
A robust data completion method for two dimensional Cauchy problems associated with the Laplace equation

Franck Delvare* — Alain Cimetière**

* *Laboratoire PRISME, ENSI de Bourges*
88, Boulevard Lahitolle, F-18020 Bourges cedex
franck.delvare@ensi-bourges.fr

** *Institut Pprime, CNRS-ENSMA-Université de Poitiers, UPR 3346*
Boulevard Marie et Pierre Curie, BP 30179
F-86962 Chasseneuil Futuroscope cedex
alain.cimetiere@univ-poitiers.fr

ABSTRACT. Our aim is to propose an improved regularization method for data completion problems. This method is presented on the Cauchy problem for the Laplace equation in 2D situations. This method is an iterative one, uses a regularization with fading effect and penalization terms which take into account the fact that, under some regularity assumptions, the partial derivatives of a harmonic function is also harmonic. Many numerical simulations using the finite element method highlight the efficiency, accuracy, stability when data are noisy and the ability of the method to take into account and deblur noisy data.

RÉSUMÉ. Cet article propose une amélioration d'une méthode de régularisation pour les problèmes de complétion des données. Cette méthode est présentée, en dimension deux, sur le problème de Cauchy associé à l'équation de Laplace. Elle repose sur un processus itératif et utilise une régularisation à effet évanescent et des termes de pénalisation qui prennent en compte le fait que les dérivées partielles d'une fonction harmonique sont aussi harmoniques. Des simulations numériques, utilisant la méthode des éléments finis, mettent en évidence l'efficacité, la précision et la stabilité de la méthode, ainsi que sa capacité à débruiter les données.

KEYWORDS: Cauchy problems, inverse problems, data completion, Laplace equation, regularization.

MOTS-CLÉS : problèmes de Cauchy, problèmes inverses, complétion de données, équation de Laplace, régularisation.

DOI:10.3166/EJCM.20.309-340 © 2011 Lavoisier, Paris

1. Introduction

Inverse problems arise in many engineering fields, such as, in thermal sciences, electrostatics, solid mechanics and find applications in non-destructive testing or in medical investigations. Kubo (1988) has defined inverse problems in opposition to direct problems. Direct problems are problems for which the geometry of the domain, the equilibrium equations, the constitutive equations, the boundary conditions on the whole boundary of the domain and the initial conditions are all given. For Kubo, inverse problems can be characterized by the lack of at least one of these elements of information. According to this definition, many physical problems, for example, the identification of material parameters, the identification of unknown boundaries (such as corrosion zones, cavities or cracks), the identification of inclusions, the identification of initial boundary conditions, the identification of inaccessible boundary conditions can be considered as inverse problems. In a mathematical sense, direct problems can be considered as well-posed problems. In linear cases, these problems have a unique solution which is stable (continuously dependent on the data). Conversely, inverse problems are generally ill-posed problems in the Hadamard sense (Hadamard, 1923), since the existence or uniqueness or the continuous dependence on the data of their solutions may not be ensured.

The purpose of this paper is to examine an inverse boundary value problem associated with the Laplace equation. It consists in recovering missing data (for instance, both temperature and heat flux are unknown) on some part of the boundary of a domain from overspecified data (both temperature and heat flux are given) on the other part. In this case, the equilibrium equation, the constitutive equation, the domain and its boundary are known. This problem is namely known as a Cauchy problem and is a data completion problem. In order to solve the Cauchy problem for the Laplace equation, many regularization methods have been introduced (Andrieux *et al.*, 2006; Bourgeois, 2005; Chen *et al.*, 2009; Cimetière *et al.*, 2000; Cimetière *et al.*, 2001; Cimetière *et al.*, 2005; Delvare *et al.*, 2002; Delvare *et al.*, 2008; Engl *et al.*, 1996; Hayashi *et al.*, 2002; Jourhmane *et al.*, 2004; Klibanov, 1991; Kozlov *et al.*, 1991; Lattès *et al.*, 1967; Lesnic *et al.*, 1997; Marin, 2005; Marin, 2009b; Marin, 2009a; Marin, 2011). These methods can be classified as Tikhonov type methods (Chen *et al.*, 2009; Hayashi *et al.*, 2002; Marin, 2005; Tikhonov *et al.*, 1977), quasi-reversibility type methods (Bourgeois, 2005; Klibanov, 1991; Lattès *et al.*, 1967), iterative methods (Andrieux *et al.*, 2006; Cimetière *et al.*, 2000; Cimetière *et al.*, 2001; Cimetière *et al.*, 2005; Delvare *et al.*, 2002; Delvare *et al.*, 2008; Engl *et al.*, 1996; Jourhmane *et al.*, 2004; Kozlov *et al.*, 1991; Lesnic *et al.*, 1997; Marin, 2009b; Marin, 2009a; Marin, 2011; Hao *et al.*, 2000)... Quasi reversibility methods and Tikhonov regularization methods present the advantage of leading to well posed problems after modifying the equilibrium equation. Some iterative methods are based on the use of a sequence of well-posed problems and others on the minimization of an energy-like functional. Numerical algorithms are implemented using different numerical methods, such as the finite element method (FEM) (Andrieux *et al.*, 2006; Bourgeois, 2005; Cimetière *et al.*, 2000; Cimetière *et al.*, 2001; Cimetière *et al.*, 2005; Cimetière *et al.*, 2002), the boundary element method

(BEM) (Delvare *et al.*, 2002; Delvare *et al.*, 2008; Jourhmane *et al.*, 2004; Kozlov *et al.*, 1991; Lesnic *et al.*, 1997; Cimetière *et al.*, 2002), the finite difference method (Klibanov, 1991) or meshless methods in conjunction with Tikhonov regularization techniques (Chen *et al.*, 2009; Marin, 2005; Marin, 2009b; Marin, 2009a; Marin, 2011) or in conjunction with a singular value decomposition (Chen *et al.*, 2009).

A somewhat different resolution approach was first introduced in (Cimetière *et al.*, 2000) to solve the Cauchy problem for the Laplace equation. This approach, also used in (Cimetière *et al.*, 2001; Delvare *et al.*, 2002), reduces the resolution of the Cauchy problem to the resolution of a sequence of optimization problems under equality constraints. The functional is composed of two terms. At each step of the resolution, the first term gives the gap between the optimal element and the overspecified boundary data (relaxation term), the second one the gap between the optimal element and the previous optimal element (regularization term). The equality constraints characterize the equilibrium. So, at each step an optimal element is obtained which is an exact solution to the equilibrium equation and is nearer to the overspecified data than the previous optimal element calculated. In the case of compatible data, it was also proved that the sequence converges to the solution of the Cauchy problem. The additional regularization term tends to zero as iterations continue. It is the reason why this method is called the evanescent regularization method. This inverse technique was extended to the Cauchy problem in linear elasticity (Delvare *et al.*, 2010).

In order to improve the reconstruction of the normal derivative, a first-order method was introduced (Delvare *et al.*, 2008). This method connects the determination of the solution of the Cauchy problem with the determination of two auxiliary functions, which are the partial derivatives of the solution when the data are compatible. The first-order problem needs additional boundary conditions which have to be derived from the given boundary conditions of the Cauchy problem. The evaluation of these additional boundary conditions requires the boundary conditions to be tangentially differentiated. These tangential derivatives have to be numerically evaluated.

The purpose of this paper is to propose a resolution method for the Cauchy problem designed to reconstruct, in a very precise way, regular harmonic functions when the boundary is non-regular. This method is based on the same principle as the first-order method but requiring no numerical evaluations of additional data. This method relies on a system of two weak integral formulations which connect the two partial derivatives of a harmonic function and not their normal derivatives. Section 2 is devoted to the formulation of the Cauchy problem for the Laplace equation. In Section 3 the system of weak formulations which connects the partial derivatives of a harmonic function is introduced. Section 4 describes the inverse method and Section 5 describes its numerical implementation using the finite element method. In Section 6 several numerical simulations are presented.

2. The Cauchy problem for the Laplace equation

Let us consider an open set Ω in \mathbb{R}^2 , assuming that its boundary Γ is divided in two parts Γ_d and Γ_u , where $\Gamma_d \cup \Gamma_u = \Gamma$ and $\Gamma_d \cap \Gamma_u = \emptyset$. The subscript d denotes the data and the subscript u denotes the unknowns. With no source term, the equilibrium equation is given by:

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = 0 \quad x \in \Omega \quad [1]$$

At a point $x \in \Gamma$, $n(x)$ denotes the outward unit normal vector and the normal derivative u' is defined by:

$$u' = \frac{\partial u}{\partial n} = \frac{\partial u}{\partial x_1} n_1 + \frac{\partial u}{\partial x_2} n_2$$

It is assumed that both the function u and the normal derivative u' are given or known on the part of boundary Γ_d but no condition is prescribed on the remaining part Γ_u :

$$\begin{aligned} u(x) &= \phi_d & x \in \Gamma_d \\ u'(x) &= \psi_d & x \in \Gamma_d \end{aligned} \quad [2]$$

where ϕ_d and ψ_d are prescribed functions.

The equilibrium equation [1] and the boundary conditions [2] lead to the formulation of the Cauchy problem for the Laplace equation:

$$\begin{cases} \Delta u = 0 & x \in \Omega \\ u = \phi_d & x \in \Gamma_d \\ u' = \psi_d & x \in \Gamma_d \end{cases} \quad [3]$$

This problem is difficult to solve, since it is ill-posed. When it admits a solution, its solution is unique, but it is known to be very sensitive (Hadamard, 1923; Engl *et al.*, 1996; Tikhonov *et al.*, 1977) to small perturbations on boundary conditions [2]. Thereafter, the Cauchy problem [3] is regarded as a problem without dimension.

3. System of weak formulations connecting the partial derivatives and reformulation of the Cauchy problem

3.1. Weak formulation of the Cauchy problem

Let u be a harmonic function. We have the following weak formulation:

$$\langle r(u, u'), v \rangle \equiv \int_{\Omega} \text{grad } u \text{ grad } v \, d\Omega - \int_{\Gamma} u' v \, ds = 0 \quad \forall v \in H^1(\Omega) \quad [4]$$

We can deduce the following weak formulation of the Cauchy problem [3]:

$$\begin{cases} \langle r(u, u'), v \rangle \equiv 0 & \forall v \in H^1(\Omega) \\ u = \phi_d & x \in \Gamma_d \\ u' = \psi_d & x \in \Gamma_d \end{cases} \quad [5]$$

3.2. Two weak equations connecting the partial derivatives of a harmonic function

Under the regularity assumption $u \in C^2(\overline{\Omega}) \cap C^3(\Omega)$, we obtain

$$\forall i = 1, 2 \quad \frac{\partial u}{\partial x_i} \text{ is harmonic on } \Omega \quad [6]$$

i.e.

$$\Delta \left(\frac{\partial u}{\partial x_i} \right) = \frac{\partial (\Delta u)}{\partial x_i} = 0 \quad \forall x \in \Omega \quad [7]$$

Next, using $\Delta u = 0$ on the boundary Γ , we obtain

$$\frac{\partial}{\partial n} \left(\frac{\partial u}{\partial x_2} \right) = \frac{\partial}{\partial s} \left(\frac{\partial u}{\partial x_1} \right) \quad \forall x \in \Gamma \quad [8]$$

and

$$\frac{\partial}{\partial n} \left(\frac{\partial u}{\partial x_1} \right) = -\frac{\partial}{\partial s} \left(\frac{\partial u}{\partial x_2} \right) \quad \forall x \in \Gamma \quad [9]$$

The weak formulation [4] can be written for $\frac{\partial u}{\partial x_1}$:

$$\begin{aligned} \langle R_1 \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \right), v \rangle &\equiv \int_{\Omega} \text{grad} \left(\frac{\partial u}{\partial x_1} \right) \text{grad} v \, d\Omega \\ &+ \int_{\Gamma} \frac{\partial}{\partial s} \left(\frac{\partial u}{\partial x_2} \right) v \, ds = 0 \quad \forall v \in H^1(\Omega) \end{aligned} \quad [10]$$

and for $\frac{\partial u}{\partial x_2}$:

$$\begin{aligned} \langle R_2 \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \right), v \rangle &\equiv \int_{\Omega} \text{grad} \left(\frac{\partial u}{\partial x_2} \right) \text{grad} v \, d\Omega \\ &- \int_{\Gamma} \frac{\partial}{\partial s} \left(\frac{\partial u}{\partial x_1} \right) v \, ds = 0 \quad \forall v \in H^1(\Omega) \end{aligned} \quad [11]$$

3.3. Reformulation of the Cauchy problem

The following problem [12] is introduced:

$$\begin{cases} \langle r(u, u_1 n_1 + u_2 n_2), v \rangle = 0 & \forall v \in H^1(\Omega) & (a) \\ u = \phi_d & \text{on } \Gamma_d & (b) \\ u_1 n_1 + u_2 n_2 = \psi_d & \text{on } \Gamma_d & (c) \\ \langle R_1(u_1, u_2), v \rangle = 0 & \forall v \in H^1(\Omega) & (d) \\ \langle R_2(u_1, u_2), v \rangle = 0 & \forall v \in H^1(\Omega) & (e) \end{cases} \quad [12]$$

It can be proved that for compatible data, u_1 and u_2 are respectively the partial derivatives of u and u is the solution of the Cauchy problem [3].

4. An iterative Tikhonov type algorithm with penalization

Let us introduce the space $H(\Omega)$ of solutions of the weak formulation [12a]:

$$H(\Omega) = \{ (u, u_1, u_2) / \langle r(u, u_1 n_1 + u_2 n_2), v \rangle = 0 \quad \forall v \in H^1(\Omega) \}$$

Next, let us denote $H(\Gamma)$ the space composed of the restrictions on Γ of elements (u, u_1, u_2) in $H(\Omega)$.

As in (Cimetière *et al.*, 2001), the reliable information (equilibrium equation [12a]) and the uncertain informations (boundary conditions [12b and 12c]) are distinguished. An iterative Tikhonov type algorithm is introduced where the equations [12d and 12e] are taken into account through penalization terms.

Let consider $c > 0$ and $\mathbf{U}^0 = \mathbf{0}$

$$\begin{cases} \text{Find } \mathbf{U}^{k+1} = (u^{k+1}, u_1^{k+1}, u_2^{k+1}) \in H(\Gamma) \text{ such that} \\ J^k(\mathbf{U}^{k+1}) \leq J^k(\mathbf{W}) \quad \forall \mathbf{W} \in H(\Gamma) \text{ with:} \\ J^k(\mathbf{W}) = \|w - \phi_d\|_{\Gamma_d}^2 + \|(w_1 n_1 + w_2 n_2) - \psi_d\|_{\Gamma_d}^2 + c \|w - u^k\|_{\Gamma}^2 \\ \quad + \|R_1(w_1, w_2)\|_{\Gamma}^2 + \|R_2(w_1, w_2)\|_{\Gamma}^2 \end{cases} \quad [13]$$

where the norms are defined by:

$$\|v\|_{\Gamma_d}^2 = \int_{\Gamma_d} v^2 ds$$

$$\|v\|_{\Gamma}^2 = \int_{\Gamma} v^2 ds$$

In the iterative process [13], the equilibrium equation [1] (or in a weak sense relation [12a]) is exactly taken into account since at each step the search for the optimal

element is performed in space $H(\Gamma)$. The functional is composed of different terms which play different roles. The terms $\|w - \phi_d\|_{\Gamma_d}^2$ and $\|(w_1 n_1 + w_2 n_2) - \psi_d\|_{\Gamma_d}^2$ act only on Γ_d and represent the gap between the optimal element and the overspecified boundary data. They relax the overspecified data which can possibly be blurred by measurement noises (relaxation terms). The term $c\|w - u^k\|_{\Gamma}^2$ acts on the whole boundary Γ and not only on the boundary Γ_u where the boundary conditions are to be completed. This term is a regularization term and controls the distance between the new optimal element and the previous optimal element. This term tends to zero as the iterations continue. The terms $\|R_1(w_1, w_2)\|_{\Gamma}^2$ and $\|R_2(w_1, w_2)\|_{\Gamma}^2$ are penalization terms. At each step the optimal element obtained is an exact solution of the equilibrium equation [12a] and is near to the overspecified data ϕ_d and ψ_d .

5. FEM implementation

5.1. Discretization of the relation [12a]- Discrete solutions space

The first issue in this section is to discretize the weak formulation [12a]. Our main concern has been to make use of any ordinary finite element code, not to work out a specific one or a specific finite element. As a matter of fact, computations were run using Cast3M¹ (CASTEM2000, 1998) and piecewise linear finite elements, which means a piecewise constant approximation for the normal derivative. Let us now discretize the domain Ω , h being the discretization parameter standing for the element size, leading to n nodes and n elements on the boundary, and m nodes inside the domain. Let V_h be the space of continuous piecewise linear functions with respect to the mesh, and let us define $V_h(\Gamma)$ and $W_h(\Gamma)$ as the space of continuous piecewise linear functions and the space of piecewise constant functions on the boundary. Traces of functions belonging to $V_h(\Omega)$ span the space $V_h(\Gamma)$, whereas the associated normal derivatives belong to the space $W_h(\Gamma)$ of piecewise constant functions. Defining U , U_1 , U_2 and U' as the n vectors standing respectively for the n nodal values of u , u_1 , u_2 and the n discrete values of u' on the boundary, and U^* the m -vector of internal nodal values of u , the discrete equilibrium equations read as:

$$\begin{bmatrix} A_{ii} & A_{ei}^T \\ A_{ei} & A_{ee} \end{bmatrix} \begin{pmatrix} U^* \\ U \end{pmatrix} = \begin{pmatrix} 0 \\ -BU' \end{pmatrix} \quad [14]$$

A_{ii} is the stiffness matrix corresponding to the Dirichlet problem and thus is invertible. Expressing the internal unknowns U^* in terms of the boundary ones U , i.e performing a condensation, Equation [14] reduces to:

$$(A_{ee} - A_{ei} A_{ii}^{-1} A_{ei}^T)U + BU' = 0 \quad [15]$$

1. The Finite Element code Cast3M is developed by the Department of Mechanics and Technology (DMT) of the French Atomic Energy Agency (CEA - DEN/DM2S/SEMT), <http://www-cast3m.cea.fr>

The matrix form of [15] reads:

$$\begin{bmatrix} A & B \end{bmatrix} \begin{pmatrix} U \\ U' \end{pmatrix} = 0 \quad [16]$$

U' can be expressed as a function of U_1 and U_2 , and of N_i , the discrete components of the normal vector n , and leads to the following system of linear equations:

$$AU + B_1U_1 + B_2U_2 = 0 \quad [17]$$

where the matrices A , B_1 and B_2 only depend on the domain mesh. The linear system [17] of n equations for $3n$ unknowns defines the discrete formulation of the space $H(\Gamma)$ and leads to the definition of the following discrete solutions space:

$$H_h(\Gamma) = \left\{ \begin{array}{l} (U, U_1, U_2) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \text{ such that} \\ E_h(U, U_1, U_2) = AU + B_1U_1 + B_2U_2 = 0 \end{array} \right\} \quad [18]$$

where E_h denotes a linear operator mapping $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ onto \mathbb{R}^n .

5.2. Discretization of the relations [12d] and [12e]

Next, we need to discretize the relations $\langle R_1(u_1, u_2), v \rangle \equiv 0$ and $\langle R_2(u_1, u_2), v \rangle \equiv 0$ using the finite element method. Considering the same finite element discretization as in the previous section, the system of n linear equations with $2n$ unknowns [19] (respectively the system [20]) represents the discrete formulation of Equation [10] (respectively of Equation [11]):

$$AU_1 + CU_2 = 0 \quad [19]$$

$$AU_2 - CU_1 = 0 \quad [20]$$

For $u = x_1 + x_2$, we obtain $U_1 = (1, \dots, 1)$ and $U_2 = (1, \dots, 1)$. These two vectors satisfy the system of $2n$ equations with $2n$ unknowns composed of the linear system [19] and of the linear system [20]. So the lines of the linear systems [19-20] are not independent. This explains why the relations [12c] and [12d] (the systems [19] and [20] in the discrete form of the algorithm) are taken into account in the functional to be minimized and not as equality constraints.

5.3. Discretization of the $(k + 1)^{th}$ iteration

Given now that $c > 0$ and $(U^0, U_1^0, U_2^0) = (0, 0, 0) \in H_h(\Gamma)$, iteration $(k + 1)$ of the discretized iterative algorithm reads as follows:

$$\left\{ \begin{array}{l} \text{Find } \Psi^{k+1} = (U^{k+1}, U_1^{k+1}, U_2^{k+1}) \in R^{3N} \text{ such that} \\ J^k(\Psi^{k+1}) \leq J^k(\Psi) \quad \forall \Psi \in R^{3N} \\ \text{under the equality constraints} \\ AU + B_1U_1 + B_2U_2 = 0 \end{array} \right. \quad [21]$$

Problem [21] is a minimization problem in R^{3n} under the n equality constraints expressed by [17]. Its solution is given by:

$$\left| \begin{array}{l} \text{Find } (U^{k+1}, U_1^{k+1}, U_2^{k+1}, \lambda^{k+1}) \in \mathbb{R}^{4n} \text{ such that :} \\ \nabla J_c^k(U^{k+1}, U_1^{k+1}, U_2^{k+1}) + (\lambda^{k+1})^T \nabla E_h(U^{k+1}, U_1^{k+1}, U_2^{k+1}) = 0 \\ E_h(U^{k+1}, U_1^{k+1}, U_2^{k+1}) = 0 \end{array} \right. \quad [22]$$

where λ^{k+1} is a n -vector of Lagrange multipliers introduced to take the equality constraints [16] into account.

Each iteration in the iterative algorithm must solve a system of $4n$ linear equations with $4n$ unknowns. The matrix of this linear system is independent of the iterations and needs to be computed only once. For this reason, a direct algorithm (the Crout factorization) has been preferred to iterative methods. The factorization, which is obtained at the first step, is also used at each following step.

The corresponding system for the first-order method (Delvare *et al.*, 2008) involves a system of $9n + 1$ equations for $9n + 1$ unknowns and moreover needs to numerically differentiate ϕ_d twice and ψ_d once.

6. Numerical simulations

The purpose of this section is to present the numerical results obtained with the method introduced. The procedure used during the numerical simulations is as follows:

- (i) The meshing of the boundary is made using SEG2 elements. The SEG2 element is a finite element with two nodes which leads to a linear interpolation of the functions u , u_1 and u_2 .
- (ii) The user specifies the meshing of the boundary specifying the number and the distribution of the finite elements on each part of the boundary.
- (iii) Then, the mesh of the entire domain is generated automatically by a routine included in the Cast3m software (CASTEM2000, 1998). This mesh is constituted of 4-node quadrilaterals and 3-node triangles.
- (iv) The computation and the assembly of the stiffness matrix corresponding to the domain is performed thanks to the standard routines of Cast3m software (CASTEM2000, 1998).
- (v) This software creates a superelement based on the boundary and computes the corresponding stiffness matrix. This leads to the condensed stiffness matrix A.
- (vi) The stiffness matrix A is then used by the specific code that implements the inverse method introduced.

Note that the inverse method only involves values on the boundary. Therefore, only for the different numerical tests will the discretization of the boundary Γ be specified. Moreover, note that all numerical computations were performed on a machine with a 2.20GHz Intel® Core™ 2 Duo processor T7500.

Different cases are considered for which an analytical solution u^{an} is known. The following control quantities are used to assess the accuracy of the method:

(i) the $L^2(\Gamma)$ gap between an optimal element u and the analytical solution u^{an} on Γ :

$$g_{\Gamma} = \int_{\Gamma} (u - u^{an})^2 ds = \|u - u^{an}\|_{\Gamma}^2$$

(ii) the $L^2(\Gamma)$ gap between the normal derivative u' of an optimal element and the normal derivative of the analytical solution on Γ :

$$g'_{\Gamma} = \int_{\Gamma} (u' - u'^{an})^2 ds = \|u' - u'^{an}\|_{\Gamma}^2$$

(iii) the $L^2(\Gamma_d)$ gap between an optimal element u and the data ϕ_d on Γ_d :

$$g_{\Gamma_d} = \int_{\Gamma_d} (u - \phi_d)^2 ds = \|u - \phi_d\|_{\Gamma_d}^2$$

(iv) the $L^2(\Gamma_d)$ gap between the normal derivative u' and the data ψ_d on Γ_d :

$$g'_{\Gamma_d} = \int_{\Gamma_d} (u' - \psi_d)^2 ds = \|u' - \psi_d\|_{\Gamma_d}^2$$

(v) the $L^2(\Gamma)$ relative error (in %) made on u :

$$u_{error} = \sqrt{\frac{\int_{\Gamma} (u - u^{an})^2 ds}{\int_{\Gamma} (u^{an})^2 ds}} = \sqrt{\frac{\|u - u^{an}\|_{\Gamma}^2}{\|u^{an}\|_{\Gamma}^2}}$$

(vi) the $L^2(\Gamma)$ relative error (in %) made on u' :

$$u'_{error} = \sqrt{\frac{\int_{\Gamma} (u' - u'^{an})^2 ds}{\int_{\Gamma} (u'^{an})^2 ds}} = \sqrt{\frac{\|u' - u'^{an}\|_{\Gamma}^2}{\|u'^{an}\|_{\Gamma}^2}}$$

6.1. Example 1

Numerical simulations are performed on a square domain ($\Omega =]0, 1[\times]0, 1[$) (Figure 1). The boundary part Γ_d is composed of the two sides ($x_2 = 0$ and $x_1 = 1$). The boundary part Γ_u is composed of the two sides ($x_1 = 0$ and $x_2 = 1$). Each side of the square domain is discretized using N finite elements SEG2. All the finite elements have the same length and the nodes are uniformly distributed. This discretization used

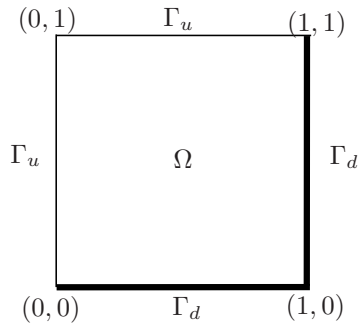


Figure 1. Square domain Ω

to find the function u and its normal derivative u' on Γ_u , from the knowledge of function u and its normal derivative u' on Γ_d . The data ϕ_d and ψ_d are built using the analytical solution:

$$u^{an}(x_1, x_2) = \cos(x_1) \cosh(x_2) + \sin(x_1) \sinh(x_2) \quad [23]$$

6.1.1. Comparison with other methods

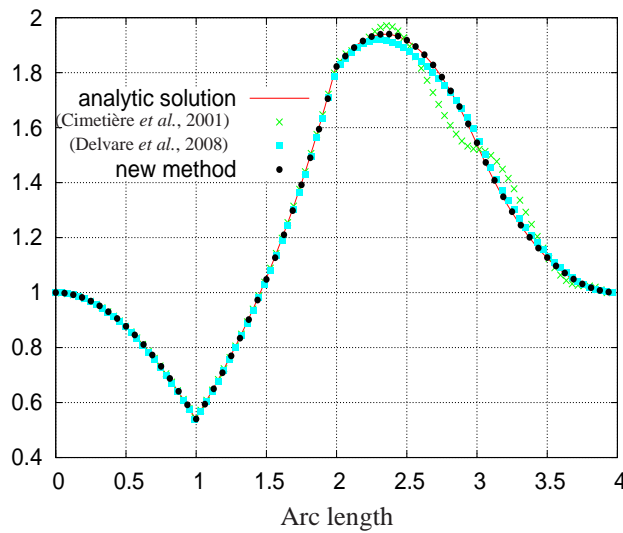


Figure 2. u reconstructions (comparison of different methods)

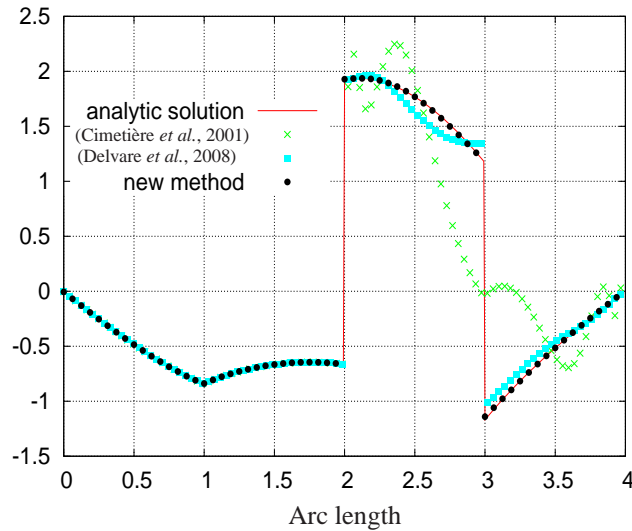


Figure 3. u' reconstructions (comparison of different methods)

In this section, the reconstructions obtained with this new method are compared with those obtained by other methods. This example was studied using a Tikhonov like method (Hayashi *et al.*, 2002), using a fading regularization method (Cimetière *et al.*, 2001) and using an iterative method (Lesnic *et al.*, 1997). All these methods gave similar results. This example was also studied using a first-order method (Delvare *et al.*, 2008). Figures 2 and 3 display all these reconstructions. The new method gives more accurate normal derivative reconstructions than the others.

6.1.2. Evolution of the control quantities during the iterative process

In the present case, the data on Γ_d are generated from the analytical solution (Equation [23]) and are not blurred by measurement noise. The boundaries Γ_d and Γ_u are both approximated with 160 SEG2 finite elements. The value of the parameter c is arbitrarily fixed at 0.01.

Figure 4 represents the evolution of different control quantities g_Γ , g'_Γ , g_{Γ_d} and g'_{Γ_d} versus the number of iterations k . Figure 5 represents the evolution of the errors u_{error} and u'_{error} versus the number of iterations k . We notice from the 300th iteration that the quantities g_Γ , g'_Γ , g_{Γ_d} , g'_{Γ_d} , u_{error} and u'_{error} remain constant. This confirms that the method converges. The iterative process is stopped when two successive optimal elements are identical. The residual terms g_{Γ_d} and g'_{Γ_d} (relaxation terms) decrease during the iterative process and after convergence these terms remain constant.

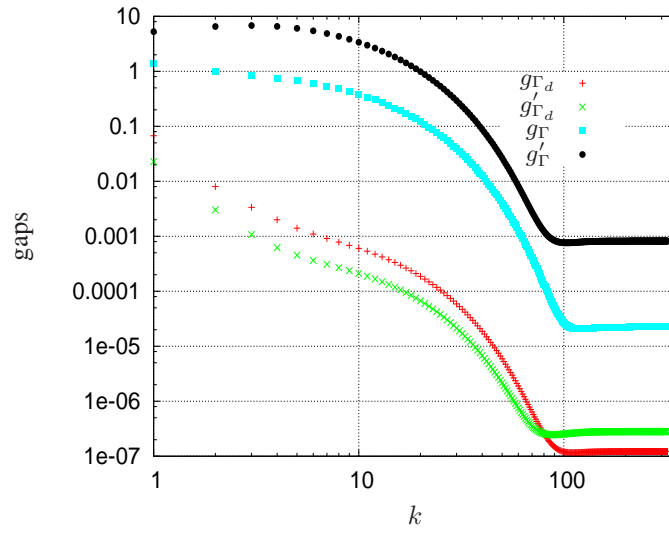


Figure 4. Evolution of control quantities versus the number of iterations k

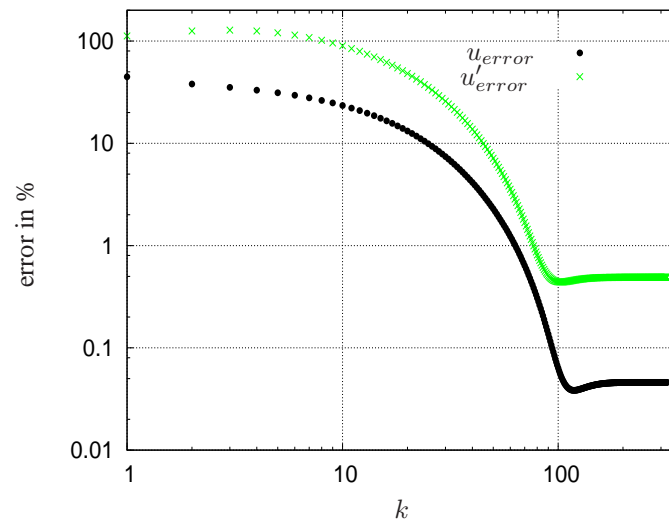


Figure 5. Evolution of errors versus the number of iterations k

It can also be observed that the error in the numerical normal derivative obtained (u'_{error}) using this method is larger than that corresponding to the reconstructed function (u_{error}). This last remark is also valid for all the following numerical examples which will be analysed.

6.1.3. Behaviour with respect to c parameter

It is necessary to look at the influence of parameter c . For the same discretization as in the previous example, different values of the parameter c are tested. Table I lists the results obtained for each value of parameter c by specifying the number of iterations necessary to achieve convergence, the u_{error} , the u'_{error} and the CPU time. The errors on u and u' are identical for each value of the parameter c .

Table 1. Influence of c on the number of iterations k to achieve convergence - influence of c on the u_{error} and on the u'_{error}

c	k	u_{error} in %	u'_{error} in %	CPU time in s
1.	16832	0.04564	0.48925	116.95
0.1	1931	0.04564	0.48925	14.87
0.01	320	0.04564	0.48926	3.83
0.001	35	0.04564	0.48925	1.88
0.0001	11	0.04564	0.48925	1.72
1E-5	7	0.04564	0.48925	1.69
1E-6	4	0.04563	0.48922	1.67
1E-7	4	0.04565	0.48969	1.67
1E-8	4	0.04620	0.49821	1.67

This confirms that the algorithm converges to the same solution whatever the value of c . However the choice of the parameter c affects the number k of iterations needed to obtain convergence. From the evolution of the CPU time with c (or with the number of iterations k) it can be deduced that the CPU time taken by each iteration for $k > 1$ is roughly $6.85 \cdot 10^{-3} s$. This CPU time is lower than the CPU time taken to achieve both the preliminary computations and the first iteration (roughly $1.64 s$). Subsequently, the value of c and the number of iterations required to achieve convergence will be no longer specified.

6.1.4. Behaviour with respect to the mesh refinement

It is necessary to see how the reconstructions are influenced when the mesh refinement increases (depending on the number of finite elements N on each side of the square domain). Figure 6 shows the reconstructions on Γ of the function u obtained with $N = 5$, $N = 80$ and $N = 150$. Figure 7 shows the corresponding reconstructions of the normal derivative. Table II lists the errors u_{error} and u'_{error} for different mesh refinements. It can be observed that the error level decreases as the mesh re-

finement increases. This confirms that the inverse method is stable with respect to the mesh refinement. It can also be observed that reconstructions using a coarse mesh ($N = 5$) are also accurate.

Table 2. Influence of mesh refinement on the u_{error} and on the u'_{error}

N	u_{error} in %	u'_{error} in %
5	0.376	2.843
10	0.185	1.443
20	0.108	1.086
30	0.074	0.746
40	0.063	0.676
50	0.053	0.584
60	0.049	0.543
80	0.046	0.489
100	0.035	0.398
120	0.027	0.355
150	0.024	0.299

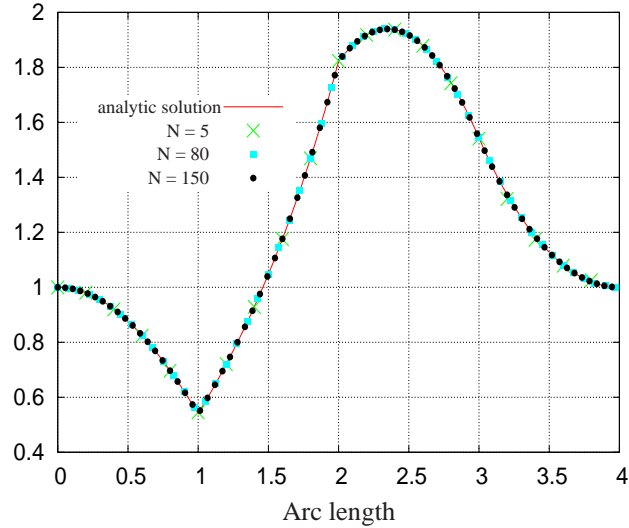


Figure 6. u reconstructions using different boundary mesh refinements

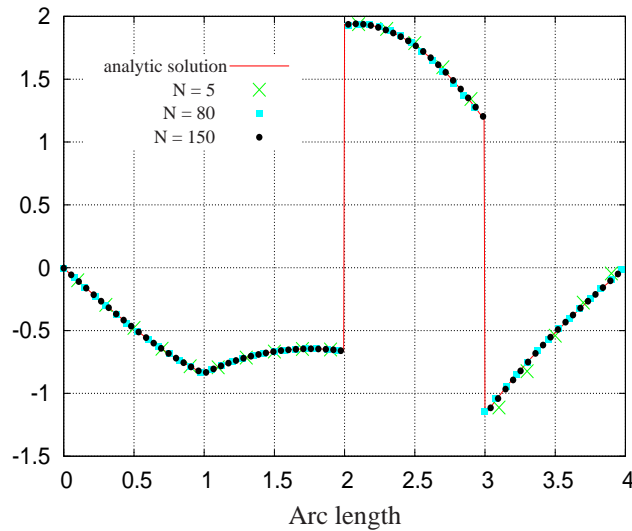


Figure 7. u' reconstructions using different boundary mesh refinements

6.1.5. Behaviour with respect to noisy data ϕ_d

It is necessary to see how the reconstructions are influenced when data are noisy. The noisy displacement data ϕ_d are generated by:

$$\phi_d = \phi_d^{an} + \delta \eta \phi_d^{max} \quad [24]$$

where $-1 \leq \eta \leq 1$ is a random value, δ is the noise level in % and ϕ_d^{max} is the maximal value of the data of Γ_d .

The boundary parts Γ_d and Γ_u are both approximated using 160 finite elements. The value of parameter c is arbitrarily fixed at 0.1.

The first step is to verify that the stopping criterion (g_{Γ_d} and g'_{Γ_d} become constant) remains reliable when the data are noisy. Figure 8 represents the evolution of the control quantity g_{Γ_d} versus the number of iterations k when the noise level is respectively fixed at 0%, 1%, 5%, 10%, 20% and 50%. Figure 9 represents the evolution of the control quantity g'_{Γ_d} versus the number of iterations k when the noise level is respectively fixed at 0%, 1%, 5%, 10%, 20% and 50%. It can be noticed that the quantities g_{Γ_d} and g'_{Γ_d} decrease at the beginning and then remain constant. The iterative process is stopped at this stage.

Figure 10 represents the evolution of the control quantity u_{error} versus the number of iterations k when the noise level is respectively fixed at 0%, 1%, 5%, 10%, 20% and 50%. Figure 11 represents the evolution of the control quantity u'_{error} versus the number of iterations k when the noise level is respectively fixed at 0%, 1%, 5%,

Table 3. Influence of the noise level on the gaps g_{Γ_d} and g'_{Γ_d} and on the errors u_{error} and u'_{error}

δ in %	g_{Γ_d}	g'_{Γ_d}	u_{error} in %	u'_{error} in %
0	1.22E-7	2.80E-7	0.05	0.49
1	2.21E-4	5.72E-7	0.27	1.77
2	8.84E-4	1.35E-6	0.42	2.23
3	1.99E-3	2.61E-6	0.59	2.74
5	5.53E-3	6.58E-6	0.96	3.83
10	2.21E-2	2.50E-5	1.88	6.69
20	8.83E-2	9.81E-5	3.73	12.54
25	0.14	1.52E-4	4.66	15.47
30	0.20	2.19E-4	5.59	18.42
40	0.35	3.89E-4	7.45	24.30
45	0.45	4.92E-4	8.38	27.25
50	0.55	6.07E-4	9.31	30.20
75	1.24	1.36E-3	13.96	44.94
80	1.79	1.56E-3	14.89	47.89
100	2.21	2.42E-3	18.61	59.68

10%, 20% and 50%. It can be noticed that the quantities u_{error} and u'_{error} have the same evolutions as the quantities g_{Γ_d} and g'_{Γ_d} . This confirms that the chosen stopping criterion (g_{Γ_d} and g'_{Γ_d} become constant) remains reliable when the data ϕ_d is noisy.

Figure 12 represents the reconstructions of u on the whole boundary Γ when the noise level is respectively fixed at 5% and at 10%. On the same figure, the noisy data ϕ_d used when the noise level is fixed at 10% is also specified. It can be noticed that the reconstructions correspond to the analytical solution u^{an} and that the noise in the data has been deleted by the algorithm. Figure 13 gives the corresponding reconstructions of the normal derivative. The reconstructions obtained can be seen to be very accurate.

The functional is composed of different terms with different roles. As in most inverse methods, there is a regularization term which tends to zero as the iterations continue. But, in the present method, there is also a relaxation term that allows data blurred by noise to be taken into account. We therefore seek a solution which is close to the data but not a solution that exactly fits the data. The algorithm then recomputes, at each step, a solution on the whole boundary. Residual terms are equal to the relaxation terms g_{Γ_d} and g'_{Γ_d} . These terms correspond to the gap on Γ_d between the deblurred reconstruction and the noisy data.

Figure 14 gives the evolution of the residual gaps g_{Γ_d} and g'_{Γ_d} obtained after convergence of the iterative process with respect to the noise level δ . Figure 15 gives the evolution of the errors u_{error} and u'_{error} obtained after convergence of the iterative

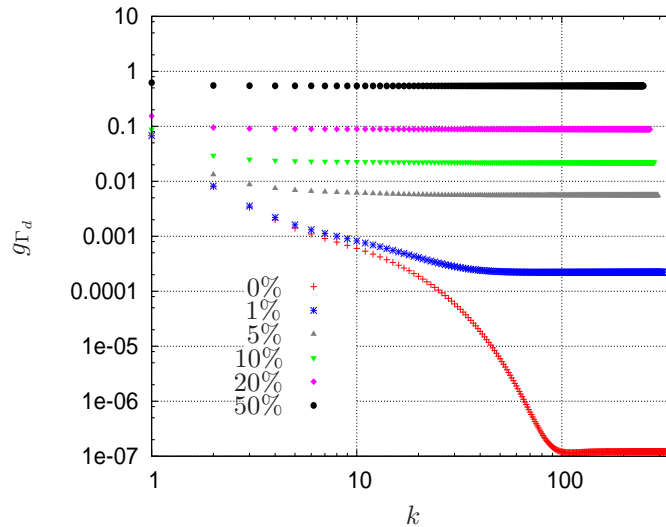


Figure 8. Evolution of the gap g_{Γ_d} versus the number of iterations k for different u noise levels (in %)

process with respect to the noise level δ . Table III lists the values of the residual errors u_{error} and u'_{error} and the value of the residual gaps g_{Γ_d} and g'_{Γ_d} obtained after convergence for the different noise levels considered.

6.1.6. Behaviour with respect to noisy data ψ_d

A similar study has been performed with noisy normal derivative data. The noisy data ψ_d are generated by:

$$\psi_d = \psi_d^{an} + \delta \eta \psi_d^{max} \tag{25}$$

Figure 16 gives the reconstructions of u on the whole boundary Γ when the noise level is respectively fixed at 5% and at 10%. Figure 17 represents the corresponding reconstructions of the normal derivative on the whole boundary Γ . On the same figure, the noisy data ψ_d used when the noise level is fixed at 10% is also specified. It can be noticed that the reconstructions corresponds to the analytical solution and that the noise in the data has been deleted by the algorithm. The reconstructions obtained can be seen to be very accurate.

Figure 18 gives the evolution of the errors u_{error} and u'_{error} obtained after convergence of the iterative process with respect to the noise level δ . Figure 19 gives the evolution of the residual gaps g_{Γ_d} and g'_{Γ_d} obtained after convergence of the iterative process with respect to the noise level δ .

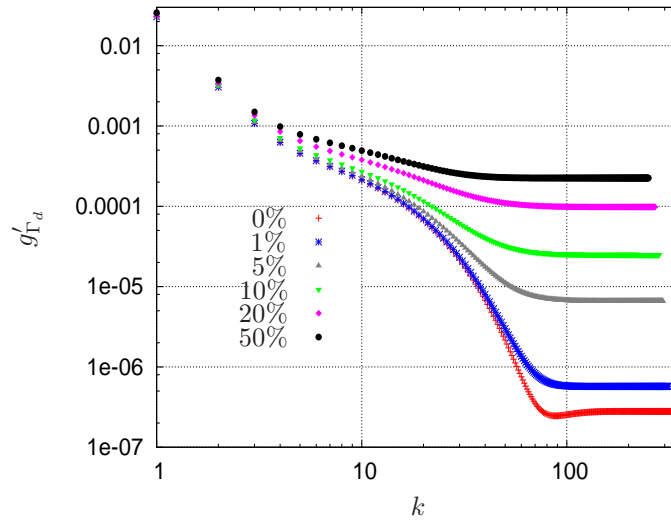


Figure 9. Evolution of the gap $g_{\Gamma_d}^l$ versus the number of iterations k for different u noise levels (in %)

6.2. Example 2

In this example, the function u to be reconstructed is given by:

$$u(x_1, x_2) = \frac{x_2}{(x_1 - \frac{1}{2})^2 + x_2^2} \tag{26}$$

The domain Ω and its boundary Γ are indicated in Figure 20.

For the first numerical simulations, the boundary part Γ_d is defined by $\Gamma_d = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ and is discretized using 120 finite elements SEG2. The boundary part Γ_u is defined by $\Gamma_u = \Gamma_4 \cup \Gamma_5$ and is discretized using 360 finite elements SEG2. Both the data ϕ_d and ψ_d are noisy. Figure 21 gives the reconstructions of the function on the boundary part Γ_u when the noise level on the data ϕ_d and ψ_d is respectively fixed at 0%, 1% and at 10%. Figure 22 gives the corresponding reconstructions of the normal function derivative on the boundary part Γ_u . Figure 23 gives the noisy data ϕ_d when the noise level is fixed at 10% as well as the corresponding reconstruction of the function on the boundary part Γ_d . Figure 24 gives the noisy data ψ_d when the noise level is fixed at 10% as well as the corresponding reconstruction of the normal derivative on the boundary part Γ_d . It is found in this case that the reconstructions are accurate on the whole boundary (on both boundary parts Γ_u and Γ_d) and that the noise in the data has been deleted by the algorithm.

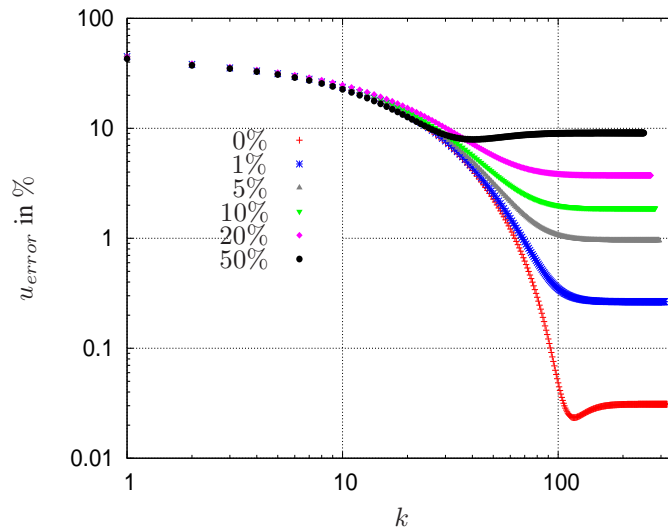


Figure 10. Evolution of u_{error} versus the number of iterations k for different u noise levels (in %)

For the second numerical simulations, we consider a situation where the numbers of available data on Γ_d for u (N_1 values) and for u' (N_2 values) are different. We examine the following cases:

- case 1: $N_1 = 40$ and $N_2 = 120$,
- case 2: $N_1 = 120$ and $N_2 = 40$,
- case 3: $N_1 = N_2 = 120$,
- case 4: $N_1 = N_2 = 40$.

In a sake of simplicity, we assume that these data are uniformly distributed on Γ_d but are noisy (the noise level on the data ϕ_d and ψ_d is fixed at 10%). Before making the data completion in case 2 (respectively in case 3), 80 auxiliary data are generated for u (respectively for u') by interpolating the available data in order to have the same number of data for u and u' and to use the same finite element discretization as the one used in the first simulations. For case 4, the boundary part Γ_d is discretized using a coarser mesh with 40 finite elements SEG2. Figure 25 gives the reconstructions of the function on the boundary part Γ_u for the four cases investigated. Figure 26 gives the corresponding reconstructions of the normal function derivative. It can be observed that the reconstructions in case 1 and in case 4 are slightly less accurate than that obtained in case 2 and in case 3.

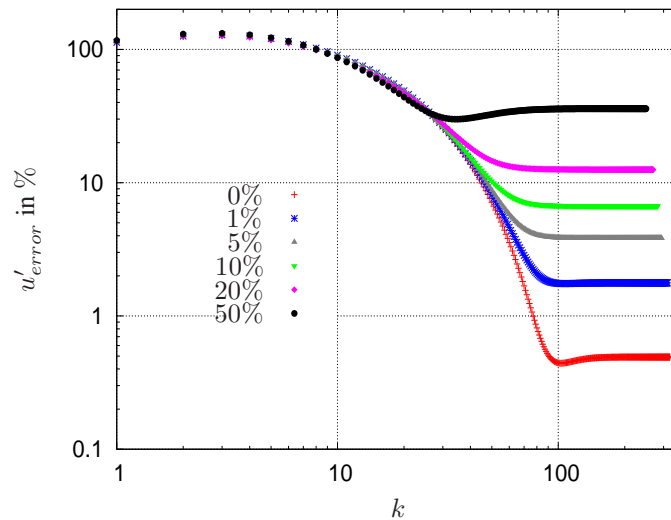


Figure 11. Evolution of u'_{error} versus the number of iterations k for different u noise levels (in %)

In the last numerical simulation, the robustness of the inverse method with respect to the number of corners on Γ_u is studied. In this case, the boundary part Γ_d extension is reduced and is limited to Γ_2 . Γ_d is discretized using 90 finite elements SEG2. The boundary part Γ_u presents 3 singular points and is defined by $\Gamma_u = \Gamma_3 \cup \Gamma_4 \cup \Gamma_5 \cup \Gamma_1$. Γ_1 is discretized using 15 finite elements SEG2, Γ_3 is discretized using 15 finite elements SEG2 and $\Gamma_4 \cup \Gamma_5$ is discretized using 360 finite elements SEG2. The data ϕ_d and ψ_d are exact. Figure 27 gives the reconstruction of the function on the boundary part Γ_u . Figure 28 gives the corresponding reconstruction of the normal derivative on the boundary part Γ_u .

It is found in all cases that the reconstructions are accurate when the domain has a non-regular boundary. It may also be noticed that for these different numerical simulations, the number of data (number of nodes on Γ_d) was less than the number of unknowns sought (number of nodes on Γ_u).

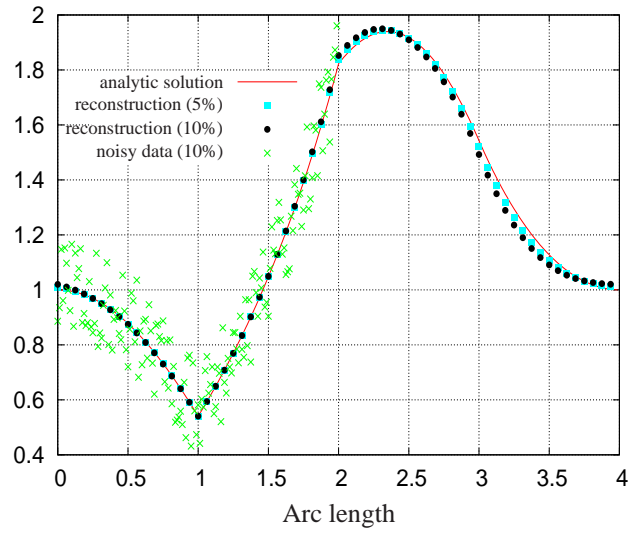


Figure 12. The noisy data ϕ_d when the noise level is 10% and the u reconstructions for different u noise levels

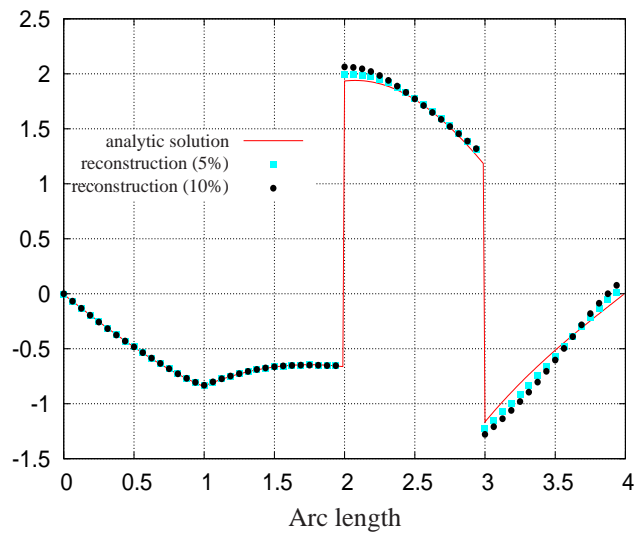


Figure 13. The u' reconstructions for different u noise levels

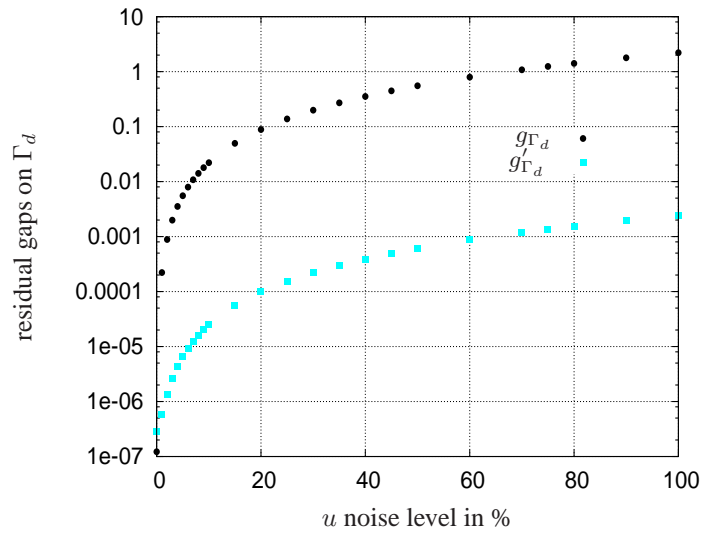


Figure 14. Influence of u noise level (in %) on the residual gaps g_{Γ_d} and g'_{Γ_d} after convergence

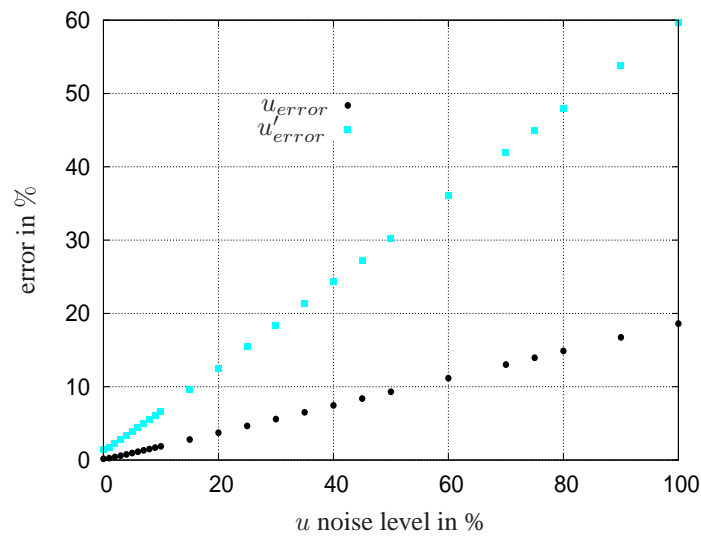


Figure 15. Influence of u noise level (in %) on the errors u_{error} and u'_{error}

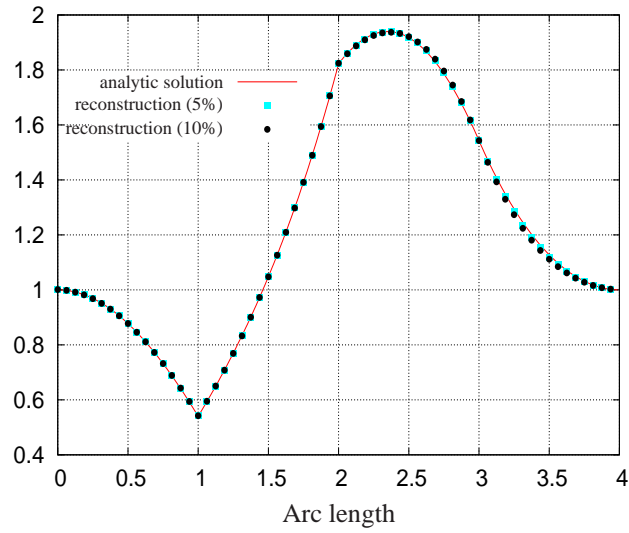


Figure 16. u reconstructions for different u' noise levels

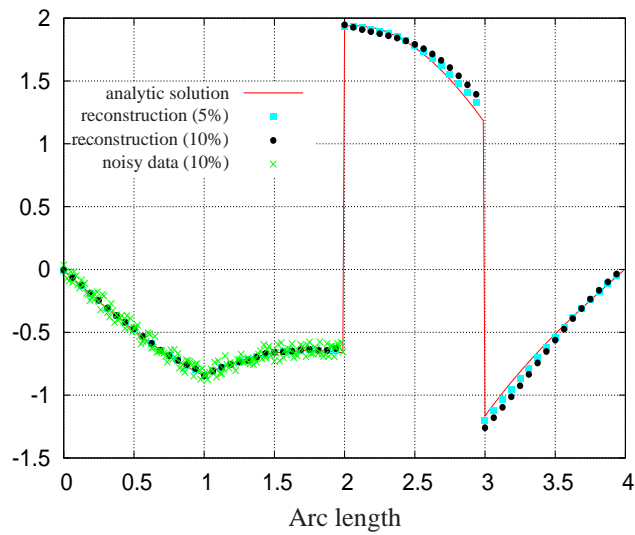


Figure 17. The noisy data ψ_d when the noise level is 10% and u' reconstructions for different u' noise levels

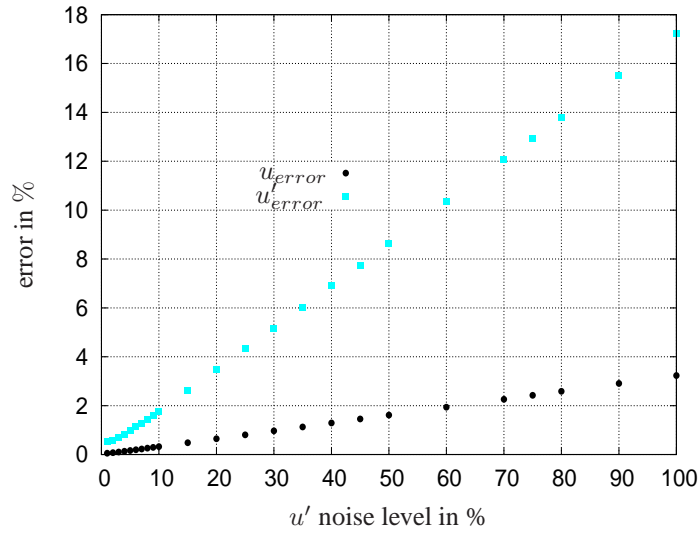


Figure 18. Influence of u' noise level (in %) on the errors u_{error} and u'_{error}

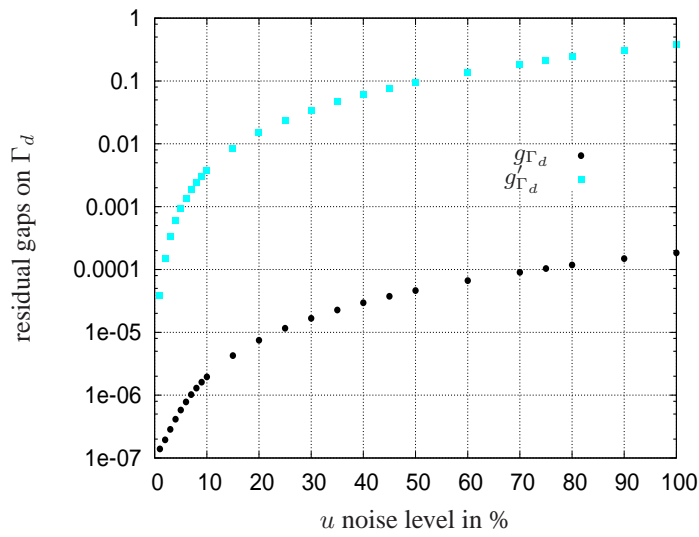


Figure 19. Influence of u' noise level (in %) on the residual gaps g_{Γ_d} and g'_{Γ_d} after convergence

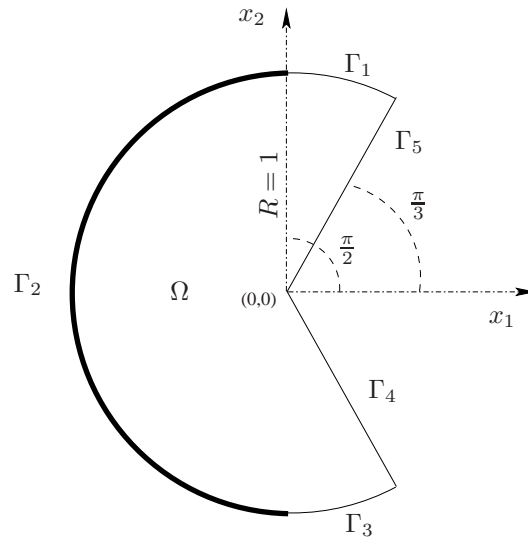


Figure 20. Domain Ω

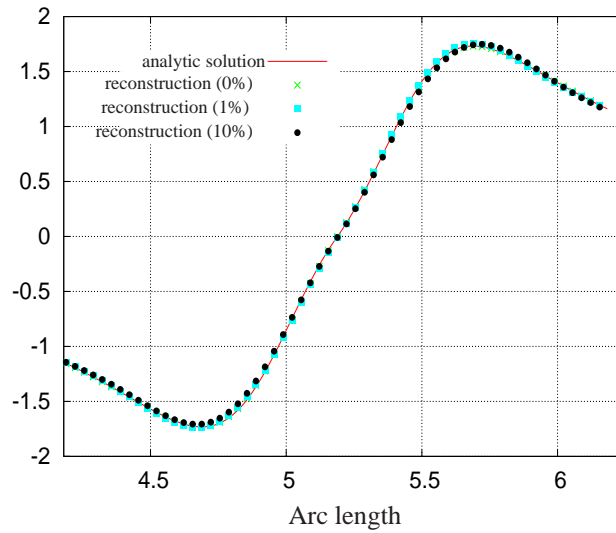


Figure 21. u reconstructions on Γ_u when both data ϕ_d and ψ_d are noisy (different noise levels)

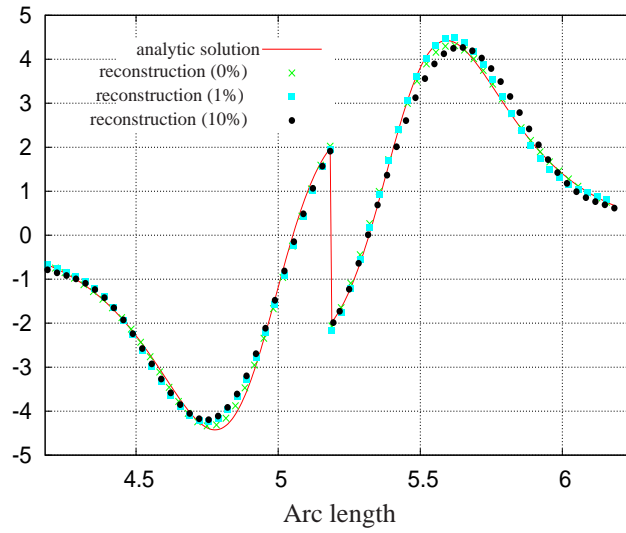


Figure 22. u' reconstructions on Γ_u when both data ϕ_d and ψ_d are noisy (different noise levels)

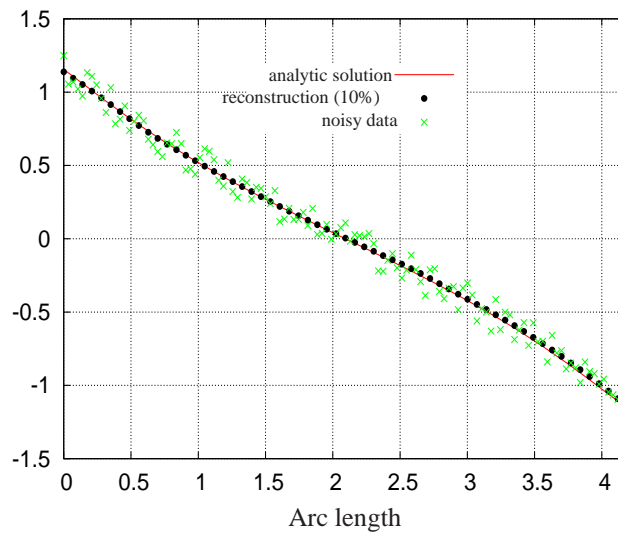


Figure 23. u reconstruction on Γ_d and the noisy data ϕ_d when the noise level is 10%

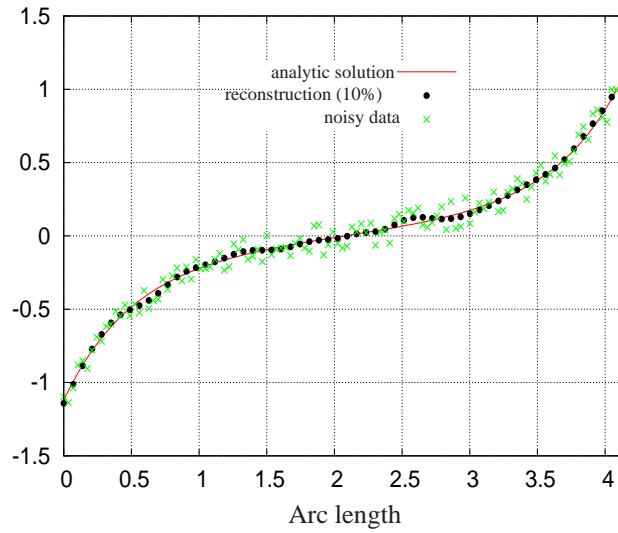


Figure 24. u' reconstruction on Γ_d and the noisy data ψ_d when the noise level is 10%

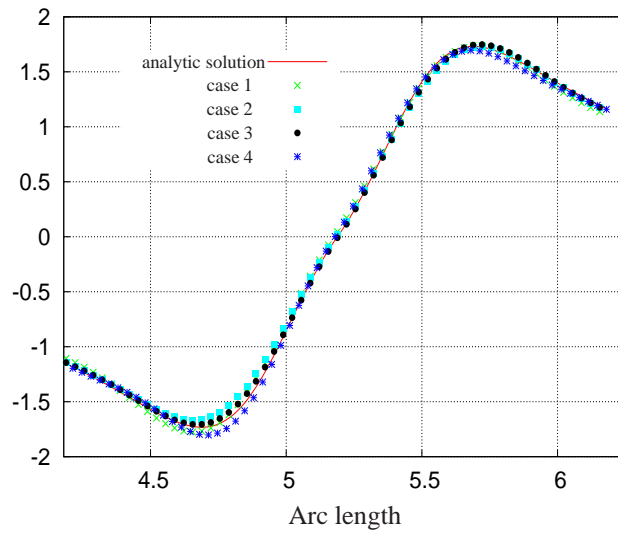


Figure 25. u reconstruction on Γ_u

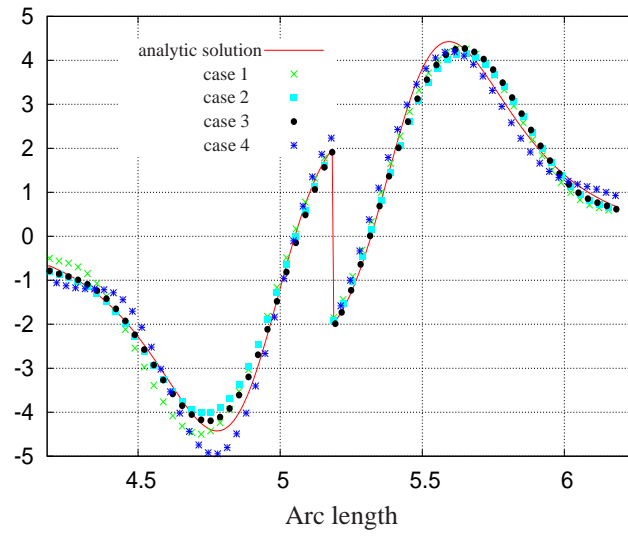


Figure 26. u' reconstruction on Γ_u

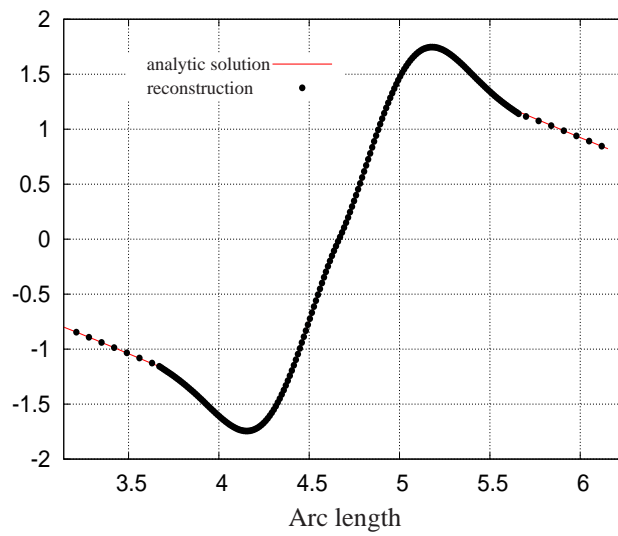


Figure 27. u reconstruction on Γ_u

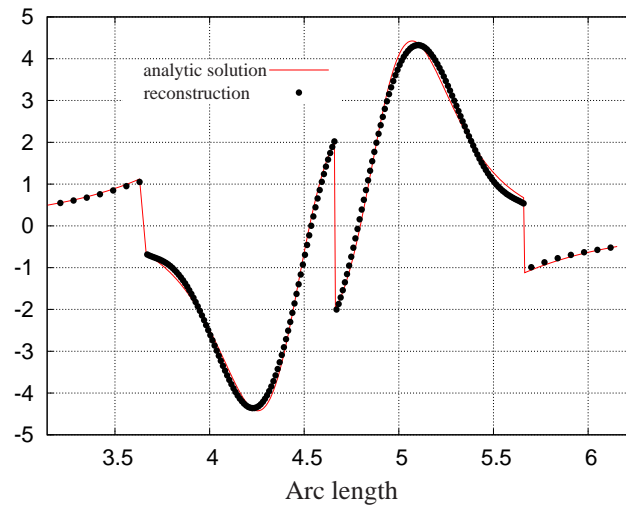


Figure 28. u' reconstruction on Γ_u

7. Concluding remarks

In this paper, an inverse method has been introduced for solving 2D Cauchy problems associated with the Laplace equation. This method is based on the one hand on the fading regularization effect introduced by Cimetière et Al. (Cimetière *et al.*, 2001) and on the other hand on the same principle as the first-order inverse method introduced by Delvare and Cimetière (Delvare *et al.*, 2008) but this method does not require the numerical evaluations of additional data. This method relies on a system of two weak integral formulations which connect the two partial derivatives of a harmonic function and not their normal derivatives. This system is used to introduce first order penalization terms in the functional to be minimized. The numerical simulations in 2D situations prove the efficiency and the robustness of this approach when dealing with regular solutions and that the method is more accurate than methods introduced in previous papers (Cimetière *et al.*, 2001; Cimetière *et al.*, 2002; Delvare *et al.*, 2008). In particular, it gives u' with increased accuracy when the boundary has corners, it is stable with respect to strong perturbations on the data and is able to deblur the two boundary data ϕ_d and ψ_d when they are noisy. The problem remains open whether the method can be adapted to reconstruct non-regular solutions.

Acknowledgements

The authors would like to thank the reviewers for their valuable comments.

8. References

- Andrieux S., Baranger T. N., Ben Abda A., “Solving Cauchy problems by minimizing an energy-like functional”, *Inverse Problems*, vol. 22, p. 115-133, 2006.
- Bourgeois L., “A mixed formulation of quasi-reversibility to solve the Cauchy problem for the Laplace equation”, *Inverse Problems*, vol. 21, p. 1087-1104, 2005.
- CASTEM2000, *Code de calcul pour l'analyse de structures par la méthode des éléments finis. Guide d'utilisation.*, Commissariat l'Energie Atomique, DEN/DM2S/SEMT/LM2S, 1998.
- Chen K. H., Kao J. H., Chen J. T., Wu K. L., “Desingularized meshless method for solving Laplace equation with over-specified boundary conditions using regularization techniques”, *Computational Mechanics*, vol. 43, p. 827-837, 2009.
- Cimetière A., Delvare F., Jaoua M., Pons F., “Solution of the Cauchy problem using iterated Tikhonov regularization”, *Inverse Problems*, vol. 17, n° 3, p. 553-570, 2001.
- Cimetière A., Delvare F., Jaoua M., Pons F., “An inversion method for harmonic functions reconstruction”, *International Journal of Thermal Sciences*, vol. 41, p. 509-516, 2002.
- Cimetière A., Delvare F., Pons F., “Une méthode inverse avec régularisation évanescence”, *C.R. Acad. Sci. Paris Tome IIb*, vol. 328, p. 639-644, 2000.
- Cimetière A., Delvare F., Pons F., “Une méthode inverse d'ordre un pour les problèmes de complétion de données”, *Comptes Rendus Mécanique*, vol. 333, p. 123-126, 2005.
- Delvare F., Cimetière A., “A first order method for the Cauchy problem for the Laplace equation using BEM”, *Computational Mechanics*, vol. 41, p. 789-796, 2008.
- Delvare F., Cimetière A., Hanus J. L., Bailly P., “An iterative method for the Cauchy problem in linear elasticity with fading regularization effect”, *Comput. Methods Appl. Mech. Engrg.*, vol. 199, p. 3336-3344, 2010.
- Delvare F., Cimetière A., Pons F., “An iterative boundary element method for Cauchy inverse problems”, *Computational Mechanics*, vol. 28, p. 291-302, 2002.
- Engl H. W., Hanke M., Neubauer A., *Regularization of Inverse problems*, Kluwer Academic Publishers, Dordrecht, 1996.
- Hadamard J., *Lectures on Cauchy's problem in linear partial differential equations*, Yale University Press, New Haven, 1923.
- Hao D. N., Lesnic D., “The Cauchy problem for the Laplace's equation via the conjugate gradient method”, *IMA Journal of Applied Mathematics*, vol. 65, p. 199-217, 2000.
- Hayashi K., Ohura Y., Onishi K., “Direct method of solution for general boundary value problem of the Laplace equation”, *Engineering Analysis with Boundary Elements*, vol. 26, p. 763-771, 2002.
- Jourhmane M., Lesnic D., Mera N. S., “Relaxation procedures for an iterative algorithm for solving the cauchy problem for the laplace equation.”, *Engineering Analysis with Boundary Elements*, vol. 28, p. 655-665, 2004.
- Klibanov M. V. and F. Santosa F., “A computational quasi-reversibility method for Cauchy problems for Laplace's equation”, *SIAM Journal of Applied Mathematics*, vol. 51, p. 1653-1675, 1991.
- Kozlov V. A., Maz'ya V. G., Fomin A. F., “An iterative method for solving the Cauchy problem for elliptic equations”, *Comput. Math. Phys.*, vol. 31, n° 1, p. 45-52, 1991.
- Lattès R., Lions J.-L., *Méthode de quasi-réversibilité et applications*, Dunod, Paris, 1967.

- Lesnic D., Elliott L., Ingham D. B., “An iterative boundary element method for solving the Cauchy problem for the Laplace equation”, *Engineering Analysis with Boundary Elements*, vol. 20, p. 123-133, 1997.
- Marin L., “Numerical solution of the Cauchy problem for steady-state heat transfer in two-dimensional functionally graded materials”, *International Journal of Solids and Structures*, vol. 42, p. 4338-4358, 2005.
- Marin L., “An alternating iterative MFS algorithm for the Cauchy problem in two-dimensional anisotropic heat conduction”, *Computers, Materials & Continua*, vol. 12, p. 71-100, 2009a.
- Marin L., “An iterative MFS algorithm for the Cauchy problem associated with the Laplace equation”, *Computer Modeling in Engineering & Sciences*, vol. 48, p. 121-152, 2009b.
- Marin L., “Relaxation procedures for an iterative MFS algorithm for two-dimensional steady-state isotropic heat conduction Cauchy problems”, *Engineering Analysis with Boundary Elements*, vol. 35, p. 415-429, 2011.
- Tikhonov A. N., Arsenin V. Y., *Solution of ill-posed problems*, John Wiley and Sons, New York, 1977.

Received: 16 February 2011

Accepted: 26 June 2011