Genetic Algorithm-Based Optimization Approach for Solving a Class of Inverse Problems with Tikhonov Regularization

JAMAL DAOUDI[1], CHAKIR TAJANI[2*]
[1,2] SMAD Team, Department of Mathematics
Polydisciplinary Faculty of Larache
Abdelmalek Essaadi University
MOROCCO

Abstract: -In this paper, we are interested in solving the data completion problem for the Laplace equation. It consists to determine the missing data on the inaccessible part of the boundary from overspecified conditions in the accessible part. Knowing that this problem is severely ill-posed, we consider its formulation as an optimization problem using Tikhonov regularization. Then, we consider an optimization approach based on adapted Real Coded Genetic Algorithm (RCGA) to minimize the cost function and recover the missing data. The performed numerical simulations, with different domains, illustrate the accuracy and efficiency of the proposed method with an adequate regularization parameter, in addition to the good agreement between the numerical solutions and different noise level of the given data.

Key-Words: Inverse Problem, Tikhonov regularization , Genetic Algorithm, Data completion problem, Optimization

Received: July 9, 2023. Revised: September 28, 2023. Accepted: October 23, 2023. Published: November 14, 2023.

1 Introduction

According to [1], two problems are said to be "inverses" of each other if the formulation of one of them has implications for the other. This definition is somewhat arbitrary, but it does give the two problems a symmetrical role. A more operational way to define an inverse problem is that it consists to determine causes knowing effects. In other words, an inverse problem is the opposite of a direct problem, which consists of deducing effects given known causes.

The data completion problem for the Laplace equation is a important type of inverse problem which aims to retrieve Cauchy data for the inaccessible part of a boundary using measurements obtained from the accessible one. This problem arises in various applications. For instance, in Electrocardiography (ECG), [2], where the objective is to estimate the electrical activity within the heart based on surface measurements, assuming constant conductivity. In Electroencephalography (EEG), [3], the goal is to estimate the current source generated by neuronal activity, considering the electrical conductivity of tissue and measurements taken at specific points on the head's surface. In fissure detection, [4], the aim is to pinpoint and characterize cracks in a material using measurements obtained at the material's boundaries. In corrosion detection, [5], the emphasis is to deduce the shape of the corrosive boundary based on the collected data.

The data completion problem for the Laplace equation is known to be severely ill-posed problem. It does not satisfy the conditions of the well-posed problem, [6], as it is sensitive to measurement errors and can lead to unstable and inaccurate results. To remedy this, we need to introduce regularization methods. Among the well-known methods for inverse problems, we can cite; sequential function specification methods, [7], iterative methods [8], [9], and Tikhonov regularization, [10]. These methods try to find a balance between accuracy and stability.

In inverse problems, we aim to find the model parameters that best explain the observed data. This can be done by defining an objective function which measures the discrepancy between the model output and the data, and then minimizing that function. There are two main classes of optimization methods: Deterministic optimization methods are best suited for continuously differentiable objective functions with a limited number of local extrema. In contrast, stochastic optimization methods involve random sampling of the objective function across the entire feasible space and can be applied to both differentiable and non-differentiable objective functions.

Genetic algorithms (GAs) represent a subset of metaheuristics, which includes artificial bee colony, [11], particle swarm optimization, [12], ant colony optimization, [13], and the Bat algorithm, [14], among others. These methodologies draw inspiration from natural phenomena and have demonstrated their effectiveness in addressing a multitude of challenges spanning various domains. Although, GAs do not offer a guarantee of identifying the absolute best solu-

tion, they excel in identifying solutions that closely approach the optimal outcome within a reasonable time frame.

GAs have demonstrated notable success in tackling a range of inverse problems. For instance, they have been applied to estimate interaction mechanisms among system components, [15], solve groundwater source identification challenges, [16], address nonlinear IHCP scenarios, [17], compute minimizers of Tikhonov functionals, [18], detect the shape of unknown boundary segments in the solution domain for the Laplace equation, [19], identification of the Robin coefficient, [20], solve support vector machines (SVMs), [21], reconstruct epicardial potentials from body surface potentials, [22], identify isotropic inclu-sions through a single boundary measurement, [23], solve IHCP by minimizing a modified version of Tikhonov's functional, [24], and address the inverse problem of aerodynamic shape optimization, [25].

The main feature of our approach is the use of an optimization approach based on a Real Coded Genetic Algorithm (RCGA) as a powerful optimization tool capable of handling complex, non-monotonic objective functions. In [26], the authors addressed the data completion problem for the Laplace equation by reconstructing the Dirichlet condition in the inaccessible part of the boundary of a regular domain, without resorting to Tikhonov regularization. In this study, we extend this approach to two scenarios: regular and irregular domains, aiming to estimate the unknown Dirichlet and Neumann conditions. In addition, we assess the stability of the proposed approach by examining the optimal choice of the regularization parameter. Finally, we conduct a thorough analysis of the solution's stability under varying levels of noise.

The rest of this paper is organized as follows: Section 2 provides the mathematical formulation of the problem, including the differences between the forward and inverse problem, and its formulation as an optimization one. Section 3 provides a brief overview of GAs, in addition to the proposed approach based on adapted RCGA . Section 4 presents numerical results and discussion. Finally, Section 5, summarizes the main results and provides concluding thoughts.

2 Mathematical formulation

2.1 Forward and inverse problem

We consider the following Elliptic Cauchy problem:

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \\ u = f & \text{on } \Gamma_c \\ \partial_n u = g & \text{on } \Gamma_c \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is an open bounded set with Lipschitz continuous boundary Γ, which consists of tow disjointed parts $\Gamma = \Gamma_c \cup \Gamma_i$, with Γ_c is the accessible part of the boundary and Γ_i is the inaccessible part (Fig.1).

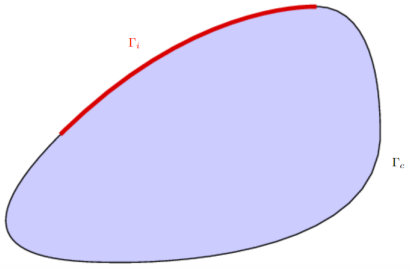

Figure 1: Example of the domain

The functions f and g are the given Dirichlet and Neumann conditions, and $\partial_n u$ is the normal derivative of the unknown function u.

Forward problem: the aim of the forward problem is to find u, by solving the Cauchy problem for Laplace equations from knowledge of the Cauchy data $u_D = u/_{\Gamma_i}$ or the $u_N = \partial_n u/_{\Gamma_i}$.

Inverse problem: the aim of the inverse problem is to estimate the Cauchy data (u_D, u_N) on the inaccessible part of the boundary Γ_i from the knowledge of the Cauchy data f and g on the accessible part of the boundary Γ_c.

2.2 Optimization problem

Since the u_D and u_N on the boundary Γ_i is to be determined, two direct problems are considered:

$$(P_D): \begin{cases} -\Delta u = 0 & \text{in } \Omega \\ \partial_n u = g & \text{on } \Gamma_c \\ u = u_D & \text{on } \Gamma_i \end{cases} \tag{2}$$

$$(P_N): \begin{cases} -\Delta u = 0 & \text{in } \Omega \\ u = f & \text{on } \Gamma_c \\ \partial_n u = u_N & \text{on } \Gamma_i \end{cases} \tag{3}$$

It is important to note that if $u_D \in H^{1/2}(\Gamma_i)$ and $g \in H^{-1/2}(\Gamma_c)$ (respectively, $f \in H^{1/2}(\Gamma_c)$ and $u_N \in H^{-1/2}(\Gamma_i)$), then it exists a unique solution $u(u_D, g)$ (respectively, $u(u_N, f)$) for the direct problem Eq.(2) (respectively, Eq.(3)). Our goal is to find u_D (respectively, u_N) that satisfies:

$$\begin{aligned} u(u_D, g) &= f \quad \text{on} \quad \Gamma_c \\ (\text{respectively}, \partial_n u(u_N, f) &= g \quad \text{on} \quad \Gamma_c) \end{aligned} \tag{4}$$

In doing so, we attempt to minimize the two well-posed least-squares functional $\mathcal{J}_{\mathcal{D}}$ and $\mathcal{J}_{\mathcal{N}}$ defined

by:

$$\mathcal{J}_{\mathcal{D}}(u) = \frac{1}{2}\|u(u_D, g) - f\|^2_{L^2(\Gamma_c)} + \frac{\alpha_1}{2}\|u_D\|^2_{L^2(\Gamma_i)} \tag{5}$$

$$\mathcal{J}_{\mathcal{N}}(u) = \frac{1}{2}\|\partial_n u(u_N, f) - g\|^2_{L^2(\Gamma_c)} + \frac{\alpha_2}{2}\|u_N\|^2_{L^2(\Gamma_i)} \tag{6}$$

where, α_1 and α_2 are the regularization parameters and $\frac{\alpha_1}{2}\|u_D\|^2_{L^2(\Gamma_i)}$ and $\frac{\alpha_2}{2}\|u_N\|^2_{L^2(\Gamma_i)}$ are the well-known Tikhonov regularization functional. The regularization serves to stabilize the solution. The process of minimizing $\mathcal{J}_{\mathcal{D}}$ and $\mathcal{J}_{\mathcal{N}}$ represents a delicate balance between achieving data fidelity and stabilizing the solution. The selected values for α_1 and α_2 directly impact the stability of the solution.

3 Genetic Approach for the inverse problem

3.1 Overview of Genetics Algorithms

Metaheuristics are a class of optimization algorithms that are designed to solve complex optimization problems for which traditional methods might be impractical or ineffective, [27]. They provide a general framework for exploring and searching large solution spaces to find near-optimal solutions. Metaheuristics are particularly useful when dealing with problems that involve non-linearities, discontinuities, and other challenging characteristics. Metaheuristic algorithms are usually inspired by natural phenomena, social behaviors, or problem-specific strategies. They don't guarantee finding the global optimal solution but aim to find good approximations in a reasonable amount of time.

Genetic Algorithms (GAs) are a classic example of metaheuristic algorithms based on the mechanisms of natural selection and genetics, [28]. It combines a "survival of the fittest" strategy with a random structured exchange of information. For a problem for which a solution is unknown, a set of possible solutions is created randomly. The characteristics (or variables to be determined) are then used in gene sequences that will be combined with other genes to form chromosomes and then individuals. Each solution is associated with an individual, and this individual is evaluated and classified according to its resemblance to the best, but still unknown, solution to the problem. It can be shown that through a natural selection process inspired by Darwin's theory of evolution, this method gradually approaches a solution. Natural selection is a potent force in evolution, leading to the development of intricate and well-adapted organisms. It operates on the principle that organisms with traits well-suited to their environment are more likely to survive and reproduce. This dynamic ultimately yields a population finely attuned to their surroundings. Genetic algorithms replicate this process by establishing a pool of potential solutions and favoring the reproduction of the most successful ones. Selection is based on higher fitness scores, and as iterations progress, the population refines, producing solutions with increasingly superior fitness scores, approaching an optimal solution, Fig.2 shows a flowchart of the sequential basic steps of a genetic algorithm.

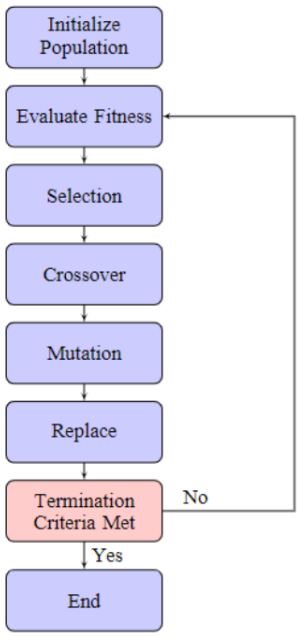

Figure 2: Flowchart of genetic algorithm

GAs are a powerful tool for solving difficult problems. It is often used in practice and has been shown to be effective in a wide variety of problems. However, it is important to note that the GA is not guaranteed to find the optimal solution to a problem. It is also important to choose the parameters of the genetic algorithm carefully, such as the size of the population and the adapted crossover and mutation.

3.2 A REAL CODED GENETIC ALGORITHM

In this article, we employ a real-coded genetic algorithm (RCGA) to address the data completion problem for the Laplace equation. In a RCGA, each chromosome represents a vector of real parameters, where each gene corresponds to a real number, and each allele corresponds to a real value.

We have chosen to use an RCGA because, as a general rule, RCGA demonstrate superior performance compared to binary-coded GAs when it comes to high-precision optimization problems. Additionally,

the encoding of variables as floating-point numbers allows for a seamless implementation of finely-tuned local optimization processes, [29], [30]. In this algorithm, we incorporate a specialized operator that leverages the floating-point representation of the solution space, thereby enhancing the convergence rate of the algorithm

3.3 Genetic operators

3.3.1 Arithmitic crossover

Crossover is a genetic operation that combines two parents to create offspring. Crossover works by swapping parts of the chromosomes between the parents. However, crossover is not performed on every pair of parents. The probability of crossover is controlled by a parameter called the crossover probability p_c.
Typically, parents are denoted as:

$$\begin{aligned} \mathcal{F}^{(1)} &= \left(\mathcal{F}_1^{(1)}, \ldots, \mathcal{F}_n^{(1)}\right) \\ \mathcal{F}^{(2)} &= \left(\mathcal{F}_1^{(2)}, \ldots, \mathcal{F}_n^{(2)}\right) \end{aligned} \tag{7}$$

Similar representation is also used for offspring:

$$\begin{aligned} \mathcal{C}^{(1)} &= \left(\mathcal{C}_1^{(1)}, \ldots., \mathcal{C}_n^{(1)}\right) \\ \mathcal{C}^{(2)} &= \left(\mathcal{C}_1^{(2)}, \ldots, \mathcal{C}_n^{(2)}\right) \end{aligned} \tag{8}$$

The number of parents and offspring may vary depending on the crossover operator considered. In arithmetic crossover, [31], two parents produce two offspring. Thus the parents defined by equation Eq.(7), produce two offspring, which can be determined as follows:

$$\begin{aligned} \mathcal{C}_i^{(1)} &= \alpha_i \mathcal{F}_i^{(1)} + (1-\alpha_i)\, \mathcal{F}_i^{(2)}, \\ \mathcal{C}_i^{(2)} &= \alpha_i \mathcal{F}_i^{(2)} + (1-\alpha_i)\, \mathcal{F}_i^{(1)}, \end{aligned} \tag{9}$$

The α_i are uniform random numbers between 0 and 1.
It should be noted that, we talk about the uniform arithmetical crossover when the α_i are constant. However, in the non-uniform arithmetical crossover, the α_i may vary with the progression of generations.

3.3.2 Power mutation

The mutation operator introduces a potential modification in selected elements of specific chromosomes. The original chromosome, denoted by $\mathcal{C}$, is defined as:

$$\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_i, \ldots, \mathcal{C}_n) \tag{10}$$

where, $\mathcal{C}_i$ refers to the element to be mutated. The chromosome $\mathcal{C}$ after mutation is given by:

$$\mathcal{C}' = (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_i', \ldots, \mathcal{C}_n) \tag{11}$$

In this new configuration, $\mathcal{C}_i'$ represents the mutated value of $\mathcal{C}$.
The Power Mutation (PM), [32], for real-coded genetic algorithm is one of the commonly used mutation operators. This operator relies on a power distribution, defined by the following function:

$$f(x) = \xi x^{\xi-1}, \quad 0 \leq x \leq 1 \tag{12}$$

The corresponding density function is:

$$F(x) = x^{\xi}, \quad 0 \leq x \leq 1 \tag{13}$$

In these equations, ξ signifies the distribution index. This mutation serves to generate a solution $\mathcal{C}'$ in proximity to a parent solution $\mathcal{C}$. It starts by generating a uniform random number t between 0 and 1. Additionally, it creates a random number ζ that adheres to the aforementioned distribution.
To compute the mutated solution, the following formula is applied:

$$\mathcal{C}_i' = \begin{cases} \mathcal{C}_i + \zeta \cdot (u_i - \mathcal{C}_i) & \text{if } t \geq \eta \\ \mathcal{C}_i - \zeta \cdot (\mathcal{C}_i - l_i) & \text{if } t < \eta \end{cases} \tag{14}$$

where, l_i and u_i denote the lower and upper bounds of the decision variable, respectively. η is a uniformly distributed random number ranging from 0 to 1.

The extent of mutation is governed by the mutation index ξ. When ξ is small, one can expect less disruption in the solution. Conversely, with larger ξ values, greater diversity is achieved. The likelihood of generating a mutated solution $\mathcal{C}'$ to the left (or right) of $\mathcal{C}$ is proportionate to the distance of $\mathcal{C}$ from l_i (or u_i), ensuring that the mutated solution always remains feasible.

3.4 Process to estimate the Cauchy data

In this section, we will outline the computational steps involved in using the RCGA to address the considered inverse problem. The problem can be formally stated as follows: Given the Cauchy data f and g on the accessible part of the boundary Γ_c, our objective is to propose a polynomial approximation of the Cauchy data u_D and u_N on Γ_i using the RCGA. The RCGA will systematically explore potential solutions through iterative updates, evaluating each individual's fitness using the Finite Element Method (FEM) to derive corresponding fitness values. Ultimately, the optimal set of parameters will be determined by selecting the individual with the highest fitness value.

Algorithm 1 Algorithm for solving the data completion problem for the Laplace equation via GA

1: **Parameter setting:**
- N: Population size,
- p_c: Probability of Crossover,
- p_m: Probability of Mutation,
- k: Maximum number of Generations,
- l_i and u_i are the lower and the upper bounds, respectively.

2: Initialize population: $\left(\mathcal{U}_{\mathcal{D}}{}^{(0)}\right)_p \sim \mathcal{U}(l_i,\, u_i)$ for $p = 0, \cdots, N$.
3: **for** $it = 1, \cdots, k$ **do**
4: **for** $p = 0, \cdots, N$ **do**
5: Solve the direct problem Eq.(15).

$$P_p : \begin{cases} \Delta u_p = 0 & \text{in} \quad \Omega \\ u_p = \left(\mathcal{U}_{\mathcal{D}}{}^{(it-1)}\right)_p & \text{on} \quad \Gamma_i \\ \partial_n u_p = g & \text{on} \quad \Gamma_c \end{cases} \tag{15}$$

6: Evaluate each individual using Eq.(5).
7: **end for**
8: $U^{it} = [u_0, \cdots, u_N]$
9: Apply selection $u_s = S_e(U^{it})$
10: Apply crossover $u_c = C_r(u_s)$ using Eq.(9).
11: Apply mutation $u_m = M_u(u_c)$ using Eq.(4).
12: Sort and select the best individual $(\mathcal{U}_{\mathcal{D}}{}^{(it)})_p = \text{argmin}\,(\mathcal{J}_{\mathcal{D}}(u_m), \mathcal{J}_{\mathcal{D}}(u_c))$.
13: **end for**

4 Numerical Results and Discussion

In this section, we present the numerical results achieved through the genetic procedure introduced in Section 3, coupled with the Finite Element Method (FEM), for solving the two mixed well-posed problems, Eq.(2) and Eq.(3), corresponding to the Laplace equation. This analysis encompasses two case regular and irregular 2D domains. Furthermore, we assess the stability of the genetic procedure under data perturbation induced by noise.
When dealing with inverse problems in the real world, the boundary data is obtained through experimental measurements, which makes it susceptible to measurement errors. In our testing scenarios, we use the following equation to produce synthetic noisy data:

$$u_{per} = u_{exact} \times (1 + \nu \times \theta) \qquad \text{on} \qquad \Gamma_c \tag{16}$$

where θ is a random number that follows a uniform distribution between -1 and 1, and the level of noise is determined by the parameter ν.

The genetic operators and the parameters used for this genetic algorithm were taken to be as follows:

- Maximum number of Generations $k = 200$,
- Population size $N_p = 70$,
- Rate of crossover $R_c = 0.8$,
- Non-uniform Arithmetic Crossover with $p_c = 0.9$,
- Random selection,
- Power mutation with $p_m = 0.1$.

4.1 Example 1: Irregular domain

In the first case, the numerical tests are made on a unit square domain $\Omega = (0,\, 1)^2$ (Fig.3). The boundary Γ is divided into tow parts:

$$\begin{aligned} \Gamma_i &= \{(0, y) : 0 < y < 1\}, \\ \Gamma_c &= \Gamma \backslash \Gamma_i. \end{aligned} \tag{17}$$

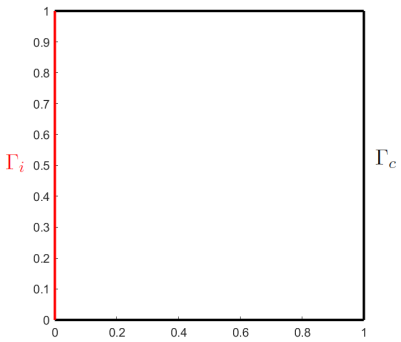

Figure 3: Unit square.

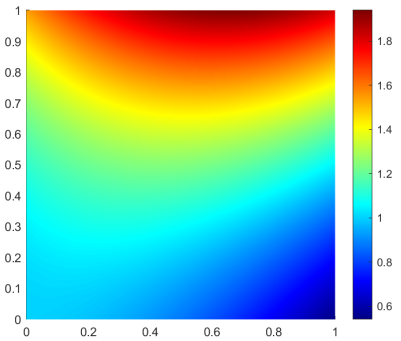

Figure 4: The analytical solution in 2D.

The over-specified Cauchy data in Γ_c are extracted from the analytic solution, given by:

$$u_{ex}(x, y) = \cos(x)\cosh(y) + \sin(x)\sinh(y) \tag{18}$$

Fig.4 presents the analytical solution in the whole domain.

4.1.1 Choice of regulation parameter

Fig.5 and Fig.6 show the objective functions $\mathcal{J}_D$ and $\mathcal{J}_N$ with respect to various values of the regularization parameters α_1 and α_2, respectively. These results are obtained for the data completion problem using exact boundary data extracted from Eq. (18). These figures reveal that, considering $\mathcal{J}_D$ and $\mathcal{J}_N$, the optimal values for the regularization parameters α_1 and α_2 are 1×10^{-5} and 1×10^{-4}, respectively, for u_D and u_N on Γ_i.

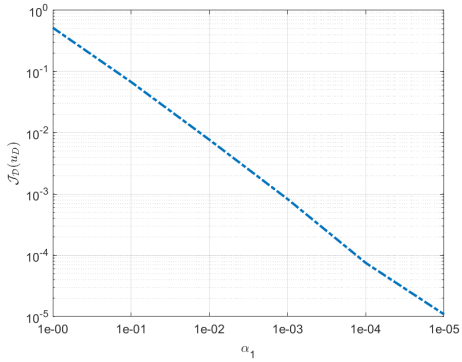

Figure 5: The cost function $\mathcal{J}_\mathcal{D}$ for various values of α_1.

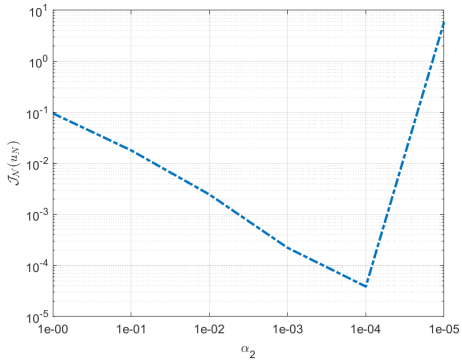

Figure 6: The Cost function $\mathcal{J}_\mathcal{N}$ for various values of α_2.

Additionally, Fig.7 and Fig.8 show that the objective functions $\mathcal{J}_\mathcal{D}$ and $\mathcal{J}_\mathcal{N}$ significantly decrease during the initial iterations. This decrease indicates that the algorithm is successfully approaching the minimum of the objective function. As the iterations progress, the convergence rate of the objective function gradually slows down, but it eventually reaches a low value by iteration $k = 200$. These observations suggest that the obtained solution is highly accurate and provides an excellent fit to the available data.

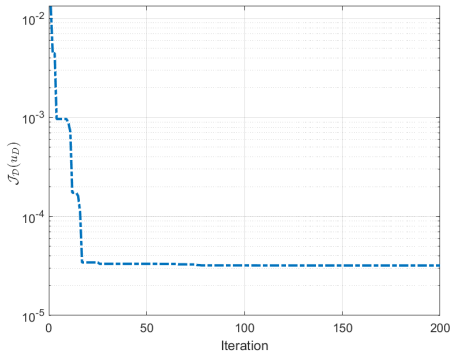

Figure 7: The convergence of the cost function $\mathcal{J}_D$ during the genetic process.

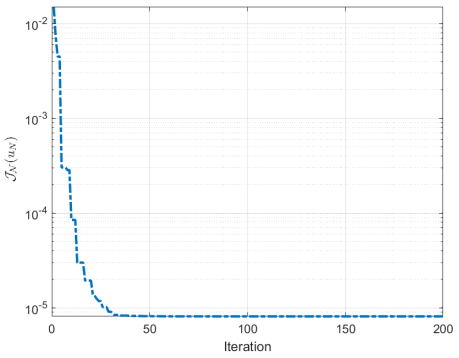

Figure 8: The convergence of the cost function $\mathcal{J}_N$ during the genetic process.

Fig.9 and Fig.10 show how the numerical solution gets closer and closer to the analytical solution as the number of iterations increases. Initially, the numerical solution is significantly different from the analytical solution, but the difference quickly decreases with each iteration. This shows that the iterative method is effective at solving the problem under study.

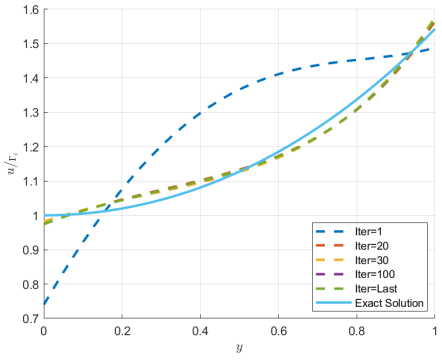

Figure 9: The numerical solution u_D on Γ_i for a specific iterations.

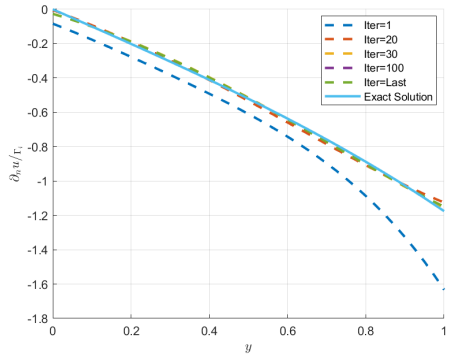

Figure 10: The numerical solution u_N on Γ_i for a specific iterations.

4.1.2 Stability of the Proposed Method

Fig.11 and Fig.12 compare the numerical solution to the analytical solution for different levels of noise in the measurement data $\nu = 1\%, 2\%, 5\%, 7\%$. As the noise level increases, the numerical solution deviates slightly from the exact solution. However, the difference between the numerical and exact solutions remains small, even when the noise level reaches a high value of 7%. This suggests that the numerical solutions obtained for problem Eq.(18) in this study are stable with respect to the amount of noise ν added to the input data.

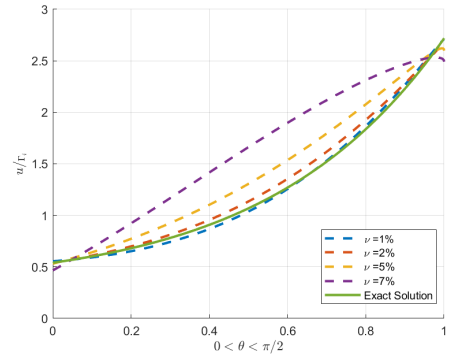

Figure 11: The numerical solution u_D for various levels of noise.

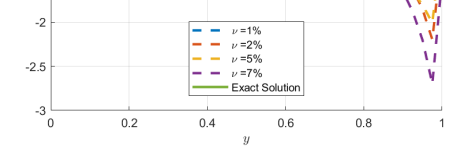

Figure 12: The numerical solution u_N for various levels of noise.

Fig.13 and Fig.14 show the cost function for different noise levels, $\nu = 1\%, 2\%, 5\%$, and 7%. The figures indicate that as the noise level increases, the cost function also increases, which means that the fit to the data becomes less precise. However, even with higher noise levels, the cost function remains relatively low, suggesting that the numerical solution still provides a satisfactory fit to the data.

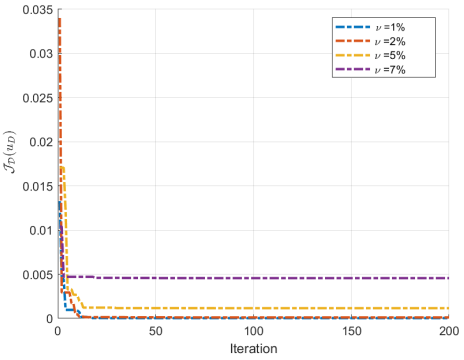

Figure 13: $\mathcal{J}_{\mathcal{D}}$ for various levels of noise.

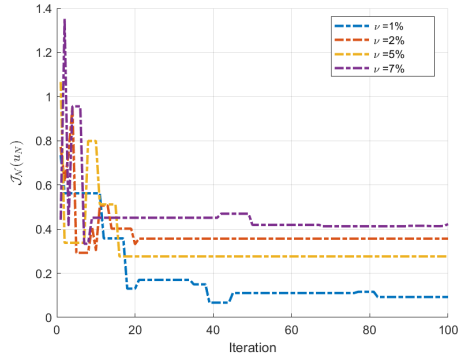

Figure 14: $\mathcal{J}_{\mathcal{N}}$ for various levels of noise.

4.2 Example 2: regular domain

In the second case, the numerical tests are performed on an unit disc Fig.15.
The boundary of this domain is divided into two parts:

$$\Gamma_i = \left\{(x,y) : x^2+y^2 = 1, \quad y > 0 \quad , \quad x > 0\right\},$$
$$\Gamma_c = \Gamma \backslash \Gamma_i. \tag{19}$$

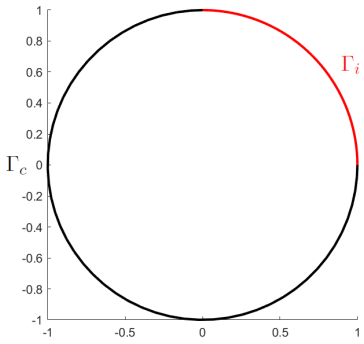

Figure 15: Unit disc.

The over-specified Cauchy data in Γ_c are extracted from the analytic solution:

$$u_{ex}(x,y) = \cos(x)\exp(y) \tag{20}$$

Fig.16 shows the analytical solution in whole domain.

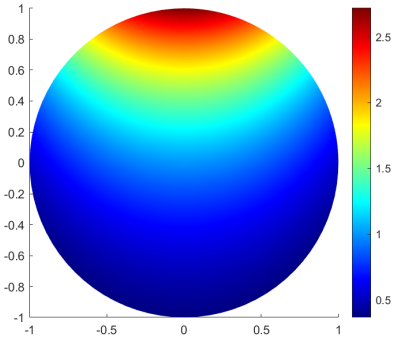

Figure 16: The analytical solution in 2D.

4.2.1 Choice of regulation parameter

Fig.17 and Fig.18 show the objective functions $\mathcal{J}_D$ and $\mathcal{J}_N$ with respect to various values of the regularization parameters α_1 and α_2, respectively, for the data completion problem using exact boundary data extracted from Eq. (20). The optimal value for the regularization parameters α_1 and α_2 is 1×10^{-5} for u_D and u_N on Γ_i, considering both $\mathcal{J}_D$ and $\mathcal{J}_N$.

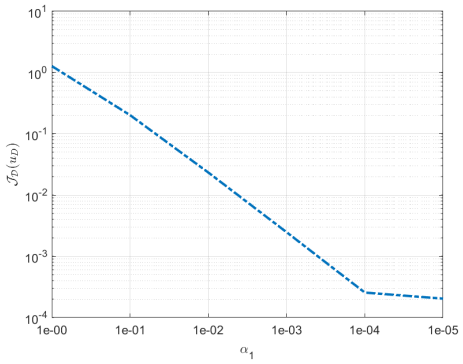

Figure 17: Objective function $\mathcal{J}_{\mathcal{D}}$ for various values of α_1.

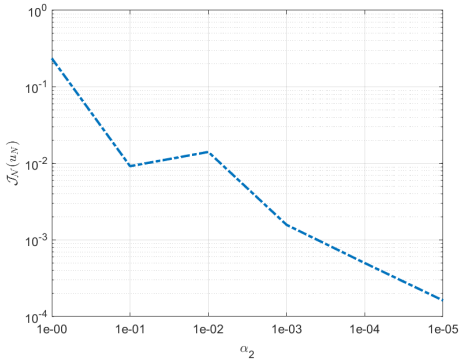

Figure 18: Objective function $\mathcal{J}_{\mathcal{N}}$ for various values of α_2.

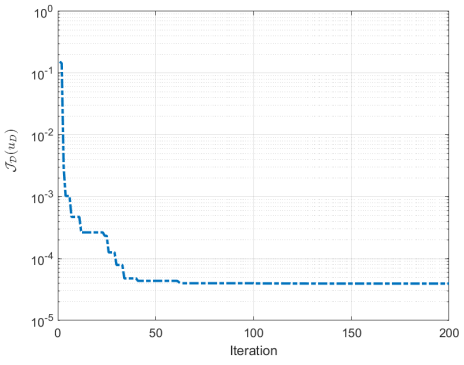

Figure 19: The convergence of the cost function $\mathcal{J}_D$ during the genetic process.

Fig.19 and Fig.20 show that the objective functions $\mathcal{J}_{\mathcal{D}}$ and $\mathcal{J}_{\mathcal{N}}$ decrease significantly in the early iterations. This indicates that the algorithm is successfully converging to the minimum of the objective function. As the iterations progress, the convergence rate slows down, but the objective function eventually

reaches a low value by iteration $k = 200$. These observations suggest that the obtained solution is highly accurate and provides a good fit to the available data.

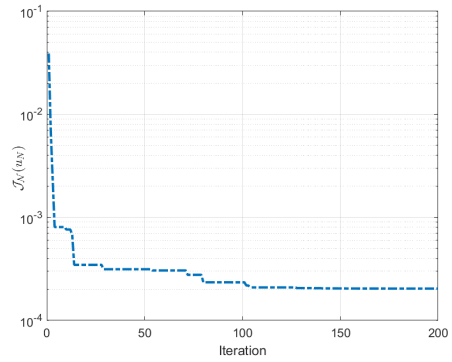

Figure 20: The convergence of the cost function $\mathcal{J}_N$ during the genetic process.

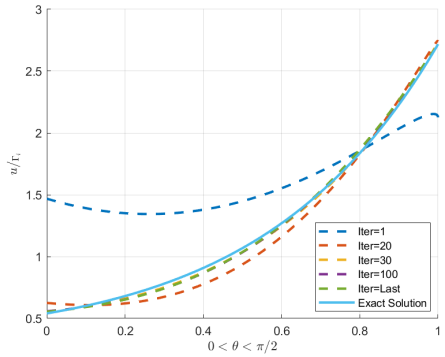

Figure 21: The numerical solution u_D on Γ_i for a specific iterations.

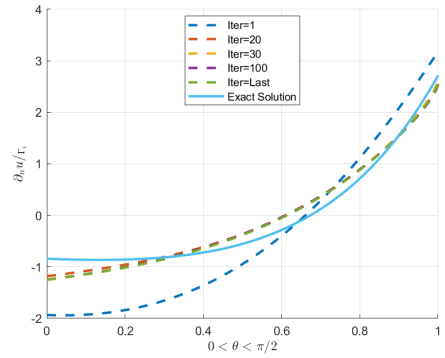

Figure 22: The numerical solution u_N on Γ_i for a specific iterations.

Fig.21 and Fig.22 illustrate the progressive convergence of the numerical solution to the analytical solution as the number of iterations increases. Initially, the numerical solution exhibits significant deviation from the analytical solution, but this discrepancy diminishes rapidly with each iteration. This demonstrates the efficacy of the iterative method in effectively resolving the problem under consideration.

4.2.2 Stability of the Proposed Method

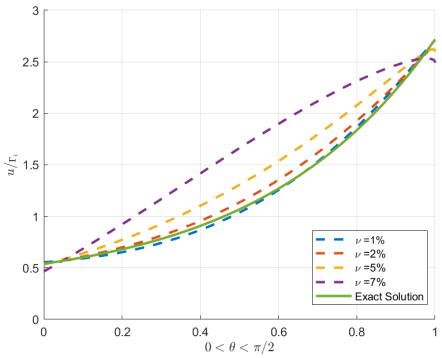

Figure 23: The numerical solution u_D for various levels of noise.

Fig.23 and Fig.24 compare the numerical solution to the analytical solution for different noise levels in the measurement data $\nu = 1\%, 2\%, 5\%, 7\%$. The numerical solution deviates slightly from the analytical solution as the noise level increases. However, the difference between the numerical and analytical solutions remains small, even when the noise level reaches a high value of 7%. This suggests that the numerical solutions obtained for problem Eq.(20) in this study are stable with respect to the amount of noise ν added to the input data.

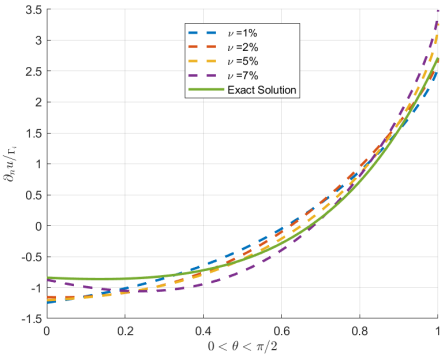

Figure 24: The numerical solution u_N for various levels of noise.

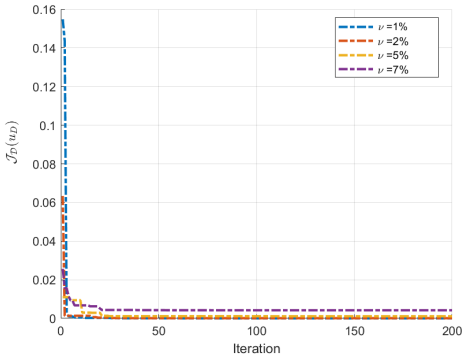

Figure 25: $\mathcal{J}_{\mathcal{D}}$ for various levels of noise.

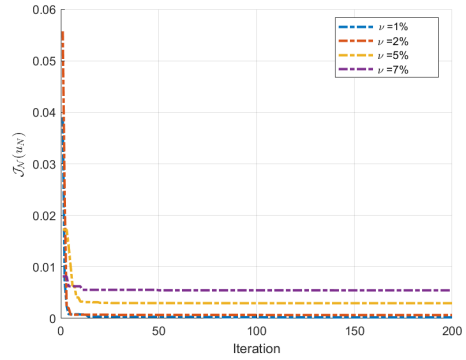

Figure 26: $\mathcal{J}_{\mathcal{N}}$ for various levels of noise.

Fig.25 and Fig.26 show the cost function for different noise levels, $\nu = 1\%, 2\%, 5\%$, and 7%. The figures indicate that the cost function increases as the noise level increases. This means that the fit to the data becomes less precise as the noise level increases. However, even with higher noise levels, the cost function remains relatively low, suggesting that the numerical solution still provides a satisfactory fit to the data.

5 Conclusion

In this paper, we address the challenging ill-posed inverse problem associated with the Cauchy problem for the Laplace equation. We propose a novel approach to solve this problem by converting it into an optimization problem. We use the RCGA with Tikhonov regularization to solve the optimization problem. The effectiveness of our approach is evaluated through numerical experiments conducted on both regular and irregular domains. The results demonstrate the efficacy of RCGA, enhanced with adapted genetic operators, in successfully solving the Cauchy problem associated with the Laplace equation.

Acknowledgment:
The authors would like to thank the editors and the anonymous reviewers for their comments and suggestions.

References:

[1] Martinez-Luaces, V., Fernández-Plaza, J. & Rico, L., About the Notion of Inverse Problem in STEM Education, *Active Learning-Research and Practice for STEAM and Social Sciences Education*, IntechOpen, 2022.

[2] Gulrajani, R., The forward and inverse problems of electrocardiography, *IEEE Engineering in Medicine and Biology Magazine*, Vol.17, No.5, 1998, pp. 84-101.

[3] El Badia, A. & Farah, M., Identification of dipole sources in an elliptic equation from boundary measurements: application to the inverse EEG problem, *Journal of Inverse & Ill-Posed Problems*, Vol.14, No.4, 2006, pp. 331-353.

[4] Friedman, A. & Vogelius, M., Determining cracks by boundary measurements, *Indiana University Mathematics Journal*, Vol.38, No.2, 1989, pp.497-525.

[5] Yang, F., Yan, L. & Wei, T., Reconstruction of the corrosion boundary for the Laplace equation by using a boundary collocation method, *Mathematics and Computers in Simulation*, Vol.79, No.7, 2009, pp. 2148-2156.

[6] Hadamard, J., *Lectures on Cauchy's problem in linear partial differential equations*, Yale university press, 1923.

[7] Beck, J. & Murio, D., Combined function specification-regularization procedure for solution of inverse heat conduction problem, *AIAA Journal*, Vol.24, No.1, 1986, pp. 180-185.

[8] Avdonin, S., Kozlov, V., Maxwell, D. & Truffer, M., Iterative methods for solving a nonlinear boundary inverse problem in glaciology, *Journal of Inverse and Ill-posed Problems*, Vol.17, No.4, 2009, pp. 239-258.

[9] Tajani, C. & Abouchabaka, J., On the data completion problem for Laplace's equation, *Annals of the University of Craiova-Mathematics and Computer Science Series*, Vol.45, No.1, 2018, pp. 11-36.

[10] A. N. Tikhonov, V. J. Arsenin, V., *Solutions of ill-posed problems*, Vh Winston, 1977.

[11] Karaboga, D., Gorkemli, B., Ozturk, C. & Karaboga, N., A comprehensive survey: artificial bee colony (ABC) algorithm and applications, *Artificial Intelligence Review*, Vol.42, 2014, pp. 21-57.

[12] Kennedy, J. & Eberhart, R., Particle swarm optimization, *Proceedings Of ICNN'95-international Conference On Neural Networks*, Vol.4, 1995, pp. 1942-1948.

[13] Socha, K. & Dorigo, M., Ant colony optimization for continuous domains, *European Journal of Operational Research*, Vol.185, No.3, 2008, pp. 1155-1173.

[14] Yang, X. & Hossein Gandomi, A., Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, Vol.29, No.5, 2012, pp. 464-483.

[15] Tominaga D., Koga N. & Okamoto M., Efficient numerical optimization algorithm based on genetic algorithm for inverse problem, Pro-ceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, 2000, pp. 251-258.

[16] Mahinthakumar, G. & Sayeed, M., Hybrid genetic algorithm—local search methods for solving groundwater source identification inverse problems, *Journal of Water Resources Planning and Management*. Vol.131, No.1, 2005, pp. 45-57.

[17] Raudenskỳ, M., Woodbury, K., Kral, J. & Brezina, T., Genetic algorithm in solution of inverse heat conduction problems, *Numerical Heat Transfer, Part B Fundamentals*, Vol.28, No.3, 1995, pp. 293-306.

[18] Mera, N., Elliott, L. & Ingham, D., A multi-population genetic algorithm approach for solving ill-posed problems, *Computational Mechanics*, Vol.33, No.4, 2004, pp. 254-262.

[19] Mera, N., Elliott, L. & Ingham, D. (2004). Numerical solution of a boundary detection problem using genetic algorithms, *Engineering Analysis with Boundary Elements*, Vol.28, No.4, 2004, pp. 405-411.

[20] Jouilik, B., Daoudi, J., Tajani, C. & Abouchabaka, J., Optimization approach based on genetic algorithm for a robin coefficient problem, *Palestine Journal of Mathematics*, Vol.11, No.3, 2022, pp. 708-718.

[21] Wang, X., He, Q., Chen, D. & Yeung, D., A genetic algorithm for solving the inverse problem of support vector machines, *Neurocomputing*, Vol.68, 2005, pp. 225-238.

[22] Jiang, M., Xia, L. & Shou, G., The use of genetic algorithms for solving the inverse problem of electrocardiography, *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 3907-3910.

[23] Mera, N., Elliott, L. & Ingham, D., Comparison of a genetic algorithm and a gradient based optimisation technique for the detection of subsurface inclusions, *Acta Universitatis Apulensis. Mathematics-Informatics*, Vol.4, 2002, pp. 123-134.

[24] Suram, S., Bryden, K. & Ashlock, D., An evolutionary algorithm to estimate unknown heat flux in a one-dimensional inverse heat conduction problem, *Proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice*, Vol.11, 2005, pp. 15.

[25] Yamamoto, K. & Inoue, O., Applications of genetic algorithm to aerodynamic shape optimization, *12th Computational Fluid Dynamics Conference*, 1995, pp. 1650.

[26] Jouilik, B., Daoudi, J., Tajani, C. & Abouchabaka, J., Numerical optimization algorithm based on genetic algorithm for a data completion problem, *TWMS Journal of Applied and Engineering Mathematics*, Vol.13, No.1, 2023, pp. 86-97.

[27] Hussain, K., Mohd Salleh, M., Cheng, S. & Shi, Y., Metaheuristic research: a comprehensive survey, *Artificial Intelligence Review*, Vol.52, 2019, pp. 2191-2233.

[28] Holland, J., Genetic algorithms, Scientific American (June 1992).[12] Intergovernmental Panel on Climate Change, Climate Change: the IPCC Assessment, eds. JT Houghton, GJ Jenkins and JJ Ephraums, Cambridge University Press, 1990.

[29] Wright, A., Genetic algorithms for real parameter optimization, *Foundations of Genetic Algorithms*, Vol.1, 1991, pp. 205-218.

[30] Herrea, F., Tackling real-coded genetic algorithms: operators and tools for behavioral analysis, *Artificial Intelligence Review*, Vol.12, No.4, 1998, pp. 265-319.

[31] Kora, P. & Yadlapalli, P., Crossover operators in genetic algorithms: A review, *International Journal of Computer Applications*, Vol.162, No.10, 2017, pp. 34-36.

[32] Deep, K. & Thakur, M., A new crossover operator for real coded genetic algorithms, *Applied Mathematics and Computation*, Vol.188, No.1, 2007, pp. 895-911.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.