S©MR

‘€´€‘€. . . ´<. . . ’´´>. . .
’. . . ’€—. . . ‘€. . . €‡,. . . “’€
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Vol. 20, No. 2, pp. 144–145 (2024) “„ ???.??
https://doi.org/10.33048/semi.2023.20.??? MSC 74G75, 74G15, 80M20, 80A20

THE CAUCHY PROBLEM FOR THE 3D POISSON EQUATION: HORIZONTALLY DIAGONALIZE AND FIT METHOD VS. FINITE DIFFERENCE METHOD WITH SEVERAL ITERATIVE TECHNIQUES

S. CHANDRAGIRI, **M.A. SHISHLENIN**, AND **S.I. KABANIKHIN**

Communicated by S.I. Kabanikhin

Abstract: The Finite difference method (FDM) is proposed to solve the ill-posed Cauchy problem for the three dimensional Poisson equation with data given on the part of the boundary (a continuation problem). The idea of Finite difference method is to discretize the PDE by replacing the partial derivatives with their approximations, that is finite differences. This approximation will be used to form a system of linear equations that gives solutions at the internal points of the domain and we consider a specific implementation of the Jacobi, Gauss-Seidel and SOR(α_{opt}) iterative methods used to solve the linear system; we then consider the convergence behavior of the iterative methods as the size of the grid increases. Through MATLAB, our experiments show that FDM is applicable to large scale problems and for $n \geq 2500$ is significantly more effective than horizontally diagonalize and fit method.

Chandragiri, S., Shishlenin, M.A., Kabanikhin, S.I., The Cauchy problem for the 3D Poisson equation: Horizontally diagonalize and fit method vs. finite difference method with several iterative techniques.

The work is supported by the Mathematical Centre in Akademgorodok under the agreement No. 075-15-2022-281 with the Ministry of Science and Higher Education of the Russian Federation.

Received January, 1, 2024, Published December, 31, 2024.

Keywords: continuation problem, inverse and ill-posed problem, partial differential equations, numerical analysis, regularization.

1 Introduction

The Cauchy problem for Poisson equation has been investigated started from the sixties by Douglas (1960, [1]), Cannon (1964, [2]), Cannon and Miller (1965, [3]).

An algorithm for solving the Cauchy problem for the Laplace equation was independently and simultaneously developed, conditional stability of the Cauchy problem solution for the Laplace equation was studied by Lavrentiev (1955, [4]), and by Lavrentiev et al. (1986, [5]).

A little earlier, for the Laplace equation, the well-posedness of the Cauchy problem in the class of bounded functions with the corresponding estimates was proved Mergelyan (1956, [6]) using approximation theory.

Lavrentiev (1956, [7]) proposed several methods for solving the problem in the class of bounded functions based on the Carleman formula and, for the spatial problem, derived estimates that characterize their stability in the class of bounded solutions. Soon after Lavrentiev (1957, [8]) appeared, where well-posedness was proved in other ways.

Conditional stability of the problem was investigated by Payne, Payne et al. (1960, 1967, 1970, 1975, [9]-[12]), Tautenhahn (1990, [13]), Reinhardt et al. (1999, [14]).

The uniqueness in Cauchy problems for elliptic systems was considered by Douglis (1960, [15]), Schaefer (1965, [16]). Unique continuation for elliptic equations was considered by Protter (1960, [17]) and by Watanabe et al. (1977, [18]). The solution uniqueness of the Cauchy problem and the unique continuation theorem for elliptic equations was established by Kumano-go (1962, [19]).

Stability of finite-difference schemes for an ill-posed Cauchy problem with constant coefficients was first investigated by Chudov (1962, [20]). Error estimates and convergence results for the solution of discretized continuation problems were obtained by Falk et al. (1986, [21]), Falk (1990, [22]), Han et al. (1997, [23]).

A special iterative procedure for solving the Cauchy problem for elliptic equations was proposed by Kozlov, Mazya, and Fomin (1991, [24]). The proposed iterative procedure consists of successive solution steps of correct mixed boundary value problems for the original equation. One of the advantages of this method is that the original equation is preserved. The convergence and regularizing properties of the method are guaranteed by selecting appropriate boundary conditions.

Hào et al. (1992, [25]) developed a regularization of the Cauchy problems for the Laplace equation based on the theory of pseudo differential operators with real analytic symbols and the approximation theory.

Hào et al. (2000, [26]) proposed another variation approach for solving the Cauchy problem for the Laplace equation. Given boundary conditions on one part of the boundary were considered as a control in the formulation of a direct problem for determining Cauchy data on the remaining boundary part.

Modified method of fundamental solutions for the Cauchy problem connected with the Laplace equation was studied by Sun (2014, [27]). The method keeps a very basic natural property, i.e. the invariance under trivial coordinate changes in the problem description and it combines Newton's method and classic Tikhonov regularization to solve an inverse problem. Even for high noise levels some examples are analysed using the modified MFS with convergence, accuracy, and stability with respect to increasing the number of source points, and decreasing the amount of noise added into the input data.

A 2D steady state heat conduction problem was considered by Berntsson (2001, [28]). There, the Laplace equation was posed in a domain with a hole. Temperature and heat-flux data were specified on the outer boundary, and the temperature on the inner boundary was unknown.

Marin et al. (2002, [29]) analysed an iterative algorithm based on the conjugate gradient method combined with the boundary element method to obtain approximate solutions to the Cauchy problem in linear elasticity.

Jourhmane et al. (2002, [30]) applied accelerated the process proposed by Kozlov, Mazya and Fomin [24] for solving the Cauchy problem for the Poisson equation.

Alessandrini et al. (2009, [31]) studied the stability for the Cauchy problem for elliptic equations. An ill-posed Cauchy problem for elliptic equations was discussed, which is pervasive in inverse boundary value problems modeled by elliptic equations and provided essentially optimal stability results, in wide generality and under substantially minimal assumptions, and showed that all such stability results can be derived by the use of a single building brick, the three-spheres inequality.

The conjugate gradient method was applied to the Cauchy problem for general elliptic linear partial differential equations of the second order, in which Dirichlet data were available on a larger part of the boundary of the bounded domain than the boundary part on which the Neumann data by Hào et al. (2010, [32]).

The Cauchy problem was rewritten as an operator equation on the boundary using the Dirichlet-to-Neumann map by Helsing and Johansson (2011, [33]).

Note that Maxwell (2014, [34]) showed that the alternating method of Kozlov, Mazya, and Fomin [24] can be recast as a form of Landweber iteration.

A finite element method for the solution of the Cauchy problem for Poisson equation was introduced Hu et al. (2017, [35]) and based on least squares method with discontinuous Galerkin finite element method. Error estimates of the finite element solution were derived.

Hào et al. (2018, [36]) introduced the concept of very weak solution to a Cauchy problem for elliptic equations. The Cauchy problem was regularized by a well-posed non-local boundary value problem whose solution was also understood in a very weak sense. A stable finite difference scheme was suggested for solving the non-local boundary value problem and then applied to stabilizing the Cauchy problem.

The method of spectral expansion in eigen functions of the Cauchy problem for equations with deviating argument established a criterion of the strong solvability of the considered elliptic Cauchy problem.

Variational methods coupled with Tikhonov's regularization for solving the Cauchy problem for Poisson equation were suggested and studied by Thanh and Steinbach (2020, [37]). The penalty term was evaluated by some appropriate boundary integral operators. The optimality condition in the form of boundary integral equations was derived and then discretized by the Galerkin boundary element method.

A weak Galerkin least squares finite element method for the Cauchy problem was introduced by Wang and others (2022, [38]). The optimal order of convergence was proved in an energy norm.

We also note the following methods that were used for numerical solutions of the Cauchy problem for the elliptic equations: iterative (Kabanikhin et al. 1995, 2013, 2014 ([39], [40], [41]); Hào et al. 2000, [26]; quasi-reversibility method (Klibanov et al. 1991, [42]); Bourgeois et al. 2005, 2006, 2010, ([43]-[46]), regularizing methods: Backus-Gilbert method (Cheng et al. 2001, [47]; Hon et al. 2001, [48]); Mann's iterative regularization method (Engl et al. 2001, [49]); fourth-order derivative regularization method (Qian et al. 2006, [50]); Fourier method (Fu et al. 2008, [51]; Yang 2011, [52]); wavelet method (Qiu et al. 2008, [53]); level-set method (Leitão et al. 2007, [54]); method of expanding compacts (Titarenko et al. 2002, [55]), method of fundamental solutions (Marin 2005, [56]), numerical procedure based on the rational Krylov method (Eldén et al. 2009, [57]).

Cannon and DuChateau studied approximating the solution to the Cauchy problem for Laplace's equation (1977, [58]). Cauchy problem for Laplace's equation solved by Yarmukhamedov (1977, [59]) and the Cauchy problem for the Laplace equation and application to image inpainting were shown by Belaid et al. (2011, [60]).

The Cauchy problem for the Laplace equation in an infinite strip was proved by Ivanov (1965, [61]). An efficient direct method for numerically solving the Cauchy problem for Laplace's equation was investigated by Sorokin (2019, [62]).

Liu (2011, [63]) investigated an analytical method for the inverse Cauchy problem of Laplace equation in a rectangular plate.

Mazya and Khavin obtained the solutions of the Cauchy problem for Laplace's equation (1974, [64]) and Romanovich et al. (1987, [65]) studied the solvability of the Cauchy problem for the Laplace operator.

Belgacem et al. (2005, [66]) studied a variational steklov-poincaré theory on Cauchy's problem, and why is the Cauchy problem severly ill-posed was investigated by Belgacem (2007, [67]).

Calderón showed uniqueness in the Cauchy problem for partial differential equations (1958, [68]) and Cimetiere et al. (2001, [69]) obtained the solution of the Cauchy problem using iterated tikhonov regularization.

The Finite element method was studied by Han (1982, [70]) and proved in a family of improperly posed problems.

John wrote a note on "improper" problems in partial differential equations (1955, [71]) and also he showed Continuous dependence on data for solutions of partial differential equations with a prescribed boundaries (1960, [72]).

Klibanov proposed the regularization of ill-posed Cauchy problems by Carleman estimates (2015, [73]). Shaydurov et al. (2023, [74]) proposed two algorithms for refinement of approximate solutions in the regularization method.

Poisson's equation is a steady state, time-independent variation of Laplace equation (2004, [75]) that models energy distribution in equilibrium systems. Solutions of this equation model heat distribution through a region, concentration of particles after diffusion, electrostatic potential, and Newtonian gravity potential (2002, [76]). Although actual solutions to Poisson's equation are known, these solutions are complicated and difficult to calculate. Numerical methods become necessary to efficiently model solutions of this partial differential equation.

In this article, a new approach is proposed to solve the ill-posed Cauchy problem for the Laplace and Poisson equations in 3D with the data given on a part of the boundary. Within this approach, we will approximate Poisson's equation with a finite difference method with Dirichlet and Neumann boundary conditions in a cube. This approximation will be used to form a system of linear equations $Aq = f$ that gives solutions at internal points of our domain. First, we will identify patterns in this system for a small number of subintervals. With these patterns, we will define algorithms to construct the matrices and vectors of this equation with a general number of subintervals along the x, y, and z axes.

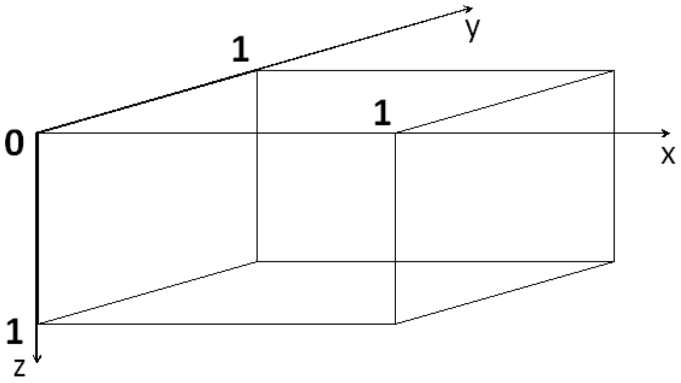

FIG. 1. Domain Ω. Variable z means the depth, variables x and y are horizontal ones

2 Statement of the problem

2.1. Cauchy problem (continuation problem). Let us consider the ill-posed [77] continuation problem in which the unknown function $u(x,y,z)$ satisfies the following boundary-value problem (BVP):

$$\Delta u = p(x,y,z), (x,y,z) \in \Omega = (0,1)^3 \tag{1}$$

$$u(0,y,z) = u(1,y,z) = 0, \tag{2}$$

$$u(x,0,z) = u(x,1,z) = 0, \tag{3}$$

$$u(x,y,0) = f(x,y), \frac{\partial u}{\partial z}(x,y,0) = g(x,y). \tag{4}$$

Here $p(x,y,z), f(x,y)$, and $g(x,y)$, are given functions and

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

2.2. Inverse problem. Let us consider the following direct (well-posed) problem (DP)

$$\Delta u = p(x,y,z), (x,y,z) \in \Omega, \tag{5}$$

$$u(0,y,z) = u(1,y,z) = 0, \tag{6}$$

$$u(x,0,z) = u(x,1,z) = 0, \tag{7}$$

$$\frac{\partial u}{\partial z}(x,y,0) = g(x,y), \tag{8}$$

$$u(x,y,1) = q(x,y). \tag{9}$$

The direct problem (5)-(9) is to find u for given p, g and q. We suppose that function $q(x,y)$ is unknown. But instead of $q(x,y)$ we have the following additional information concerning the solution $u(x,y,z)$ of DP:

$$u(x,y,0) = f(x,y). \tag{10}$$

The inverse problem (IP) consists in finding the function $q(x,y)$ from (5)-(10). The inverse problem (5)-(10) is equivalent to the continuation problem

(1)-(4) in the following sense. If we solve the continuation problem then we can find $q(x, y) = u(x, y, 1)$, i.e., the solution of inverse problem q. Vice versa if we solve inverse problem and find the solution of inverse problem, we can set $u(x, y, 1) = q(x, y)$ and solve the direct problem (5)-(9) and find $u(x, y, 1) \in \Omega$ - the solution of the continuation problem.

2.3. Operator form of the Inverse problem. The inverse problem (5)-(10) can be formulated in operator form:

$$Aq = f \tag{11}$$

Here $A : q(x, y) \to u(x, y, 0)$ and $u(x, y, z)$ is the solution of the direct problem (5)-(9).

Note that [78] $A : L_2((0, 1) \times (0, 1)) \to L_2((0, 1) \times (0, 1))$.

2.4. Limitations of Direct Methods. The number of operations needed to solve the equation sets increases rapidly as the size of the coefficient matrix increases. Iterative methods are usually the preferred choice for large systems where direct methods would be too computationally expensive. When the system becomes increasingly large and complex, it is no longer feasible to solve the systems with direct methods due to limitations of computer memory or solve the system in a reasonable amount of time and the number of arithmetic operations that must be carried out.

2.5. Iterative Methods. Iterative methods can be used to solve the system of equations that arise from finite difference approximation of partial differential equations, which have large and sparse coefficient matrices. Another reason for using iterative methods is that they are far easier to implement on parallel computers. This is becoming increasingly important as inexpensive powerful parallel computers become broadly available. In iterative methods, the process starts with an initial approximation to the unknown vector q of $Aq = f$ and then the successive approximations will be improved by an iterative process

$$q^{(n+1)} = Rq^{(n)} + C, \ \ n = 0, 1, 2 \dots \tag{12}$$

where $q^{(n)}$ and $q^{(n+1)}$ are the n^{th} and $(n+1)^{\text{th}}$ approximations for the solution of the linear set of equations. Given $q^{(0)}$, the classical methods generate a sequence $q^{(n)}$ converging to the solution $A^{-1}f$, where $q^{(n+1)}$ is computed from $q^{(n)}$ by iterating (12). The iterative method strategy generates a sequence of approximate solution vectors $q^{(0)}, q^{(1)}, q^{(2)}, ., q^{(n)}$ for the system $Aq = f$.

R: is called the non-singular iteration matrix depending on A

C: is called the constant column vector.

The process can be stopped when

$$\|q^{(n+1)} - q^{(n)}\|_\infty < \varepsilon \tag{13}$$

in the limiting case, when $n \to \infty, q^{(n)}$ converges to the exact solution $q = A^{-1}f$. From (13) we find that the exact solution, q, is a stationary point, i.e., if $q^{(n)}$ is equal to the exact solution of the equation set then $q^{(n+1)}$ will be equal to the exact solution as well.

2.6. Classical Iterative Methods. The classic iterative methods are built on the principle that the matrix A can be written as a sum of other matrices. There are several ways to divide the matrix; two of them are the origin of the Jacobi and the Gauss-Seidel method. The successive over-relaxation method is an improved version of the Gauss-Seidel method. The classic iterative methods do in general have quite a low convergence rate.

The splitting of the matrix A is divided into two matrices M and K such that $M + K = A$. M is a diagonal matrix with the same entries as A has on the main diagonal, and K has zeros on the diagonal and the off-diagonal entries are equal to the rest of the entries in A. Applying this to the set of linear equations we find that

$$Aq = f.$$

$$(M + K)q = f,$$

where M is non-singular and solve for q, we obtain

$$q = Rq + C, \tag{14}$$

where $R = -M^{-1}K$ and $C = M^{-1}f$. Here, R is called iteration matrix.

Definition 1. [79][80] *The matrix $A = (a_{ij})$ is an M-matrix if $a_{ij} \leq 0$ for all $i \neq j$, A is non singular and $A^{-1} \geq 0$. If A is irreducible and $|a_{ii}| \geq \sum |a_{ij}|$ with strict inequality for at least on i, then A is an M-matrix.*

We write (14) in the component form, which gives the following expression

$$q_i = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j\neq i}}^{k} a_{ij}q_j - f_i \right), \ i = 1, 2, \ldots, k. \tag{15}$$

Lemma 1. *Let $\|\cdot\|$ be any operator norm $\left(\|R\| \underset{q\neq 0}{max} \frac{\|Rq\|}{\|q\|} \right)$. Then, $\|R\| < 1$ iff $q^{(n+1)} = Rq^{(n)} + C$ converges for any $q^{(0)}$.*

Proof *Using (14), we have*

$$\|q^{(n+1)} - q\| = \|Rq^{(n)} + C - Rq - C\|$$

$$\|Rq^{(n)} - Rq\| \leq \|R\|\|q^{(n)} - q\| \leq \|R\|^{(n+1)}\|q^{(0)} - q\|.$$

If $\|R\| < 1, \|R\|^{(n+1)} \to 0$, as $n \to \infty$. Thus, $\|q^{(n+1)} - q\| \to 0$, as $n \to \infty$. For the converse, if $q^{(n+1)} \to q$, then $\|R\| < 1$. The spectral radius of a matrix A, denoted by

$$\rho(A) = max\{|\lambda|\},$$

where the maximum is taken overall eigenvalues λ of A.

Lemma 2. [81] *For all operator norms $\|\cdot\|$, $\rho(R) \leq \|R\|$.*

Corollary 1. *The iteration $q^{(n+1)} = Rq^{(n)} + C$ converges to the solution $Aq = f$ for all initial $q^{(0)}$ and for all f iff $\rho(R) < 1$.*

Proof *This corollary follows from Lemmas (1) and (2).*

The rate of convergence of the iterative scheme $q^{(n+1)} = Rq^{(n)} + C$ is a measure of the number of iterations needed to converge to some given tolerance. It defined as $r(R) = -log\rho(R)$. This tells us, the smaller $\rho(R)$, the higher is the rate of convergence. Thus, the method is said to be efficient if we choose a splitting $A = M + K$ so that

1. $R = -M^{-1}K$ and $C = M^{-1}f$ are easy to evaluate,
2. $\rho(R)$ is small.

The splitting for the methods discussed in this section all share the following notation

$$A = M + K = D + L + U, \tag{16}$$

where D is the diagonal of A, L is the strictly lower triangular part of A, U is the strictly upper triangular part of A.

2.6.1. Jacobi Iterative Method. In the Jacobi method all the entries in the approximation will be updated based on the values in the previous approximation. The splitting of the coefficient matrix A for Jacobi's method is $M = D, K = L + U$ and its iteration gives

$$q^{(n+1)} = R_J q^{(n)} + C_J, \tag{17}$$

where $R_J = -M^{-1}K = -D^{-1}(L + U),\ C_J = M^{-1}f = D^{-1}f$.

The component form of equation (17) is (see [82],[83])

$$q_i^{(n+1)} = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^{k} a_{ij} q_j^n - f_i \right),\ i = 1, 2, \ldots, k \textit{ and } n \geq 0, \tag{18}$$

where the initial guess $q^{(0)} = (q_1^{(0)}, q_2^{(0)}, q_3^{(0)}, \ldots, q_k^{(0)})$ for the solution can be chosen arbitrarily.

Always we are starting with the zero initial vector $q^{(0)} = (0, 0, 0, \ldots, 0)$. The Jacobi iteration, obviously, consists of starting with an initial approximation $q^{(0)}$, and repeatedly applying the Jacobi update, creating a sequence $q^{(0)}, q^{(1)}, q^{(2)}, \ldots$ which converges to the exact solution. The control that makes sense to apply to the iteration checks the residual, that is, having computed the n^{th} iterate $q^{(n)}$, we define the residual $\varepsilon^{(n)}$ defined by

$$\varepsilon^{(n)} = Aq^{(n)} - f,$$

and control error using the RMS norm $\frac{\|\varepsilon^{(n)}\|}{\sqrt{k}}$. However, if we don't know the true solution, monitoring the residual is the proper way to control and terminate an iteration.

The error is defined as the difference between the approximate solution (18) and the exact solution (15)

$$\left(q_i^{(n+1)} - q_i\right) = -\sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| \left(q_i^{(n)} - q^i\right), \quad i = 1, 2, \ldots, k \quad and \quad n \geq 0.$$

Each component of the error satisfies

$$\varepsilon_i^{(n+1)} = -\sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| \varepsilon_i^{(n)}, \quad i = 1, 2, \ldots, k \quad and \quad n \geq 0$$

where $q_i^{(n)} - q_i = \varepsilon_i^{(n)}, n \geq 0$ is the error at the n^{th} approximation, so

$$\left|\varepsilon_i^{(n+1)}\right| = -\sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| \left|\varepsilon_i^{(n)}\right|, \quad i = 1, 2, \ldots, k \quad and \quad n \geq 0.$$

$$\|\varepsilon^{(n+1)}\|_\infty \leq \underset{1\leq i\leq k}{max} \sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| \|\varepsilon^{(n)}\|_\infty, \quad i = 1, 2, \ldots, k \quad and \quad n \geq 0.$$

$$\|\varepsilon^{(n+1)}\|_\infty \leq R\|\varepsilon^{(n)}\|_\infty \tag{19}$$

where

$$R = \underset{1\leq i\leq k}{max} \sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| = \|R\|_\infty.$$

This shows that the rate of convergence is linear. Equation (19) implies that

$$\|\varepsilon^1\|_\infty \leq \|R\|\|\varepsilon^{(0)}\|_\infty$$

$$\|\varepsilon^2\|_\infty \leq \|R\|\|\varepsilon^{(1)}\|_\infty \leq \|R\|^2\|\varepsilon^{(0)}\|_\infty$$

$$\|\varepsilon^3\|_\infty \leq \|R\|\|\varepsilon^{(2)}\|_\infty \leq \|R\|^3\|\varepsilon^{(0)}\|_\infty$$

and so, on

$$\|\varepsilon^n\|_\infty \leq \|R\|^n\|\varepsilon^{(0)}\|_\infty.$$

If $\|R\| < 1, \|R\|^n \to 0$, as $n \to \infty$, then $\|\varepsilon\|^n \to 0$ as $n \to \infty$, the Jacobi iteration method converges.

Definition 2. [80] *The matrix A is said to be diagonally dominant if and only if $\|R\| < 1$. In order for $\|R\| < 1$ to be true, the coefficient matrix A must be diagonally dominant, that is*

$$\underset{1\leq i\leq k}{max} \sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| < 1 \implies \sum_{\substack{j=1\\ j\neq i}}^{k} \left|\frac{a_{ij}}{a_{ii}}\right| < 1, \quad i = 1, 2, \ldots, k$$

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{k} |a_{ij}|\,, \quad i = 1, 2, \ldots, k.$$

Therefore, if A is diagonally dominant, the Jacobi method converges. Thus, the Jacobi iterative method converges if the given system of linear equations is strictly diagonally dominant by rows.

2.6.2. Gauss-Seidel Iterative Method. The Gauss-Seidel method uses the previously updated values in the current approximation to find the rest of them. Carrying out the same derivations as for equation (17) for this splitting of the matrix A for Gauss-Seidel (GS) method is $M = D+L, K = U$ and its iteration gives

$$q^{(n+1)} = R_{GS}q^{(n)} + C_{GS}, \tag{20}$$

where $R_{GS} = -M^{-1}K = -(D+L)^{-1}(U),\ C_{GS} = M^{-1}f = (D+L)^{-1}f.$

The component form of equation (20) is (see [82],[83])

$$q_i^{(n+1)} = -\frac{1}{a_{ii}}\left(\sum_{j=1}^{i-1} a_{ij}q_j^{(n+1)} + \sum_{j=i+1}^{k} a_{ij}q_j^{(n)} - f_i\right),\ \ i = 1, 2, \ldots, k \ \ and \ \ n \geq 0 \tag{21}$$

where $q_i^{(n+1)}$ indicates that it is a new value of the current iteration.

2.6.3. SOR(α_{opt}) Iterative Method. An improvement to the Gauss-Seidel method can be made by anticipating future corrections to the approximation by making an over correction at each iterative step. The method is called the Successive over relaxation method SOR(α_{opt}). This method is based on the matrix splitting

$$\alpha A = (D + \alpha L) + ((\alpha - 1)D + \alpha U).$$

Applying this to the set of linear equations we find that

$$((D + \alpha L) + ((\alpha - 1)D + \alpha U))q = \alpha f.$$

The matrices D, L and U are the same as for the Gauss-Seidel method, and α is the successive over relaxation parameter. This matrix splitting results in the following iterative method $(D+\alpha L)q^{(n+1)} = -[(\alpha-1)D+\alpha U]q^{(n)} + \alpha f,\ M = (D + \alpha L),\ K = [(\alpha - 1)D + \alpha U]$ and its iteration gives

$$q^{(n+1)} = R_{SOR(\alpha)}q^{(n)} + C_{SOR(\alpha)}, \tag{22}$$

where $R_{SOR(\alpha)} = -M^{-1}K = -(D + \alpha L)^{-1}[(\alpha - 1)D + \alpha U], C_{SOR(\alpha)} = M^{-1}f = (D + \alpha L)^{-1}\alpha f.$

Here α is called relaxation parameter. If $\alpha = 1$, the method is equivalent to the Gauss-Seidel method. If $\alpha < 1$, the method is called under-relaxation, and $\alpha > 1$ is called over-relaxation.

The component form of equation (22) is see([82],[84])

$$q_i^{(n+1)} = (1-\alpha)q_i^{(n)} - \frac{\alpha}{a_{ii}}\left(\sum_{j=1}^{i-1} a_{ij}q_j^{(n+1)} + \sum_{j=i+1}^{k} a_{ij}q_j^{(n)} - f_i\right), \quad i = 1, 2, \ldots, k \ \mathit{and}\ n \geq 0 \qquad (23)$$

where $q_i^{(n+1)}$ indicates that it is a new value of the current iteration.

It can be shown that this converges for $0 < \alpha < 2$ (see [85]). When $\alpha = 1$ this is just the Gauss-Seidel method, $\alpha < 1$ is under-relaxation (which slows the convergence), and $1 < \alpha < 2$ is over-relaxation. The convergence rate depends on the value of α; choosing a value that is too large is as bad as choosing one that is too small because the solution will overshoot the final value. SOR(α_{opt}) is very easy to program but does require determining the relaxation parameter α (although this can be estimated empirically, since if α is too large the solution will oscillate). Note that R_J, R_{GS} and $R_{SOR(\alpha)}$ are known as iteration matrix of Jacobi, Gauss-Seidel and SOR(α_{opt}), respectively.

2.7. Convergence. The convergence rate indicates the number of iterations that is needed for an iterative method to find an approximation of the solution that is within a certain range of the exact solution. Both the rate of convergence and the choice of optimal over relaxation parameter are heavily dependent on finding the spectral radius of the iteration matrix, or at least an upper bound for it. All the iterative methods discussed in the previous sections are on the form

$$q^{(n+1)} = Rq^{(n)} + \vec{f}. \qquad (24)$$

The matrix R is the iteration matrix, and $\vec{f}$ is a vector. The iteration matrices for the three methods are given in Table I.

Method	Iteration Matrix
Jacobi	$-D^{-1}(L+U)$
Gauss-Seidel	$-(D+L)^{-1}(U)$
SOR	$-(D+\alpha L)^{-1}((\alpha-1)D+\alpha U)$

Table 1. The iteration matrices of the classic iterative methods.

The iterative methods will converge if the spectral radius of the iterative matrix is less than one for stability. For any matrix norm the following inequality is upheld (see [82]).

$$\rho(R) \leq \|R\|,$$

where $\rho(R)$ is the spectral radius of the matrix R.

Lemma 3. [86] *If A is irreducible and weakly row diagonally dominant, then both Jacobi's and Gauss-Seidel methods converges, and $\rho(R_{GS}) < \rho(R_J) < 1$.*

It is shown that the spectral radius of the iteration matrix of Gauss-Seidel method is simply the square of the spectral radius of the iteration matrix of the Jacobi method (see [80],[87])

$$\rho(R_{GS}) = (\rho(R_J))^2.$$

Lemma 4. [86] *If $R_{SOR(\alpha)}$ converges, then $\rho(R_{SOR(\alpha)}) \geq |\alpha - 1|$. Thus, $0 < \alpha < 2$ is required for convergence.*

Lemma 5. [86] *If A is symmetric positive definite, then $\rho(R_{SOR(\alpha)}) < 1$ for all $0 < \alpha < 2$.*

The spectral radius of the SOR(α_{opt}) method is found in the following subsection since it is dependent on the choice of over-relaxation parameter.

2.8. Over-relaxation Parameter. The over-relaxation parameter must be within a narrow range around the optimal value for the SOR(α_{opt}) algorithm to convergence considerably faster than the standard Gauss-Seidel algorithm. The optimal value for the over-relaxation parameter can be found based on the following analytical expression (see [88])

$$\alpha = \frac{2}{1 + \sqrt{1 - \rho(R_{GS})}}.$$

The spectral radius of the SOR(α_{opt}) iteration matrix for the optimal choice of α is given by the following expression,

$$\rho(R_{SOR(\alpha)}) = \frac{\rho(R_{GS})}{(1 + \sqrt{1 - \rho(R_{GS})})^2}$$

The over-relaxation parameter can be found experimentally. Finding it numerically is often the only choice since it is not always possible to derive the spectral radius of the Gauss-Seidel iteration matrix analytically. That is, GS method twice faster than Jacobi's method and, SOR(α_{opt}) is faster than both Jacobi and Gauss-Seidel for the system of equations (11).

3 Finite Difference Method

As in the case of ordinary differential equation, the idea of Finite difference method is to discretize the PDE by replacing the partial derivatives with their approximations, that is finite differences (see [89],[90]). We will illustrate the scheme with Poisson's equation. The effectiveness of this method is tested for some of Poisson's equations with known analytical solution using MATLAB software and the derived numerical results show that the method produces accurate results. In real-world systems, numerical methods can be used to provide precise results.

Let us divide a three-dimensional region into small regions with the increments in the x, y and z directions given as $\Delta x, \Delta y$ and Δz as shown in the above-mentioned figure. Each nodal point is designed by a numbering scheme i, j and k where i indicates x increment and j indicates y increment and k indicates z increment as shown in the Fig.1. In a case study on temperature distribution, the temperature at each nodal point (x_i, y_j, z_k) is the average temperature of the surrounding hatched region. A finite difference equation suitable for the interior nodes of a steady three-dimensional system can be obtained by considering Poisson's equation at the nodal point i, j, k as

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,j,k} + \left.\frac{\partial^2 u}{\partial y^2}\right|_{i,j,k} + \left.\frac{\partial^2 u}{\partial z^2}\right|_{i,j,k} = p(x_i, y_j, z_k).$$

The second order central difference scheme at the nodal point (i, j, k) can be approximated as

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,j,k} \approx \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{(\Delta x)^2}, \quad \left.\frac{\partial^2 u}{\partial y^2}\right|_{i,j,k} \approx \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{(\Delta y)^2},$$
$$\left.\frac{\partial^2 u}{\partial z^2}\right|_{i,j,k} \approx \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{(\Delta z)^2}.$$

The finite difference approximation of Poisson's equation for interior regions can be expressed as

$$\begin{aligned} 2u_{i,j,k}((\Delta y)^2(\Delta z)^2+(\Delta x)^2(\Delta z)^2+(\Delta x)^2(\Delta y)^2)-(\Delta y)^2(\Delta z)^2(u_{i+1,j,k}+u_{i-1,j,k}) \\ - (\Delta x)^2(\Delta z)^2(u_{i,j+1,k} + u_{i,j-1,k}) - (\Delta x)^2(\Delta y)^2(u_{i,j,k+1} + u_{i,j,k-1}) = \\ = -(\Delta x)^2(\Delta y)^2(\Delta z)^2 p(x_i, y_j, z_k). \quad (25) \end{aligned}$$

Let $\Delta x = \Delta y = \Delta z = h$. We can write equation (25) as

$$u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k}+u_{i,j,k+1}+u_{i,j,k-1}-6u_{i,j,k} = h^2 p(x_i, y_j, z_k). \quad (26)$$

More accurate higher order approximations for interior nodes and boundary nodes are also obtained in a same manner.

The Cauchy problem for elliptic equations occurs in the study of many practical problems and the Cauchy problem for Poisson equation has been investigated in many works (see [1],[2],[3]). The solution of a Cauchy problem for the Poisson equation is ill-posed in the sense of Hadamard [5],[91] and it will exist only if strong compatibility or smoothness conditions are imposed on the initial data. It was Hadamard who showed that unless a certain compatibility relation holds among the Cauchy data, no global solution can exist. He further showed that even if the data are such that a classical solution exists, this solution will not depend continuously on the data. These problems are well known to be ill-posed in the sense of Hadamard and many attempts, and many investigations from various aspects, as existence-uniqueness theorems, regularization, least square methods [12] for such problems have been discussed.

The purpose of this paper is to investigate regularization approaches and develop numerical methods for solving certain ill-posed problems for the 3D Poisson equation.

3.1. Reducing to Cube Domains. We first discretize BVP (5)-(8), (10) in all the three (x, y, z) dimensions on a uniform grid with $n_x \times n_y \times n_z$ grid points for that we consider a cube domain where $L = W = H$. If $\Delta x \neq \Delta y \neq \Delta z$, then we can separate our region into $n_x = \frac{L}{\Delta x}, n_y = \frac{W}{\Delta y}$ and $n_z = \frac{H}{\Delta z}$ subintervals along the x, y, and z axis. The goal is to approximate all the solutions, $u_{i,j,k}$ where $0 \leq i \leq n_x, 0 \leq j \leq n_y$, and $0 \leq k \leq n_z$. As we have seen from equation (26), any point $u_{i,j,k}$ in the region is related to the six points surrounding it. Consider a sketch of a region where $n_x = 3, n_y = 3$ and $n_z = 3$. Here we can view the cross sections of our cube at different z values.

Note that many of the values in this region are already defined. From the boundary conditions, it is known that $u_{0,j,k} = 0, u_{n_x,j,k} = 0, u_{i,0,k} = 0, u_{i,n_y,k} = 0, u_{i,j,0} = f(x_i, y_j)$, and $u_z(x_i, y_j, 0) = g(x_i, y_j)$. The remaining $(n_x - 1)(n_y - 1)(n_z)$ points will be approximated by building a linear system of equation. We will create a system of $(n_x - 1)(n_y - 1)(n_z) = 12$ equations, one for each solution at an internal point of our cube by iterating through all possible values of i, j and k, where $0 < i < n_x, 0 < j < n_y$, and $0 < k \leq n_z$.

Corollary 2. *Problem (5)-(8), (10) can be solved by discretization in all the three dimensions and transforming the discretized problem (26) by the eigenvector basis of the matrix A using the forward finite difference for the boundary conditions*

$$u_z(x_i, y_j, 0) \approx \frac{u_{i,j,1} - u_{i,j,0}}{\Delta z} = g(x_i, y_j).$$

For example, if working with $n_x = 3, n_y = 3$ and $n_z = 3$ subintervals, we can turn the system of linear equation into corresponding matrices and vectors as

$$A\vec{u} = \vec{f} - h^2\vec{p}, \tag{27}$$

where $\vec{u}$ is the vector of approximate solutions at each point, A is the coefficient matrix of these solutions, $\vec{f}$ is the vector of boundary conditions at these points, and $\vec{p}$ is the source function. Although this equation is same as two-dimensional case, the A coefficient matrix and boundary condition vector $\vec{f}$ will have different patterns.

3.2. Direct or Iterative Solution. A solution of the above system of equation may be obtained from direct Gaussian elimination method for a system of small $(n_x \times n_y \times n_z)$ unknown. For a large system of unknown, iterative methods achieve a better result. For all grid based numerical schemes according to [92], the accuracy of the numerical results depends greatly on the computational grid. Thus, a grid converged solution would be preferred for accuracy (i.e., when more grid points are used, solution does not change significantly as one approaches a tolerance point). Three iterative methods are proposed to use for this work. Details of each iterative methods are provided below.

If we apply equation (17) or (18) to solve the system of finite difference equations for 3D-Poisson equation, we obtain the Jacobi iteration formula (see [78])

$$u_{i,j,k}^{n+1} = \frac{1}{6}\left(u_{i+1,j,k}^{n} + u_{i-1,j,k}^{n} + u_{i,j+1,k}^{n} + u_{i,j-1,k}^{n} + u_{i,j,k+1}^{n} + u_{i,j,k-1}^{n} - h^2 p_{i,j,k}\right). \tag{28}$$

The superscript n is an iterative index. The initial iterative guess can be set at $n = 0$, to produce $u_{i,j,k}^{0}$ and then successively improve it according to the iteration. From equation (28) above, the next iteration $(n+1)$ can be found for each grid point (i, j, k) across all the grid points in the horizontal rows. On completion of the iteration for all interior grid points, the difference between the vectors of the next iteration u^{n+1} and the previous iteration u^n is computed. The iteration terminates once the predefined condition (tolerance) set for the iteration to converge is met and the solution to (28) is u^{n+1}, otherwise the iterations continue. i.e.

$$\left|u^{n+1} - u^{n}\right| < tolerance.$$

If we apply equation (20) or (21) to solve the system of finite difference equations for 3D-Poisson equation, we obtain the Gauss-Seidel iteration formula (see [92])

$$u_{i,j,k}^{n+1} = \frac{1}{6}\left(u_{i+1,j,k}^{n} + u_{i-1,j,k}^{n+1} + u_{i,j+1,k}^{n} + u_{i,j-1,k}^{n+1} + u_{i,j,k+1}^{n} + u_{i,j,k-1}^{n+1} - h^2 p_{i,j,k}\right). \tag{29}$$

As can be seen from equation (29), the values $u_{i-1,j,k}, u_{i,j-1,k}$, and $u_{i,j,k-1}$, updated already as one moves through the grids to reach the grid point (i, j, k). The implementation of this iteration scheme is the same as in Jacobi scheme.

The SOR(α_{opt}) method is the most used iteration method and is embedded in the Gauss-Seidel Method. A relaxation parameter α is included in the Gauss-Seidel iteration formula with the aim of quickening the convergence. Using (22) or (23), we obtain SOR(α_{opt}) iteration formula (see [93])

$$\begin{aligned} u_{i,j,k}^{n+1} &= (1-\alpha)u_{i,j,k}^{n} + \\ &+ \frac{\alpha}{6}\left(u_{i+1,j,k}^{n} + u_{i-1,j,k}^{n+1} + u_{i,j+1,k}^{n} + u_{i,j-1,k}^{n+1} + u_{i,j,k+1}^{n} + u_{i,j,k-1}^{n+1} - h^2 p_{i,j,k}\right). \end{aligned} \tag{30}$$

The relaxation parameter is of the range $0 < \alpha < 2$. Implementation of SOR(α_{opt}) proceeds the same way as the first two methods.

MATLAB programs are developed for the three iteration methods using Dirichlet and Neumann boundary conditions which are applied at the boundary of the domain. We can examine the results of our discretization and iterative approximations for the sample problem with different grid sizes. Our Jacobi, Gauss-seidel, and SOR(α_{opt}) iterations will use an RMS residual tolerance of 10^{-6}. We compare the exact solution to the continuous problem against the solution to the discretized problem computed directly and with the iteration techniques.

4 Numerical Experiments

4.1. Test problem 1: a known analytical solution. In this test problem we set in (5)-(8), (10)

$$p(x, y, z) = -2\pi^2 sin(\pi x) sin(\pi y)(z + 1),$$

$$f(x, y) = sin(\pi x) sin(\pi y),$$

$$g(x, y) = sin(\pi x) sin(\pi y).$$

These functions are chosen such that problem (5)-(8), (10) has a known analytical solution

$$u(x, y, z) = sin(\pi x) sin(\pi y)(z + 1).$$

4.2. Results and Discussions. From Table 2 we can compare the numerical performance of the Jacobi, the Gauss-Seidel, and the SOR(α_{opt}) methods in three dimensions. We observe that the Jacobi method is slower than previous methods but fails to converge in a reasonable amount of time for the higher values of n_x, n_y and n_z. The second panel of this table shows the numerical results from the Gauss-Seidel method on solving the Poisson equation for the same values of n_x, n_y and n_z. The Gauss-Seidel method requires about half as many iterations and correspondingly less time than the Jacobi method. But for the larger grid size, the required numbers of iterations are still unacceptable.

The SOR(α_{opt}) method has results shown in the third panel of Table 2. Here, we observe that this iterative method can solve larger systems than Jacobi, and the Gauss-Seidel methods and converges for higher choice of n_x, n_y and n_z is dramatically faster than the other classical iterative methods. We see that the SOR(α_{opt}) has a much lower iteration count, and has a lower runtime than the Jacobi and the Gauss-Seidel methods. Specifically, for $n_x = 160, n_y = 176$, and $n_z = 144$, the Gauss-Seidel method took 44633 iterations and 3104.330192 seconds versus the SOR(α_{opt}) method, which took only 766 iterations and just 55.142233 seconds.

3D - Jacobi Method :

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	4.3680e-07	3.2609e-07	0.0062561	297	0.041594
20 ×22 × 18	3.2723e-07	2.3450e-07	0.0013862	1302	0.322852
30 ×33 × 27	2.6889e-07	1.9182e-07	5.9147e-04	2993	1.366138
40 ×44 × 36	2.3380e-07	1.6642e-07	3.2541e-04	5357	5.281894
80 ×88 × 72	1.6600e-07	1.1787e-07	7.7028e-05	21409	149.910765
160 ×176 × 144	1.1763e-07	1.1196e-07	1.6926e-05	84358	5216.835643

3D - Gauss-Seidel Method :

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	4.2216e-07	2.9054e-07	0.0062564	162	0.026728
20 ×22 × 18	3.1859e-07	2.2298e-07	0.0013868	696	0.204045
30 ×33 × 27	2.6551e-07	1.8536e-07	5.9219e-04	1591	0.836068
40 ×44 × 36	2.3124e-07	1.6192e-07	3.2625e-04	2842	2.937105
80 ×88 × 72	1.6532e-07	1.1643e-07	7.8230e-05	11330	85.456343
160 ×176 × 144	1.1741e-07	1.1084e-07	1.8235e-05	44633	3104.330192

3D - SOR(α_{opt}) Method :

Grid size $n_x \times n_y \times n_z$	α	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	1.62	2.3452e-07	1.2369e-07	0.0062568	31	0.011925
20 ×22 × 18	1.81	1.4170e-07	7.0185e-08	0.0013874	71	0.093955
30 ×33 × 27	1.86	8.4366e-08	6.6814e-08	5.9290e-04	99	0.117842
40 ×44 × 36	1.91	8.1616e-08	3.5652e-08	3.2709e-04	159	0.263134
80 ×88 × 72	1.96	5.3672e-08	3.5202e-08	7.9444e-05	369	3.253304
160 ×176 × 144	1.98	4.4957e-08	1.9885e-08	1.9576e-05	766	55.142233

Table 2. Numerical results obtained for Test problem 1 using Jacobi, Gauss-Seidel and SOR(α_{opt}) methods.

The top panel of Table 2. shows the results of the iterative method on this problem. The first column represents the mesh size, the second column is the numerical method error, and the third column is the inverse problem error. The fourth column represents the spectral radius of the matrix A, and the fifth column shows the number of iterations taken by the iterative method until convergence to the chosen tolerance of 10^{-6} in the relative residual. The last column is the wall clock time of each run.

4.3. Comparison of Iterative Methods. As mentioned above, we have demonstrated that iterative methods that are sophisticated have converged in fewer iterations within a shorter wall clock run time. We now aim to further characterize the differences in convergence rate in terms of iterations. In order to do so we have taken the spectral radius of the iteration matrix, from each method with different grid sizes of $n_x \times n_y \times n_z$. Then, we plot the relative residuals versus the iteration number and compare the results from each method, which is shown in Figure 2 and Figure 3. Figure 2(a) depicts the convergence of each method for the grid size $10 \times 11 \times 9$ and (b) shows the same data, but with grid $40 \times 44 \times 36$. In Figure 3 here we have two visualizations of this result. Figure 3(a) depicts the convergence of each method for the grid size $160 \times 176 \times 144$ and (b) shows the same data, but only for the first 1000 iterations, so that we can take a closer look at the methods that converge very fast.

From the Figures 2 and 3 we can see that the Jacobi and Gauss-Seidel methods are very slow than SOR(α_{opt}) method. They clearly take many more iterations to converge.

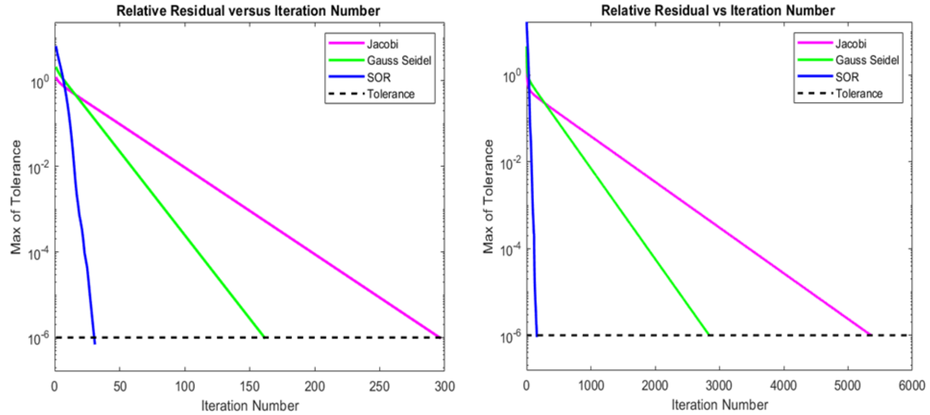

FIG. 2. Comparison of iterative methods in Test problem 1: (a) The relative residual versus the iteration number for each of the iterative methods with grid $10 \times 11 \times 9$ on a semilog plot (b) the same data as (a), but with grid $40 \times 44 \times 36$.

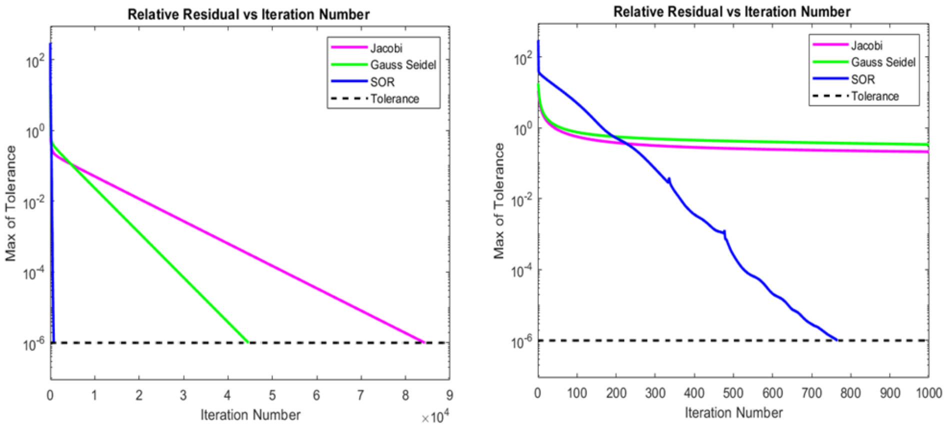

FIG. 3. Comparison of iterative methods in Test problem 1: (a) The relative residual versus the iteration number for each of the iterative methods with grid $160 \times 176 \times 144$ on a semilog plot (b) the same data as (a), but only the first 1000 iterations.

Assumption 1. *Given $\omega \in [-1, 1]$ and the function f_δ defined as follows*

$$f_0 = f \quad \text{and} \quad f_\delta := f(x,y)\Big(1 + \omega(x,y)\frac{\delta}{100}\Big). \tag{31}$$

Here ω is randomly uniformly number on [-1, 1] for each x and y and δ is a percentage of the noise levels.

4.4. The noise data case. In what follows we address the noisy data case $\delta > 0$. We begin by defining the quadratic (square residual) functional $f_\delta : q \to f_\delta(q) := \frac{1}{2}\|Aq - f_\delta\|^2 \in \mathbb{R}^+$. The Jacobi method in the noisy data case reads: $q_\delta^{(n+1)} = R_J q_\delta^{(n)} + C_J$, the Gauss-Seidel method in the noisy data case reads: $q_\delta^{(n+1)} = R_{GS} q_\delta^{(n)} + C_{GS}$, and the SOR($\alpha_{opt}$) method in the noisy data case reads: $q_\delta^{(n+1)} = R_{SOR(\alpha)} q_\delta^{(n)} + C_{SOR(\alpha)}$.

4.5. Experiments with noisy data. The numerical experiments with noisy data discussed in this section. Here $\Omega = (0,1)^3$ and the three iterative methods are compared with different levels of noisy data. The setup of our numerical experiments is detailed as follows:

Solve problem (12) with $q = q^*$, and compute the exact data f. Add 5% and 10% of uniformly distributed random noise to the exact data, generating the noisy data f_δ.

4.6. Test problem 1 (5% noise). We present in Table 3 how the Jacobi, Gauss-Seidel and SOR(α_{opt}) methods perform under a 5% noise level. The stopping criteria are reached specifically, for $n_x = 160, n_y = 176$, and $n_z = 144$, the Gauss-Seidel method took 44633 iterations and 3106.350241 seconds versus the SOR(α_{opt}) method, which took only 766 iterations and just 55.83125 seconds.

For the noise level of 5%, the stability was seen for the grid $10 \times 11 \times 9$ at every value of q when $-0.1 \le \omega \le 0.1$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 5%. The stability was seen for the grids $20 \times 22 \times 18$ and $30 \times 33 \times 27$ at every value of q when $-0.01 \le \omega \le 0.01$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 5%. The stability was seen for the grids $40 \times 44 \times 36$ and $80 \times 88 \times 72$ at every value of q when $-0.001 \le \omega \le 0.001$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 5%. The stability was seen for the grid $160 \times 176 \times 144$ at every value of q when $-0.0001 \le \omega \le 0.0001$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 5%.

3D - Jacobi Method with 5% noise level :

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 $\times$11 $\times$ 9	4.3680e-07	3.2609e-07	9.5613e-07	297	0.042561
20 $\times$22 $\times$ 18	3.2723e-07	2.3450e-07	9.9625e-07	1302	0.326872
30 $\times$33 $\times$ 27	2.6889e-07	1.9182e-07	9.9751e-07	2993	1.384139
40 $\times$44 $\times$ 36	2.3380e-07	1.6642e-07	9.9905e-07	5357	5.421682
80 $\times$88 $\times$ 72	1.6600e-07	1.1787e-07	9.9953e-07	21409	149.961275
160 $\times$176 $\times$ 144	1.1763e-07	1.1196e-07	9.9991e-07	84358	5217.236945

3D - Gauss-Seidel Method with 5% noise level :

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 $\times$11 $\times$ 9	4.2216e-07	2.9054e-07	9.4660e-07	162	0.026167
20 $\times$22 $\times$ 18	3.1859e-07	2.2298e-07	9.8351e-07	696	0.217043
30 $\times$33 $\times$ 27	2.6551e-07	1.8536e-07	9.9453e-07	1591	0.842048
40 $\times$44 $\times$ 36	2.3124e-07	1.6192e-07	9.9542e-07	2842	2.952106
80 $\times$88 $\times$ 72	1.6532e-07	1.1643e-07	9.9921e-07	11330	86.257393
160 $\times$176 $\times$ 144	1.1741e-07	1.1084e-07	9.9992e-07	44633	3106.350241

3D - SOR(α_{opt}) Method with 5% noise level :

Grid size $n_x \times n_y \times n_z$	α	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 $\times$11 $\times$ 9	1.62	2.3452e-07	1.2369e-07	6.8754e-07	31	0.013625
20 $\times$22 $\times$ 18	1.81	1.4170e-07	7.0185e-08	9.0406e-07	71	0.096985
30 $\times$33 $\times$ 27	1.86	8.4366e-08	6.6814e-08	9.3266e-07	99	0.119442
40 $\times$44 $\times$ 36	1.91	8.1616e-08	3.5652e-08	9.0723e-07	159	0.265172
80 $\times$88 $\times$ 72	1.96	5.3672e-08	3.5202e-08	9.6601e-07	369	3.289307
160 $\times$176 $\times$ 144	1.98	4.4957e-08	1.9885e-08	9.9842e-07	766	55.831253

Table 3. Numerical results obtained for Test problem 1 with 5% noise level using Jacobi, Gauss-Seidel and SOR(α_{opt}) methods.

The top panel of Table 3. shows the results of the iterative method with 5% noise level on this problem. The first column represents the mesh size, the second column is the numerical method error, and the third column is the inverse problem error. The fourth column represents the maximum of tolerance, and the fifth column shows the number of iterations taken by the iterative method until convergence to the chosen tolerance of 10^{-6} in the relative residual. The last column is the wall clock time of each run.

4.7. Test problem 1 (10% noise). Similarly to the previous case, we present in Table 4 how the Jacobi, Gauss-Seidel and SOR(α_{opt}) methods perform under a 10% noise level. The stopping criteria are reached specifically, for $n_x = 160, n_y = 176$, and $n_z = 144$, the Gauss-Seidel method took 44633 iterations and 3108.070139 seconds versus the SOR(α_{opt}) method, which took only 766 iterations and just 56.24028 seconds.

For the noise level of 10%, the stability was seen for the grid $10 \times 11 \times 9$ at every value of q when $-0.1 \leq \omega \leq 0.1$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 10%. The stability was seen for the grids $20 \times 22 \times 18$ and $30 \times 33 \times 27$ at every value of q when $-0.01 \leq \omega \leq 0.01$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 10%. The stability was seen for the grids $40 \times 44 \times 36$ and $80 \times 88 \times 72$ at every value of q when $-0.001 \leq \omega \leq 0.001$, while $\omega =$ -0.1e-05 was the best choice for the noise level of 10%. The stability was seen for the grid 160 × 176 × 144 at every value of q when $-0.0001 \leq \omega \leq 0.0001$, while ω = -0.1e-05 was the best choice for the noise level of 5%.

3D - Jacobi Method with 10% noise level:

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 ×11 × 9	4.3680e-07	3.2609e-07	9.5613e-07	297	0.042853
20 ×22 × 18	3.2723e-07	2.3450e-07	9.9625e-07	1302	0.337251
30 ×33 × 27	2.6889e-07	1.9182e-07	9.9751e-07	2993	1.386017
40 ×44 × 36	2.3380e-07	1.6642e-07	9.9905e-07	5357	5.641304
80 ×88 × 72	1.6600e-07	1.1787e-07	9.9953e-07	21409	150.328731
160 ×176 × 144	1.1763e-07	1.1196e-07	9.9991e-07	84358	5219.136948

3D - Gauss-Seidel Method with 10% noise level :

Grid size $n_x \times n_y \times n_z$	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 ×11 × 9	4.2216e-07	2.9054e-07	9.4660e-07	162	0.045082
20 ×22 × 18	3.1859e-07	2.2298e-07	9.8351e-07	696	0.254073
30 ×33 × 27	2.6551e-07	1.8536e-07	9.9453e-07	1591	0.845172
40 ×44 × 36	2.3124e-07	1.6192e-07	9.9542e-07	2842	2.985087
80 ×88 × 72	1.6532e-07	1.1643e-07	9.9921e-07	11330	86.402342
160 ×176 × 144	1.1741e-07	1.1084e-07	9.9992e-07	44633	3108.070139

3D - SOR(α_{opt}) Method with 10% noise level :

Grid size $n_x \times n_y \times n_z$	α	$\|u - u_{exact}\|$	$\|q - q_{exact}\|$	Max Tolerance	Iterations	Run time (s)
10 ×11 × 9	1.62	2.3452e-07	1.2369e-07	6.8754e-07	31	0.013967
20 ×22 × 18	1.81	1.4170e-07	7.0185e-08	9.0406e-07	71	0.097024
30 ×33 × 27	1.86	8.4366e-08	6.6814e-08	9.3266e-07	99	0.127549
40 ×44 × 36	1.91	8.1616e-08	3.5652e-08	9.0723e-07	159	0.283304
80 ×88 × 72	1.96	5.3672e-08	3.5202e-08	9.6601e-07	369	3.355109
160 ×176 × 144	1.98	4.4957e-08	1.9885e-08	9.9842e-07	766	56.240282

Table 4. Numerical results obtained for Test problem 1 with 10% noise level using Jacobi, Gauss-Seidel and SOR(α_{opt}) methods.

The top panel of Table 4. shows the results of the iterative method with 10% noise level on this problem. The first column represents the mesh size, the second column is the numerical method error, and the third column is the inverse problem error. The fourth column represents the maximum of tolerance, and the fifth column shows the number of iterations taken by the iterative method until convergence to the chosen tolerance of 10^{-6} in the relative residual. The last column is the wall clock time of each run.

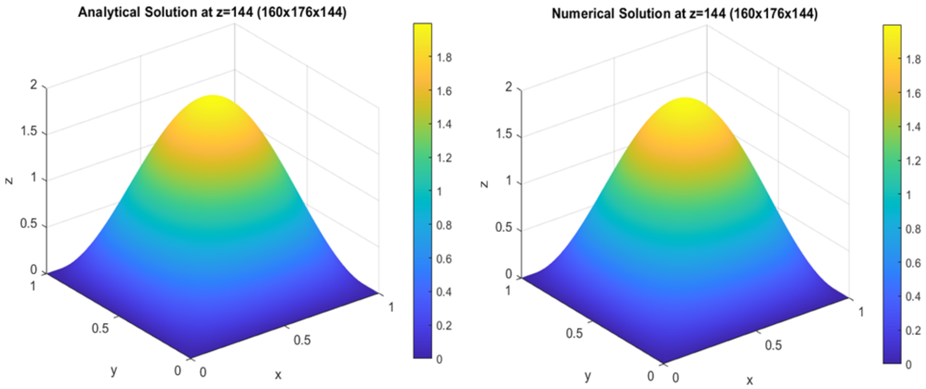

FIG. 4. For the grid $160 \times 176 \times 144$: (a) Analytical solution at $z = 144$ (b) Numerical solution at $z = 144$.

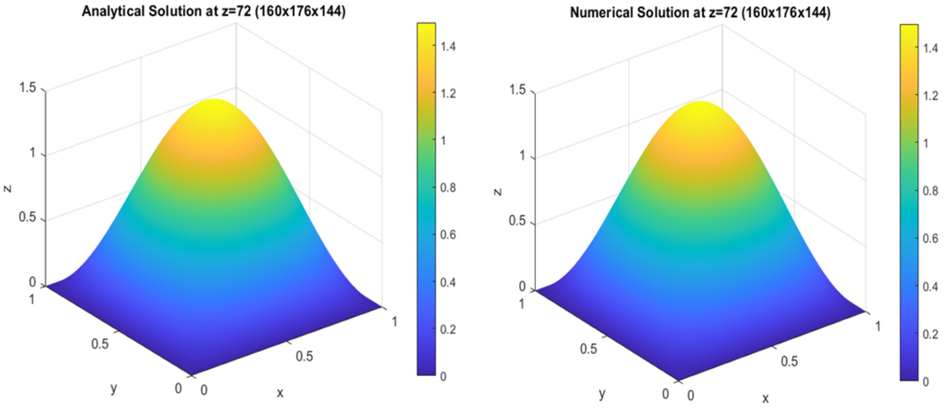

FIG. 5. For the grid $160 \times 176 \times 144$: (a) Analytical solution at $z = 72$ (b) Numerical solution at $z = 72$.

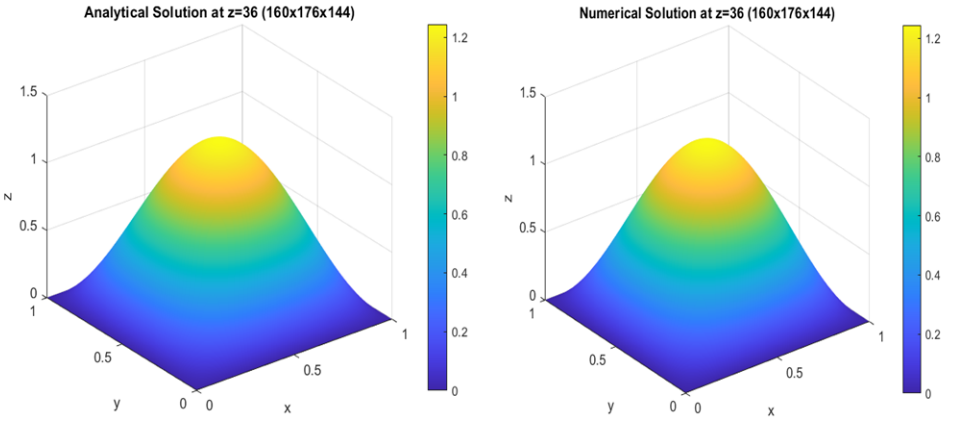

FIG. 6. For the grid $160 \times 176 \times 144$: (a) Analytical solution at $z = 36$ (b) Numerical solution at $z = 36$.

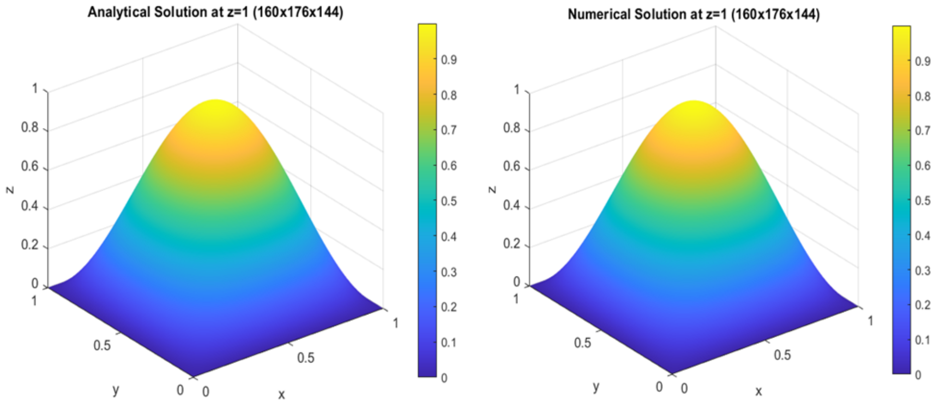

FIG. 7. For the grid $160 \times 176 \times 144$: (a) Analytical solution at $z = 1$ (b) Numerical solution at $z = 1$.

4.8. Maximum Error between Analytical and Numerical solutions. We present in Table 5 the direct and inverse problem errors, with 5% noise and with 10% noise for the different grid sizes using SOR(α_{opt}) method.

3D - SOR(α_{opt}) Method :

Grid size $n_x \times n_y \times n_z$	Direct problem Error	Inverse problem Error
$10 \times 11 \times 9$	0.0152	0.0156
$20 \times 22 \times 18$	0.0035	0.0036
$30 \times 33 \times 27$	0.0015	0.0016
$40 \times 44 \times 36$	8.5395e-04	8.6927e-04
$80 \times 88 \times 72$	2.0955e-04	2.1313e-04
$160 \times 176 \times 144$	5.1889e-05	5.2760e-05

3D - SOR(α_{opt}) Method with 5% noise level :

Grid size $n_x \times n_y \times n_z$	Direct problem Error	Inverse problem Error
$10 \times 11 \times 9$	0.0152	0.0156
$20 \times 22 \times 18$	0.0035	0.0036
$30 \times 33 \times 27$	0.0015	0.0016
$40 \times 44 \times 36$	8.5395e-04	8.6927e-04
$80 \times 88 \times 72$	2.0955e-04	2.1313e-04
$160 \times 176 \times 144$	5.1888e-05	5.2760e-05

3D - SOR(α_{opt}) Method with 10% noise level :

Grid size $n_x \times n_y \times n_z$	Direct problem Error	Inverse problem Error
$10 \times 11 \times 9$	0.0152	0.0156
$20 \times 22 \times 18$	0.0035	0.0036
$30 \times 33 \times 27$	0.0015	0.0016
$40 \times 44 \times 36$	8.5395e-04	8.6927e-04
$80 \times 88 \times 72$	2.0955e-04	2.1313e-04
$160 \times 176 \times 144$	5.1887e-05	5.2760e-05

Table 5. Maximum Error between Analytical and Numerical solution on Test problem 2 using SOR(α_{opt}) method.

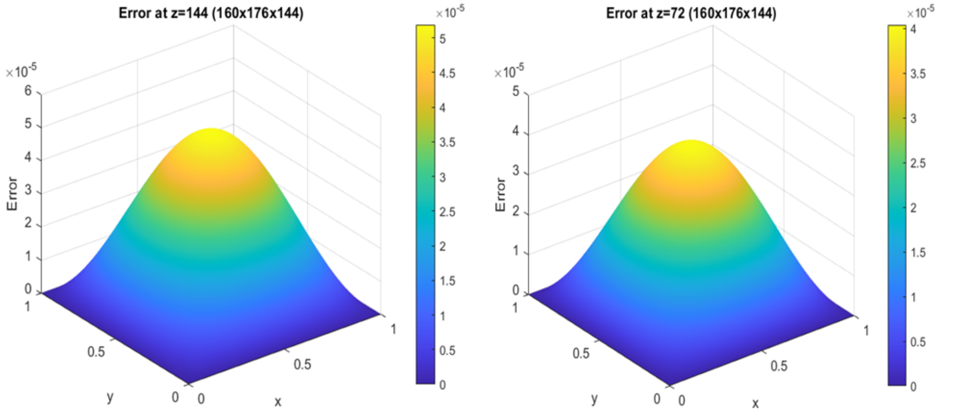

FIG. 8. Error between analytical and numerical solution$(160 \times 176 \times 144)$: (a) at $z = 144$ (b) at $z = 72$.

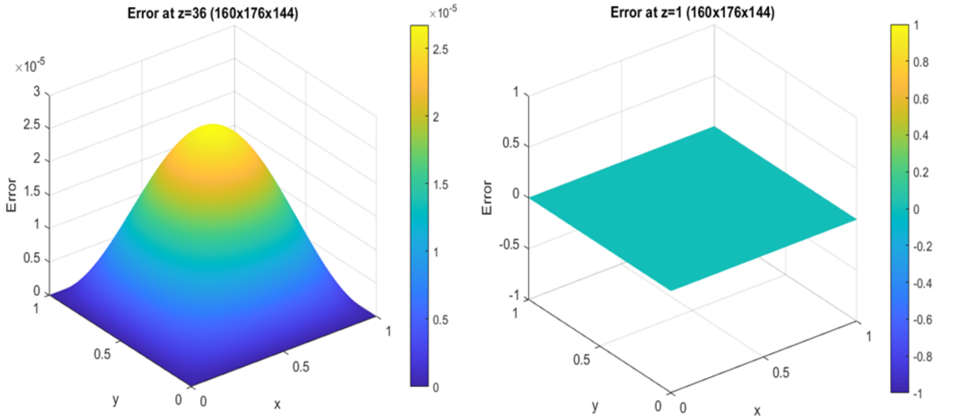

FIG. 9. Error between analytical and numerical solution$(160 \times 176 \times 144)$: (a) at $z = 36$ (b) at $z = 1$.

5 FDM vs HDF vs Landweber iteration

In [94], the Landweber iteration and horizontally diagonalize and fit methods were applied to approximate the 3D Poisson equation. It is already a proven theory that in one dimensional case the Landweber iteration method is effective for $n \geq 2500$ whereas in two dimensional case the HDF method is much more effective for $n \geq 2500$ than Landweber iteration method. We checked for the same grid sizes as shown in the HDF method using the Finite difference method with Jacobi, Gauss-Seidel and SOR(α_{opt}) iterative techniques. Based on our experimental results the Finite difference method is significantly more effective for $n \geq 2500$ than HDF method.

6 Continuation problem in geophysics (analyticity)

Quantitative regularities of the structure of geophysical and geochemical fields mathematically described by PDE of the second order. So, the gravitational and magnetic field are described by the Laplace or Poisson equation, electromagnetic field - by Maxwell equations, seismic field - by equations of the theory of elasticity, concentration field - diffusion equation. Fundamental in the continuation problems is the theorem of Cauchy-Kovalevskaya. The existence and uniqueness of analytic continuation of the gravitational field and the concentration field or chemical elements are guaranteed by this theorem. Assuming analyticity of the coefficients in the equations of geophysical and geochemical fields have the unique analytical solution. Analyticity of the coefficients of the equations implies the smoothness of the change of physical properties.

6.1. Application. The continuation problem of the gravimetry. The inhomogeneity of the density distribution below the surface of the Earth causes the gravitational field strength on the Earth's surface, which deviates from its average value. The deviations are small in percentage terms, but they are fixed by physical devices (gravimeters). Gravity data are used in mineral exploration. The purpose of gravimetric exploration - determine the location and shape of subsurface heterogeneities by known gravimetric measurement data.

7 Statement of the problem

7.1. Cauchy problem (continuation problem). Let us consider the ill-posed [77] continuation problem in which the unknown function $v(x, y, z)$ satisfies the following boundary-value problem (BVP):

$$\Delta v = p_1(x, y, z), (x, y, z) \in \Omega = (0, 5)^3 \tag{32}$$

$$v(0, y, z) = v(5, y, z) = 0, \tag{33}$$

$$v(x, 0, z) = v(x, 5, z) = 0, \tag{34}$$

$$v(x, y, 0) = f_1(x, y), \frac{\partial v}{\partial z}(x, y, 0) = g_1(x, y). \tag{35}$$

Here $p_1(x, y, z), f_1(x, y)$, and $g_1(x, y)$, are given functions and

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

7.2. Inverse problem. Let us consider the following direct (well-posed) problem (DP)

$$\Delta v = p_1(x, y, z), (x, y, z) \in \Omega, \tag{36}$$

$$v(0, y, z) = v(5, y, z) = 0, \tag{37}$$

$$v(x, 0, z) = v(x, 5, z) = 0, \tag{38}$$

$$\frac{\partial v}{\partial z}(x, y, 0) = g_1(x, y), \tag{39}$$

$$v(x, y, 5) = q_1(x, y). \tag{40}$$

The direct problem (36)-(40) is to find v for given p_1, g_1 and q_1. We suppose that function $q_1(x, y)$ is unknown. But instead of $q_1(x, y)$ we have the following additional information concerning the solution $v(x, y, z)$ of DP:

$$v(x, y, 0) = f_1(x, y). \tag{41}$$

The inverse problem (IP) consists in finding the function $q_1(x, y)$ from (36)-(41). The inverse problem (36)-(41) is equivalent to the continuation problem (32)-(35) in the following sense. If we solve the continuation problem then we can find $q_1(x, y) = v(x, y, 5)$, i.e., the solution of inverse problem q_1. Vice versa if we solve inverse problem and find the solution of inverse problem, we can set $v(x, y, 5) = q_1(x, y)$ and solve the direct problem (36)-(40) and find $v(x, y, 5) \in \Omega$ - the solution of the continuation problem.

7.3. Operator form of the Inverse problem. The inverse problem (36)-(41) can be formulated in operator form:

$$Aq_1 = f \tag{42}$$

Here $A : q_1(x, y) \rightarrow v(x, y, 0)$ and $v(x, y, z)$ is the solution of the direct problem (36)-(40).

Note that [78] $A : L_2((0, 5) \times (0, 5)) \rightarrow L_2((0, 5) \times (0, 5))$.

8 Numerical Experiments

8.1. Test problem 2: a known analytical solution. In this test problem we set in (36)-(39), (41)

$$p_1(x, y, z) = -2\pi^2 sin(\pi x) sin(\pi y)(z + 1),$$
$$f_1(x, y) = sin(\pi x) sin(\pi y),$$
$$g_1(x, y) = sin(\pi x) sin(\pi y).$$

These functions are chosen such that problem (36)-(39), (41) has a known analytical solution

$$v(x, y, z) = sin(\pi x) sin(\pi y)(z + 1).$$

8.2. Results and Discussions. From Table 6 we can compare the numerical performance of the Jacobi, the Gauss-Seidel, and the SOR(α_{opt}) methods in three dimensions. We observe that the Jacobi method is slower than previous methods but fails to converge in a reasonable amount of time for the higher values of n_x, n_y and n_z. The second panel of this table shows the numerical results from the Gauss-Seidel method on solving the Poisson equation for the same values of n_x, n_y and n_z. The Gauss-Seidel method requires about half as many iterations and correspondingly less time than the Jacobi method. But for the larger grid size, the required numbers of iterations are still unacceptable.

The SOR(α_{opt}) method has results shown in the third panel of Table 6. Here, we observe that this iterative method can solve larger systems than Jacobi, and the Gauss-Seidel methods and converges for higher choice of n_x, n_y and n_z is dramatically faster than the other classical iterative methods. We see that the SOR(α_{opt}) has a much lower iteration count, and has a lower runtime than the Jacobi and the Gauss-Seidel methods. Specifically, for $n_x = 160, n_y = 176$, and $n_z = 144$, the Gauss-Seidel method took 3944 iterations and 439.811217 seconds versus the SOR(α_{opt}) method, which took only 860 iterations and just 81.644467 seconds.

3D - Jacobi Method :

Grid size $n_x \times n_y \times n_z$	$\|v - v_{exact}\|$	$\|q_1 - q_{1exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	1.1091e-07	9.6210e-08	1.1531685	13	0.057656
20 ×22 × 18	2.9548e-07	2.9536e-07	0.1565441	73	0.182718
30 ×33 × 27	2.5206e-07	2.5203e-07	0.0347732	173	0.412693
40 ×44 × 36	2.3428e-07	2.2179e-07	0.0122092	312	0.731160
80 ×88 × 72	1.6769e-07	1.5906e-07	0.0010201	1264	17.459969
160 ×176 × 144	1.1909e-07	1.1342e-07	0.0008782	5022	550.569554

3D - Gauss-Seidel Method :

Grid size $n_x \times n_y \times n_z$	$\|v - v_{exact}\|$	$\|q_1 - q_{1exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	4.1609e-07	3.0506e-07	1.1531685	46	0.102597
20 ×22 × 18	3.0596e-07	2.6429e-07	0.1565441	141	0.277619
30 ×33 × 27	2.6018e-07	2.2724e-07	0.0347732	272	0.504966
40 ×44 × 36	2.2864e-07	2.0031e-07	0.0122092	441	1.304330
80 ×88 × 72	1.6548e-07	1.5299e-07	0.0010201	1378	19.242459
160 ×176 × 144	1.1898e-07	1.1441e-07	0.0008782	3944	439.811217

3D - SOR(α_{opt}) Method :

Grid size $n_x \times n_y \times n_z$	α	$\|v - v_{exact}\|$	$\|q_1 - q_{1exact}\|$	L_2 norm	Iterations	Run time (s)
10 ×11 × 9	1.62	1.2234e-07	4.1188e-08	1.1531685	38	0.087859
20 ×22 × 18	1.81	8.2793e-08	2.5612e-08	0.1565441	84	0.233653
30 ×33 × 27	1.86	5.2227e-08	2.2013e-08	0.0347732	122	0.613844
40 ×44 × 36	1.91	7.4979e-08	3.0396e-08	0.0122092	184	0.876061
80 ×88 × 72	1.96	5.9029e-08	3.0217e-08	0.0010201	417	6.327349
160 ×176 × 144	1.98	4.4517e-08	2.3042e-08	0.0008782	860	81.644467

Table 6. Numerical results obtained for Test problem 2 using Jacobi, Gauss-Seidel and SOR(α_{opt}) methods.

The top panel of Table 6. shows the results of the iterative method on this problem. The first column represents the mesh size, the second column is the numerical method error, and the third column is the inverse problem error. The fourth column represents the spectral radius of the matrix A, and the fifth column shows the number of iterations taken by the iterative method until convergence to the chosen tolerance of 10^{-6} in the relative residual. The last column is the wall clock time of each run.

8.3. Comparison of Iterative Methods. As mentioned above, we have demonstrated that iterative methods that are sophisticated have converged in fewer iterations within a shorter wall clock run time. We now aim to further characterize the differences in convergence rate in terms of iterations. In order to do so we have taken the spectral radius of the iteration matrix, from each method with different grid sizes of $n_x \times n_y \times n_z$. Then, we plot the relative residuals versus the iteration number and compare the results from each method, which is shown in Figure 10 and Figure 11. Figure 10(a) depicts the convergence of each method for the grid size 10 × 11 × 9 and (b) shows the same data, but with grid 40 × 44 × 36. In Figure 11 here we have two visualizations of this result. Figure 11(a) depicts the convergence of each method for the grid size 160 × 176 × 144 and (b) shows the same data, but only for the first 1200 iterations, so that we can take a closer look at the methods that converge very fast.

From the Figures 10(b) and 11 we can see that the Jacobi and Gauss-Seidel methods are very slow than SOR(α_{opt}) method. They clearly take many more iterations to converge.

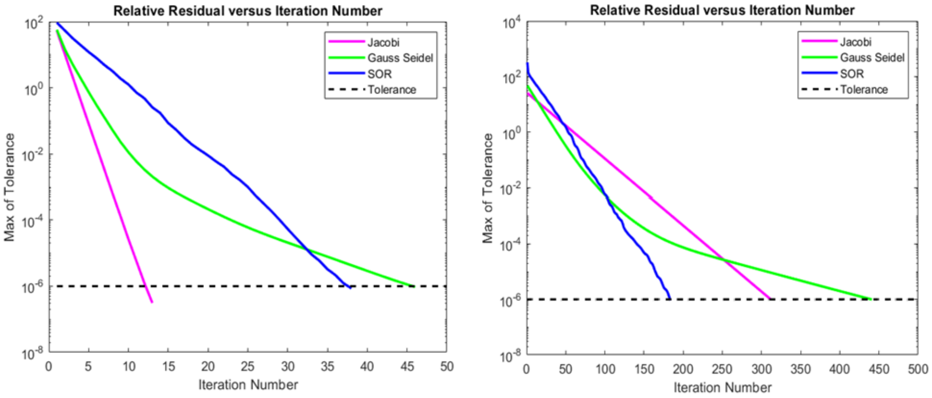

Fig. 10. Comparison of iterative methods in Test problem 2: (a) The relative residual versus the iteration number for each of the iterative methods with grid 10 × 11 × 9 on a semilog plot (b) the same data as (a), but with grid 40 × 44 × 36.

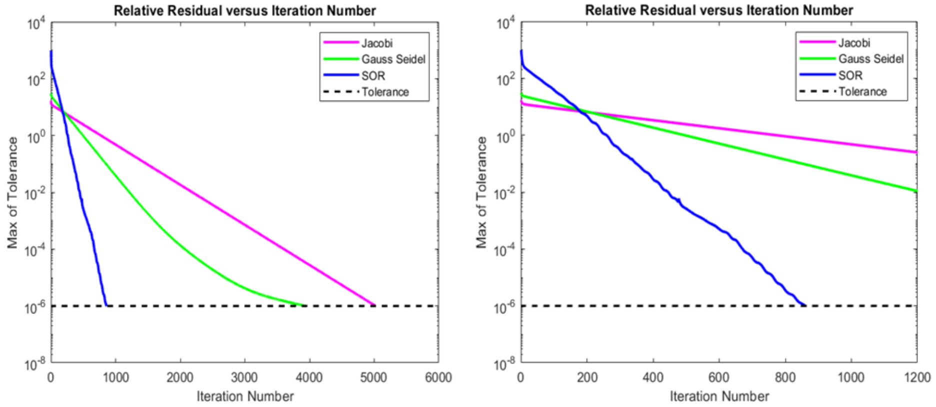

FIG. 11. Comparison of iterative methods in Test problem 2: (a) The relative residual versus the iteration number for each of the iterative methods with grid 160 × 176 × 144 on a semilog plot (b) the same data as (a), but only the first 1200 iterations.

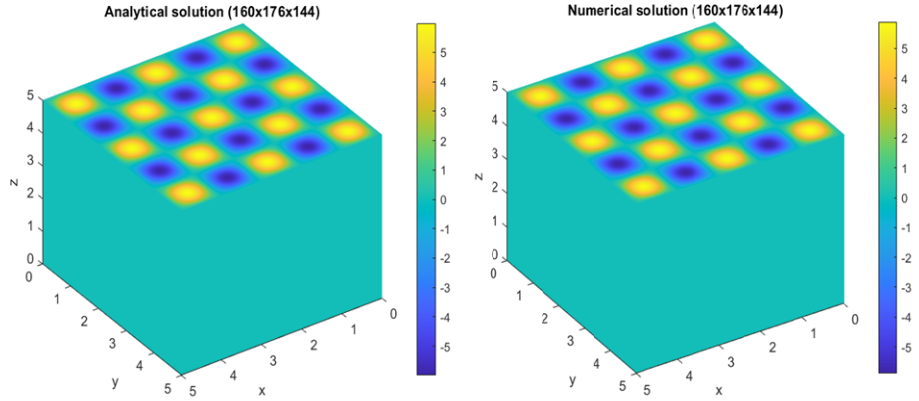

FIG. 12. For the grid 160×176×144: (a) Analytical solution (b) Numerical solution.

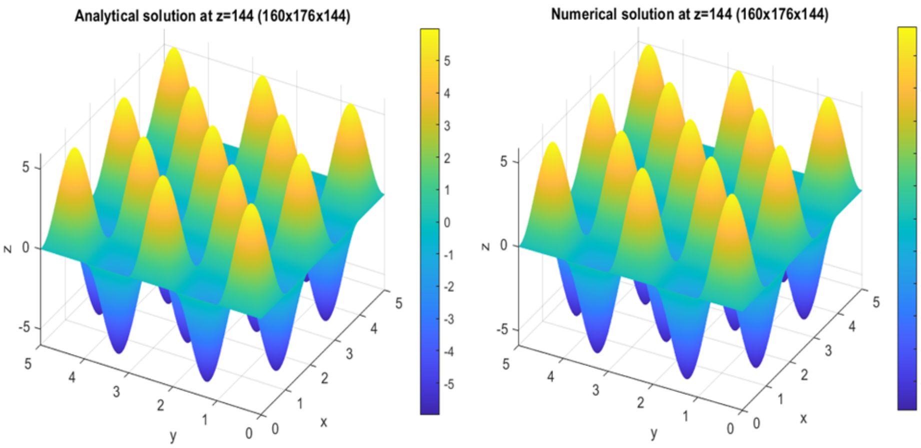

Fig. 13. For the grid 160 × 176 × 144: (a) Analytical solution at $z = 144$ (b) Numerical solution at $z = 144$.

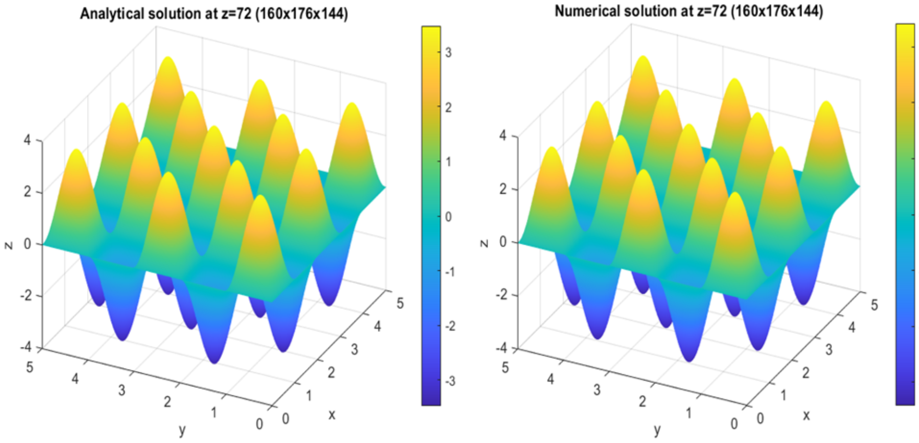

Fig. 14. For the grid 160 × 176 × 144: (a) Analytical solution at $z = 72$ (b) Numerical solution at $z = 72$.

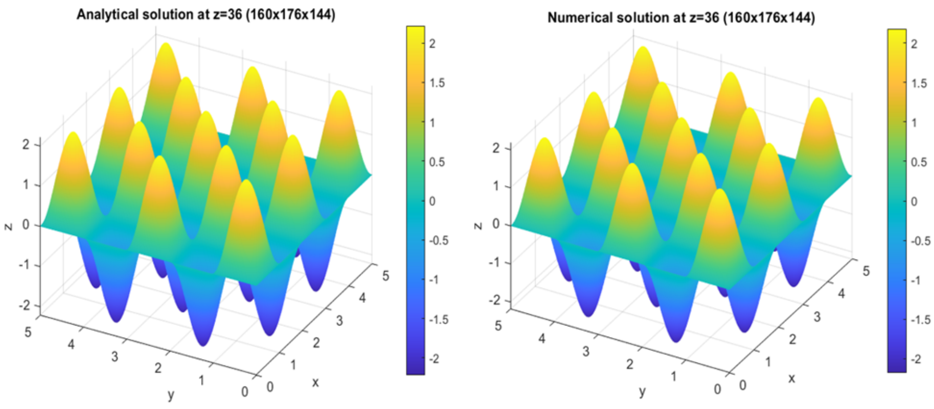

FIG. 15. For the grid 160 × 176 × 144: (a) Analytical solution at $z = 36$ (b) Numerical solution at $z = 36$.

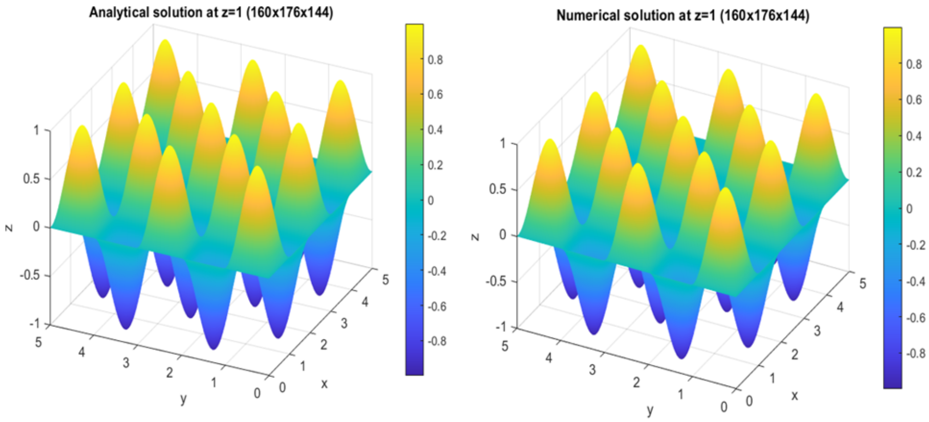

FIG. 16. For the grid 160 × 176 × 144: (a) Analytical solution at $z = 1$ (b) Numerical solution at $z = 1$.

8.4. Maximum Error between Analytical and Numerical solutions. We present in Table 7 the direct and inverse problem errors for the different grid sizes using SOR(α_{opt}) method.

3D - SOR(α_{opt}) Method :		
Grid size $n_x \times n_y \times n_z$	Direct problem Error	Inverse problem Error
10 $\times$11 $\times$ 9	1.3788613274	1.3788615707
20 $\times$22 $\times$ 18	0.2962412384	0.2962412368
30 $\times$33 $\times$ 27	0.1274467291	0.1274467273
40 $\times$44 $\times$ 36	0.0705250209	0.0705250212
80 $\times$88 $\times$ 72	0.0172666865	0.0172666880
160 $\times$176 $\times$ 144	0.0042709326	0.0042709306

Table 7. Maximum Error between Analytical and Numerical solution on Test problem 2 using SOR(α_{opt}) method.

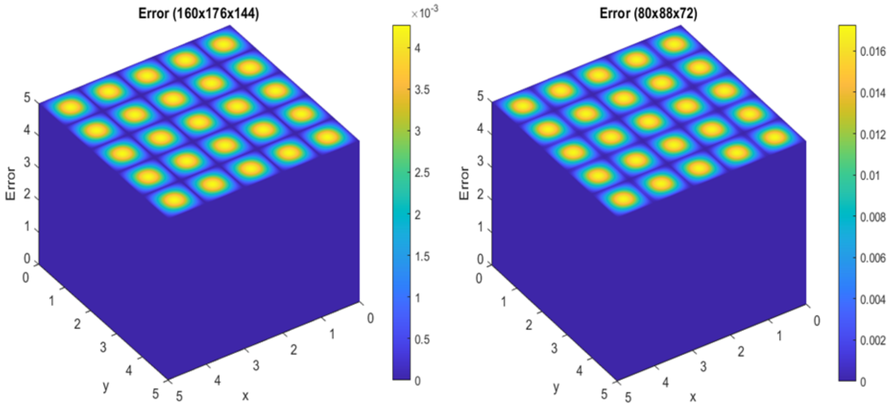

Fig. 17. Error between analytical and numerical solution: (a) for $160 \times 176 \times 144$ (b) for $80 \times 88 \times 72$.

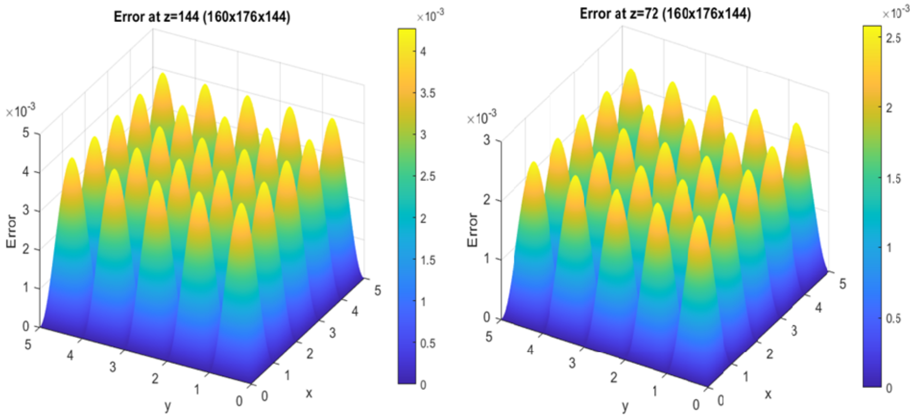

Fig. 18. Error between analytical and numerical solution($160 \times 176 \times 144$): (a) at $z = 144$ (b) at $z = 72$.

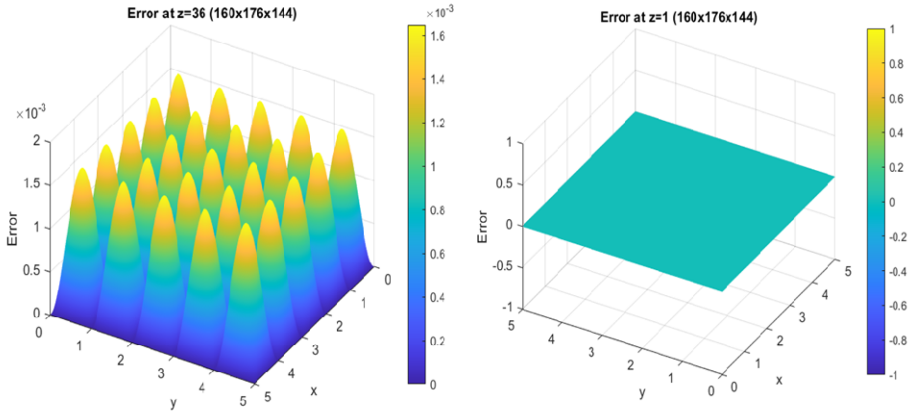

FIG. 19. Error between analytical and numerical solution($160 \times 176 \times 144$): (a) at $z = 36$ (b) at $z = 1$.

9 Conclusion

This research considered the distribution of electric potential in a cube through the numerical solution of Poisson's equation. Three different iteration schemes under Finite Difference Method (FDM) were compared which are the Jacobi, the Gauss Seidel and the SOR(α_{opt}) iteration methods. The results obtained showed clearly that the SOR(α_{opt}) iteration method is more efficient in terms of accuracy and speedy convergence.

References

[1] J. Douglas, *A numerical method for analytic continuation, Boundary Value Problems in Differential Equations*, Univ. of Wisconsin Press, Madison (1960), 179–189.

[2] J. R. Cannon, *Error estimates for some unstable continuation problems*, Journal of the Society for Industrial and Applied Mathematics, **12**:2 (1964), 270–284.

[3] J. R. Cannon, K. Miller, *Some problems in numerical analytic continuation*, J. SIAM Numer. Anal. **2**:1 (1965), 87–98.

[4] M. M. Lavrentiev, *On Cauchy problem for Laplace equation*, Reports of the USSR Academy of Sciences, **102** (1955), 205–206.

[5] M. M. Lavrentiev, V.G. Romanov, S.P.Shishat ski, *ill- posed problems of mathematical physics and analysis*, Transl. Math. Monographs. Amer. Math. Soc., Providence, R.I., (1986).

[6] S. N. Mergelyan, *Harmonic approximation and approximate solution of the Cauchy problem for the Laplace equation*, Successes Of Mathematical Sciences, **5**:71 (1956), 3–26.

[7] M. M. Lavrentiev, *On Cauchy problem for Laplace equation*, Izvestia of the USSR Academy of Sciences, **20** (1956), 819–842.

[8] M. M. Lavrentiev, *On the Cauchy problem for linear elliptic equations of second order*, Dokl. Akad. Nauk SSSR, **112** (1957), 195–197.

[9] L. E. Payne, *Bounds in the Cauchy problem for the Laplace equation*, Rational Mechanics and Analysis, **5** (1960), 35–45.

[10] L. E. Payne, D. Sather, *On some improperly posed problems for quasilinear equations of mixed type*, Trans. Amer. Math. Soc., **128** (1967), 135–141.

[11] L. E. Payne, *On a priori bounds in the cauchy problem for elliptic equations*, SIAM J. Math. Anal., **1** (1970), 82–89.

[12] L.E. Payne, *Improperly posed problems in Partial Differential Equation*, SIAM, Philadelphia, PA (1975).

[13] U. Tautenhahn, *Optimal stable solution of Cauchy problems for elliptic equations*, J. Anal. Appl., **4** (1990), 961–984.

[14] J.-J. Reinhardt, H. Han, D. N. Hao, *Stability and regularization of discrete approximation to the Cauchy problem for the Laplace's equation*, SIAM Numer. Anal., **36**:3 (1999), 890–905.

[15] A. Douglis, *Uniqueness in Cauchy problems for elliptic systems of equations*, Comm. Pure Appl. Math., **13** (1960), 593–608.

[16] P. W. Schaefer, *On the cauchy problem for an elliptic system*, Arch. Rational Mech. Anal., **20** (1965), 391–412.

[17] M. H. Protter, *Unique continuation for elliptic equations*, Trans. Amer. Math. Soc., **95** (1960), 81–91.

[18] K. Watanabe, C. Zuily, *On the uniqueness of the Cauchy problem for elliptic differential operators with smooth charateristics of variable multiplicity*, Comm. Partial Differential Equations, **2** (1977), 831–855.

[19] H. Kumano-go, *On the uniqueness of the solution of the Cauchy problem and the unique continuation theorem for elliptic equations*, Osaka Math. J., **14** (1962), 182–212.

[20] L. A. Chudov, *Difference methods for solving Cauchy problem for Laplace's equation*, Soviet Math. Dokl., **3** (1962), 499–503.

[21] R. S. Falk, P. B. Monk, *Logarithmic convexity for discrete harmonic functions and the approximation of the Cauchy problem for Poisson's equation*, Math. Comp., 47 **175** (1986), 135–149.

[22] R. S. Falk, *Approximation of inverse problems*, SIAM, Philadelphia, PA (1990), 7–16.

[23] H. Han, J.-J. Reinhardt, *Some stability estimates for Cauchy problems of elliptic equations*, J. Inverse Ill-Posed Probl., **5** (1997), 437–454.

[24] V. A. Kozlov, V. G. Mazya, A. V. Fomin, *An adaptive shifted power method for computing generalized tensor eigenpairs*, Zh. Vychisl. Mat. Fiz., **31**:1 (1991), 64–74.

[25] D. N. Hào, T. V. Duc, R. Gorenflo, *Towards the Cauchy problem for the Laplace equation*, Banach Center Publications, **27**:1 (1992), 111–128.

[26] D. N. Hào, D. Lesnic, *The Cauchy problem for Laplace's equation via the conjugate gradient method*, IMA J. Applied Math, **65**:2 (2000), 199–217.

[27] Y. Sun, *Modified method of fundamental solutions for the Cauchy problem connected with the Laplace equation*, Int Journal Computational Mathematics, **91**:10 (2014), 2185–2198.

[28] F. Berntsson and L. Eldén, *Numerical solution of a cauchy problem for the laplace equation, Inverse Problems*, **17** (2001), 839–853.

[29] L. Marin, D. N. Hào, D. Lesnic, *Conjugate gradient-boundary element method for the Cauchy problem in elasticity*, The Quarterly Journal of Mechanics and Applied Mathematics, **55**:2 (2002), 227–247.

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

[31] G. Alessandrini, L. Rondi, E. Rosset, S. Vessella, *The stability for the Cauchy problem for elliptic equations*, Inverse Problems, **25** (2009), 123004 (47 pages).

[32] D. N. Hào, B. Johansson, D. Lesnic, P. M. Hien, *A variational method and approximations of a Cauchy problem for elliptic equations*, Journal of Algorithms and Computational Technology, **4**:1 (2010), 89–120.

[33] J. Helsing, B. Johansson, *Fast reconstruction of harmonic functions from cauchy data using the dirichlet-to-neumann map and integral equations*, Inverse Problems in Science and Engineering, **19**: 5 (2011), 717–727.

[34] D. Maxwell, Kozlov-mazya, *Iteration as a form of Landweber iteration*, Inverse Problems Imaging, **8**:2 (2014), 537–560.

[35] X. Hu, L. Mu, X. Ye, *A simple Finite element method of the Cauchy problem for Poisson equation*, International Journal of Numerical Analysis and Modeling, **14**:4-5 (2017), 591–603.

[36] D. N. Hào, L. T. T. Giang, S. I. Kabanikhin, M. A. Shishlenin, *A Finite difference method for the very weak solution to a Cauchy problem for an elliptic equation*, Journal of Inverse and Ill-Posed Problems, **26**:6 (2018), 835–857.

[37] P. X. Thanh, O. Steinbach, *Energy space approaches to the Cauchy problem for Poisson's equation*, Acta Mathematica Vietnamica, **45** (2020), 693–707.

[38] X. Wang, X. Ye, S. Zhang, P. Zhu, *A weak galerkin least squares finite element method of cauchy problem for poisson equation*, Journal of Computational and Applied Mathematics, **401**:12 (2022), 113767.

[39] S. I. Kabanikhin, A. L. Karchevsky, *Method for solving the Cauchy problem for an elliptic equation*, Journal of Inverse and Ill-posed Problems, **1** (1995), 21–46.

[40] S. I. Kabanikhin, Y. S. Gasimov, D. B. Nurseitov, M. A. Shishlenin, B. B. Sholpanbaev, S. Kasenov, *Regularization of the continuation problem for elliptic equations*, Journal of Inverse and Ill-Posed Problems, **21**:6 (2013), 871–884.

[41] S. I. Kabanikhin, M. A. Shishlenin, D. B. Nurseitov, A. T. Nurseitova, S. E. Kasenov, *Comparative analysis of methods for regularizing an initial boundary value problem for the helmholtz equation*, Journal of Applied Mathematics, (2014), 786326 (7 pages).

[42] M. V. Klibanov, F. Santosa, *Cauchy problems for Laplace's equation*, SIAM J. Appl. Math., **51**:6 (1991), 1653–1675.

[43] L. Bourgeois, *A mixed formulation of quasi-reversibility to solve the cauchy problem for laplace's equation*, Inverse Problems, **21** (2005), 1087.

[44] L. Bourgeois, *Convergence rates for the quasi-reversibility method to solve the Cauchy Problem for laplace's equation*, Inverse Problems, **22** (2006), 413.

[45] L. Bourgeois, *About stability and regularization of ill-posed elliptic Cauchy problems: the case of $c^{1,1}$ domains*, ESAIM: Mathematical Modelling and Numerical Analysis, **44**:4 (2010), 715–735.

[46] L. Bourgeois, J. Dardé, *A duality-based method of quasi-reversibility to solve the cauchy problem in the presence of noisy data*, Inverse Problems, **26**:9 (2010), 095016.

[47] J. Cheng, Y. C. Hon, T. Wei, and M. Yamamoto, *Numerical computation of a cauchy problem for laplace's equation*, Zeitschrift fu€r Angewandte Mathematik und Mechanik, **81**:10 (2001), 665–674.

[48] Y. C. Hon, T. Wei, *Backus-gilbert algorithm for the Cauchy problem of the Laplace equation*, Inverse Problems, **17**:2 (2001), 261–271.

[49] H. W. Engl, A. Leitão, *A mann iterative regularization method for elliptic Cauchy problems*, Numer. Funct. Anal. and Optimiz., **22** (2001), 861–884.

[50] Z. Qian, C.-L. Fu, X.-T. Xiong, *Fourth-order modified method for the Cauchy Problem for the Laplace equation*, Journal of Computational and Applied Mathematics, **192** (2006), 205–218.

[51] C.-L. Fu, H.-F. Li, Z. Qian, X.-T. Xiong, *Fourier regularization method for solving a Cauchy problem for the Laplace equation*, Inverse Problems in Science and Engineering, **16** (2008), 159–169.

[52] F. Yang, *The truncation method for identifying an unknown source in the Poisson equation*, Applied Mathematics and Computation, **217**:22 (2011), 9334–9339.

[53] C.-Y. Qiu, C.-L. Fu, *Wavelets and regularization of the Cauchy problem for the Laplace equation*, Journal of Mathematical Analysis and Applications, **338**:2 (2008), 1440–1447.

[54] A. Leitão, M. M. Alves, *On level set type methods for elliptic Cauchy problems*, Inverse Problems, **23**:5 (2007), 2207–2222.

[55] V. N. Titarenko, A. G. Yagola, *Cauchy problems for Laplace equation on compact sets*, Inverse Problems in Engineering, **10**:3 (2002), 235–254.

[56] L. Marin, *A meshless method for solving the cauchy problem in three-dimensional elastostatics*, Computers and Mathematics with Applications, **50** (2005), 73–92.

[57] L. Eldén, V. Simoncini, *A numerical solution of a Cauchy problem for an elliptic equation by krylov subspaces*, Inverse Problems, **91** (2009), 2187–2205.

[58] J. R. Cannon, P. DuChateau, *Approximating the solution to the Cauchy problem for laplace's equation*, SIAM Journal on Numerical Analysis, **14**:3 (1977), 473–483.

[59] S. Yarmukhamedov, *On the Cauchy problem for Laplace's equation*, Soviet Math. Dokl., **18** (1977), 882–939.

[60] A. Belaid, B. Abda, N. Malki, *The Cauchy problem for the Laplace equation and application to image inpainting*, ISRN Math. Anal., (2011), 11.

[61] V.K. Ivanov, *The Cauchy problem for the Laplace equation in an infinite strip*, Differentsialnye Uravneniya, **1** (1965), 131–136.

[62] S. B. Sorokin, *An efficient direct method for numerically solving the Cauchy problem for Laplace's equation*, Numerical Analysis and Applications, **12** (2019), 87–103.

[63] C.-S. Liu, *An analytical method for the inverse Cauchy problem of Laplace equation in a rectangular plate*, Journal of Mechanics, **27**:4 (2011), 575–584.

[64] V. G. Mazya, V. P. Khavin, *On the solutions of the Cauchy problem for Laplace's equation (solvability, normality, approximation)*, Trans, Moscow Math. Soc., **30** (1974), 65–117.

[65] A. Romanovich, A. V. Fursikov, *On the question of the solvability of the Cauchy problem for the Laplace operator*, Vestnik Moskov. Gos. Univ. Ser. I Mat. Mekh., **3** (1987), 78–80.

[66] F. B. Belgacem, H.E. Fekih, *On cauchy's problem: I. a variational steklov-poincaré theory*, Inverse Problems, **21** (2005), 1915.

[67] F. B. Belgacem, *Why is the Cauchy problem severly ill-posed?*, Inverse Problems, **23** (2007), 823–836.

[68] A. P. Calderón, *Uniqueness in the Cauchy problem for partial differential equations*, Amer. J. Math., **80** (1958), 16–36.

[69] A. Cimetiere, F. Delvare, M. Jaoua, F. Pons, *Solution of the Cauchy problem using iterated tikhonov regularization*. Inverse Problems, **17**:3 (2001), 553–570.

[70] H. Han, *The Finite element method in a family of improperly posed problems*, Math. Comp., **38**:157 (1982), 55–65.

[71] F. John, *A note on "improper" problems in partial differential equations*, Communications on Pure and Applied Mathematics, (1955), 494–495.

[72] F. John, *Continuous dependence on data for solutions of partial differential equations with a prescribed boundaries*, Communications on Pure and Applied Mathematics, (1960), 551–585.

[73] M. V. Klibanov, *Carleman estimates for the regularization of ill-posed Cauchy problems*, Applied Numerical Mathematics, **94** (2015), 46–74.

[74] V. Shaydurov, S. Kabanikhin, V. Petrakova, *Two algorithms for refinement of approximate solutions in the regularization method*, Lobachevskii Journal of Mathematics, **44**:1 (2023), 434–445.

[75] R. Haberman, *Applied Partial Differential Equations: With Fourier Series and Boundary Value Problems.* Upper Saddle River, NJ: Pearson Prentice Hall, 4 editions (2004).

[76] R. E. Hunt, *Chapter 2: Poisson's equation*, Department of Applied Mathematics and Theoretical Physics, University of Cambridge (2002).

[77] S. I. Kabanikhin, *Definitions and examples of inverse and ill-posed problems*, Journal of Inverse and Ill-Posed Problems. **16**:4 (2008), 317–357.

[78] D. M. Causon, C. G. Mingham, *Introductory Finite Difference Methods for PDES*, Ventus Publishing Aps (2010).

[79] O. Axelsson, *Iterative solution methods*, Cambridge university press, Cambridge, UK (1994).

[80] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York (1971).

[81] J. C. Strikwerda, *Finite difference Schemes and Partial Differential Equations*, 2nd Edition, SIAM (2004).

[82] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia (2003).

[83] G. D. Smith, *Numerical solution of partial differential equations Finite difference methods*, 3rd Edition, Oxford university press, New York (1985).

[84] R. L. Burden, J. F Douglas, *Numerical analysis*, Canada: cengage learning (2010).

[85] W. F. Ames, *Numerical methods for partial differential equations*, 2nd edition, Academic Press, New York (1977).

[86] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, (1997).

[87] R. S. Varga, *Matrix iterative analysis*, Springer, Berlin (2000).

[88] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P Flannery, *Numerical Recipes Third Edition*, Cambridge University Press, New York (2007).

[89] J. R. Nagel, *Numerical solutions to Poisson equations using the Finite difference method*, IEEE An. and Pr. Magazine, **56**:4 (2014), 209–224.

[90] R. Dambrose, *Algorithms to Approximate Solutions of Poisson's Equation in Three Dimensions*, Rose-Hulman Undergraduate Mathematics Journal, **20**:1, (2019).

[91] J. Hadamard, *Lectures on Cauchy's problem in Linear Partial differential equations*, Yale Univ. Press, New Haven (1923).

[92] C. J. Vendromin, *Numerical Solutions of Laplace's Equation for Various Physical Situations*, Brock University, St. Catharine's Canada, (2017).

[93] X. I. A. Yang and R. Mittal, *Acceleration of the Jacobi Iterative Method by factors exceeding 100 using Scheduled Relaxation*, Journal of Computational Physics, **274** (2014), 695–708.

[94] M. A. Botchev, S. I. Kabanikhin, M. A. Shishlenin, E. E. Tyrtyshnikov, *The Cauchy problem for the 3D Poisson equation: Landweber iteration vs. horizontally diagonalize and fit method, J. of Inverse and Ill-posed Problems*, **31**:2 (2023), 203–221.

Sreelatha Chandragiri
Sobolev Institute of Mathematics,
pr. Koptyuga, 4,
630090, Novosibirsk, Russia
Email address: srilathasami@math.nsc.ru

Maxim Alexandrovich Shishlenin
Sobolev Institute of Mathematics,
pr. Koptyuga, 4,
630090, Novosibirsk, Russia
Email address: m.a.shishlenin@mail.ru

Sergey Igorevich Kabanikhin
Sobolev Institute of Mathematics,
pr. Koptyuga, 4,
630090, Novosibirsk, Russia
Email address: ksi52@mail.ru