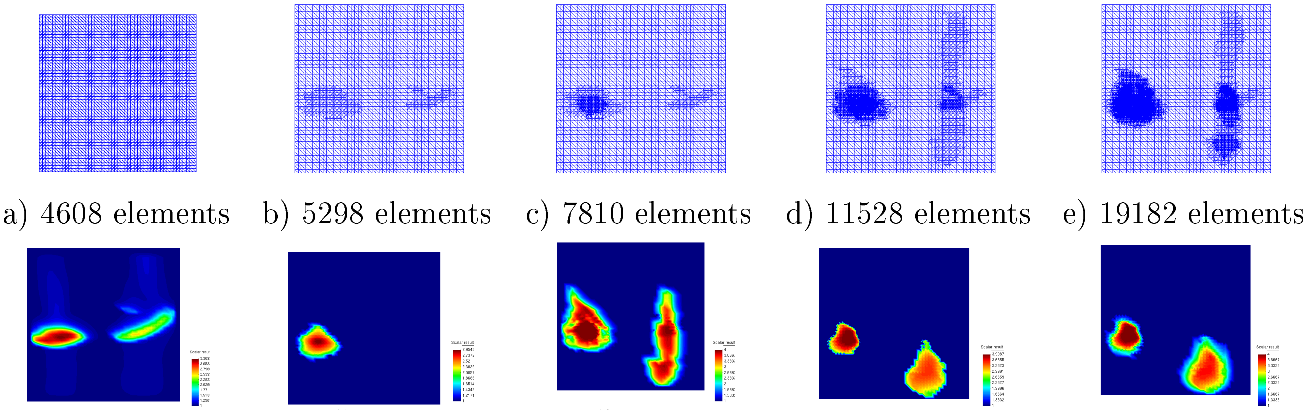
f) 4608 elements g) 5340 elements h) 6356 elements i) 10058 elements j) 14586 elements

Figure 8. Test 1.2: Adaptively refined computational meshes on a)-e) and spatial distribution of the parameter c_h with $\sigma = 5\%$, which corresponds to these meshes, on f)-j).

The plane wave is initialized on the top boundary of the domain G, which is the large rectangle of Fig. 7.1b. The starting point for the adaptivity algorithm is the function $c_{11,2}(x) := c_{glob}(x)$, which corresponds to Figure 4-c). We have performed numerical experiments with introducing $\sigma = 5\%$ of the multiplicative random noise in the function $g(x,t)$ in an adaptive procedure. First, the function $c_{glob}(x)$ was taken on the initial coarse mesh is shown on Figure 8-a) and the quasi-Newton method is applied on this mesh. Figure 8-f shows that the image was not improved when the same mesh was used as one on the globally convergent stage. Next, the mesh was adaptively refined four times using the above described procedure (subsection 6.4). Adaptively refined meshes shown on Figure 8-a-e). Table 1 presents computed L_2-norms of $||u\ |_{\Gamma_T}\ -g||_{L_2(\Gamma_T)}$. We observe that norms at the boundary decrease as meshes are refined. Then they slightly increase and are finally stabilized for all refinements $n > 3$ of the initial mesh. Thus, using this table, we conclude that on the three times refined mesh we get the final solution of our inverse problem, which corresponds to Figure 8-j) . One can see on Figure 8-j) that we are able to accurately reconstruct locations of both small squares. At the same time, an accurate inclusion/background contrast obtained on the globally convergent stage is preserved. This contrast turns out to be now 4:1=max $c_f(x)$: 1 instead of 3.8:1 calculated on the first stage, where $c_f(x)$ is the final imaged coefficient. The value

opt.it.	4608 elements	5298 elements	7810 elements	11528 elements	19182 elements
1	0.0992683	0.0976474	0.0976851	0.089979	0.0977153
2	0.0988798		0.0974385	0.0901018	0.097487
3	0.0959911			0.0901153	0.0975039

Table 2. Test 2.2: $||u\ |_{\Gamma_T} - g||_{L_2(\Gamma_T)}$ on adaptively refined meshes. The number of stored corrections in the quasi-Newton method is $m = 15$. Computations was performed with the noise level $\sigma = 5\%$ and with the regularization parameter $\gamma = 0.01$.



a) 4608 elements b) 5298 elements c) 7810 elements d) 11528 elements e) 19182 elements

f) 4608 elements g) 5298 elements h) 7810 elements i) 11528 elements j) 19182 elements

Figure 9. Test 2.2: Adaptively refined computational meshes on a)-e) and spatial distribution of the parameter c_h with $\sigma = 5\%$, which corresponds to these meshes, on f)-j).

of the coefficient $c_f(x) = 1$ outside of small squares is also imaged well.

We have used the smoothing indicator procedure applied to the reconstructed coefficient $c(x)$ on the all adaptively refined meshes. As it was stated in subsection 7.3 of [2], this procedure consists in a local averaging of computed values of $c_{n,i}(x)$. The use of the smoothing indicator for the reconstructed coefficient c_h has helped us to obtain more accurate images as well as to get a lesser number of finite elements in computational meshes.

Test 2.2

Now we test the synthesis of the globally convergent numerical method with the adaptivity with the starting point on the coarse mesh taken from the result of Test 2.1 and with the plane wave initialized at the bottom boundary of the computational domain G. The initial guess for the adaptive algorithm on the coarse mesh is the computed coefficient $c_{12,1}(x)$ presented on Figure 5-c). The boundary data g is taken the same as in Test 2.1.

As Test 2.1, we have used four (4) times adaptively refined meshes shown on Figure 9-a)-e). Figures 9-a),f) show that when the same mesh is used as one for the globally convergent stage, the image quality is not improved, which coincides with the observation of Test 2.1. However, adaptively refined meshes do improve the quality of the reconstruction, see Figures 9-f)-j). In Table 2 we present computed norms of $||u\ |_{\Gamma_T}\ - g||_{L_2(\Gamma_T)}$. We observe that these norms decrease as meshes are refined. They decrease until the third refinement. On the fourth refinement they slightly increase and then they stabilize. Further mesh refinements are not necessary since norms $||u\ |_{S_T}\ - g||_{L_2(S_T)}$ are stabilized for all refinements with $n > 3$ of the initial mesh, and we get the same reconstruction result with further refinements. Thus, using Table 2, we conclude that on the three times refined mesh we get the final solution of our inverse problem. One can see from Figure 9-j) that locations of both small squares are imaged accurately, so as the contrast $\max c_f(x) : 1 = 4 : 1$ in them, as well as the value of the final reconsructed coefficient $c_f(x) = 1$ outside of these two small squares.

Acknowledgements

This work was supported by the U.S. Army Research Laboratory and U.S. Army Research Office grants W911NF-05-1-0378 and W911NF-08-1-0470.

References

[1] H. Ammari, E. Iakovleva, and D. Lesselier. Music-type electromagnetic imaging of a collection of small three dimensional inclusions. *SIAM J.Sci.Comp.*, 29:674 709, 2007.

[2] L. Beilina and M. V. Klibanov, A globally convergent numerical method for a coefficient inverse problem, *SIAM J. Sci. Comp.*, 31(1):478-509, 2008.

[3] L. Beilina and M. V. Klibanov, A globally convergent numerical method and adaptivity for a hyperbolic coefficient inverse problem, available on-line at Chalmers Preprint Series ISSN 1652-9715, 2009:6 and at http://www.ma.utexas.edu/mp_arc/, 2009.

[4] L. Beilina and M. V. Klibanov, A globally convergent numerical method and the adaptivity technique for a hyperbolic coefficient inverse problem. Part I: analytical study, available on-line at http://www.ma.utexas.edu/mp_arc/, 2009.

[5] L. Beilina and M. V. Klibanov, A globally convergent numerical method and the adaptivity technique for a hyperbolic coefficient inverse problem. Part II: numerical studies, available on-line at http://www.ma.utexas.edu/mp_arc/, 2009.

[6] L. Beilina, K. Samuelsson, and K. Åhlander. Efficiency of a hybrid method for the wave equation. In *Inter- national Conference on Finite Element Methods*, Gakuto International Series Mathematical Sciences and Applications. Gakkotosho CO.,LTD, 2001.

[7] M. I. Belishev. Boundary control in reconstruction of manifolds and metrics (the bc method). *Inverse Problems*, 13, R1-R45, 1997.

[8] M. I. Belishev and V. Yu Gotlib. Dynamical variant of the bc-method: theory and numerical testing. *Inverse and Ill-Posed Problems*, 7:221 240, 1999.

[9] V. A. Burov, S. A. Morozov, and O. D. Rumyantseva. Reconstruction of fine-scale structure of acoustical scatterers on large-scale contrast background. *Acoustical Imaging*, 26:231 238, 2002.

[10] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31:629 651, 1977.

[11] M. V. Klibanov and A. Timonov. *Carleman Estimates for Coefficient Inverse Problems and Numerical Appli- cations.* VSP, Utrecht, The Netherlands, 2004

[12] J. Mueller and S. Siltanen. Direct reconstructions of conductivities from boundary measurements. *SIAM J. Sci. Comp.*, 24:1232 1266, 2003.

[13] R. G. Novikov. Multidimensional inverse spectral problem for the equation $-\triangle\psi + (v(x) - Eu(x))\psi = 0$. *Functional Analysis and Its Applications*, 22:11 22, 1988.

[14] R. G. Novikov. The $\overline{\partial}$ approach to approximate inverse scattering at fixed energy in three dimensions. *Inter- national Math. Research Papers*, 6:287 349, 2005.

[15] J. Nocedal, Updating quasi-Newton matrices with limited storage, *Mathematics of Comp.*, V.35, N.151, 773–782, 1991.

[16] O. Pironneau. *Optimal shape design for elliptic systems*, Springer Verlag, Berlin, 1984.