# An iterative approach to the solution of an inverse problem in linear elasticity

Abdellatif Ellabib, Abdeljalil Nachaoui

# An iterative approach to the solution of an inverse problem in linear elasticity

A. Ellabib [a], A. Nachaoui [b]

[a] *Université Cadi Ayyad, Faculté des Sciences et Techniques, Département de Mathématiques et Informatique, Avenue Abdelkrim Elkhattabi, B.P 549, Guéliz Marrakech, Maroc.*

[b] *Laboratoire de Mathématiques Jean Leray, UMR 6629, Université de Nantes/CNRS/ECN, 2 rue de la Houssinière, BP 92208, 44322 Nantes, France.*

**Abstract**

This paper presents an iterative alternating algorithm for solving an inverse problem in linear elasticity. A relaxation procedure is developed in order to increase the rate of convergence of the algorithm and two selection criteria for the variable relaxation factors are provided. The boundary element method is used in order to implement numerically the constructing algorithm. We discuss this implementation, mention the use of Krylov methods to solve the obtained linear algebraic systems of equations and investigate the convergence and the stability when the data is perturbed by noise.

*Key words:* Boundary elements, Inverse Problem, Elasticity equations, LU decomposition, Iterative methods.

## 1 Introduction

A vast body of engineering experience shows that the theory of linear elasticity allows an accurate modeling of many natural or manufactured solid materials (civil engineering structures, transportation vehicles, machines, the Earth's mantle, rocks mechanics [9]) and provides an essential tool for analysis and design.

When the governing system of partial differential equations, i.e. the equilibrium, constitutive and kinematics equations, have to be solved with the appropriate initial and boundary conditions for the displacement and/or traction vectors, i.e. Dirichlet, Neumann or mixed boundary conditions the associated problems are called direct problems and their existence and uniqueness have been well established. When one or more of the conditions for solving the direct problem are partially or entirely

*Email addresses:* ellabib@fstg-marrakech.ac.ma (A. Ellabib), nachaoui@math.univ-nantes.fr (A. Nachaoui).

unknown then an inverse problem may be formulated to determine the unknowns from specified or measured system responses.

The main type of inverse problems that arise in the context of linear elasticity, and more generally of the mechanics of deformable solids, are similar to those encountered in other areas of physics involving continuous media and distributed physical quantities, e.g., acoustics, electrostatics and electromagnetism. They are usually motivated by the desire or need to overcome a lack of information concerning the properties of the system (a deformable solid body or structure). It should be noted that most of the inverse problems are ill-posed and hence they are more difficult to solve than the direct problems. It is well known that they are generally instable, i.e. the existence, uniqueness and stability of their solutions are not always guaranteed, see e.g. Hadamard [4]. Identification of inaccessible boundary values (Cauchy problem in elasticity) is a classical example of inverse problem. This inverse problem, in which both displacement and traction boundary conditions are prescribed only on a part of the boundary of the solution domain whilst no information is available on the remaining part of the boundary, can be encountered in many situations [5,16].

Recently, an approximate solution to the Cauchy problem for Poison equation has been determined by one of the authors, [7,11], using an alternating iterative method which reduced the problem to solving a sequence of well-posed boundary value problems. Our goal in this paper is to extend this algorithm in conjunction with the boundary element method (BEM) to the Cauchy problem in elasticity.

The paper is organized as follows. In the next section, we present the direct and an inverse problem for linear elasticity. Then, we give an iterative approach for the Cauchy problem for linear elasticity equation and also we expose two automatic selection of the relaxation factor. We describe in section 4 boundary element method for elasticity equations. In section 5, the technique of implementation of this iterative approach are detailed. Numerical results are presented in section 6 which explore the convergence and stability of algorithm and also we compare some linear iterative method.

## 2 Mathematical model

### *2.1 Direct problem statement*

The mathematical formulation of the 2D elasticity problem in the case of an isotropic linear elastic material which occupies an open bounded domain $\Omega \subset \mathbb{R}^2$ with boundary $\Gamma$ such that $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1, \Gamma_2 \neq \emptyset$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$ is described as follows.

Let $w = (u, v)^T$ be the displacement vector and $b$ the volume force vector. Here $(.,.)^T$ denotes the transpose of a vector or a matrix. Let us define the matrices

$$\mathcal{D} = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 & \dfrac{\partial}{\partial y} \\ 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{pmatrix}, \mathcal{E} = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{2\partial y} & \dfrac{\partial}{2\partial x} \end{pmatrix}, \mathcal{C} = \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix} \tag{1}$$

Then the strain vector $\varepsilon = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12})^T$ is given by

$$\varepsilon = \mathcal{E}w \tag{2}$$

The strain tensor $\varepsilon$ is related to the stress vector $\sigma = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T$ by the constitutive law

$$\sigma = \frac{2G}{1-2\nu}\mathcal{C}\varepsilon \tag{3}$$

where $G$ and $\nu$ are respectively the Shear modulus and Poisson ratio.

The equilibrium equations are given by

$$\mathcal{D}\sigma = b \tag{4}$$

If we now substitute the constitutive law (3) into the equilibrium equation (4), and use the kinematic relations (2) of the elasticity tensor for an isotropic linear elastic material, we obtain the following Lamé system or the Navier equations

$$\begin{cases} G\Delta u + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 v}{\partial x \partial y}\right) = b_1 \text{ in } \Omega \\ G\Delta v + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u}{\partial x \partial y} + \dfrac{\partial^2 v}{\partial y^2}\right) = b_2 \text{ in } \Omega \end{cases} \tag{5}$$

The solution of Eqs. (5) must satisfy prescribed boundary conditions on the boundary $\Gamma$ of the body, which are based either on the displacements $u$ and $v$, or the boundary traction $t$ and $s$. The boundary conditions can be written into the following types

$$u(X) = \tilde{u}(X),\ v(X) = \tilde{v}(X) \text{ for } X \in \Gamma_1 \tag{6}$$

and

$$t(X) = \tilde{t}(X),\ s(X) = \tilde{s}(X) \text{ for } X \in \Gamma_2 \tag{7}$$

where $(t(X), s(X))$ is the traction vector at a point $X \in \Gamma_2$ with $\tilde{u}$, $\tilde{v}$, $\tilde{t}$ and $\tilde{s}$ prescribed quantities.

The knowledge of the geometry (the domain $\Omega$) of the problem, the material constants $G$ and $\nu$ and the prescribed quantities $\tilde{u}$, $\tilde{v}$, $\tilde{t}$ and $\tilde{s}$ enable us to determine the displacement vector $w(x)$ and the strain and the stress tensors in the domain $\Omega$. In this case the problem is called direct problem. Different inverse problems can be considered for this direct problem. In all cases, part of the data which is known for the well posed direct problem is not known. In order to find this unknown data, supplementary information have to be provided.

In this work, we are interested by a reconstruction inverse problem where the geometry of the problem and the material constants are determined, but the boundary conditions are not completely known. This problem arises in cases where a portion of the boundary is exposed to environmental conditions which can not be assessed due to physical difficulties or geometrical inaccessibility. The aim in the reconstruction inverse problem is to find the unknown boundary conditions based on the supplementary data provided on the boundary and/or the domain.

Consider the problem where no conditions are prescribed on $\Gamma_2$ and assume that it is possible to measure the traction vector on $\Gamma_1$. This gives arise to the supplementary boundary conditions

$$t(X) = \tilde{t}(X) \text{ and } s(X) = \tilde{s}(X) \text{ for } X \in \Gamma_1 \tag{8}$$

where $\tilde{t}$ and $\tilde{s}$ are given functions.

In the next section, we describe an iterative method to solve numerically the reconstruction problem (5), (6) and (8), which is ill-posed and cannot be solved efficiently by a direct approach.

## 3 Description of the alternating algorithm

An alternating algorithm for solving Cauchy problems for elliptic equations was introduced by Kozlov et al. [8]. This algorithm was the subject of several studies which addressed various numerical and theoretical aspects (see for example [1,6,7,11]). We extend here this procedure to the reconstruction problem described above. The iterative algorithm investigated is based on reducing this ill-posed problem to a sequence of mixed well-posed boundary value problems and consists of the following steps.

Giving $\omega^0$ and $z^0$, initial approximation of the solution on $\Gamma_2$, we construct a sequence of approximation $u^k, v^k$ by solving alternately the following mixed well-posed direct problems until a prescribed stopping criterion is satisfied.

$u^{2k}$ and $v^{2k}$ are obtained as the solution of

$$\begin{cases} G\Delta u^{2k} + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u^{2k}}{\partial x^2} + \dfrac{\partial^2 v^{2k}}{\partial x \partial y}\right) = b_1 \text{ in } \Omega \\ G\Delta v^{2k} + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u^{2k}}{\partial x \partial y} + \dfrac{\partial^2 v^{2k}}{\partial y^2}\right) = b_2 \text{ in } \Omega \\ t^{2k} = \tilde{t}, \;\; s^{2k} = \tilde{s} \quad \text{on } \Gamma_1 \text{ and } u^{2k} = \omega^k, \;\; v^{2k} = z^k \quad \text{on } \Gamma_2 \end{cases} \tag{9}$$

Having constructed $u^{2k}$ and $v^{2k}$ we can obtain $u^{2k+1}$ and $v^{2k+1}$ by solving the problem

$$\begin{cases} G\Delta u^{2k+1} + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u^{2k+1}}{\partial x^2} + \dfrac{\partial^2 v^{2k+1}}{\partial x \partial y}\right) = b_1 \text{ in } \Omega \\ G\Delta v^{2k+1} + \dfrac{G}{1-2\nu}\left(\dfrac{\partial^2 u^{2k+1}}{\partial x \partial y} + \dfrac{\partial^2 v^{2k+1}}{\partial y^2}\right) = b_2 \text{ in } \Omega \\ u^{2k+1} = \tilde{u}, \;\; v^{2k+1} = \tilde{v} \quad \text{on } \Gamma_1 \text{ and } t^{2k+1} = t^{2k}, \;\; s^{2k+1} = s^{2k} \quad \text{on } \Gamma_2 \end{cases} \tag{10}$$

The sequence $\omega^k$ and $z^k$ are constructed as follows

$$\omega^k = F_1(\omega^{k-1}) \text{ and } z^k = F_2(z^{k-1}) \tag{11}$$

where $F_1$ and $F_2$ are two relaxation operators that will be determined in order to ensure and possibly accelerate the convergence of the iterative procedure. Note that the similar Kozlov-Maz'ya-Fomin's schemes [8] for elasticity problem is obtained by taking $F_1(\omega^{k-1}) = u^{2k-1}$ and $F_2(z^{k-1}) = v^{2k-1}$.

### *3.1 Selection criteria for the relaxation factor based on convex combination*

To solve Cauchy Problems for Poisson equation Nachaoui et al. [7] established a relaxation algorithm by the use of a convex combination of the successive solutions on $\Gamma_2$ which produces a convergent and stable numerical solution.

We extend this idea to the reconstructing algorithm (9), (10). This can be done by defining $F_1$ and $F_2$ in (11) as follows:

$$F_1(\omega^{k-1}) = \theta_1 u^{2k-1}_{|\Gamma_2} + (1-\theta_1)\omega^{k-1} \text{ and } F_2(z^{k-1}) = \theta_2 v^{2k-1}_{|\Gamma_2} + (1-\theta_2) z^{k-1} \tag{12}$$

where $\theta_1$ and $\theta_2$ are two parameters that will be determined in order to ensure and possibly accelerate the convergence of the iterative scheme. Note that the equivalent of Kozlov-Maz'ya-Fomin's schemes [8] for elasticity problem is obtained by taking $\theta_1 = 1$ and $\theta_2 = 1$.

As in [7] the numerical tests performed revealed that the algorithm with constants relaxation factors $\theta_1$ end $\theta_2$ is convergent but there are large variation in the rate of convergence. Therefore we developed selection criteria for the relaxation factors.

Consider that constant relaxation factors are applied in the relaxation algorithm associated to (12) and let $\omega^k$ defined by (11). Note that a good indicator of the level of accuracy achieved is given by the functions

$$\varphi_1(\theta_1) = \|w^k - w^{k-1}\|_{L^2(\Gamma_2)} \text{ and } \varphi_2(\theta_2) = \|z^k - z^{k-1}\|_{L^2(\Gamma_2)},$$

since $\varphi_1$ and $\varphi_2$ tend to zero as the convergence of the iterative algorithm is achieved. Therefore the relaxation factors are selected such that the functions $\varphi_1$ and $\varphi_2$ are minimized. For this we require that $\dfrac{\partial \varphi_1}{\partial \theta_1} = 0$ and $\dfrac{\partial \varphi_2}{\partial \theta_2} = 0$ which yield

$$\theta_1^{k+1} = \frac{\langle e_1^{2k}, e_1^{2k} - e_1^{2k+1} \rangle}{\|e_1^{2k+1} - e_1^{2k}\|^2_{L^2(\Gamma_2)}} \text{ and } \theta_2^{k+1} = \frac{\langle e_2^{2k}, e_2^{2k} - e_2^{2k+1} \rangle}{\|e_2^{2k+1} - e_2^{2k}\|^2_{L^2(\Gamma_2)}} \, \forall\, k \geq 1, \tag{13}$$

where $e_1^{2k} = u^{2k}_{|\Gamma_2} - u^{2k-2}_{|\Gamma_2}$, $e_1^{2k+1} = u^{2k+1}_{|\Gamma_2} - u^{2k-1}_{|\Gamma_2}$, $e_2^{2k} = v^{2k}_{|\Gamma_2} - v^{2k-2}_{|\Gamma_2}$, $e_2^{2k+1} = v^{2k+1}_{|\Gamma_2} - v^{2k-1}_{|\Gamma_2}$, and $\langle \cdot, \cdot \rangle$ denotes the inner product in $L^2(\Gamma_2)$.

Note that the automatic selection of the relaxation factors given in (13) requires 2 inner products at each iteration which are equivalent in discreet form to a number of operations of order $O(N)$ and this is negligible compared to the number of operations needed to solve the two direct problem (9) and (10).

### *3.2 Selection criteria for the relaxation factor based on fixed point operator*

In this section we develop a second relaxation scheme based on some least-residual strategy. Let $F_1$ and $F_2$ be the mappings from $L^2(\Gamma_2)$ to $L^2(\Gamma_2)$ defined by solving successively the problems (9), (10) and taking $F_1(\omega^k) = u^{2k+1}_{|\Gamma_2}$, $F_2(z^k) = v^{2k+1}_{|\Gamma_2}$.

Let $L_1$ and $L_2$ be two linear operator from $L^2(\Gamma_2)$ to $L^2(\Gamma_2)$ defined as follows. For any $w_1$, $w_2 \in L^2(\Gamma_2)$, $L_1 w_1$ and $L_2 w_2$ are the solutions respectively of the two well posed linear problems (9) and (10) where $w_1$ and $w_2$ play respectively the role of $\omega^k$ and $z^k$ but with the volume force equal to zero and homogeneous boundary conditions on $\Gamma_1$. Let $w_n$ and $w_d$ computed by solving respectively the two well posed linear problems (9) and (10) with homogeneous boundary conditions on $\Gamma_2$. This implies that $F_1$ and $F_2$ can be written as:

$$F_1(\omega) = L_1\omega + w_n \text{ and } F_2(z) = L_2 z + w_d. \tag{14}$$

Then the marching condition for the displacements on $\Gamma_2$ in (9) can be relaxed as follows

$$\omega^{k+1} = \omega^k + \theta_1(L_1\omega^k + w_n - \omega^k) \text{ and } z^{k+1} = z^k + \theta_2(L_2 z^k + w_d - z^k). \tag{15}$$

Let us define the following vectors

$$r_1^k = L_1\omega^k + w_n - \omega^k,\ r_2^k = L_2 z^k + w_d - z^k, \tag{16}$$

$$\omega^k(\theta_1) = \omega^k + \theta_1 r_1^k,\ z^k(\theta_2) = z^k + \theta_2 r_2^k, \tag{17}$$

$$r_1^k(\theta_1) = L_1\omega^k(\theta_1) + w_n - \omega^k(\theta_1),\ r_2^k(\theta_2) = L_2 z^k(\theta_2) + w_d - z^k(\theta_2). \tag{18}$$

Note that here a good indicator of the level of accuracy achieved is given by the functions

$$\phi_1(\theta_1) = \|r_1^k(\theta_1)\|_{L^2(\Gamma_2)} \quad \text{and } \phi_2(\theta_2) = \|r_2^k(\theta_2)\|_{L^2(\Gamma_2)} \tag{19}$$

since $\phi_1$ and $\phi_2$ tend to zero as the convergence of the iterative algorithm is achieved. Therefore the relaxation factors are selected such that the functions $\phi_1$ and $\phi_2$ are minimized.

From (16), (17) and (18) we obtain

$$r_1^k(\theta_1) = r_1^k + \theta_1(L_1 r_1^k - r_1^k) \text{ and } r_2^k(\theta_2) = r_2^k + \theta_2(L_2 r_2^k - r_2^k). \tag{20}$$

Then $\phi_1$ and $\phi_2$ can be written as follows

$$\phi_1(\theta_1) = \|r_1^k\|^2_{L^2(\Gamma_2)} + 2\theta_1\langle r_1^k, L_1 r_1^k - r_1^k\rangle + \theta_1^2\|L_1 r_1^k - r_1^k\|^2_{L^2(\Gamma_2)} \tag{21}$$

$$\phi_1(\theta_1) = \|r_2^k\|^2_{L^2(\Gamma_2)} + 2\theta_2\langle r_2^k, L_1 r_2^k - r_2^k\rangle + \theta_2^2\|L_2 r_2^k - r_2^k\|^2_{L^2(\Gamma_2)} \tag{22}$$

For the functions $\phi_1$ and $\phi_2$ to be minimized we require that $\dfrac{\partial\phi_1}{\partial\theta_1} = 0$ and $\dfrac{\partial\phi_2}{\partial\theta_2} = 0$ which yield

$$\theta_1^k = \frac{\langle r_1^k, r_1^k - L_1 r_1^k\rangle}{\|L_1 r_1^k - r_1^k\|^2_{L^2(\Gamma_2)}} \text{ and } \theta_2^k = \frac{\langle r_2^k, r_2^k - L_2 r_2^k\rangle}{\|L_2 r_2^k - r_2^k\|^2_{L^2(\Gamma_2)}}. \tag{23}$$

Thus the automatic adjustment of $\omega^{k+1}$ and $z^{k+1}$ is obtained as follows

$$\omega^{k+1} = \omega^k + \theta_1^k r_1^k \text{ and } z^{k+1} = z^k + \theta_2^k r_2^k, \tag{24}$$

where $\theta_1^k$ and $\theta_2^k$ are given by (23).

Note that this scheme requires the solution of the two well posed problems (9) and (10) and the computation of $L_1 r_1^k$ and $L_2 r_2^k$ which are equivalent in the discreet form to matrix-vector product.

The boundary element method is a very apt tool to solve the auxiliary problems (9) and (10), since the boundary conditions are the main unknown of the problem and the statement of these problems in term of boundary integral equation reduces the modeling effort to a minimum. Moreover, the BEM determines simultaneously the boundary displacement $u$, $v$ and its traction $t$, $s$, this allows us to solve problem (10) without the need or further finite difference, as one would employ if using the finite element or the finite difference method.

We describe the boundary element method in the next section.

## 4 Integral equation formulation and boundary element

### *4.1 Integral equation formulation*

The linear elasticity problem (5) in two-dimensional case can be formulated in integral form [2] as follows

$$\begin{aligned}\int_\Gamma U_{ij}(P,Q)\{\mathcal{T}\}_j(Q)\,d\Gamma - \int_\Gamma T_{ij}(P,Q)\{\mathcal{U}\}_j(Q)\,d\Gamma + \int_\Omega U_{ij}(P,Q)b_j(Q)d\Omega \\ = \begin{cases} \{\mathcal{U}\}_i(P) \text{ if } P\in\Omega \\ \dfrac{1}{2}\{\mathcal{U}\}_i(P) \text{ if } P\in\Gamma \end{cases}\end{aligned} \tag{25}$$

for $i,j=1,2$, where $U_{ij}$ and $T_{ij}$ denote the fundamental displacements and traction for the two-dimensional isotropic linear elasticity [2] and they are given by

$$\begin{aligned} U_{ij}(P,Q) &= \frac{1}{8\pi G(1-\nu)}\left[\delta_{ij}(4\nu-3)\ln r + \frac{r_{,i}r_{,j}}{r^2}\right] \\ T_{ij}(P,Q) &= \frac{1}{4\pi(1-\nu)r}\left[\frac{\partial r}{\partial n}\left\{(2\nu-1)\delta_{ij} - \frac{2r_{,i}r_{,j}}{r^2}\right\} + (2\nu-1)\frac{n_i r_{,j} - n_j r_{,i}}{r}\right] \end{aligned} \tag{26}$$

where $r$ is the distance between the source point $P$ and field point $Q$, $n=(n_1,n_2)$ denotes the outer normal vector to $\Gamma$, $r_{,i}=Q_i-P_i$.

### *4.2 Boundary element method for elasticity equations*

The boundary integral equations (25) are solved using boundary element method with constant boundary elements. The boundary is divided into $N$ constant elements. Thus the distribution of the displacements and traction are taken constant on each element and equal to their value at the nodal point, which lies at the midpoint of the element. Denoting by $\{\mathcal{U}\}^i=\{u^i,v^i\}^T$ and $\{\mathcal{T}\}^i=\{t^i,s^i\}^T$ the displacements and traction at the $i^{th}$ node and taking into account that the boundary is smooth at the nodal point of the constant element. Then, the discretized form of Eq. (25) can be written as

$$\frac{1}{2}\{\mathcal{U}\}^i + \sum_{j=1}^{N}\hat{H}^{ij}\{\mathcal{U}\}^j = \sum_{j=1}^{N}G^{ij}\{\mathcal{T}\}^j + \mathcal{F} \tag{27}$$

where $G^{ij}$ and $\hat{H}^{ij}$ are $2\times 2$ matrices such that

$$(G^{ij})_{lm} = \int_{\Gamma_j} U_{lm}(P^i,Q)\,d\Gamma(Q) \text{ and } (\hat{H}^{ij})_{lm} = \int_{\Gamma_j} T_{lm}(P^i,Q)\,d\Gamma(Q) \text{ for } l,m=1,2. \tag{28}$$

Eq. (27) relates the displacements of the $i^{th}$ node to the displacements and the traction of all the nodes including the $i^{th}$ node. Applying this equation to all the boundary nodal points yields $2N$ equations, which can be set in matrix form as $H\,\mathcal{U} = G\,\mathcal{T} + \mathcal{F}$ where $H = \hat{H} + \frac{1}{2}I$ and $I$ is $2N \times 2N$ identity matrix. The dimension of matrices $\hat{H}$ and $G$ is $2N \times 2N$, and those the vectors $\mathcal{U}$ and $\mathcal{T}$ is $2N$. They are defined as $G = (G^{ij})_{1\leq i,j\leq N}$, $\hat{H} = (\hat{H}^{ij})_{1\leq i,j\leq N}$ and $\mathcal{U} = \left(\{\mathcal{U}\}^1, \cdots, \{\mathcal{U}\}^N\right)$, $\mathcal{T} = \left(\{\mathcal{T}\}^1, \cdots, \{\mathcal{T}\}^N\right)$. The $2N$ equations within the matrix Eq. (27) contain $4N$ boundary values, that is $2N$ values of displacements and $2N$ values of traction. However, a total of $2N$ values are known from the boundary conditions. Consequently, Eq. (27) can be used to determine the $2N$ unknown boundary values.

It should be noted that rearrangement of the unknowns is necessary for mixed boundary conditions. After doing so the following system of $2N$ linear equations is obtained $\mathcal{A}\mathcal{X} = \mathcal{R} + \mathcal{F}$ where $\mathcal{A}$ is a square coefficient matrix having dimensions $2N \times 2N$, $\mathcal{R}$ is a vector resulting as the sum of the columns of the matrices $G$ and $H$ multiplied by the respective known boundary values. The main inconvenient with integral formulations is that its time consuming. To make the proposed method more appealing, effort must be devoted to the implementation of algorithm (9)-(10) with (12) or (24) and in particular to solving large dense linear systems efficiently. In general, the following factors are considered in choosing an implementation technique :
1. The accuracy of the calculation, or the quality of the approximations;
2. The computational effort involved, or the efficiency of the method; and
3. The ease-of-implementation

We have tested iterative Krylov methods, method based on LU factorization and the exploitation of the nature of algorithm (9)-(10).

## 5 Implementation

The resulting systems of equations, when the boundary element discretization is applied to the reconstructing algorithm (9)-(10), will be of the form

$$\mathcal{A}_1^k \mathcal{X}^{2k} = \mathcal{B}_1^k \tag{29}$$

$$\mathcal{A}_2^k \mathcal{X}^{2k+1} = \mathcal{B}_2^k \tag{30}$$

where $\mathcal{A}_i^k$ and $\mathcal{B}_i^k$, $i = 1, 2$ are constructed from the discreet form of (9) and (10) by boundary element method. Note that, from Eq. (28), $\mathcal{A}_1^k$ and $\mathcal{A}_2^k$ are geometry dependent matrices and depend on the type of the boundary conditions, but not on their values. Therefore $\mathcal{A}_1^k = \mathcal{A}_1^0$ and $\mathcal{A}_2^k = \mathcal{A}_2^0$. These matrices can have the factored form : $\mathcal{A}_1^0 = \mathcal{L}_1^0\mathcal{R}_1^0$, $\mathcal{A}_2^0 = \mathcal{L}_2^0\mathcal{R}_2^0$ where $\mathcal{L}_1^0$, $\mathcal{L}_2^0$ are lower triangular matrices and $\mathcal{R}_1^0$, $\mathcal{R}_2^0$ are upper triangular matrices. Now from (29) and (30), $\mathcal{X}^{2k}$ and $\mathcal{X}^{2k+1}$ can be obtained by backward followed by forward substitutions requiring only $4N^2$ operation. This gives arise to the following algorithm :

**Algorithm 1**
*1. set* $k = 0$ *and choose the initial estimate* $(\omega^0, z^0)$
*2. Compute* $\mathcal{H}$ *and* $\mathcal{G}$
*3. Compute* $\mathcal{A}_1^0$, $\mathcal{B}_1^0$, $\mathcal{A}_2^0$ *and* $\mathcal{B}_2^0$
*4. Compute* $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{R}_1$*and* $\mathcal{R}_2$
*5. Solve systems (29) and (30) replacing* $\mathcal{A}_1^0$ *and* $\mathcal{A}_2^0$ *by their factorized forms*
*6. until convergence do*
*7.* $k = k + 1$*, compute* $\omega^k$, $z^k$ *and* $\mathcal{B}_1^k$
*8. replace* $\mathcal{A}_1^k$ *by* $\mathcal{L}_1\mathcal{R}_1$ *and solve (29)*
*9. compute* $\mathcal{B}_2^k$*, replace* $\mathcal{A}_2^k$ *by* $\mathcal{L}_2\mathcal{R}_2$ *and solve (30)*
*10. End do.*

The efficiency of the proposed method is illustrated for different orders of system. The efficiency is compared based on the CPU times carried out using the Gaussian elimination method at each iteration.

Due to the properties of matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ (are nonsingular and non-symmetric), one might consider solving $\mathcal{A}\mathcal{X} = \mathcal{B}$ by applying CG to the normal equations $\mathcal{A}^T\mathcal{A}\mathcal{X} = \mathcal{A}^T\mathcal{B}$. This approach is called CGNR [3].

Alternatively, one could solve $\mathcal{A}\mathcal{A}^T\mathcal{Y} = \mathcal{B}$ and then set $\mathcal{X} = \mathcal{A}^T\mathcal{Y}$ to solve $\mathcal{A}\mathcal{X} = \mathcal{B}$. This approach is now called CGNE [3].

There are three disadvantages that may or may not be serious. The first is that the condition number of the coefficient matrix $\mathcal{A}^T\mathcal{A}$ is the square of that of $\mathcal{A}$. The second is that two matrix-vector products are needed for each CG iterate. The third, more important, disadvantage is that one must compute the action of $\mathcal{A}^T$ on a vector as part of the matrix-vector product involving $\mathcal{A}^T\mathcal{A}$.

We used the bi-conjugate gradient stabilized (BICGSTAB) [15], which was developed to have the same convergence rate as the conjugate gradient squared (CGS) [12] at its best, without having the same difficulties (irregular convergence behavior). An advantage of BI-CGSTAB over other Krylov method such as the generalized minimal residual method (GMRES)[13] is that it has limited computation and storage requirement in each iteration step. Comparison of these methods can be found in [10,14].

An alternative to Algorithm 1 is an implementation of the reconstructing algorithm (9) and (10) where all the linear system are solved by an iterative solver for each iteration. The proposed method summarized in the next algorithm :

**Algorithm 2**
*1. set* $k = 0$ *and choose the initial estimate* $(\omega^0, z^0)$ *and a tolerance for the iterative solver*
*2. Compute* $\mathcal{H}$ *and* $\mathcal{G}$
*3. Compute* $\mathcal{A}_1^0$, $\mathcal{B}_1^0$, $\mathcal{A}_2^0$ *and* $\mathcal{B}_2^0$
*4. Solve systems (29) and (30) using an iterative solver*
*5.* $k = k + 1$
*6. compute* $w^k$, $z^k$, $\mathcal{B}_1^k$
*7. solve* $\mathcal{A}_1^0\mathcal{X}^{2k} = \mathcal{B}_1^k$ *using an iterative solver*

*8. compute $\mathcal{B}_2^k$ and solve $\mathcal{A}_2^0 \mathcal{X}^{2k+1} = \mathcal{B}_2^k$ using an iterative solver*
*9. repeat steps 5-8 until convergence*
*10. End do.*

The following stopping criterion can be used for iterative solvers of linear systems

$$\frac{\|\mathcal{B} - \mathcal{A}\mathcal{X}\|_2}{\|\mathcal{B}\|_2} < \alpha \tag{31}$$

Note that to compare Algorithm 2 with Algorithm 1, we take the same tolerance $\alpha$ for all $k$ but this is not necessary for the convergence of Algorithm 2. Note also that Algorithm 2 converges when $\alpha$ is not very small.

## 6 Numerical results

In order to illustrate the performance of the numerical method described above, we solve the inverse elasticity problem (5), (6) and (8) in two-dimensional annular domain $\Omega$ given by

$$\Omega = \left\{(x, y) \in \mathbb{R}^2,\, 1 < x^2 + y^2 < 16\right\}.$$

We assume that the boundary is split into two parts

$$\Gamma_1 = \left\{(x, y) \in \mathbb{R}^2,\, x^2 + y^2 = 16\right\} \text{ and } \Gamma_2 = \left\{(x, y) \in \mathbb{R}^2,\, x^2 + y^2 = 1\right\}.$$

The exact solution of the direct problem is given by

$$\begin{aligned} u(x, y) &= \frac{1}{2G(1+\nu)} \left( V(1-\nu)x - W(1+\nu)\frac{x}{x^2+y^2} \right) \\ v(x, y) &= \frac{1}{2G(1+\nu)} \left( V(1-\nu)y - W(1+\nu)\frac{y}{x^2+y^2} \right) \end{aligned} \tag{32}$$

where $V = -\dfrac{\sigma_0 - 16\sigma_1}{15}$ and $W = \dfrac{16(\sigma_0 - \sigma_1)}{15}$ and stress tensor is given by

$$\begin{aligned} &\sigma_{11}(x, y) = V + W\frac{x^2 - y^2}{(x^2+y^2)^2}, \quad \sigma_{22}(x, y) = V - W\frac{x^2 - y^2}{(x^2+y^2)^2} \\ &\text{and } \sigma_{12}(x, y) = 2W\frac{xy}{(x^2+y^2)^2} \end{aligned} \tag{33}$$

with $\sigma_1 = 10^{10}$, $\sigma_0 = 2 \times 10^{10}$, $G = 3.35 \times 10^{10}$ and $\nu = 0.34$.

The auxiliary problems (9) and (10) corresponding to this example are discretized by boundary element method using a piecewise constant polynomial interpolation. The number of boundary elements

used for the discretization of the boundary $\Gamma$ is taken to be $N \in \{80, 160, 320, 640, 1280\}$. We denote by $\|\cdot\|_{0,\Gamma_2}$ the discreet $L^2$ norm defined on $\Gamma_2$.

The convergence of the algorithm may be investigated by evaluating at every iteration the error

$$G_k^u = \|u^{2k} - u^{an}\|_{0,\Gamma_2},\ G_k^v = \|v^{2k} - v^{an}\|_{0,\Gamma_2} \tag{34}$$

where $u^{2k}$ and $v^{2k}$ are the numerical displacements on the boundary $\Gamma_2$ obtained after $k$ iterations and $u^{an}$, $v^{an}$ are the exact displacements of the problem given by (32). In a similar way we may evaluate the errors in retrieving the traction on the boundary $\Gamma_2$ given by

$$G_k^t = \|t^{2k+1} - t^{an}\|_{0,\Gamma_2},\ G_k^s = \|s^{2k+1} - s^{an}\|_{0,\Gamma_2} \tag{35}$$

The behavior of the method is investigated by evaluation the difference between two consecutive approximations for the displacements solutions and its traction on the boundary $\Gamma_2$ given by

$$E_k^u = \|u^{2k} - u^{2k-2}\|_{0,\Gamma_2}^2/N_2,\ E_k^v = \|v^{2k} - v^{2k-2}\|_{0,\Gamma_2}^2/N_2 \tag{36}$$

$$E_k^t = \|t^{2k+1} - t^{2k-1}\|_{0,\Gamma_2}^2/N_2,\ E_k^s = \|s^{2k+1} - s^{2k-1}\|_{0,\Gamma_2}^2/N_2 \tag{37}$$

Based on absolute errors the following stopping criterion is considered

$$\max(E_k^u, E_k^v) < \eta \tag{38}$$

where $\eta$ is a small prescribed positive quantity. Note that (38) express that the sequence $(u^{2k}, v^{2k})$ converge in Sobolev spaces $H^{\frac{1}{2}}(\Gamma_2) \times H^{\frac{1}{2}}(\Gamma_2)$. For all numerical experiments, we take $\eta = 10^{-7}$.

This test is used to analyze the behavior with respect to accuracy and efficiency of the techniques considered in this work when applied to the approximation solution of boundary element method systems of algebraic equations.

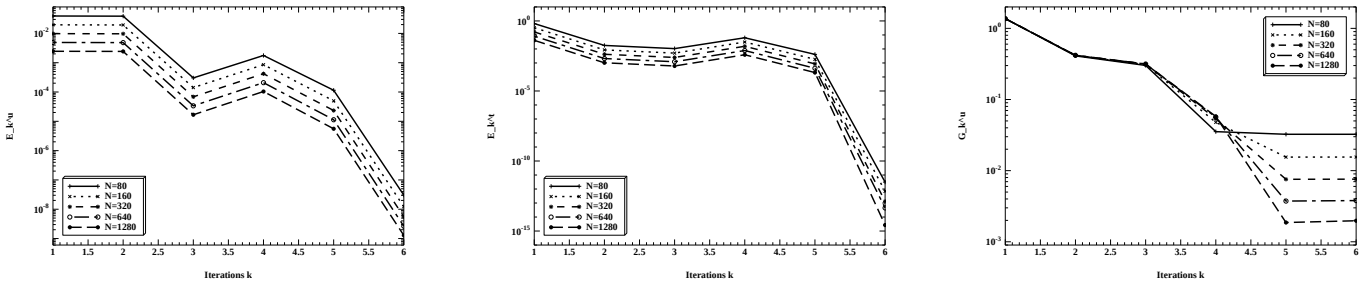
### *6.1 Comparative result for various parameter relaxation*

Table 1 presents results obtained based on Algorithm 1 for various number of boundary elements. We denote by $k_1$, $k_2$ and $k_3$ respectively the number of iterations required to achieve the convergence using $\theta_1 = \theta_2 = 1$, $\theta_1$, $\theta_2$ computed by (13) and $\theta_1$, $\theta_2$ computed by (23) respectively. We denote by $CPU_1$, $CPU_2$, $CPU_3$ the CPU time required for the convergence in the three cases.

We observe from Table 1 that the reconstructing algorithm is very efficient when used with the automatic adjustment of $\theta_1$, $\theta_2$ given by (13) or (23).

Table 1
CPU time for fixed parameter, dynamically estimated parameter by (13) and (23) using Algorithm 1.

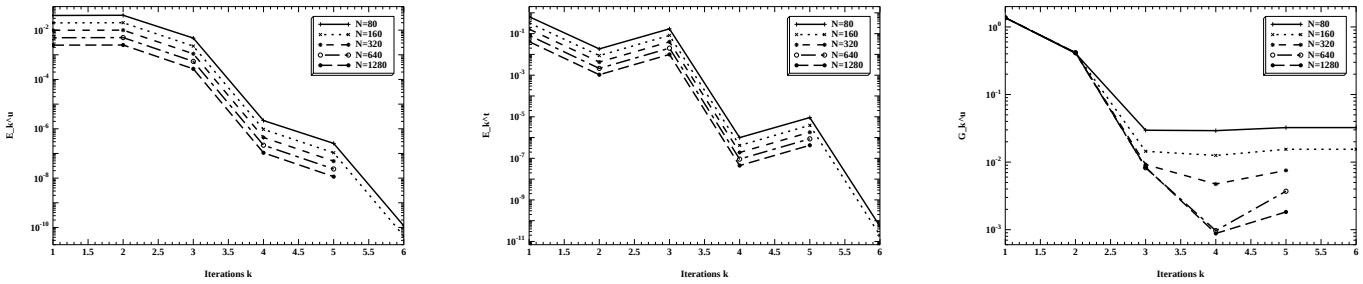
| N | $CPU_1$ | $CPU_2$ | $CPU_3$ | $k_1$ | $k_2$ | $k_3$ |
|---|---|---|---|---|---|---|
| 80 | 0.07199003 | 0.05799199 | 0.06399098 | 18 | 6 | 6 |
| 160 | 0.48792499 | 0.44793200 | 0.47592701 | 16 | 6 | 6 |
| 320 | 14.6957663 | 13.953879 | 14.0178692 | 15 | 6 | 5 |
| 640 | 224.173124 | 122.282414 | 122.070447 | 14 | 6 | 5 |
| 1280 | 1386.6820 | 907.992332 | 913.368518 | 13 | 6 | 5 |




(a) (b) (c)

Fig. 1. $E_k^u$ (a), $E_k^t$ (b) and $G_k^u$ (c) of first dynamical choice of relaxation parameter for different $N$.

The behavior of numerical solution of the first components of the displacement and traction vectors ($u$ and $t$) are similar to that of the second components ($v$ and $s$ respectively) and therefore they have not been presented here.

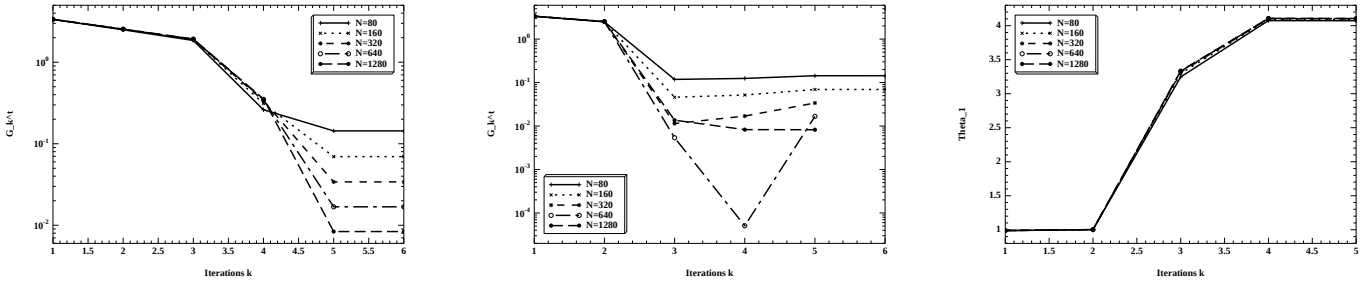



(a) (b) (c)

Fig. 2. $E_k^u$ (a), $E_k^t$ (b) and $G_k^u$ (c) of second dynamical choice of relaxation parameter for different $N$.

Fig.1(a),(b)-2(a),(b) show, on a semi-log scale, the corresponding successive difference $E_k^u$ and $E_k^t$ for automatic selection of $\theta_1$, $\theta_2$ given by (13) or (23) as functions of the number of iterations $k$.

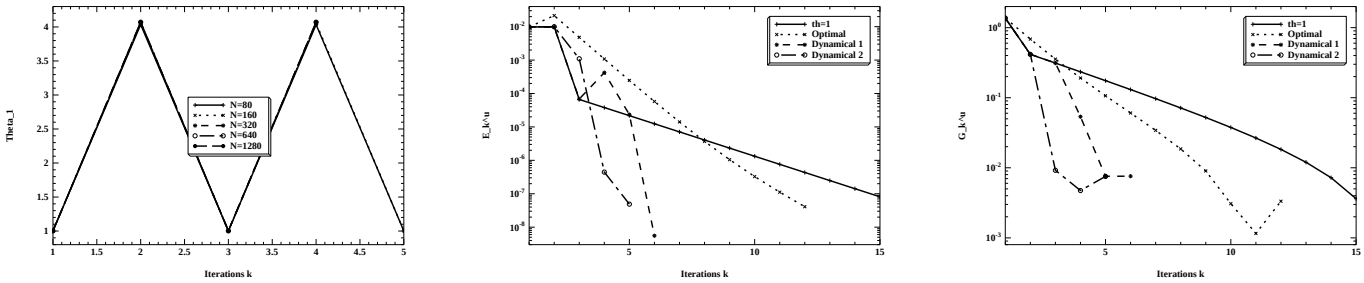
Fig.1 (c), 2 (c), 3 (a)-(b) show, on a semi-log scale, the corresponding sequences $G_k^u$ and $G_k^t$ for automatic selection of $\theta_1$, $\theta_2$ given by (13) or (23) as functions of the number of iterations $k$ for various number of boundary elements $N$. We can see easily that the quantities $G_k^u$ and $G_k^t$ decrease when $N$ increases and they remain constant after a few iteration. Therefore the method are stable with respect to the number of boundary element.




(a) (b) (c)

Fig. 3. $G_k^t$ of two dynamical choice of relaxation parameter (a), (b) for different $N$ and variation of first dynamical estimated relaxation parameter (c) as a function of number of iteration.

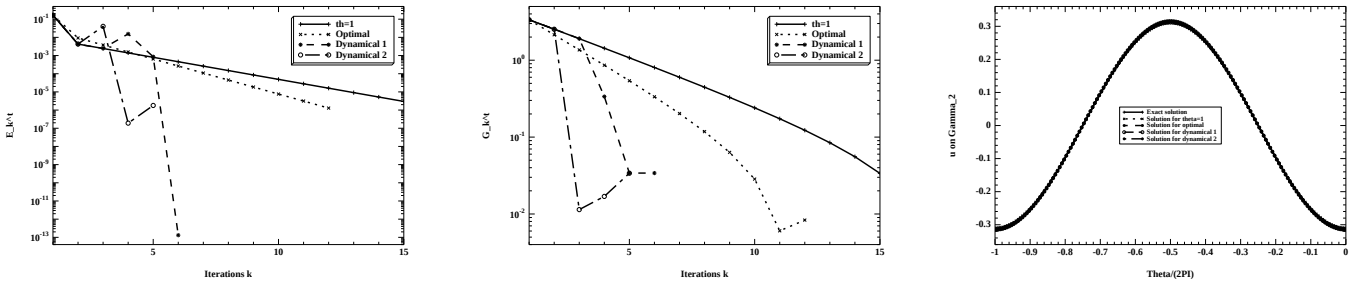
According to these results, we observe a weak oscillation of $E_k^u$ and $E_k^t$ due to the automatic search of the relaxation parameter at each iteration, see Fig. 3(c), 4 (a). This slight instability in $E_k^u$ and $E_k^t$ does not affects the behavior of the errors $G_k^u$ and $G_k^t$ as it is illustrated in Fig. 1 (c), 2 (c), 3 (a) - (b).




(a) (b) (c)

Fig. 4. Variation of second dynamical estimated relaxation parameter (a) as a function of number of iteration and $E_k^u$ (b), $G_k^u$ (c) of four relaxation parameter.

For $N = 320$, we observe from Fig.4 (b), (c), 5 (a), (b) that when the algorithm is implemented with the dynamically estimated relaxation parameter given by (13) or (23), the prescribed criterion (36) is satisfied after 6 iterations for automatic selection given by (13), after 5 iterations for automatic selection given by (23), while it is required 15 iterations for $\theta_1 = \theta_2 = 1$. With the fixed optimal parameter the convergence is obtained after 13 iterations.



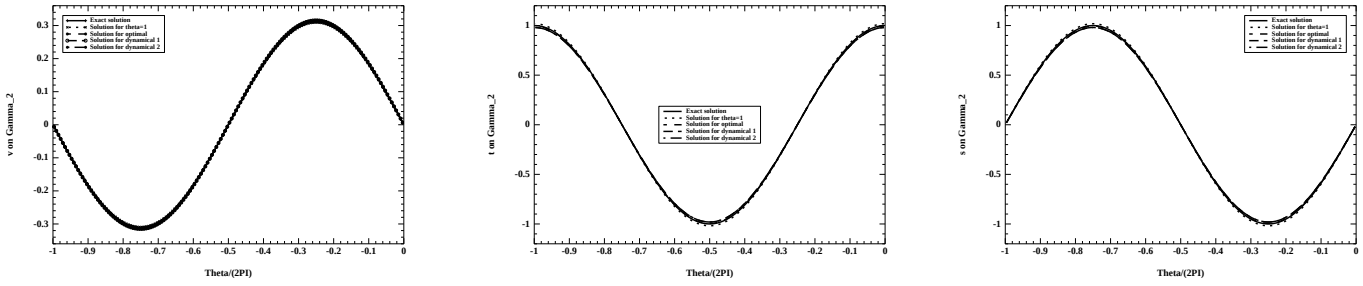

(a) (b) (c)

Fig. 5. $E_k^t$ (a), $G_k^u$ (b) of four relaxation parameter and numerical, exact solution for displacement $u$ (c) with $N = 320$.

We notice that at the convergence, the same precision is reached for the displacements solutions and traction with the different relaxation parameters, see Fig. 5 (c) and Fig. 6. These figures show the

numerical solutions and their tractions obtained on the boundary $\Gamma_2$ for four relaxation parameters, namely the two dynamically estimated parameter given by (13) or (23), the fixed optimal parameter and the fixed parameter $\theta_1 = \theta_2 = 1$.

As it can be seen from Fig. 5 (c) - 6 the displacement and traction solutions corresponding to 320 of boundary elements is in good agreement with the analytical solution.

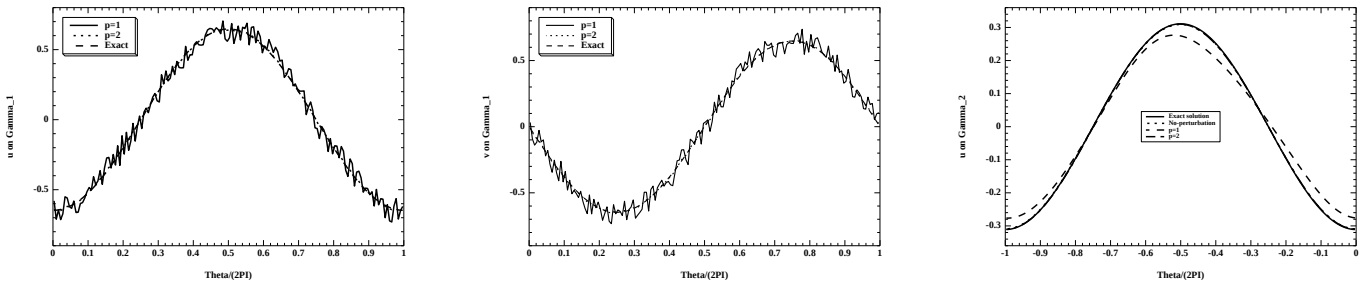



(a) (b) (c)

Fig. 6. Numerical and exact solutions for displacement $v$ (a) and tractions $t$ (b), $s$ (c) for $N = 320$

### *6.2 The effect of noise*

We discuss the numerical algorithm stability according to disturbed displacement and tractions data. For this we add the following quantity $10^{-p}(2\,rand() - 1)$ to the given data on $\Gamma_1$, where $rand$ is the FORTRAN random function.

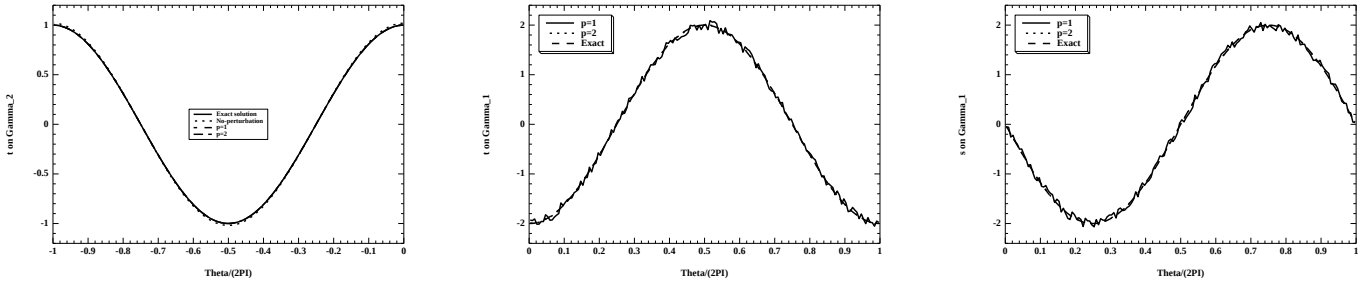



(a) (b) (c)

Fig. 7. Perturbed displacements $u$ (a), $v$ (b) data on $\Gamma_1$ and the corresponding numerical displacement $u$ (c) results with $N = 320$

#### *6.2.1 Noisy on the displacements data*

We examine the algorithm stability when $u$ and $v$ data are noisy. The noisy data on displacements are presented in Fig. 7 (a), (b) and the corresponding numerical solutions are shown in Fig. 7 (c) and Fig. 8 (a).




(a) (b) (c)

Fig. 8. Numerical traction $t$ (a) results with perturbed displacements data with $N = 320$ and perturbed tractions $t$ (b), $s$ (c) data on $\Gamma_1$.

*6.2.2 Noisy on the tractions data*

We examine the algorithm stability when $t$ and $s$ data are noisy. The noisy data on tractions are presented in Fig. 8 (b), (c) and the corresponding numerical solutions are shown in Fig. 9.

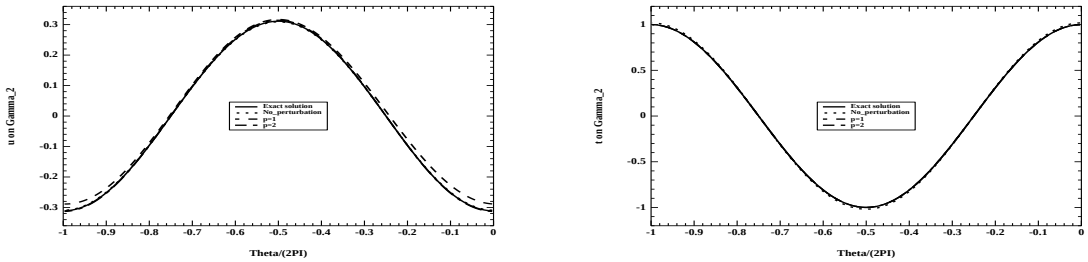



Fig. 9. Numerical displacement $u$ (left) traction $t$ (right) results obtained for perturbed tractions data and $N = 320$.

For $p \geq 3$, 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 (5), (6) and (8), is an ill-posed problem with a very oscillatory data.

*6.3 Comparison of LU decomposition and linear iterative solvers*

To compare the efficiency for different iterative solvers, five computational meshes are used to generate five systems of 160, 320, 640, 1280, 2560 linear equations.

The iterative solvers BI-CGSTAB, CGNE, CGNR, CGR and CGS, are considered.

Table 2-3 summarizes the results obtained for these solvers. The times shown are the average time in CPU times. The average number of iterative solver iterations is given in brackets. We used $\alpha = 10^{-07}$

Table 2
CPU time for different solvers (first iteration) for dynamical 1

| N | LU | BI-CGSTAB | CGNE | CGNR | CGR | CGS |
|---|---|---|---|---|---|---|
| 80 | 0.0579 | 0.0619(83) | 0.0709(99) | 0.0689(92) | 0.1549(93) | 0.0699(97) |
| 160 | 0.4479 | 0.2899(75) | 0.3239(101) | 0.3239(94) | 0.6499(77) | 0.3549(98) |
| 320 | 13.953 | 4.4043(83) | 13.510(432) | 7.7138(243) | 13.053(89) | 5.0732(93) |
| 640 | 122.28 | 21.042(85) | 85.438(611) | 32.835(227) | 50.697(78) | 23.798(97) |
| 1280 | 907.99 | 84.896(90) | 1054.1(1964) | 388.05(719) | 221.24(85) | 88.798(94) |

Table 3
CPU time for different solvers (first iteration) for dynamical 2

| N | LU | BI-CGSTAB | CGNE | CGNR | CGR | CGS |
|---|---|---|---|---|---|---|
| 80 | 0.06 | 0.12(99;177) | 0.15(98;163) | 0.12(91;108) | 0.28(89;84) | 0.11(90;96) |
| 160 | 0.47 | 0.61(79;237) | 1.10(93;330) | 0.66(93;146) | 1.33(76;80) | 0.64(86;106) |
| 320 | 14.0 | 7.13(72;83) | 19.1(305;297) | 10.7(115;230) | 18.2(72;58) | 7.66(81;71) |
| 640 | 122 | 29.2(68;55) | 149(440;636) | 50.4(88;267) | 84.3(69;58) | 36.5(77;75) |
| 1280 | 913 | 129(69;87) | 1117(718;1362) | 448(120;712) | 321(69;58) | 137(73;61) |

as convergence tolerance in (31).

All the Krylov methods worked well, some, BI-CGSTAB and CGS, performed very well, yielding the solution in a small number of iterations. CGNE and CGNR revealed some difficulty in achieving a good convergence. The results show that BI-CGSTAB and CGS are more efficient than LU decomposition procedure, but BI-CGSTAB is by far the fast solver. The normalizing technique corresponding to the CGNE method is penalized due to the worsening of the condition number of their matrix relative to the condition number of the matrix of the original linear system.

## 7 Conclusion

In this paper a numerical iterative boundary element method for solving Cauchy problem in linear elasticity equations has been developed. It can be concluded that the iterative approach produces an accurate, convergent and stable numerical solution with respect to increasing the number of boundary elements and decreasing the amount of noise. The stability of method was also investigated by perturbing the given data with various levels of noise. Following are the main conclusions drawn from the numerical experiments :

- The accuracy of iterative approach is improved by the use of automatic selection of relaxation parameter.
- The proposed iterative approach could be improved by using Krylov methods as linear solver.

- When the iterative approach is combined with the automatic selection of accelerating parameter, BI-CGSTAB is to be preferred as a solver of the linear systems appearing in iterative process.

## Acknowledgements

This work was carried out in the framework of the project 'Inverse problems and shape optimization' when the first author was a visiting Professor at Laboratoire de Mathématiques Jean Leray, Université de Nantes. The research of the first author was supported in part by 'Action Intégrée' France-Maroc MA/05/116.

## References

[1] J. Baumeister, A. Leitao, Iterative methods for ill-posed problems modeled by partial differential equations, Journal of Inverse and ill-posed problems, Journal of Inverse and ill-posed problems 9 (1) (2001) 1-17.

[2] C. A. Brebbia, J. Dominguez, Boundary Elements An Introductory course, Comp. Mech. Pub. McGraw-Hill Book Company, 1992.

[3] A. Greenbaum, Iterative methods for solving linear system, in: Frontiers in Applied Mathematics, Vol. 17, SIAM, Philadelphia, PA, 1997.

[4] J. Hadamard, Lectures on Cauchy problem in linear partial differential equations, London Oxford University Press, 1923.

[5] C. H. Huang, W. Y. Shih, An inverse problem in estimating interfacial cracks in bimaterials by boundary element technique, International Journal for Numerical Methods in Engineering 45 11 (1999) 1547-1567.

[6] M. Jourhmane, A. Nachaoui, A relaxation algorithm for solving a Cauchy problem, in: Proceedings of the second international conferences on inverse problems in engineering, Engineering Foundation, 1996.

[7] M. Jourhmane, A. Nachaoui, Convergence of an alternating method to solve Cauchy problem for Poisson's equation, Appl. Analysis 81 (2002) 1065-1083.

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

[9] B. Loret, N. Khalili, An effective stress elastic-plastic model for unsaturated porous media, Mechanics of Materials Volume 34 Issue 2 (2002) 97-116.

[10] A. Nachaoui, Iterative solution of the drift-diffusion equations, Numer. Algorithms 21 (1-4) (1999) 323-341.

[11] A. Nachaoui, Numerical linear algebra for reconstruction inverse problems, Journal of Computational and Applied Mathematics Vol. 162 1 (2004) 147-164.

[12] P. Sonneveld, CGS, a fast Lanczos-type solver for non-symmetric linear systems, SIAM J. Sci. Statist. Comput. 10 (1) (1989) 36-52.

[13] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856-869.

[14] H. A. Van der Vorst, Modern methods for the iterative solution of large systems of linear equations, Neuw Arch. Wiskd. IV Ser. 14 No. 1 (1996) 127-143.

[15] H. A. Van der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 No. 2 (1992) 631-644.

[16] W. C. Yeih, T. Koya, T. Mura, An inverse problem in elasticity with partially overspecified boundary conditions, I Theoretical approach, Transactions ASME Journal of Applied Mechanics 60 (1993) 595-600.