Advanced Mathematical Models & Applications
Vol.6, No.2, 2021, pp.162-173

PARALLEL NUMERICAL COMPUTATION OF AN ANALYTICAL METHOD FOR SOLVING AN INVERSE PROBLEM

Nachaoui Abdeljalil[1*], **Nachaoui Mourad**[2], **Bayram Gasimov**[3]

[1]Laboratoire de Mathématiques Jean Leray, Université de Nantes, Nantes, France
[2]Laboratoire de Mathématiques et Applications, Université Sultan Moulay Slimane, Béni-Mellal, Maroc
[3]Azerbaijan State University of Economics (UNEC), Baku, Azerbaijan

Abstract. We solve the inverse Cauchy problems for the Laplace and Poisson equations using a closed-form regularization analytical solution. The formula is written in the form of series whose terms are integrals. The discretization is done using a Gaussian quadrature and the finite sum thus obtained is implemented on distributed memory using the Message Passing Interface (MPI). A detailed study of the parallelization procedure is presented.

Keywords: Inverse Cauchy problem, analytical regularization solution, parallelization, Message Passing Interface.
AMS Subject Classification: 65N21, 65Y05.
Corresponding author: Abdeljalil Nachaoui, Laboratoire de Mathématiques Jean Leray, Université de Nantes, Nantes, France, e-mail: *Abdeljalil.Nachaoui@univ-nantes.fr*
Received: 13 April 2021; Revised: 28 June 2021; Accepted: 7 August 2021; Published: 24 August 2021.

1 Introduction

In this work, we are interested in solving an inverse Cauchy problem for Elliptic equation. Given a domain Ω and two separate parts Γ_0 and Γ_1 of its boundary $\partial\Omega$, this problem, understood as the boundary value completion on Γ_1, from given Cauchy data on Γ_0, of an harmonic potential defined over Ω, has been widely studied because of its various medical and engineering applications. Many examples in the literature show that this Cauchy problem can be approached as a linear inverse problem, severely ill-posed in the Hadamard sense Hadamard (1953).

To solve the Cauchy problem governed by the Poisson equation, many numerical methods were proposed Cimetière et al. (2001); Chakib & Nachaoui (2006); Chang et al. (2001); Chen et al. (1998); Chi et al. (2009); Fang et al. (2004). Among the many numerical methods, the schemes based on iteration have also been developed previously by Kozlov et al. (1991), Jourhmane & Nachaoui (1996, 1999, 2002). The procedure introduced in the last papers drastically reduced the number of iterations required to achieve convergence. It was used in elasticity Ellabib & Nachaoui (2008); Marin & Johansson (2010), and recently for Cauchy problem governed by Stocks equation Chakib et al. (2018) and for the Helmholtz equation Berdawood et al. (2020, 2021); Berdawood-Nachaoui et al. (2021) and Berdawood-Nachaoui et al., 2021. It was also used in (Essaouini et al. (2004)) and Essaouini & Nachaoui. (2004) for a non linear elliptic problem and recently in Aboud et al. (2021) for Cauchy problems on inhomogenious material. Other methods have been developed for solving Cauchy's problems. The reader can consult for example Berntsson et al. (2017); Bergam et al. (2019); Choulli (2009); Ellabib et al. (2021); Isakov (2017); Juraev (2019, 2020); Kabanikhin (2012); Kabanikhin et al. (2013); Lavrent'ev

(2013); Nachaoui (2003); Nachaoui et al. (2021); Rasheed et al. (2021) and the references therein. In contrast to those methods Liu (2011) provided an analytical regularization solution without resorting to iteration.

2 Description of the analytical method

Let us consider $\Omega =]0, L[\times]0, b[$ with its boundary $\partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ where $\Gamma_0 =]0, L[\times\{b\}$, $\Gamma_1 =]0, L[\times\{0\}$, $\Gamma_2 = \{0\}\times]0; b[$ and $\Gamma_3 = \{L\}\times]0; b[$. The Cauchy problem for Laplace equation that will be considered is as follows:

$$\begin{aligned} \Delta u(x,y) &= 0, \text{ in } \Omega, && (1)\\ u(x,y) &= 0, \quad \text{on } \Gamma_2 \cup \Gamma_3, && (2)\\ u(x,y) &= 0, \quad \text{on } \Gamma_0, && (3)\\ \partial_y u(x,y) &= h(x), \quad \text{on } \Gamma_0, && (4) \end{aligned}$$

where $h(x)$ is a given function. Problem (1)-(4) is a theoretical model describing some real phenomenon. The model clearly requires that $h(x)$ belongs to a certain class M of functions where problems (1)-(4), have a solution, and it makes sense to extend that solution over Γ_1 in the same sense in which boundary conditions over Γ_0 are given.

In this section we will take up the idea described in Liu (2011). This idea consists of the use of Fourier series in order to transform the approximation of the Cauchy problem into a first-kind Fredholm integral equation for the unknown function of data.

After some computation, one is brought to the search for a function f^α solution of a problem regularized by a parameter $\alpha > 0$ and whose resolution leads to an analytical expression which is a closed-form regularized solution of the initial Cauchy problem.

We begin by considering the following boundary condition instead of Eq. (4):

$$u(x,0) = f(x), \ 0 \le x \le L, \tag{5}$$

where $f(x)$ is an unknown function to be determined.

In other word, Suppose that the datum $f(x)$ in Eq. (5) is unknown, but the Neumann datum $h(x)$ in Eq. (4) is overspecified. Determine the unknown function $f(x)$.

In this case, by the separation of variables method, solution of the problems (1)-(3) and (5) in Ω is given by

$$u(x,y) = \sum_{k=1}^{\infty} \frac{a_k \sinh[(b-y)k\pi/L]}{\sinh(bk\pi/L)} \sin(\frac{k\pi x}{L}), \tag{6}$$

where

$$a_k = \int_0^L f(t) \sin(\frac{k\pi t}{L}) dt. \tag{7}$$

The partial derivative of (6) with respect to y combined with condition (4) and Eq. (7) leads to

$$\int_0^L K(x,t) f(t) dt = h(x), \tag{8}$$

with

$$K(x,t) = \frac{2\pi}{L^2} \sum_{k=1}^{\infty} \frac{k}{\sinh(bk\pi/L)} \sin(\frac{k\pi x}{L}) \sin(\frac{k\pi t}{L}), \tag{9}$$

is a kernel function.

Thus, the solution $f(x)$ of the Cauchy problem (1)-(4), is given by solving the first-kind Fredholm integral (8). This is however a quite difficult task, since this integral equation is highly ill-posed.

Instead of (8), Liu (2011) propose to find $f(x)$ as a solution of the second-kind Fredholm integral equation

$$\alpha f(x) + \int_0^L K(x,t)f(t)dt = h(x), \tag{10}$$

where α is a Lavrentiev regularization parameter.

2.1 Approximation of the integral equation

Assume that the kernel function can be approximated by m terms with

$$K(x,t) = \frac{2\pi}{L^2}\sum_{k=1}^{m} \frac{k}{\sinh(kb\pi/L)} \sin(\frac{k\pi x}{L}) \sin(\frac{k\pi t}{L}), \tag{11}$$

such that it can be rewritten as an inner product

$$K(x,t) = P^T(x).Q(t), \tag{12}$$

where P and Q the $m-$vectors whose components are defined by

$$P_k(x) = \frac{2\pi}{L^2}\frac{k}{\sinh(bk\pi/L)} \sin(\frac{k\pi x}{L}) \text{ and } Q_k(x) = \sin(\frac{k\pi t}{L}), \text{ for } k = 1, \cdots, m.$$

The introduction of (12), in (10) gives

$$\alpha f(x) + \int_0^x P(x)^T Q(t)f(t)dt + \int_x^L P(x)^T Q(t)f(t)dt = h(x), \tag{13}$$

which can be written as

$$\alpha f(x) + P(x)^T[u_1(x) - u_2(x)] = h(x). \tag{14}$$

where

$$u_1(x) = \int_0^x Q(t)f(t)dt, \tag{15}$$

and

$$u_2(x) = \int_x^L Q(t)f(t)dt. \tag{16}$$

The differentiation of $u_1(.)$ and $u_2(.)$ with respect to x gives rise to the following two-point boundary value problem, which can be used to solve $u_1(x)$ and $u_2(x)$, and then $f(x)$ can be calculated from Eq. (13):

$$\begin{aligned} \alpha u_1'(x) &= Q(x)P^T(x)[u_2(x) - u_1(x)] + h(x)Q(x) \\ &\quad \text{and } \ u_1(0) = 0, \end{aligned} \tag{17}$$

$$\begin{aligned} \alpha u_2'(x) &= Q(x)P^T(x)[u_2(x) - u_1(x)] + h(x)Q(x), \\ &\quad \text{and } \ u_2(L) = 0. \end{aligned} \tag{18}$$

Using the fact that $u_1(x) - u_2(x) = c$ is a constant vector, the initial conditions in the problem (17)-(18) and the following orthogonality relation

$$\int_0^L \sin(\frac{j\pi x}{L}) \sin(\frac{k\pi x}{L})dx = \frac{L}{2}\delta_{jk}, \tag{19}$$

where δ_{jk} is the Kronecker delta, we find, after some calculations, the following formula

$$f(x) = \frac{1}{\alpha}h(x) - \frac{2}{L\alpha}\sum_{k=1}^{m} \frac{k\pi}{\alpha L \sinh(bk\pi/L) + k\pi} \int_0^L \sin(\frac{k\pi x}{L}) \sin(\frac{k\pi t}{L})h(t)dt.$$

For a given $h(x)$, through some integrals one may employ the above equation to comput $f(x)$ very efficiently. Once f is calculated, we insert it into the equation (7) and we use the orthogonality relation (19) and we recover

$$a_k^\alpha = \frac{2\sinh(\frac{kb\pi}{L})}{\alpha L \sinh(\frac{kb\pi}{L}) + k\pi} \int_0^L \sin(\frac{k\pi t}{L}) h(t) dt, \tag{20}$$

This last expression in eq. (6) permits to obtain an analytically regularization solution

$$u^\alpha(x,y) = \sum_{k=1}^{\infty} a_k^\alpha \frac{\sinh[k(b-y)\pi/L]}{\sinh(kb\pi/L)} \sin(\frac{k\pi x}{L}), \tag{21}$$

We then have the following result, the proof can be found in Liu (2011).

Theorem 1. *If the function h is bounded on the interval $[0, L]$ then for all $\alpha > 0$ and for all $y_0 > 0$, the solution u^α converges uniformly to u for all (x, y) of $[0, L] \times [y_0, b]$.*

3 Numerical approximation

3.1 Approximation of the integral

Replacing in (21) a_k^α by its formula given in (20) allows to write f^α and u^α using the same formula which shows an integration on $]0, L[$ and we obtain $\forall x \in [0, L]$

$$f^\alpha(x) = \frac{1}{\alpha} h(x) - \frac{2\pi}{\alpha L} \sum_{k=1}^{\infty} \frac{k \sin\left(\frac{k\pi x}{L}\right)}{\alpha L \sinh\left(\frac{bk\pi}{L}\right) + k\pi} \int_0^L \sin\left(\frac{k\pi\xi}{L}\right) h(\xi) \mathrm{d}\xi \tag{22}$$

and $\forall (x, y) \in [0, L] \times [0, b]$

$$u^\alpha(x,y) = 2 \sum_{k=1}^{\infty} \frac{\sinh\left(\frac{(b-y)k\pi}{L}\right) \sin\left(\frac{k\pi x}{L}\right)}{\alpha L \sinh\left(\frac{bk\pi}{L}\right) + k\pi} \int_0^L \sin\left(\frac{k\pi\xi}{L}\right) h(\xi) \mathrm{d}\xi. \tag{23}$$

Then, the equations (22) and (23) reveal one-dimensional integrals which should be approached by quadrature formulas. In this context a quadrature using N_q points ζ, weighted by weights w is written:

$$\int_0^L \sin\left(\frac{k\pi\xi}{L}\right) h(\xi) \mathrm{d}\xi \approx \sum_{l=1}^{N_q} w_l \sin\left(\frac{k\pi\zeta_l}{L}\right) h(\zeta_l).$$

The key point in this quadrature approach is the trade-off between the precision and the number of points used. We opt for a Gauss-Legendre quadrature which presents, subject to a sufficient regularity of the integrand, a factorial convergence.

Note that these formulas are well prepared for a parallelization since we can calculate the value of f^α or $u^\alpha(x, y)$ for an x or a couple (x, y) independently of the other points. So to approach the solution in the domain Ω we have no constraint on the discretization. We can therefore use a uniform structured mesh composed of $N_x + 1$ points in the direction x and of $N_y + 1$ points in the direction y. The functions considered will be evaluated at the nodes (x_i, y_j) of this mesh defined as follows:

$\forall (i, j) \in \{0, ..., N_x\} \times \{0, ..., N_y\}$

$$x_i = \frac{iL}{N_x} \quad \text{et} \quad y_j = \frac{jb}{N_y}.$$

Finally, the expressions of f^α and u^α use series that have to be truncated. In practice, we will only keep m terms.

Thus, the quantities of interest f^α and u^α will be approached at the nodes (x_i, y_j) by the following discrete quantities:

$$f^\alpha(x_i) \approx \frac{1}{\alpha}h(x_i) - \frac{2\pi}{\alpha L}\sum_{k=1}^{m}\sum_{l=1}^{N_q}\frac{w_l k \sin\left(\frac{k\pi x_i}{L}\right)\sin\left(\frac{k\pi\zeta_l}{L}\right)h(\zeta_l)}{\alpha L \sinh\left(\frac{bk\pi}{L}\right) + k\pi}, \tag{24}$$

$$u^\alpha(x_i, y_j) \approx 2\sum_{k=1}^{m}\sum_{l=1}^{N_q}\frac{\sinh\left(\frac{(b-y_j)k\pi}{L}\right)\sin\left(\frac{k\pi x_i}{L}\right)}{\alpha L \sinh\left(\frac{bk\pi}{L}\right) + k\pi} w_l \sin\left(\frac{k\pi\zeta_l}{L}\right)h(\zeta_l). \tag{25}$$

3.2 Parallelization strategy

During the numerical illustration phase of the analytical method, we want to efficiently estimate the quantities f^α, u^α and the convergence result. We therefore have to parallelize the spatial dimensions, the calculation of sums and of a maximum on a tensor of dimension two.

It is first of all clear that the precision of the analytical method is independent of the mesh used, so it will not be useful in practice to lengthen the calculations with a fine mesh. Therefore we will have $m \gg \max(N_x, N_y)$ et $N_q \gg \max(N_x, N_y)$. The parallelization of a calculation expressed as an independent term sum is naturally done by distributing on each processor portions of sums as homogeneous as possible. Without additional assumptions, we should parallelize the sums on the indices k and l of the expressions (24) and (25). Now, if we assume enough regularity over h to reach the factorial convergence of the Gausse-Legendre quadrature, then the error made by the quadrature will necessarily be dominated by the error of truncation of the sum of index k and parameterized by m. Thus, for a numerical computation whose precision is parameterized by the data of a triplet (α, m, N_q) we can choose in practice $N_q \ll m$ and this without altering the performances of the analytical method. Therefore, we can only parallelize the sum bearing on the index k and parameterized by m.

The parallel calculation of a maximum on tensor can be carried out according to various methods. We opt for an approach in which each processor reproduces the maximum value of a portion of tensor. These division of tasks into independent zones makes it possible to favor integer exchanges between processor which is appreciable for the modularity of the precision of the floats.

The parallelization is therefore done in the following form:

1. The first case is when the number of processors *nprocs* is less than or equal to N_x. In this case, we share the processors on the iterations in x. If $N_x = nprocs$, then each processor will process an iteration. If $nprocs < N_x$ then some processors will process more than one iteration. We then introduce q, a quotient of Euclidean division defined by: $q = \frac{N_x}{nprocs}$ and a remainder r of this Euclidean division: $r = N_x - q \times nprocs$. We then distribute, in the same way as in the parallelizations of sums on u and f, the iterations in x.

   We then define a new quotient $\hat{q} = 0$. This allows us to calculate the parallelized sums of u and f. Indeed, in the case where $nprocs \leq N_x$, there is only one processor per iteration. Therefore, no parallelization can be performed on the sums of f and u and the processor corresponding to its iteration must perform the entire sum on the k of u and f. This is why we assign $\hat{q}$ to 0.

2. The second case is that where the number of processors *nprocs* is strictly greater than N_x. In this case, we share the iterations on the processors. If $nprocs > N_x$ then some iterations in x will be processed by several processors and this will allow access to parallelization on the sums in k of u and f.

For that, one calls the subroutine MPI (MPI_Comm_SPLIT) which makes it possible to create under groups of processors. We establish another quotient q^* defined by: $q^* = \frac{nprocs}{N_x}$ as well as the remainder of the Euclidean division: $r^* = nprocs - q^* \times N_x$. We establish a new rank $key = new\_rang = \frac{old\_rang}{N_x}$ and an attribute which allows to group the the processors together according to the number N_x : $color = old\_rang - key \times N_x$.

We then obtain a new communicator "new_comm" with new ranks for the processors. The iterations in x are then processed by several processors, depending on the *color* attribute.

We now have to gather the parts of parallelized sums of u and f calculated by the slave processors in the master processors. There is still a subtlety, because if r^* $ne0$, then for some iterations there will not be the same number of processors as for others. From the way in which the subgroups are made, we know which processors are involved. These are those for which the starting rank is included in the interval $[nprocs - r^*, nprocs)$. The affected processors will have a larger group.

Having divided the operations, it remains to decide on the mode of communication to adopt. As one must carry out standard operations of sum and maximum, it is natural to use global communications of reduction from Message Passing Interface library (`mpi_reduce`) and of diffusion (`mpi_bcast`).

4 Numerical results

We place ourselves on a rectangle such as $L = pi$, $b = 1$ which we discretize by a mesh verifying $N_x = N_y = 50$, and we consider the following Laplace problem: for $p \in \mathbb{N}^*$

$$\left\{\begin{array}{rcll} \Delta u & = & 0 & \text{dans } [0,\pi] \times [0,1] \\ u(0,y) & = & 0, & 0 \leq y \leq 1 \\ u(\pi,y) & = & 0, & 0 \leq y \leq 1 \\ u(x,1) & = & 0, & 0 \leq x \leq \pi \\ -\frac{\partial u}{\partial y}(x,1) & = & h(x) = \frac{\sin(px)}{p}, & 0 \leq x \leq \pi. \end{array}\right. \tag{26}$$

With this choice of function h, the exact solution of problem (26) is given by

$$u(x,y) = \frac{\sinh(p(1-y))}{p^2}\sin(px) \quad \text{and} \quad f(x) = \frac{\sinh(p)}{p^2}\sin(px).$$

This is a classic example illustrating the ill-posed nature of the Cauchy problem. The function h tends to zero when p becomes very large while $|u(x,y)|$ can be very large, which proves the non-continuity of the solution with respect to the data. The function h chosen being very regular, the quadrature of Gauss-Legendre satisfied (see Kovvali (2011); Rappaz et al. (2004) for the detail of the general proof): $\forall (k,p) \in \{1,...,m\} \times \mathbb{N}^*$

$$\int_0^L \sin\left(\frac{k\pi x}{L}\right)\frac{\sin(px)}{p}\mathrm{d}x - \sum_{l=1}^{N_q} w_l \sin\left(\frac{k\pi\zeta_l}{L}\right)\frac{\sin(p\zeta_l)}{p}$$
$$= \frac{\mathrm{d}^{2N_q+1}}{\mathrm{d}x^{2N_q+1}}\left[\sin\left(\frac{k\pi x}{L}\right)\frac{\sin(px)}{p}\right](\xi)\frac{L^{2N_q+1}(N_q!)^4}{(2N_q+1)(2N_q)!^3}$$
$$\text{with} \quad \xi \in [0,L]$$

If we calculate the derivative $2N_q + 1$ of the integrand we find: $\forall (k,p) \in \{1,...,m\} \times \mathbb{N}^*$

$$\left|\int_0^L \sin\left(\frac{k\pi x}{L}\right)\frac{\sin(px)}{p}\mathrm{d}x - \sum_{l=1}^{N_q} w_l \sin\left(\frac{k\pi\zeta_l}{L}\right)\frac{\sin(p\zeta_l)}{p}\right| \leq$$
$$\frac{1}{2p}\left(\left(\frac{k\pi}{L}-p\right)^{2N_q+1} + \left(\frac{k\pi}{L}+p\right)^{2N_q+1}\right)\frac{L^{2N_q+1}(N_q!)^4}{(2N_q+1)(2N_q)!^3}$$

$$\leq \frac{1}{2p}\left(\left(\frac{m\pi}{L}-p\right)^{2N_q+1}+\left(\frac{m\pi}{L}+p\right)^{2N_q+1}\right)\frac{L^{2N_q+1}(N_q!)^4}{(2N_q+1)(2N_q)!^3}$$

Then, one can show that the asymptotic behavior of this upper bound for large m and N_q is:

$$\frac{(m\pi)^{2Nq+1}}{16\pi p(2Nq+1)N_q 2^{6N_q}}\left(\frac{e}{N_q}\right)^{2N_q}.$$

From this, one can concludes that at large fixed m, the truncation error dominates the quadrature error.

4.1 Parallelization efficiency

The first interesting point to observe is the performance gain in terms of computing time provided by parallelization. To compare these performances to a sequential code, we measure the CPU time necessary for the reconstruction of the edge condition f^α by taking the triplet $m = 10, \alpha = 10^{-2}, p = 5$. We obtain the following results:

Table 1: Speed of execution of the code according to the number of processors

Nombre Processeurs	Temps
1	1.15
2	0.57
4	0.28
8	0.15
12	9.58E-002
20	6.95E-002

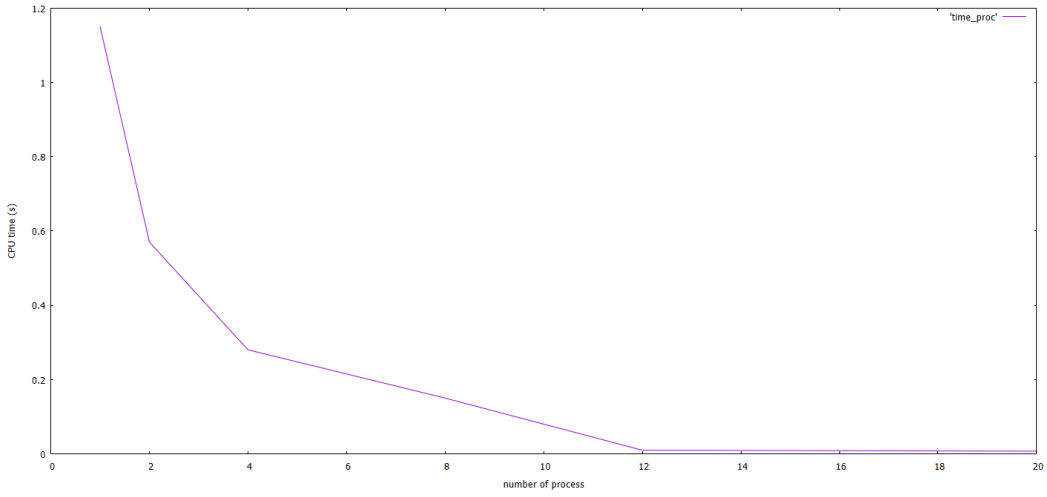

Figure 1: CPU parallelization efficiency

In table 1 and figure 1, we display the results for a first case where the number of points is greater than the number of processors.

We observe that the execution time is practically halved each time we multiply the number of processors by two with a slight difference which is due to the communication. It continues to decrease as the number of processors increases to 12 processors and then stagnates. Cesi

is in accordance with forecasts at a time the communication between the processors becomes important in front of the saving in time obtained by the increase in the number of processors. In table 2 we present the results for the case where the number of points is low and where the

Table 2: Computing time for number of point less than number of process

Nombre Processeurs	Temps
4	5.3E-002
8	4.9E-002
12	4.7E-002
20	4.1E-002

number of processors is possibly greater than this number of points. We take $Nx = 7$ and $m = 100$. Here the execution time is too low to hope to see an increase in performance when the number of processors increases. Increase the number of terms m by the sum. We observe

Table 3: Computing time for large m

Nombre Processeurs	Temps
4	0.76
8	0.69
12	0.67
20	0.38

from table 3 an increase in performance as the number of processors increases. However this increase is less strong than in the case where the number of points is higher than the number of processors and where we only parallelize the sum in m. This is explained by the fact the number d The operation required in the parlization of the mesh is more important.

4.2 Boundary data reconstruction

The second interesting point to observe is the reconstraction of the regularized estimator f^α. For different values of triplets (m, p, α). The following graphs are obtained:

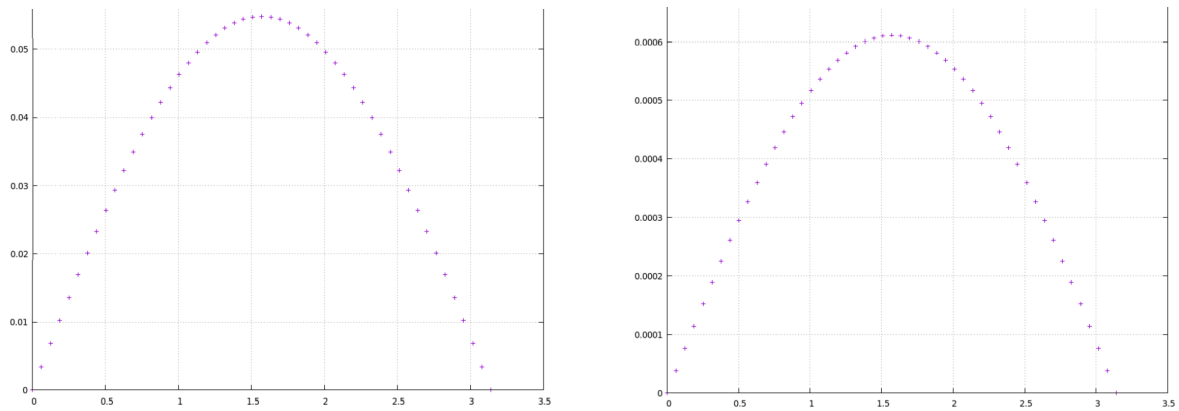

Figure 2: Approximation error for f^α, with $m = 10$ and $\alpha = 0.00001$ left : $p = 1$, right: $p = 2$

First, we fixed m and we varied α. We have displayed the error in the approximation of f^α in the graphs Fig. 2 and Fig. 3 . We observe from these graphs that the reconstruction of f^α in the case where the problem solution (26) is not very oscillating is very correct even for large values of α.

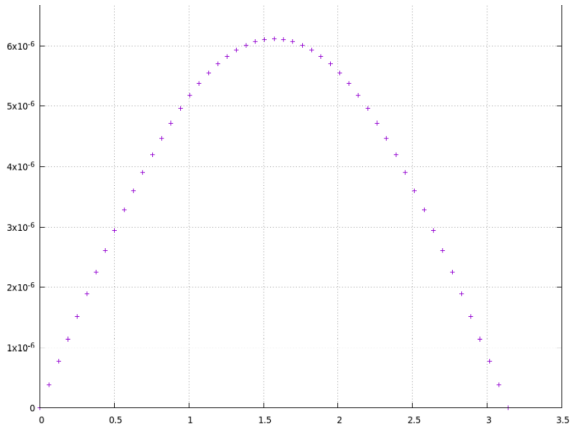

Figure 3: Approximation error for f^α, with $\alpha = 0.00001$, $m = 10$

Secondly, we set $\alpha = 0.001$ and we varied m. We have noticed that for $m = 1, 10, 100$, the approximation error of f^α does not vary very much. This is well in line with the theory. Indeed in this case the analytical solution coincides with the exact solution from the first term, increasing m does not bring anything to the level of precision in this case, it is well what we have observed.

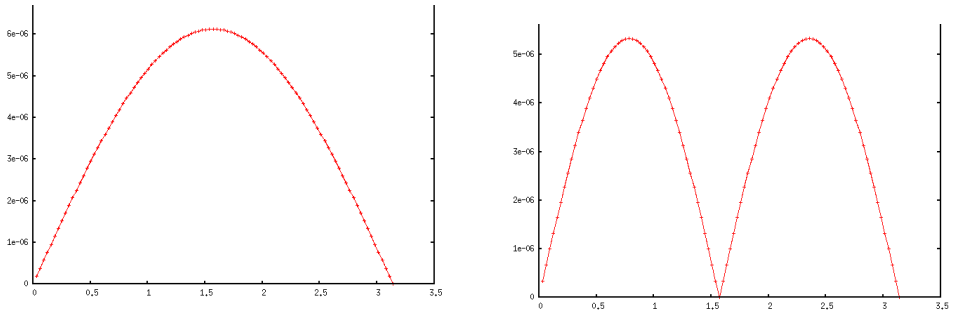

Figure 4: Approximation error for f^α, with $m = 10$ and $\alpha = 0.00001$ left : $p = 1$, right: $p = 2$

We also carried out numerical tests where we took $m = 5$ and we varied p and α. The error corresponding to f^α is displayed in the graph fig. 4 for and $alpha = 0.00001$ and p taking the value 1 and 2. We notice that the error is very good for both cases. We have observed on other results that the approximation error decreases as α decreases.

We noticed also that the ill-posed character of the problem (26) appears quickly (from $p = 10$) and in this case it is imperative to select very small values of α.

5 Conclusion

It is first of all clear that parallelization makes it possible to drastically reduce the computing time. However, this saving of time is not arithmetic with the number of processors engaged and this due to the fact of an increasing communication time. This increasing communication time is undoubtedly also at the origin of the increasing dispersion of the calculation times observed.

Thus in order to take advantage of parallelization, it would therefore be necessary not to take too large a number of processors, this makes it possible to accelerate the time of calculation and to avoid its disturbance by the increase of the time of communication.

Overall, we observe that the order of convergence can depend on the ill-posed nature of the problem. On the other hand, this character influences for several decades the absolute error committed. This probably comes from the fact that the exact solution u becomes very oscillating and has large amplitudes as p increases.

6 Acknowledgement

All the calculations were made on dual processor hexa-core Xeon (XE340) computing nodes at 2.66 GHz at **CCIPL** (The Intensive Computing Center of Pays de la Loire).

References

Aboud, F., Nachaoui, A. & Nachaoui, M. (2021). On the approximation of a Cauchy problem in a non-homogeneous medium. *J. Phys.: Conf. Ser.*, 1743 012003.

Berdawood, K.A., Nachaoui, A., Saeed, R., Nachaoui, M. & Aboud, F. (2020). An alternating procedure with dynamic relaxation for Cauchy problems governed by the modified Helmholtz equation. *Advanced Mathematical Models & Applications, 5*(1), 131-139.

Berdawood, K.A., Nachaoui, A., Saeed, R., Nachaoui, M. & Aboud, F. (2021). An efficient D-N alternating algorithm for solving an inverse problem for Helmholtz equation. *Discrete & Continuous Dynamical Systems - S*, (1), 131-139.

Berdawood, K.A., Nachaoui, A., Nachaoui, M. & Aboud, F. (2021). An effective relaxed alternating procedure for Cauchy problem connected with Helmholtz Equation. *Numer Methods Partial Differential Eq.* 1–27. https://doi.org/10.1002/num.22793

Bergam, A., Chakib, A., Nachaoui, A. & Nachaoui, M. (2019). Adaptive mesh techniques based on a posteriori error estimates for an inverse Cauchy problem. *Applied Mathematics and Computation*, *346*, 865-878.

Berntsson, F., Kozlov, V.A., Mpinganzima, L. & Turesson, B.O. (2017). Iterative Tikhonov regularization for the Cauchy problem for the Helmholtz equation. *Comput. Math. Appl., 73*(1), 163-172.

Cimetière, A., Delvare, F., Jaoua, M. & Pons, F. (2001) Solution of the Cauchy Problem Using Iterated Tikhonov Regularization, *Inverse Problems, 17*, 553-570.

Chakib, A. & Nachaoui, A. (2006). Convergence analysis for finite element approximation to an inverse Cauchy problem. *Inverse Problems, 22*(4), 1191–1206.

Chakib, A., Nachaoui, A., Nachaoui, M. & Ouaissa, H. (2018). On a fixed point study of an inverse problem governed by Stokes equation. *Inverse Problems*, *35*(1), 015008.

Chang, J.R., Yeih, W. & Shieh, M.H. (2001). On the Modified Tikhonov's Regularization Method for the Cauchy Problem of the Laplace Equation. *Journal of Marine Science and Technology, 9*, 113-121.

Chen, J.T. & Chen, K.H. (1998). Analytical Study and Numerical Experiments for Laplace Equation with Overspecified Boundary Conditions. *Applied Mathematical Modelling, 22*, 703-725.

Chi, C.C., Yeih, W. & Liu, C.S. (2009). A Novel Method for Solving the Cauchy Problem of Laplace Equation Using the Fictitious Time Integration Method, *CMES-Computer Modeling in Engineering and Sciences, 47*, 167-190.

Choulli, M. (2009). *Une introduction aux problèmes inverses elliptiques et paraboliques*. Springer-Verlag, Berlin.

Ellabib, A. & Nachaoui, A. (2008). An iterative approach to the solution of an inverse problem in linear elasticity. *Mathematics and Computers in Simulation*, *77*(2-3), 189-201.

Ellabib, A., Nachaoui, A. & Ousaadane A. (2021). Mathematical analysis and simulation of fixed point formulation of Cauchy problem in linear elasticity. *Mathematics and Computers in Simulation*, *187*, 231-247.

Essaouini, M., Nachaoui, A. & El Hajji, S. (2004). Numerical method for solving a class of nonlinear elliptic inverse problems. *Journal of Computational and Applied Mathematics*, *162*(1), 165-181.

Essaouini, M., Nachaoui, A. & El Hajji, S. (2004). Reconstruction of boundary data for a class of nonlinear inverse problems. *Journal of Inverse and Ill-Posed Problems*, *12*(4), 369-385.

Fang, W. & Lu, M. (2004) A fast collocation method for an inverse boundary value problem. *International Journal for Numerical Methods in Engineering, 21*, 1563-1585.

Hadamard, J. (1953). *Lectures on Cauchy's problem in linear partial differential equations*. Dover Publications, New York.

Isakov, V. (2017). *Inverse problems for partial differential equations*. Springer, Cham.

Jourhmane, M. & Nachaoui, A. (1996). A relaxation algorithm for solving a Cauchy problem, *Proc. of the 2nd Internat. Conf.*, 151-158, *Foundation Editor*.

Jourhmane, M. & Nachaoui, A. (2002). Convergence of an alternating method to solve the Cauchy problem for Poisson's equation. *Applicable Analysis, 81*(5), 1065-1083.

Jourhmane, M. & Nachaoui, A. (1999). An alternating method for an inverse Cauchy problem. *Numerical Algorithms*, *21*(1-4), 247-260.

Juraev, D.A. (2019). On a regularized solution of the Cauchy problem for matrix factorizations of the Helmholtz Equation. *Advanced Mathematical Models & Applications, 4*(1), 86-96.

Juraev, D.A. (2020). The solution of the ill-posed Cauchy problem for matrix factorizations of the Helmholtz equation. *Advanced Mathematical Models & Applications, 5*(2), 205-2021.

Kabanikhin, S.I. (2012). *Inverse and ill-posed problems*. Walter de Gruyter GmbH & Co. KG, Berlin.

Kabanikhin, S.I., Gasimov, Y.S., Nurseitov, D.B., Shishlenin, M.A., Sholpanbaev, B.B., Kasemov, S. (2013). Regularization of the continuation problem for elliptic equation. *J. Inverse Ill-Posed Probl., 21*(6), 871-874.

Kovvali, N. (2012). *Theory and applications of Gaussian quadrature methods.* Synthesis Lectures on Algorithms and Software in Engineering 8. Morgan & Claypool Publishers, Williston.

Kozlov, V.A., Maz'ya, V.G. & Fomin, A.V. (1991). An iterative method for solving the Cauchy problem for elliptic equations. *Zh. Vychisl. Mat. i Mat. Fiz.*, *31*(1), 64-74.

Lavrentiev, M.M. (2013). *Some improperly posed problems of mathematical physics.* Springer Science & Business Media.

Liu, C.-S. (2011). An Analytical Method for the Inverse Cauchy Problem of Laplace Equation in a Rectangular Plat. *J. Mech., 27*(4), 575–584.

Marin, L. & Johansson, B.T. (2010). A relaxation method of an alternating iterative algorithm for the Cauchy problem in linear isotropic elasticity. *Computer Methods in Applied Mechanics and Engineering*, *199*(49-52), 3179-3196.

Nachaoui, A., Nachaoui, M. Chakib, A. & Hilal, M.A. (2021). Some novel numerical techniques for an inverse Cauchy problem. *Journal of Computational and Applied Mathematics, 381*, 113030.

Nachaoui, A. (2003). An improved implementation of an iterative method in boundary identification problems. *Numer. Algorithms*, *33*(1-4), 381-398.

Rappaz, J. & Picasso, M. (2004). *Introduction à l'analyse numérique.* Presses Polytechniques et Universitaires Romandes, Lausanne.

Rasheed, S.M., Nachaoui, A., Hama, M.F.& Jabbar, A.K. (2021). Regularized and preconditioned conjugate gradient like-methods methods for polynomial approximation of an inverse Cauchy problem. *Advanced Mathematical Models & Applications 6*(2), 89-105.