Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 162 (2004) 165–181

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Numerical method for solving a class of nonlinear elliptic inverse problems[☆]

M. Essaouini[a], A. Nachaoui[b,*], S. El Hajji[a]

[a] *Département de Mathématiques et Informatique, Faculté des sciences, Université Mohammed V, B.P. 1014, Rabat, Maroc*

[b] *Laboratoire de Mathématiques Jean Leray, CNRS UMR6629, Université de Nantes, B.P. 92208, F-44322 Nantes, France*

Received 26 December 2001; received in revised form 30 October 2002

Abstract

This paper discusses a method to solve a family of nonlinear inverse problems with Cauchy conditions on a part of the boundary and no condition at all on another part. An iterative boundary element procedure is proposed. The scheme uses a dynamically estimated relaxation parameter on the under-specified boundary. Various types of convergence, boundary condition formulations and effects of added small perturbations into the input data are investigated. The numerical results show that the method produces a stable reasonably approximate solution.

Keywords: Nonlinear inverse problem; Ill-posed problem; Iterative method; Boundary element method

1. Introduction

The nonlinear Cauchy problem for elliptic equations which is an example of nonlinear inverse problems is originating from various sciences and engineering disciplines. It can be encountered in the modeling of problems like electroencephalography [20], electrocardiography [16,17], geophysics [13], heat transfer [10]. For such problems, a part Γ_0 of the boundary is inaccessible, therefore we cannot prescribe any type of boundary conditions on Γ_0 (under-specified boundary). Instead an other part of the boundary Γ_2 is over-specified, for example the solution and its normal derivative are given on Γ_2.

[☆] This work was partially supported by: Action intégrée AI 181 MA/99.

[*] Corresponding author.

E-mail addresses: nachaoui@math.univ-nantes.fr (A. Nachaoui), elhajji@fsr.ac.ma (S. El Hajji).

doi:10.1016/j.cam.2003.08.011

There are two large groups of approaches to the solution of this kind of problems. One of the groups comprises methods based on bringing the problem into a class of well posed problem in Tikhonov's sense [1,11,12,15,18]. The other group consists of iterative methods which have recently found an even wider field of applications [7–9,14,17]. The schemes based on iterative method have the advantage of being simple to implement and not requiring any parameter selection.

In this paper, we are interested in solving the following class of nonlinear boundary ill-posed problem:

$$\begin{aligned} &-\nabla K(T)\nabla T = 0 \quad \text{on } \Omega,\\ &T|_{\Gamma_d} = f_d,\\ &T|_{\Gamma_2} = f_2,\\ &K(T)\partial_\nu T|_{\Gamma_2} = g_2,\\ &K(T)\partial_\nu T|_{\Gamma_n} = g_n, \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a domain for which the boundary Γ is such that

$$\Gamma = \Gamma_0 \cup \Gamma_d \cup \Gamma_2 \cup \Gamma_n,$$

and ∂_ν is the normal derivative operator.

First we use a transformation of the governing equation of problem (1) such that all the nonlinear aspect of the problem is transferred to the boundary of the domain, see Section 2. This reduces the nonlinear problem (1) to solve a Cauchy problem for the Laplace equation followed by a sequence of nonlinear independent scalar equations. Next, we use an iterative procedure based on a method by Jourhmane and Nachaoui [7] for linear inverse problem and propose a dynamically estimated relaxation parameter in order to increase the rate of convergence. Our proposed algorithm has the following interesting property: in order to pass from one iteration to the next, the value of the solution and its normal derivative are required only on the boundary. Thus we formulate the intermediate well-posed problems as boundary integral equations which will be discretised using boundary element method. Furthermore the solution of problem (1) for $x \in \bar{\Omega}$ has to be evaluated only after a stopping criterion has been satisfied. This may be done by solving the nonlinear scalar equation using the Newton–Raphson procedure, thus saving a substantial amount of computational time. In Section 4, various boundary condition formulations are investigated in which we discuss different aspect of the proposed algorithm including the numerical convergence and accuracy with respect to the mesh size discretization and number of iterations. Since the main issue of ill-posedness of inverse problems is the instability with respect to data errors, we tested our numerical algorithm against its behavior under data perturbation. All the computations have been performed in a square region which is an example of nonsmooth geometry.

2. Description of the algorithm

In the following we suppose that:

(1) The boundaries Γ_0, Γ_d, Γ_2 and Γ_n are smooth,
(2) $\text{meas}(\Gamma_0) \neq 0$, $\text{meas}(\Gamma_2) \neq 0$ and $\text{meas}(\Gamma_2) \geqslant \text{meas}(\Gamma_0)$,

(3) $\Gamma_0 \cap \Gamma_2 = \emptyset$,
(4) f_d, f_2, g_2 and g_n are smooth functions,
(5) K is a non negative continuous function.

We construct our approximation as follows:

The governing equation of problem (1) is transformed such that all the nonlinear aspects of the problem are transferred to the boundary of the domain. This may be done by introducing a new variable ω such that:

$$\nabla\omega = K(T)\nabla T.$$

Therefore the nonlinear inverse problem (1) is equivalent to the coupled problem (2)–(3) defined by

$$\begin{aligned} -\Delta\omega &= f && \text{on } \Omega,\\ \omega(x) &= F(f_d(x)) && \text{on } \Gamma_d,\\ \omega(x) &= F(f_2(x)) && \text{on } \Gamma_2,\\ \partial_\nu\omega &= g_2 && \text{on } \Gamma_2,\\ \partial_\nu\omega &= g_n && \text{on } \Gamma_n, \end{aligned} \tag{2}$$

and

$$F(T(x)) = \omega(x) \quad \forall x \in \Omega, \tag{3}$$

where F denotes the used transformation given by

$$F(t) = \int_c^t K(u)\,\mathrm{d}u \tag{4}$$

and c is an arbitrary real constant.

Based on the work in [7], the approximation of the linear inverse problem (2) is constructed as follows, first we specify an initial guess $v_0 \in H^{1/2}(\Gamma_0)$ on Γ_0 which in addition satisfies the following compatibility condition with the data on $\bar{\Gamma}_0 \cap \bar{\Gamma}_d$, see [4]:

$$\int_0^\mu \frac{[v_0(A - t\tau_0(t)) - f_d(A + t\tau_1)]^2}{t} < \infty \tag{5}$$

with $A \in \bar{\Gamma}_0 \cap \bar{\Gamma}_d$, $0 < \mu < \min\{\text{meas}(\Gamma_0), \text{meas}(\Gamma_d)\}$, $\tau_0(t)$ and $\tau_1(t)$ are the tangent vectors in the same direction to Γ_0 and Γ_d. Then for $k \geqslant 1$, we solve alternately the following mixed well-posed direct problems until a prescribed stopping criterion is satisfied:

$$\begin{aligned} -\Delta\omega^{(2k)} &= 0 && \text{on } \Omega,\\ \omega^{(2k)} &= v^{(k)} && \text{on } \Gamma_0,\\ \partial_\nu\omega^{(2k)} &= g_2 && \text{on } \Gamma_2,\\ \omega^{(2k)}(x) &= F(f_d(x)) && \text{on } \Gamma_d,\\ \partial_\nu\omega^{(2k)} &= g_n && \text{on } \Gamma_n, \end{aligned} \tag{6}$$

$$\begin{aligned}
&-\Delta\omega^{(2k+1)} = 0 && \text{on } \Omega,\\
&\partial_\nu\omega^{(2k+1)} = \partial_\nu\omega^{(2k)} && \text{on } \Gamma_0,\\
&\omega^{(2k+1)}(x) = F(f_2(x)) && \text{on } \Gamma_2,\\
&\omega^{(2k+1)}(x) = F(f_d(x)) && \text{on } \Gamma_d,\\
&\partial_\nu\omega^{(2k+1)} = g_n && \text{on } \Gamma_n,
\end{aligned} \tag{7}$$

with

$$v^{(0)} = v_0|_{\Gamma_0} \quad \text{and} \quad v^{(k)} = r\omega^{(2k-1)}_{|\Gamma_0} + (1-r)v^{(k-1)}_{|\Gamma_0} \quad \text{for } k \geqslant 1. \tag{8}$$

After, we solve

$$F(T(x)) = \omega^{(*)}(x), \quad \forall x \in \bar{\Omega},$$

where $\omega^{(*)}$ is the limit of the sequence obtained by solving alternately (6)–(7).

Note that the parameter r in Eq. (8) acts as a relaxation parameter and various relaxation procedure may be obtained by altering the value of the parameter r. One possible disadvantage of using a fixed parameter relaxation is the large number of iterations that may be required in order to achieve convergence (see Section 5). We give in the following a dynamical choice of the parameter r which gives convergence in a reasonably small number of iterations.

Let

$$\mathrm{e}^{(2k)} = \omega^{(2k)}|_{\Gamma_0} - \omega^{(2(k-1))}|_{\Gamma_0}, \quad \mathrm{e}^{(2k+1)} = \omega^{(2k+1)}|_{\Gamma_0} - \omega^{(2k-1)}|_{\Gamma_0},$$

and let $\langle .,. \rangle$ denotes the inner product in $L^2(\Gamma_0)$. Then, in the iteration k, the relaxation parameter $r^{(k)}$ can be given dynamically as the unique real minimizing the functional

$$\phi(r) = \|v^{(k+1)}(r) - v^{(k)}(r)\|_{L^2(\Gamma_0)}$$

given by

$$r^{(k+1)} = \frac{\langle \mathrm{e}^{(2k)}, \mathrm{e}^{(2k)} - \mathrm{e}^{(2k+1)} \rangle}{\|\mathrm{e}^{(2k)} - \mathrm{e}^{(2k+1)}\|^2_{L^2(\Gamma_0)}}. \tag{9}$$

As it will be shown in Section 4, this choice improves the rate of convergence.

The first glance at the boundary value problems (6) and (7) reveals that the main exchange is done at the boundary. Thus the boundary integral equation formulation (see [6]) is an appropriate choice to formulate the boundary value problems (6)–(7), this leads to a boundary element method discretization [3].

Remark 1. Many reasons make this choice advantageous:

(1) The boundary element method utilizes only information on the boundaries of interest and it reduces the dimension of the problem by one.
(2) In the boundary element method, only the boundary is discretised; hence the mesh generation is simpler for this method than for space discretization technique as finite elements method or finite differences method.

(3) The solution and its normal derivative are simultaneously determined, this avoid a supplementary stages of computation as it is required if we employ the finite element method or the finite differences method.

We describe the boundary element method in the next section.

3. The boundary integral equation formulation

The boundary integral equations are classical kind of formulations for partial differential equations [6]. They have been used for solving a wide variety of real world problems, [2,5,6,10]. Let Ω be a planar simply connected bounded region with the boundary Γ to which the divergence theorem can be applied. We assume that the boundary Γ is partitioned into two parts, Γ_{D} and Γ_{N}. In order to give a brief description of boundary integral equations we consider the following mixed boundary value problem:

$$\begin{aligned} &\Delta u = 0 && \text{on } \Omega, \\ &u = f && \text{on } \Gamma_{\mathrm{D}}, \\ &\partial_\nu u = g && \text{on } \Gamma_{\mathrm{N}}, \end{aligned} \tag{10}$$

where f and g are given continuous functions on Γ_{D} and Γ_{N}, respectively. It is well known that the solution of (10) verify [3]

$$\int_\Gamma \{u(y)\partial_{\nu_y} \ln|x-y|\,\mathrm{d}s_y - \partial_{\nu_y} u(y) \ln|x-y|\,\mathrm{d}s_y\} = \begin{cases} 2\pi u(x), & x \in \Omega, \\ \theta(x)u(x), & x \in \Gamma, \\ 0, & x \in \Omega_e, \end{cases} \tag{11}$$

where ν_y is the outward normal to Γ at y, $\theta(x)$ is the interior boundary angle at x and $\Omega_e = \mathbb{R}^2 \setminus \bar{\Omega}$. Using the above formulas the boundary value problem (10) can be reduced to solve

$$A\bar{u}(x) = F(x) \tag{12}$$

with

$$\bar{u}(x) = \begin{cases} u(x), & x \in \Gamma_{\mathrm{N}}, \\ \partial_\nu u, & x \in \Gamma_D, \end{cases}$$

$$A\bar{u}(x) = \begin{cases} \displaystyle\int_{\Gamma_{\mathrm{N}}} u(y)\partial_{\nu_y} \ln|x-y|\,\mathrm{d}s_y - \int_{\Gamma_{\mathrm{D}}} \partial_{\nu_y} u(y) \ln|x-y|\,\mathrm{d}s_y, \\ \displaystyle -\theta(x)u(x) + \int_{\Gamma_{\mathrm{N}}} u(y)\partial_{\nu_y} \ln|x-y|\,\mathrm{d}s_y - \int_{\Gamma_{\mathrm{D}}} \partial_{\nu_y} u(y) \ln|x-y|\,\mathrm{d}s_y \end{cases}$$

and

$$F(x) = \begin{cases} \displaystyle\theta(x)f(x) + \int_{\Gamma_{\mathrm{N}}} g(y) \ln|x-y|\,\mathrm{d}s_y - \int_{\Gamma_{\mathrm{D}}} f(y)\partial_{\nu_y} \ln|x-y|\,\mathrm{d}s_y, \\ \displaystyle\int_{\Gamma_{\mathrm{N}}} g(y) \ln|x-y|\,\mathrm{d}s_y - \int_{\Gamma_{\mathrm{D}}} f(y)\partial_{\nu_y} \ln|x-y|\,\mathrm{d}s_y. \end{cases}$$

Eq. (12) has a unique solution, thus to formulate the boundary value problems (6) and (7) with the boundary integral equations we have only to determine f, g, $\bar{u}$, Γ_{N} and Γ_{D} for even and odd iterations. Indeed, for even iteration the boundaries are: $\Gamma_{\mathrm{D}}=\Gamma_0\cup\Gamma_d$ and $\Gamma_{\mathrm{N}}=\Gamma_2\cup\Gamma_n$, the unknown is

$$\bar{u}^{(2k)}(x)=\begin{cases}\omega^{(2k)}(x) & x\in\Gamma_{\mathrm{N}},\\ \partial_\nu\omega^{(2k)}(x) & x\in\Gamma_{\mathrm{D}},\end{cases}$$

and the boundary data are defined as follows:

$$f^{(2k)}(x)=\begin{cases}F(f_d(x)) & x\in\Gamma_d,\\ v^{(k)}(x) & x\in\Gamma_0,\end{cases}\qquad g^{(2k)}(x)=\begin{cases}g_n(x) & x\in\Gamma_n,\\ g_2(x) & x\in\Gamma_2.\end{cases}$$

For odd iteration: $\Gamma_{\mathrm{D}}=\Gamma_2\cup\Gamma_d$ and $\Gamma_{\mathrm{N}}=\Gamma_0\cup\Gamma_n$ the unknown is

$$\bar{u}^{(2k+1)}(x)=\begin{cases}\omega^{(2k+1)}(x) & x\in\Gamma_{\mathrm{N}},\\ \partial_\nu\omega^{(2k+1)}(x) & x\in\Gamma_{\mathrm{D}},\end{cases}$$

and the boundary data in this case are:

$$f^{(2k+1)}(x)=\begin{cases}F(f_d(x)) & x\in\Gamma_d,\\ F(f_2(x)) & x\in\Gamma_2,\end{cases}\qquad g^{(2k+1)}(x)=\begin{cases}g_n(x) & x\in\Gamma_n,\\ \partial_\nu\omega^{(2k)}(x) & x\in\Gamma_0.\end{cases}$$

By solving the obtained equations for even and odd iterations we get both $\omega^{(k)}(x)$ and $\partial_\nu\omega^{(k)}(x)$, $\forall x\in\Gamma$. Then, we use green formula (11) to obtain $\omega^{(k)}(x)$, $\forall x\in\Omega$.

4. Numerical results and discussions

As a typical example for testing the algorithm we take

$$T(x,y)=\ln(2+x^2-y^2),$$

$$K(t)=\exp(t),$$

$$F(t)=\int_0^t \exp(u)\,\mathrm{d}u=\exp(t)-1,$$

and let Ω be the domain defined by $\Omega=(0,1)\times(0,1)$. We set $\Gamma_0=\{0\}\times(0,1)$ and $\Gamma_2=\{1\}\times(0,1)$. The parts of the boundary Γ_d and Γ_n are specified via the eventuality of boundary condition formulations that are considered. It is clear that

$$\nabla K(T)\nabla T=0.$$

The unknowns on the under-specified boundary Γ_0 are given by $T(0,y)=\ln(2-y^2)$ and $\partial_\nu K(T)$ $T(x,y)|_{\Gamma_0}=0$. The boundary data on the over-specified boundary Γ_2 are $f_2(y)=\ln(3-y^2)$ and $g_2(y)=2$. The auxiliary problems (6) and (7) corresponding to this example are discretised by the boundary element method using a piecewise constant polynomial interpolation. The number of boundary elements used for discretising the boundary Γ is taken to be $N\in\{40,80,160,200\}$. We denote by $\|.\|_{0,\Gamma_0}$ the discrete L^2 norm defined on Γ_0. The convergence of the algorithm may be investigated by evaluating at every iteration the error

$$G_k = \|F(T) - \omega^{2k+1}\|_{0,\Gamma_0}, \tag{13}$$

where ω^{2k+1} is the numerical solution on the boundary Γ_0 obtained after k iterations and $F(T)$ is the exact solution of the problem given by (2). In a similar way we may evaluate the errors in retrieving the flux on the boundary Γ_0 given by

$$g_k = \|\partial_\nu F(T) - \partial_\nu \omega^{2k+1}\|_{0,\Gamma_0}. \tag{14}$$

The behavior of the method is also investigated by evaluating the difference between two consecutive approximations for the solution and its normal derivative on the boundary Γ_0 given by

$$E_k = \|\omega^{2k+1} - \omega^{2k-1}\|_{0,\Gamma_0}, \quad e_k = \|\partial_\nu \omega^{2k+1} - \partial_\nu \omega^{2k-1}\|_{0,\Gamma_0}. \tag{15}$$

Based on absolute errors the following stopping criterion is considered:

$$\max(\|\omega^{(2k+1)} - \omega^{(2k-1)}\|_{0,\Gamma_0}, \|\omega^{(2k)} - \omega^{(2k-2)}\|_{0,\Gamma_0}) \leqslant \eta, \tag{16}$$

where η is a small prescribed positive quantity. Note that Eq. (16) express that the sequences $\omega^{(2k)}$ and $\omega^{(2k+1)}$ converge in $H^{1/2}(\Gamma_0)$.

Various types of boundary conditions formulations of determined nonlinear Cauchy problems are investigated. For all numerical experiments, we take $\eta = 10^{-5}$.

4.1. Formulation 1: ($\Gamma_d \neq \emptyset, \Gamma_n \neq \emptyset$)

In this formulation we consider the case when $\Gamma_d = (0,1) \times \{0\}$ and $\Gamma_n = (0,1) \times \{1\}$, then the known data for this formulation are given by: $f_d(x) = T(x,0) = \ln(2 + x^2)$ and $g_n(x) = -2$ for $x \in (0,1)$. As an initial guess $v_0 \in H^{1/2}(\Gamma_0)$ for the step one of the algorithm, we have chosen

$$v_0(y) = 2 - 2y, \quad y \in [0,1], \tag{17}$$

this choice corresponds to $T_0 = \ln(2 - 2y)$, satisfies the compatibility condition at the corner $\bar{\Gamma}_0 \cap \bar{\Gamma}_d$ and ensure the continuity of $\partial F(T)/\partial y$ at the corner $\bar{\Gamma}_0 \cap \bar{\Gamma}_n$. We notice that v_0 is not too close to the analytical solution $F(T)$ on the under-specified boundary Γ_0.

Fig. 1 shows, on a semi-log scale, the successive differences E_k and e_k as functions of the number of iterations k. According to these results, we observe a weak oscillation of E_k and e_k due to the automatic search of the relaxation parameter at each iteration, see Fig. 3. This slight instability in E_k and e_k does not affects the behavior of the errors G_k and g_k as it is illustrated in Fig. 2.

If we turn back to Fig. 2 which shows, on a log–log scale, the sequences G_k and g_k as functions of the number of iterations k for various number of boundary elements N, we can see easily that the quantities G_k and g_k decrease when N increases and they remain constant after a few iteration. Therefore the method is stable with respect to the number of boundary element.

The conclusions drawn from Fig. 2 are graphically enhanced in Fig. 4 which presents the numerical results obtained with the mesh $N = 160$, for the unknowns data on Γ_0 in comparison with the analytical solution, its normal derivative and the initial guess.

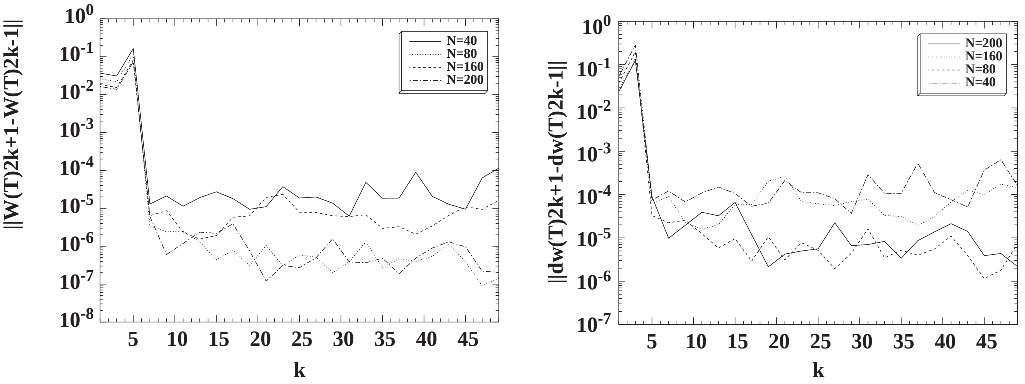

Fig. 1. Convergence errors E_k and e_k given by Eq. (15) for different N.

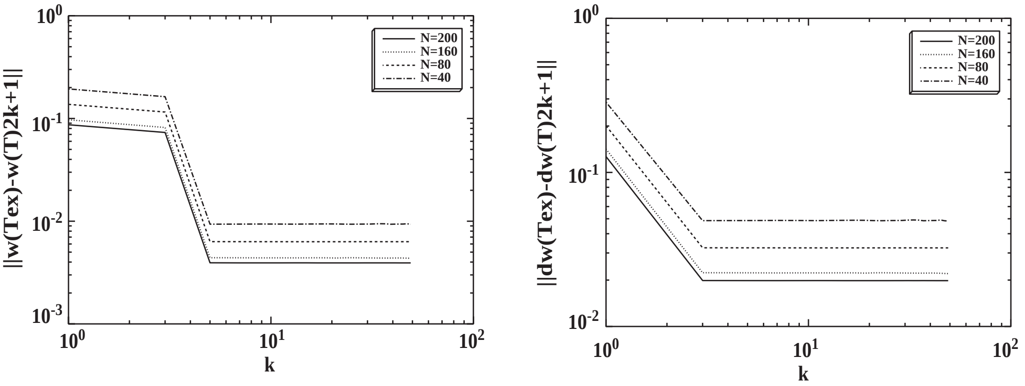

Fig. 2. The accuracy errors G_k and g_k given by Eqs. (13) and (14) as functions of the number of iterations.

We have also implemented the algorithm with various fixed relaxation parameters $r \in]0,2]$. We denote by r^* the fixed optimal parameter, i.e. r^* is the parameter such that

$$k(r^*) = \min_{r \in]0,2]} k(r), \tag{18}$$

where $k(r)$ is the number of iterations required to achieve the convergence when the algorithm is implemented with the fixed parameter r in (8). Note that taking a fixed parameter $r = 1$ in (8) gives a similar algorithm proposed by Kozlov et al. [9].

We observe from Fig. 5 that when the algorithm is implemented with the dynamically estimated relaxation parameter given by (9), the prescribed stopping criterion (16) is satisfied after five iterations, while it is required more then 100 iterations for $r = 1$. With the fixed optimal parameter r^* the convergence is obtained after 25 iterations.

We notice that at the convergence, the same precision is reached for the solution and the flux with the different relaxation parameters, see Fig. 6. Fig. 7 shows the numerical solutions and their normal derivatives obtained on the boundary Γ_0 for three relaxation parameters, namely the dynamically

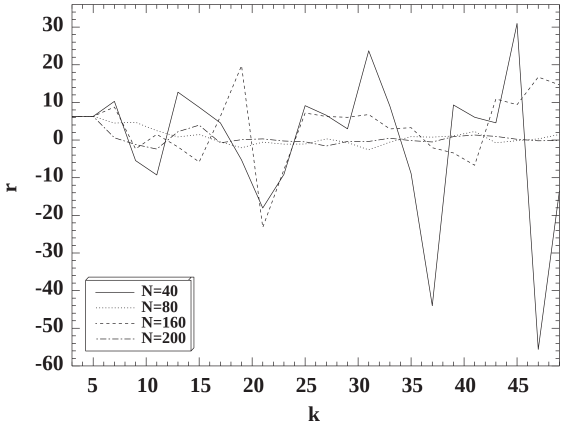

Fig. 3. Variation of the dynamically estimated relaxation parameter given by (9) as a function of the number of iterations.

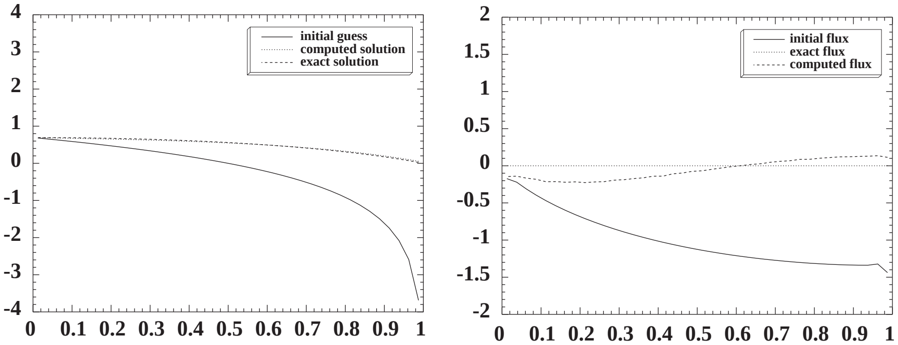

Fig. 4. The numerical results for the solution and its normal derivative obtained with $N = 160$.

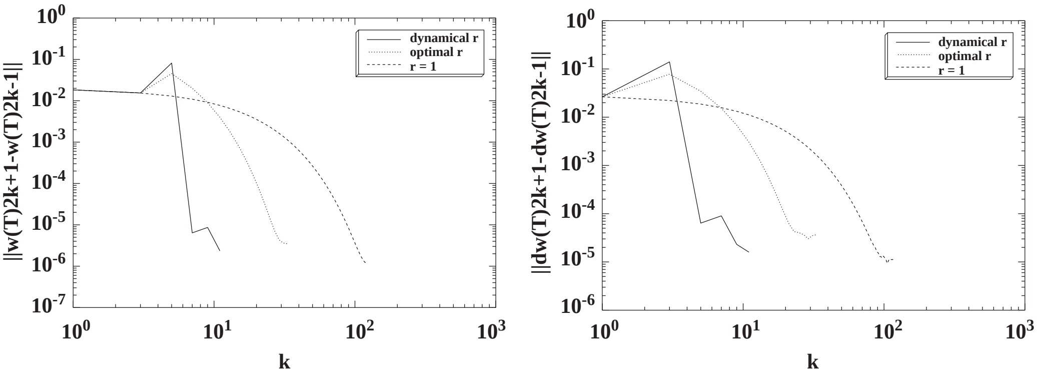

Fig. 5. Differences between the odd iterations for the solution and its flux for three relaxation parameters.

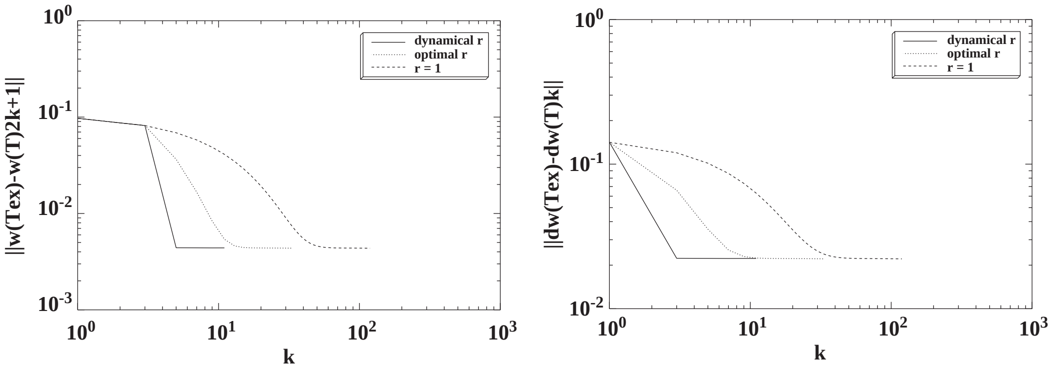

Fig. 6. The accuracy errors G_k and g_k for different relaxation parameters.

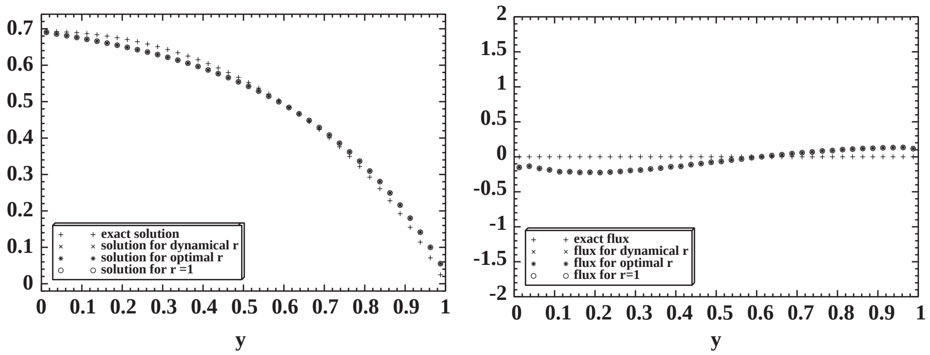

Fig. 7. Comparison of the obtained numerical results for the three relaxation parameter with $N = 160$.

estimated parameter, the fixed optimal parameter r^* and the fixed parameter $r = 1$. It can be seen from this figure that the numerical solutions and their normal derivatives obtained with the three parameters coincide. Similar results are obtained for any relaxation parameter $r \in]0, 2]$.

It should be noted at this stage that we have used an exact integration of the transformation F for all the performed computations. But in practice F may not have an explicit primitive or it should be an experimental measurements. Thus we have to solve an approached equation instead of solving (4). More precisely, using the Gauss quadrature formula we have to solve

$$\sum_{j=1}^{j=5} \theta_j (K(\phi(x) + \alpha_j) + K(-\alpha_j)) = \omega(x) \quad \forall x \in \Omega, \tag{19}$$

where θ_j and α_j are known, see [19]. If we take into consideration that we have to solve a difficult ill-posed problem in a nonsmooth region for which instabilities and a decreases in the rate of convergence should be awaited, we will not be surprised if the numerical integration of F produces a negative repercussions on the computed solution. But, as it is shown in Figs. 8 and 9, we obtain a solution with a similar precision and after the same number of iterations as in the case when F is integrated exactly. This is due the regularizing character of the algorithm.

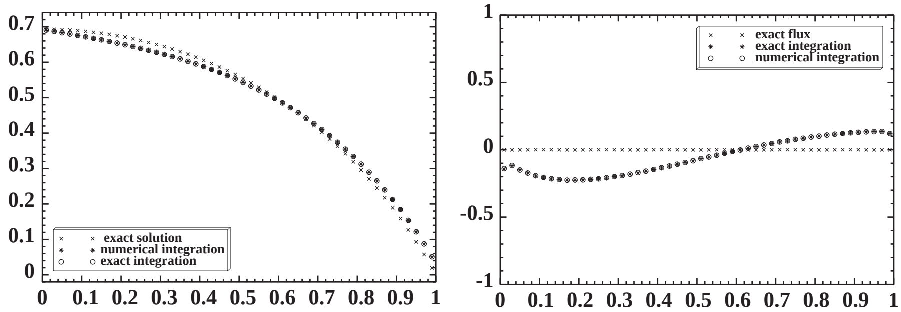

Fig. 8. Comparison of the obtained numerical results for numerical and exact integration with $N = 160$.

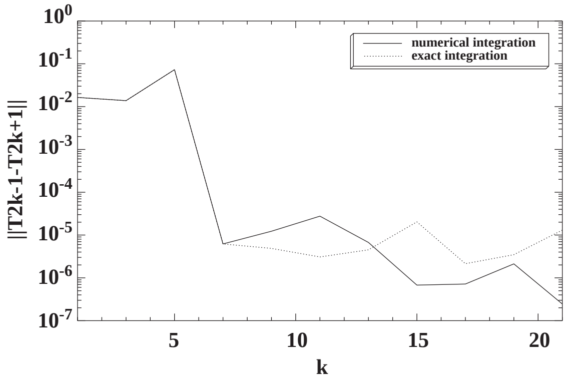

Fig. 9. Gap between the odd iteration for numerical and exact integration.

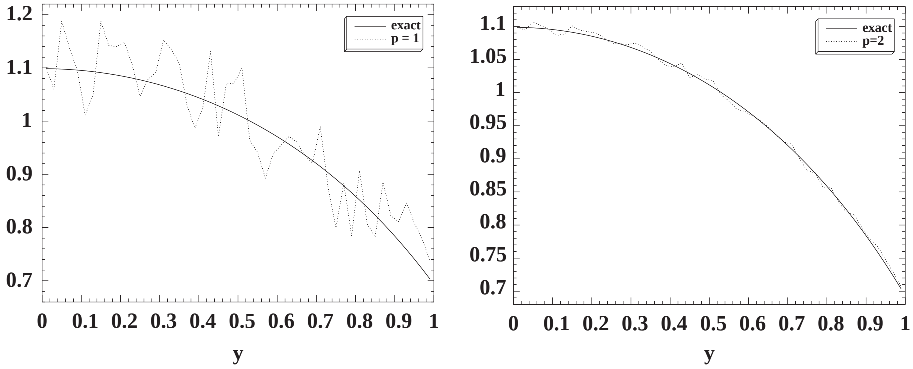

Fig. 10. Different perturbed data on Γ_2.

Note that the numerical integration of F is a particular perturbation of the boundary data. But it is more interesting to examine the behavior of the numerical solution when an arbitrary perturbation is introduced. For this, we add the following quantity $10^{-p} \times (2 \times \text{rand}(x) - 1)$ to the given data on Γ_2, where rand is the FORTRAN random function. The noisy data are presented in Fig. 10 and the corresponding numerical solutions are shown in Fig. 11.

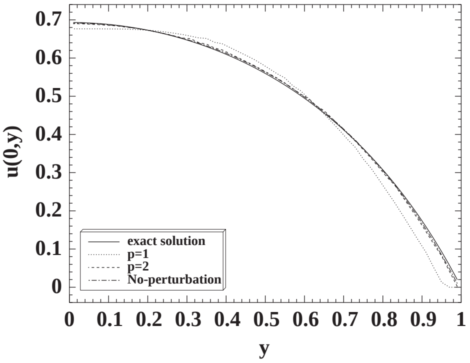

Fig. 11. Comparison of the numerical results obtained with the different perturbed data, $N = 160$.

For $p \geqslant 3$, the numerical results obtained with perturbed data coincide with the numerical results obtained with no perturbed data. Therefore this case is not presented here.

It can be seen that as p increases, the numerical solution better approximates the exact solution, whilst remaining stable. For $p = 1$, the obtained results are to be considered more than reasonable, keeping in mind that the problem (2) is an ill-posed problem with a very oscillatory data.

4.2. Formulation 2: $(\Gamma_d \neq \emptyset, \Gamma_n = \emptyset)$

In this formulation $\Gamma_d = (0,1) \times \{0\} \cup (0,1) \times \{1\}$, then the known data for this formulation are given by

$$f_d(x) = T|_{\Gamma_d} = \begin{cases} \ln(2 + x^2), \\ \ln(3 + x^2). \end{cases}$$

For step 1 of the algorithm, as an initial guess $v_0 \in H^{1/2}(\Gamma_0)$ which satisfies the compatibility condition at the two corners of $\bar{\Gamma}_0 \cap \bar{\Gamma}_d$, we have chosen

$$v_0(y) = 2y^2 - 3y + 2.$$

Fig. 12 shows the absolute errors G_k and g_k as functions of the number of iterations. The inaccuracy is about 10^{-3}. This level is reached after five iterations with the dynamical parameter relaxation. While it is required more than 100 iterations with the fixed optimal parameter and more than 1000 iterations with $r = 1$ to attain the same precision.

Fig. 13 shows the numerical results obtained, using the mesh $N = 160$ and the dynamically estimated relaxation parameter, after five iterations for the boundary solution and its normal derivative, in comparison with the analytical solutions. We have also included the initial guess v_0 and its derivative.

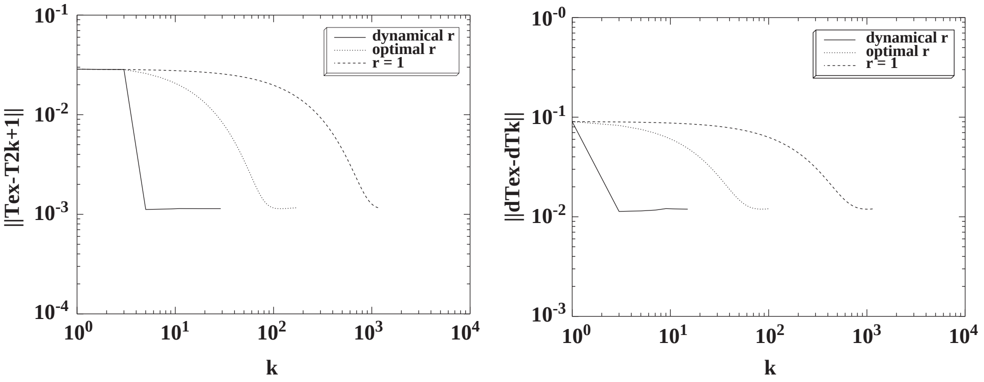

Fig. 12. The accuracy errors G_k and g_k for the three relaxation parameter with $N = 160$.

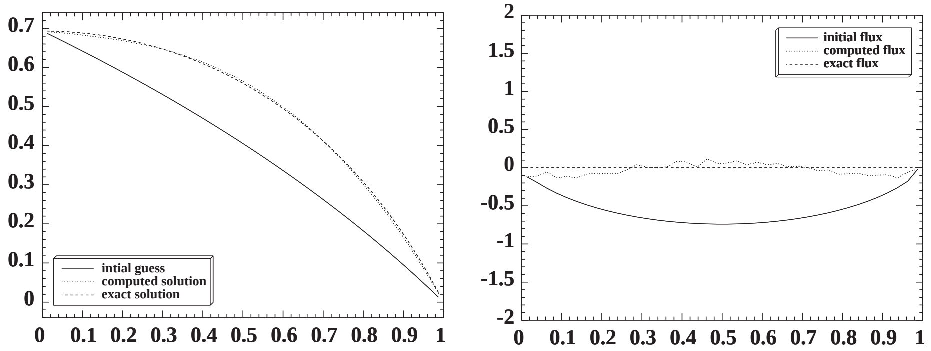

Fig. 13. Numerical results obtained for the solution and its normal derivative.

4.3. Formulation 3: ($\Gamma_d = \emptyset, \Gamma_n \neq \emptyset$)

In this formulation we consider the case when $\Gamma_n = (0,1) \times \{0\} \cup (0,1) \times \{1\}$, then the known data for this formulation are given by

$$g_n(x) = \partial_v T|_{\Gamma_n} = \begin{cases} 0 & \text{on } (0,1) \times \{0\}, \\ -2 & \text{on } (0,1) \times \{1\}. \end{cases}$$

For step 1 of the algorithm, as an initial guess $v_0 \in H^{1/2}(\Gamma_0)$ which ensures the continuity of $\partial F(T)/\partial y$ at the two corners of $\bar{\Gamma}_0 \cap \bar{\Gamma}_n$, we have chosen

$$F(v_0(y)) = \frac{4}{\pi} \cos\left(\frac{\pi}{2}\, y\right) + 1000.$$

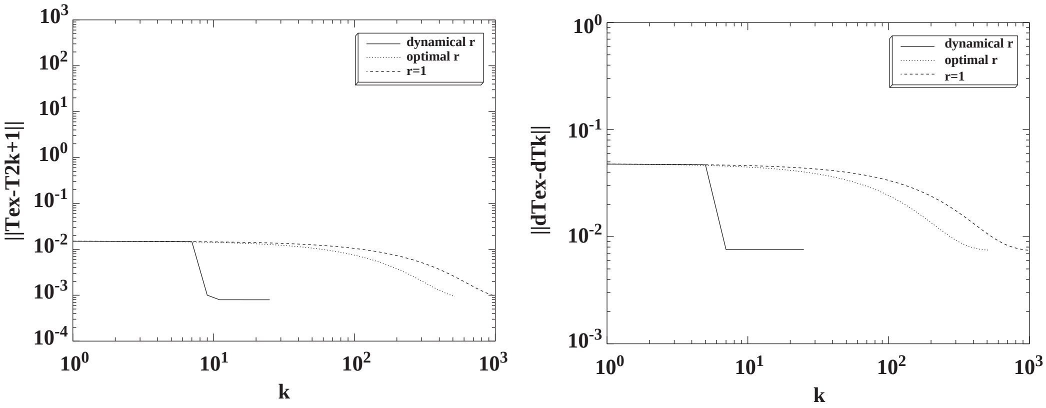

Fig. 14. The accuracy errors for the three relaxation parameter with $N = 160$.

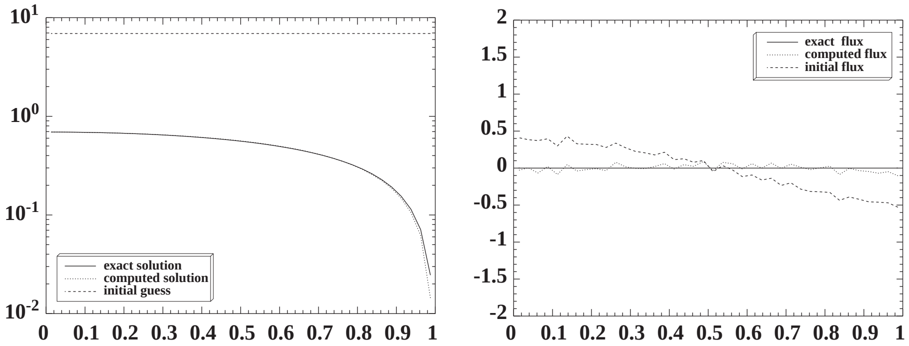

Fig. 15. Numerical results obtained for the solution and its normal derivative.

Which also ensures that the initial guess is not too close to the exact value of the solution. Fig. 14 shows the absolute errors G_k and g_k as functions of the number of the iterations. The level of the numerical accuracy for the solution is about 10^{-3}, it was obtained with the dynamically estimated relaxation parameter after 11 iterations. This level is reached after 500 iterations with the fixed optimal parameter.

From Figs. 14 and 15 it can be seen that the level of numerical accuracy attained is very good.

Over all the numerical results obtained for the formulations 1–3, we conclude that the computational cost of the algorithm with the dynamically estimated relaxation parameter is significantly less than the computational cost of the implemented method with fixed relaxation parameters.

The next formulation allows for Γ_0 and Γ_2 to be nonsmooth boundaries and therefore violates the hypotheses on which the mathematical algorithm was proved to be convergent by Jourhmane and Nachaoui [8].

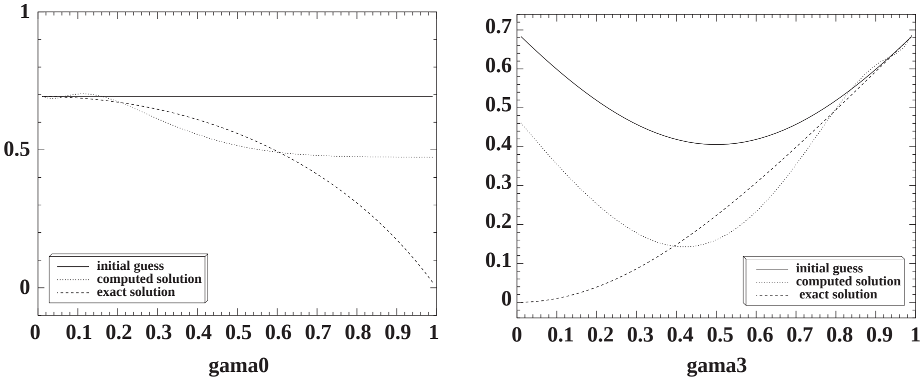

Fig. 16. The computed solution with $N = 160$.

4.4. Formulation 4: ($\Gamma_d = \emptyset, \Gamma_n = \emptyset$)

In this formulation we take $\Gamma_0 = (0,1) \times \{1\} \cup \{0\} \times (0,1)$ and $\Gamma_2 = (0,1) \times \{0\} \cup \{1\} \times (0,1)$, then the unknown data on the under-specified boundary Γ_0 are

$$T|_{\Gamma_0} = \begin{cases} \ln(2 - y^2), & x = 0, \\ \ln(1 + x^2), & y = 1, \end{cases} \quad \text{and} \quad \partial_\nu T|_{\Gamma_0} = \begin{cases} 0, & x = 0, \\ -2, & y = 1, \end{cases}$$

and the known data on the over-specified boundary Γ_2 are given by

$$T|_{\Gamma_2} = \begin{cases} \ln(2 + x^2), & y = 0, \\ \ln(3 - y^2), & x = 1, \end{cases} \quad \text{and} \quad \partial_\nu T|_{\Gamma_2} = \begin{cases} 0, & y = 0, \\ 2, & x = 1. \end{cases}$$

For step 1 of the algorithm, as an initial guess $v_0 \in H^{1/2}(\Gamma_0)$ which satisfies the compatibility condition and ensures the continuity of $\partial F(T)/\partial y$ and $\partial F(T)/\partial x$ at the corners of $\bar{\Gamma}_0 \cap \bar{\Gamma}_2$, we have taken

$$v_0(x, y) = 2x^2 - 2x + 2,$$

which also ensures that the initial guess is not too close to the exact value given by $F(T)_{|\Gamma_0}$.

Figs. 16 and 17 show the numerical results obtained using the dynamically estimated relaxation parameter, for the boundary solution and its normal derivative in comparison with the analytical solution and its normal derivative. Also the initial guess is included. A glance at Figs. 16 and 17 reveals that the level of numerical accuracy attained improves very slowly for predicting the boundary solution and its normal derivative. These error estimates are excessive compared with those obtained in the previous formulations. This can be explained by the fact that the regularity hypothesis on Γ_0 and Γ_2 on which the proof for the convergence is based (see [8]) is violated.

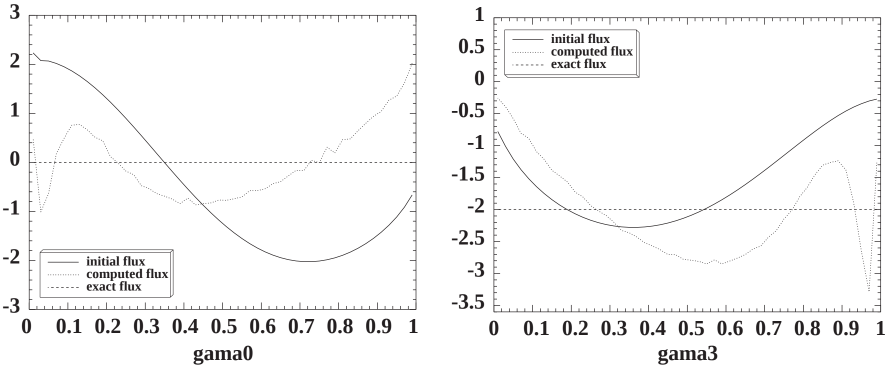

Fig. 17. The computed flux with $N = 160$.

5. Conclusion

In this paper a numerical iterative boundary element method for solving a class of nonlinear elliptic inverse problem has been developed. In Section 2, the given problem has been transformed into a linear problem followed by a sequence of nonlinear independent scalar equations. We solved the linear inverse problem using an alternating method based on the work of Jourhmane and Nachaoui [7]. In order to accelerate the convergence, we relaxed the algorithm and proposed a dynamical choice of the relaxation parameter. The auxiliary linear boundary value problems was discretised by the boundary element method using a piecewise constant polynomial interpolation and the linear scalar equations was solved by Newton method. In addition to the ill-posedness of the considered problem, the domain solution was chosen to have a nonsmooth boundary in order to test a sever case. The method was implemented with various relaxation parameters. The numerical results presented in Section 4 for the formulations 1–3 showed that:

- The dynamically relaxed algorithm produces a remarkably fast and convergent solution in comparison with that obtained by the method based on fixed relaxation parameters.
- The numerical solution is stable with respect to increasing the number of boundary elements and the absolute error remain constant even if we increase the number of iterations.
- The level of numerical inaccuracy is less than the order of the introduced perturbation. This fact was illustrate when the used transformation is evaluated by an adequate numerical integration scheme, for this situation we have obtained the same precision as in the case of an exact integration of the transformation since the perturbation was very small. This fact is due to the regularizing effects of the algorithm.
- The accuracy of the numerically obtained solution in formulation 3, was better than that obtained in formulations 1–2 which is to be expected since the imposition of the Neumann condition contains more information than the imposition of the Dirichlet condition.
- Although all the starting guess for initializing the algorithm was chosen sufficiently away from the exact solution on the under-specified boundary, the convergence was achieved in few iterations. This fact confirm the swiftness and robustness of the method.

Finally, the method was implemented for solving the problem with nonsmooth boundaries Γ_0 and Γ_2. This violates the hypothesis on which the mathematical proofs of the convergence of the algorithm are based and produces very poor solution. However, it should be mentioned that if extra information, such as $v_0(O,L)=w(0,L)$, is introduced at the corner of Γ_0, then an accurate numerical solution can be obtained.

References

[1] V. Badeva, V. Markov, Problèmes Incorrectement posés. Théorie et application, Filtrage Optimal, Contrôle Optimal, Analyse et Synthèse de Systhèmes, Reconnaissance d'Images, Masson, Paris, 1991.

[2] M. Bonnet, Boundary Integral Equations Methods for Solids and Fluids, Wiley, Chichester, 1997.

[3] C.A. Brebbia, J.C.F. Telles, L.C. Wrobel, Boundary Element Techniques: Theory and Application in Engineering, Springer, Berlin, 1984.

[4] P. Grisvard, Elliptic Problems in Non-smooth Domains, Pitman, New York, 1985.

[5] J. Hess, Panel methods in computational fluid dynamics, Ann. Rev. Fluid Mech. 22 (1990) 255–274.

[6] M.A. Jaswon, G.T. Symm, Integral equation methods in potential theory and elastostatics, Computational Mathematics and Applications, Academic Press [Harcourt Brace Jovanovich, Publishers], London, New York, 1977.

[7] M. Jourhmane, A. Nachaoui, An alternating method for an inverse Cauchy problem, Numer. Algorithms 21 (1999) 247–260.

[8] M. Jourhmane, A. Nachaoui, Convergence of an alternating method to solve Cauchy problem for Poisson's equation, Appl. Anal., to appear.

[9] V.A. Kozlov, V.G. Maz'ya, A.V. Fomin, An iterative method for solving the Cauchy problem for elliptic equations, Comput. Math. Math. Phys. 31 (1991) 45–52.

[10] K. Kurpisz, A.J. Nowak, Inverse Thermal Problems, Computational Mechanics, Southampton, Boston, 1995.

[11] M.M. Lavrent'ev, Some Ill-Posed Problems of Mathematical Physics, Izestiva Sibirskogo Otdel Akademiya Nauk SSSR, Novosibirsk, 1962.

[12] M.M. Lavrent'ev, V.G Romanov, V.G. Vasil'ev, Multidimensional inverse problems for differential equations, Izestiva Sibirskogo Otdel Akademiya Nauk SSSR, Novosibirsk, 1969.

[13] P.S. Martyshko, Inverse problems of electromagnetic geophysical fields. Inverse and Ill-Posed Problems Series, Utrecht, 1999.

[14] A. Nachaoui, Numerical linear algebra for reconstruction inverse problems, J. Comput. Appl. Math. 162 (2004) this issue.

[15] L.E. Payne, Improperly posed problems in partial differential equations, in: Regional Conference Series in Applied Mathematics, Vol. 22, SIAM, Philadelphia, PA, 1975.

[16] Y. Rudy, B.J. Messinger, The inverse problem in electrocardiography: solution in terms of epicardial potentials, CRC Crit. Rev. Biomed. Eng. 16 (1998) 215–268.

[17] A.A. Samarskii, A.N. Pavlov, P.N. Vabishchevich, Numerical method for electrocardiographic inverse problem. 2D model problem, J. Inverse Ill-Posed Probl. 4 (4) (1996) 317–329.

[18] A.N. Tikhonov, A.V. Goncharsky, V.V. Stepanov, A.G. Yagola, Numerical methods for the solution of ill-posed problems, in: Mathematics and its Applications, Vol. 328, Kluwer Academic Publishers, Dordrecht, 1995.

[19] C.W. Ueberhuber, Numerical Computation: Method, Software, and Analysis, Springer, Berlin, Heidelberg, 1997.

[20] E.V. Zakharov, Yu.M. Koptelov, Formulation and numerical solution of inverse problems of electroencephalography, Comput. Math. Model. 3 (2) (1992) 135–142.