Alfarama Journal of Basic & Applied Sciences

Faculty of Science Port Said University

https://ajbas.journals.ekb.eg
ajbas@sci.psu.edu.eg
http://sci.psu.edu.eg/en/

July 2025, Volume 6, Issue III

DOI: 10.21608/ajbas.2025.385670.1255

ISSN 2682-275X

Submitted: 21/05/2025
Accepted: 25/06/2025

Pages: 304 - 320

Bi-Conjugate Gradient Stabilized and Multigrid Approaches Based on Nonstandard Finite Differences for the Solution of Elliptic Partial Differential Equations

N. H. Sweilam[1,*], Ibrahim M. Hanafy[3], Moutaz Ramadan[2,3], Azza Eseily[3]

[1]Mathematics Department, Faculty of Science, Cairo University, Cairo, Egypt.

[2]Department of Mathematics, College of Sciences, Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia.

[3]Department of Mathematics, Faculty of Science, Port Said University, Port Said, Egypt.

***Corresponding author: azza.esilly@sci.psu.edu.eg**

ABSTRACT

This study introduces a robust and highly accurate numerical scheme for the solution of two-dimensional elliptic partial differential equations subject to Dirichlet boundary conditions. The proposed method is based on the Nonstandard Finite Difference technique, which is designed to achieve improved accuracy and stability compared to the classical Standard Finite Difference approach. To solve the resulting large, sparse linear systems, we employ two iterative solvers: the Bi-Conjugate Gradient Stabilized method and a Multigrid method. Within the multigrid framework, the Generalized Minimal Residual algorithm is utilized as a smoothing strategy to enhance convergence behavior. A comprehensive set of numerical experiments is carried out to assess and compare the performance of these approaches in terms of convergence rate, computational time, and number of iterations. The results, obtained for a range of grid resolutions and problem configurations, demonstrate the superior performance and efficiency of the proposed NSFD-based scheme, particularly on finer grids, confirming its effectiveness and reliability for solving elliptic problems.

Keywords: Iterative solvers, elliptic PDEs, Dirichlet boundary conditions; Nonstandard Finite Difference; Finite difference method; Multigrid method; Generalized minimal residual method; Biconjugate gradient stabilized method.

1. INTRODUCTION

The Poisson equation is a fundamental second-order elliptic partial differential equation (PDE) that appears often in physics and engineering problems. It is given by:

$$\nabla^2 u(x,y) = f(x,y) \tag{1}$$

where ∇^2 denotes the Laplacian operator, and u is the unknown scalar function and f is source function.

The Laplace equation a special case of the Poisson equation (where the right-hand side is zero), which implies that its solutions are infinitely differentiable in regions where boundary conditions are well-defined. Due to its wide applications, studying the behavior of its solutions, especially in complex domains or under irregular boundary conditions, is of great interest in modern computational mathematics [1-3].

In recent years, researchers have increasingly focused on expanding classical theories to more general settings—particularly when solving the Poisson equation in irregular or singular domains. For instance, the study in [4] looked at how solutions behave on bounded subanalytic manifolds, offering valuable insights that could be applied in fields like physics and geometry. Poisson's equation plays a central role in applied mathematics and engineering, especially in problems involving steady-state heat conduction and temperature distribution.

There has also been growing interest in studying key physical phenomena modeled by partial differential equations (PDEs), such as the heat and wave equations, which are commonly discussed in works like [5,6]. To solve these types of equations, researchers use a variety of mathematical and numerical techniques, as highlighted in [7]. Over time, many reliable methods have been developed for numerically solving PDEs under different conditions and domain complexities.

One of the most widely known methods is the Finite Difference Method (FDM) , which involves discretizing the domain into grid points and approximating derivatives using finite difference formulas. It's valued for its simplicity and is often used to solve problems in heat transfer, fluid flow, and electromagnetic analysis. Another powerful method is the Finite Element Method (FEM), which is built on variational principles and offers greater flexibility when dealing with complex geometries or boundary conditions.

The Nonstandard Finite Difference (NSFD) method stands out because it's designed to preserve essential qualitative features of the original differential equations like positivity, stability, or boundedness that standard methods might overlook. Meanwhile, the Boundary Element Method (BEM) transforms PDE problems into boundary integral equations, which reduces the problem's dimensionality. This makes it especially useful for problems defined in infinite or semi-infinite domains, as demonstrated in [8–12].

In particular, the Dirichlet boundary value problem for the Laplace equation involves determining a solution u within a domain D, where the boundary values of u are prescribed. A common example of this problem appears in heat conduction, where fixed temperatures are set along the boundary and the goal is to find the steady-state temperature inside the region**.** The solution to the Dirichlet problem in this context represents the equilibrium temperature profile within the domain, as in [13,14] Thus, solving the Dirichlet boundary value problem for the Laplace equation is important for understanding many steady-state physical systems.

The FDM is one of the most commonly used numerical techniques for approximating solutions to PDEs. It is also considered one of the simplest and oldest methods for solving differential equations**.** It is based on discretizing the continuous domain into a grid and approximating derivatives by differences between function values at adjacent grid points. FDM has been widely used to solve various problems in physics and engineering, such as heat conduction, fluid dynamics, and electromagnetic field analysis.[15]. NSFD methods are recognized as effective alternatives to the FDM for solving a variety of mathematical models, including algebraic equations, biological systems, and chaotic models [16]. These methods are known for their improved stability and accuracy, especially when there are issues with standard methods.

The NSFD approach was developed by Mickens [17,18] who proposed clear way for building numerical schemes that keep the key properties of the original differential equations. These methods

modify standard discretizations by incorporating nonlocal terms, using special denominator functions, or adjusting the mesh sizes to better capture the behavior of the continuous system.

Researchers have successfully applied NSFD methods in various fields**.** For example, Moaddy, Momani, and Hashim applied NSFD to solve linear fractional partial differential equations in fluid mechanics [19]. In another study, Sweilam, Al-Mekhlafi, and Baleanu developed an NSFD scheme to solve complex-order fractional Burgers' equations, demonstrating that the method could accurately preserve the physical behavior of the solution while achieving high numerical convergence [20].

The SFD and NSFD methods approximate the derivatives in the PDEs using finite difference operators [21], transforming the continuous Poisson equation into a large, sparse linear system. After discretization, the resulting linear system is solved using iterative solvers, which is especially efficient for large-scale problems where direct solvers become computationally expensive [22,23].

When equation (1) is discretized using FDM or NSFD methods it is generally converted into a linear system of the form:

$$Ax = b \tag{2}$$

where $A \in R^{n\times n}$ is the resulting coefficient matrix arising from the discretization process, $x \in R^n$ is the unknown solution vector representing the discrete values of the potential field, and $b \in R^n$ is the right-hand side vector. Efficiently solving such systems is essential, particularly when addressing large-scale problems or complex geometries. Many effective methods have been developed to solve these systems, especially when the resulting matrices are ill-conditioned. One of the most important methods is Multigrid MG, which has shown strong performance in these cases.

The MG-GMRES method combines MG and generalized minimal residual method (GMRES), with MG acting as the smoother to accelerate convergence. It efficiently addresses error components across multiple spatial scales, making it well-suited for large, complex problems [24]. MG methods depend on smoothers, such as Gauss-Seidel or Jacobi, to reduce high-frequency errors. However, when traditional smoothers become ineffective, GMRES-based smoothers provide enhanced robustness, particularly for ill-conditioned problems. However, GMRES smoothers, rather than traditional ones, can be more effective in these cases.

On the other hand, Bi-Conjugate Gradient Stabilized (BiCGStab) is another well-regarded iterative method, especially known for its stability and effectiveness in solving nonsymmetric linear systems. Unlike traditional methods, BiCGStab is specifically designed to address ill-conditioned problems more efficiently, providing strong performance even when other methods face difficulties in converging [25]. By appropriately integrating these methods, more efficient solutions to challenging linear systems can be achieved, ensuring both stability and fast convergence.

The Poisson equation is a fundamental mathematical model used in many fields, including heat transfer, fluid dynamics, and electrostatics. Because it's not easy to solve this equation exactly, numerical methods are used to find approximate solutions. One widely used approach is the FDM, known for its simplicity and ease of implementation. However, standard methods like FDM may not always provide sufficient accuracy or stability, especially for complex problems or when using very fine grids.

The main goal of this work is to enhance both the accuracy and efficiency of solving the Poisson equation. Traditional methods may fail to capture important features of the solution, especially on fine grids or near boundaries, and some iterative solvers suffer from slow or unstable convergence for large-scale problems. To address these challenges, this study proposes a novel numerical approach that combines the NSFD scheme with two advanced solvers: BiCGStab and MG methods. Furthermore, the GMRES method is used as a smoother within the MG framework to improve convergence. The effectiveness of the proposed approach is evaluated on benchmark problems and compared with standard methods in terms of accuracy and computational efficiency.

The key advantage of the proposed approach is its ability to deliver more accurate solutions with fewer iterations and lower computational cost. It performs well even for challenging problems and fine grid

resolutions, making it a practical and efficient tool for solving complex scientific and engineering problems.

2. SFD Method

The two-dimensional Poisson equation can be expressed as

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \ \ (x,y) \in R \ , \\ u(x,y) = g(x,y), \ \ (x,y) \in \partial R. \end{cases} \tag{3}$$

Here, R refers to a square domain in the 2D plane, ∂R denotes the boundary of that domain, $g(x,y)$ is the known boundary condition, and $u(x,y)$ represents the analytical solution of the problem.

To numerically approximate the solution, the second derivatives with respect to x and y are discretized using central finite difference formulas. The second derivatives with respect to x and y are approximated using central differences as follows:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}, \tag{4}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} \tag{5}$$

By inserting these expressions into the Poisson equation, we obtain the discrete approximation:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = f(x,y). \tag{6}$$

Solving this equation for $u_{i,j}$ yields the following weighted average formula:

$$u_{i,j} = \frac{1}{2\left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}\right)} \left(\frac{u_{i,j+1} + u_{i,j-1}}{(\Delta x)^2} + \frac{u_{i+1,j} + u_{i-1,j}}{(\Delta y)^2} - f(x,y)\right) \tag{7}$$

In this work, we adopt an SFD scheme over a uniform grid to investigate the numerical behavior of Poisson's equation (3). Assuming equal spacing in both coordinate directions, we let $\Delta x = \Delta y = h$.

The grid spacing is determined by $h = \frac{b-a}{n}$, where a and b denote the lower and upper bounds of the domain in both the x- and y-directions, respectively, and n is the number of subintervals along each axis. with the grid nodes defined as $x_i = ih$, $y_j = jh$ for $i, j = 1,2, \ldots, n$.

This numerical formulation is widely used for solving elliptic partial differential equations and is discussed in more detail in [26].

(a) (b)

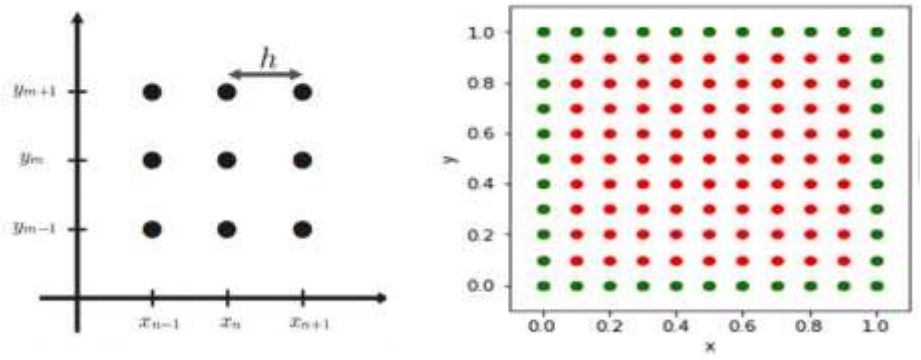

Fig 1. (a)The mesh points for the FDM grid, (b) Discrete grid points for $N = 10$ and $h = 0.1$ used in the discretization of the Poisson equation red is unknown point and green is boundary condition.

3. NSFD Method

The NSFD method, developed by Ronald E. Mickens [27], is a numerical technique designed to enhance the accuracy and stability of traditional FDM. Unlike conventional approaches, NSFD formulates special difference equations that better preserve critical properties of the original differential equation, such as positivity, symmetry, and stability. When applied to Poisson's equation, the NSFD method transforms the equation into a linear algebraic system, maintaining the correct behavior of the solution, especially near boundaries or in complex geometries.

To numerically approximate the solution of the Poisson equation, we apply an NSFD scheme following the approach proposed in [28]. This method constructs a discrete version of the Laplace operator using specially designed denominator functions, leading to a scheme that preserves the qualitative properties of the continuous model.

Based on previous studies and guided by the general NSFD framework, equation (3) is discretized as follows:

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{\varphi_1(\Delta x)} + \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{\varphi_2(\Delta y)} = f(x,y) \tag{8}$$

where:

- $u_{i,j}$ represents the numerical approximation at the grid point (x_i, y_j).
- $\varphi_1(\Delta x)$, $\varphi_2(\Delta y)$are nonstandard denominator functions satisfying:

$$\varphi_m(h) = h^2 + o(h^4), m = 1,2 \tag{9}$$

Solving (8) for $u_{i,j}$yields:

$$u_{i,j} = \frac{1}{2(\frac{1}{\varphi_1(\Delta x)}+\frac{1}{\varphi_2(\Delta y)})} \left(\frac{u_{i,j+1}+u_{i,j-1}}{\varphi_1(\Delta x)} + \frac{u_{i+1,j}+u_{i-1,j}}{\varphi_2(\Delta y)} - f(x,y)\right) \tag{10}$$

After applying both the SFD and NSFD schemes, the continuous Poisson equation is transformed into a linear algebraic system. This transformation enables the application of iterative solvers to obtain numerical approximations of the solution. In this study, particular attention is given to two prominent methods: the MG method and the BiCGStab method.

4. Multigrid (V-Cycle) with GMRES Smoother

Iterative methods are essential for the numerical solution of general elliptic PDEs. In finite difference (FD) discretizations, a common trade-off exists: finer grids offer higher accuracy but result in slower convergence, while coarser grids lead to faster convergence, at the cost of reduced accuracy.

A key challenge with traditional iterative solvers is their inefficiency in removing low-frequency error components, which tend to remain on finer grids. However, these smooth errors can be greatly reduced by transferring them to coarser grids a fundamental principle behind multigrid (MG) methods.

This principle underlies MG methods, which are among the most efficient solvers for elliptic problems. MG techniques combine relaxation (smoothing) steps on fine grids with coarse-grid correction strategies, effectively accelerating convergence by addressing all frequency components of the error. The algorithm cycles through multiple grid levels, typically using a V-cycle, W-cycle, or Full Multigrid cycle, depending on the specific strategy.

A critical component of the MG method is the smoother, which is responsible for effectively damping high-frequency errors. While classical choices include Gauss-Seidel or weighted Jacobi methods, in our case, we employ the GMRES method as the smoother due to its robustness and effectiveness, especially for more challenging or non-symmetric problems.

The standard MG algorithm consists of the following key steps:

1. Pre-smoothing: Apply a few iterations of an iterative method (in this case, GMRES) to damp high-frequency errors on the fine grid.
2. Residual Computation: Compute the residual, which represents the discrepancy between the exact and current approximate solution.
3. Restriction: Transfer (restrict) the residual to a coarser grid.
4. Coarse Grid Correction: Solve the error equation on the coarse grid (either exactly or approximately).
5. Prolongation (Interpolation): Transfer the coarse-grid correction back to the fine grid and update the fine-grid solution.
6. Post-smoothing: Apply a few more iterations of GMRES to eliminate any high-frequency errors reintroduced by the interpolation.

This process can be repeated recursively across multiple grid levels, forming the basis of multilevel solvers that are both accurate and efficient.

Algorithm (1): MG solver

Input: Matrix $A \in R^{n\times n}$ initial vector $x \in R^n$, right-hand side $b \in R^n$, error tolerance ε, number of levels λ, numbers of pre/post-relaxations steps v_{pre} , v_{post}
Output: Solution $x \in R^n$to the linear system $Ax = b$
1: Function Multigrid$(A, x, b, \varepsilon, \lambda, v_{pre}, v_{post})$:
2: Build prolongation matrices $P_1, P_2, P_3, \dots, P_\lambda$
3: $A_l \leftarrow A$
4: For $l \leftarrow 2\ to\ \lambda$ do
    $A_l \leftarrow (P_{l-1})^T A_{l-1} P_{l-1}$ %% Build level l matrix
5: end
6: repeat
7: $x \leftarrow MGI(x, b, 1)$
8: until $\| Ax - b \| \leq \varepsilon \| b \|$ %% Convergence test
9: return x
10: End Function

Algorithm (2): MG (*V*-cycle)

Input: Current iterate $x \in R^n$, right-hand side $b \in R^n$, level l
Output: New iterate $x \in R^{n_l}$
1: Function $MGI(x, b, l)$:
2: if $l < \lambda$ then
3: $y \leftarrow Relax(A_l, x, b, v_{pre})$ %%Pre-relaxation
4: $u \leftarrow MGI(0, P_l^T (b - A_l\ y), l + 1)$ %%Recursive call
5: $x \leftarrow Relax(A_l, x + P_l\, u, b, v_{post}))$ %%Post-relaxation
6: else
7: solve $A_\lambda x = b$ using a direct solver
8: end
9: return x
10: End Function

5. The BiCGStab Method

The BiCGStab algorithm is a Krylov subspace iterative method specifically designed for solving nonsymmetric and non-Hermitian linear systems. It was developed as an enhancement of the Conjugate Gradient Squared (CGS) algorithm introduced by Sonneveld [29]. The CGS algorithm itself originates from the Biconjugate Gradient (BiCG) method, which is based on the Lanczos biorthogonalization process [30].

All these algorithms fall under the broader category of Krylov subspace methods, which iteratively construct approximate solutions by projecting the original problem onto a sequence of subspaces generated by the coefficient matrix A and the initial residual vector.

The BiCG method operates by maintaining two sequences of vectors that are biorthogonal to each other through coupled recurrence relations. Although BiCG is effective in many cases, it often suffers from irregular convergence and numerical instabilities. The CGS algorithm attempts to address this by squaring the BiCG recurrence, which can improve convergence smoothness, but also tends to amplify round-off errors and may introduce significant oscillations in the residual norm.

To mitigate these issues, the BiCGStab algorithm introduces a stabilization mechanism that combines the robustness of BiCG with smoother and more reliable convergence. It modifies the CGS formulation by incorporating additional scalar parameters that stabilize the residuals, resulting in improved numerical behavior and faster convergence in many practical applications.

In the following, we present the detailed algorithmic steps of the BiCGStab method:

Algorithm 3: BiCGStab.

1: $r_0 = b - Ax_0$
2: $\rho_0 = \alpha_0 = w_0 = 1,\ v_0 = p_0 = 0$
3: for $i = 1,2,3, \ldots, do$
4: $\rho_i = r_0^T r_{n-1}$
5: $\beta = \left(\frac{\rho_i}{\rho_{i-1}}\right)\left(\frac{\alpha_{i-1}}{\omega_{i-1}}\right)$
6: $p_i = r_{i-1} + \beta_i(p_{i-1} - \omega_{i-1}v_{i-1})$
7: $v_i = Ap_i$
8: $\alpha_i = \frac{\rho_i}{r_0^T v_i}$
9: $s_i = r_{i-1} - \alpha_i v_i$
10: $t_i = As_i$
11: $\omega_i = \frac{t_i^T s_i}{t_i^T t_i}$
12: $x_i = x_{i-1} + \alpha_i p_i + \omega_i s_i$
13: $r_{i+1} = s_i - \omega_i\, t_i$

14: If x_i is accurate enough the
15: STOP
16: end if
17: end for

6. Numerical Experiment

To evaluate the accuracy of the proposed method, we examine two problems governed by Poisson's equation. These problems are discretized using SFDM, and the resulting linear systems are first solved using the MG-GMRES method. Then, we apply NSFD method, solving the corresponding linear systems with both Bi-CGSTAB and MG-GMRES. We present visual representations of the computed solutions along with their associated error distributions. For numerical comparison, key performance metrics such as iteration count (iter) and CPU time (in seconds) are reported. A uniform convergence tolerance of $\varepsilon = 10^{-15}$is enforced across different grid resolutions, denoted by N, (COND) also refers to the condition number. The precision of the computed solutions is assessed using the L_2 and L_∞ error norms, which quantify deviations from the exact analytical solutions. This framework allows for a thorough evaluation of each method's capability to accurately capture the behavior of the underlying physical system. The definitions of the L_2 and $L\infty$ norms of the solution are as follows:

$$L_2 = \|U^{exact} - U^n\|_2 = \left[\sum_{i=0}^{N} \left|U_i^{exact} - U_i^n\right|^2\right]^{\frac{1}{2}}. \quad (13)$$

$$L_\infty = \|U^{exact} - U^n\|_\infty = \max_i \left|U_i^{exact} - U_i^n\right|. \quad (14)$$

Example1:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad , \ 0 < x < \pi, \quad 0 < y < \pi. \quad (15)$$

And the boundary conditions are:

$$U(x,0) = sinh(x), \ U(x,\pi) = -sinhx$$
$$U(0,y) = 0, \ U(\pi,y) = \sinh(\pi)\cos(y). \quad (16)$$

The exact solution is given by

$$U(x,y) = \sinh(x)\cos(y). \quad (17)$$

In the NSFD scheme, the denominator functions and the term $\varphi_1$and φ_2 are defined as follows:

$$\varphi_1 = \sinh^2\left(\frac{\Delta x}{2}\right), \ \varphi_2 = \sin^2\left(\frac{\Delta y}{2}\right). \quad (18)$$

Substituting the denominator functions $\varphi_1$and $\varphi_2$in (10) yields:

$$u_{i,j} = \frac{1}{2}\frac{\sinh^2\left(\frac{\Delta x}{2}\right)\left(u_{i,j+1}+u_{i,j-1}\right)+\sin^2\left(\frac{\Delta y}{2}\right)\left(u_{i+1,j}+u_{i-1,j}\right)}{\sinh^2\left(\frac{\Delta x}{2}\right)+\sin^2\left(\frac{\Delta y}{2}\right)}. \quad (19)$$

TABLE 1. Computational results for Example 1.

N	SFD-MG				NSFD-MG				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N. iter	CPU	
16	3.486×10^{-2}	4.469×10^{-3}	2	0.067	4.832×10^{-14}	7.994×10^{-15}	2	0.053	168
32	1.805×10^{-2}	1.212×10^{-3}	4	0.096	3.487×10^{-13}	2.576×10^{-14}	4	0.072	640
64	9.177×10^{-3}	3.132×10^{-4}	5	0.254	1.168×10^{-12}	4.53×10^{-14}	5	0.166	2488
128	4.626×10^{-3}	7.958×10^{-5}	5	0.771	1.182×10^{-11}	2.061×10^{-13}	6	0.549	9806
256	2.322×10^{-3}	2.006×10^{-5}	6	1.736	1.143×10^{-10}	9.970×10^{-13}	6	1.069	38,926
512	1.163×10^{-3}	5.034×10^{-6}	6	10	1.928×10^{-11}	1.634×10^{-13}	7	16.6	16×10^4

Table 1 shows that the NSFD-MG method achieves much lower error norms than SFD-MG, especially on finer grids, reaching near machine precision. Both methods require a small, nearly constant number of iterations, but NSFD-MG provides higher accuracy with slightly more CPU time at large grid sizes.

TABLE 2. Computational results for Example 1.

N	NSFD-MG				NSFD-BicgStab				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N. iter	CPU	
16	4.832×10^{-14}	7.994×10^{-15}	2	0.053	4.53×10^{-14}	9.77×10^{-15}	48	0.061	168
32	3.487×10^{-13}	2.576×10^{-14}	4	0.072	9.789×10^{-13}	6.484×10^{-14}	85	0.08	640
64	1.168×10^{-12}	4.53×10^{-14}	5	0.166	1.959×10^{-12}	8.704×10^{-14}	162	0.13	2488
128	1.182×10^{-11}	2.061×10^{-13}	6	0.549	2.394×10^{-11}	3.739×10^{-13}	323	0.18	9806
256	1.143×10^{-10}	9.970×10^{-13}	6	1.069	1.693×10^{-10}	1.633×10^{-12}	617	1.09	38,926
512	1.928×10^{-11}	1.634×10^{-13}	7	16.6	7.521×10^{-10}	4.28×10^{-12}	1291	17.7	16×10^4

Table 2 compares NSFD-MG with NSFD-BiCGStab. NSFD-MG converges much faster, needing far fewer iterations (e.g., 7 vs. 1291 at N = 512) with better accuracy. This shows the benefit of using the multigrid method with a GMRES smoother.

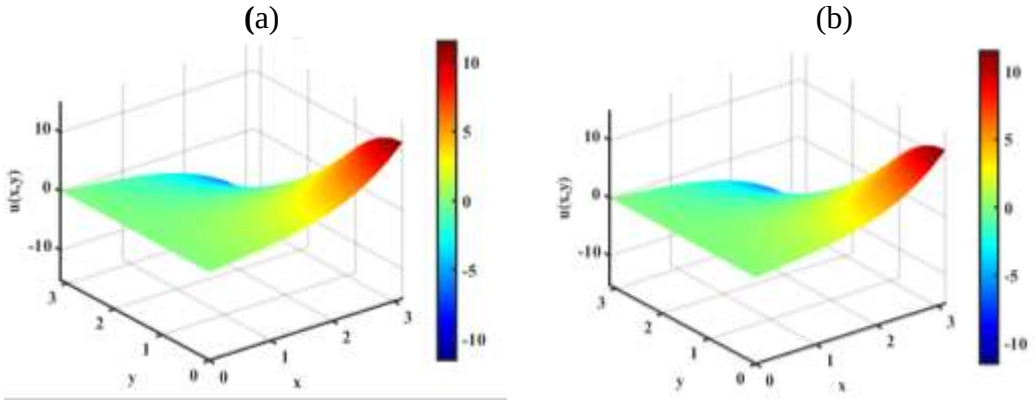

Fig 3. (a) Numerical solutions using MG-GMRES for N=64. (b) Corresponding exact solution. The close visual agreement between both solutions indicates that the proposed method accurately captures the true behavior of the problem.

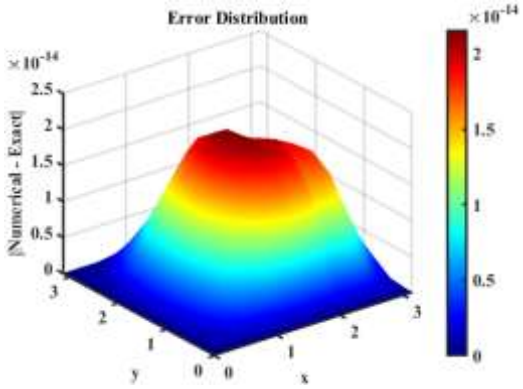

Fig 4. Shows the error surface at $N = 32$ using MG-GMRES, illustrating a smooth and very small numerical error across the domain, indicating high accuracy of the numerical method.

.

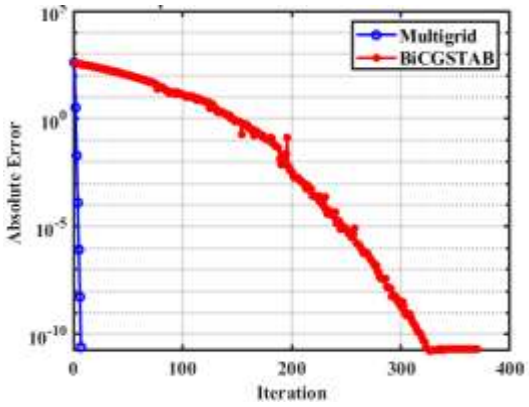

Fig 5. Comparison of absolute error vs. iteration showing faster convergence of the MG method compared to BiCGStab **at $N = 128$.** The MG method reaches a much lower error level in significantly fewer iterations than BiCGStab, clearly demonstrating its faster convergence and higher efficiency in solving the system.

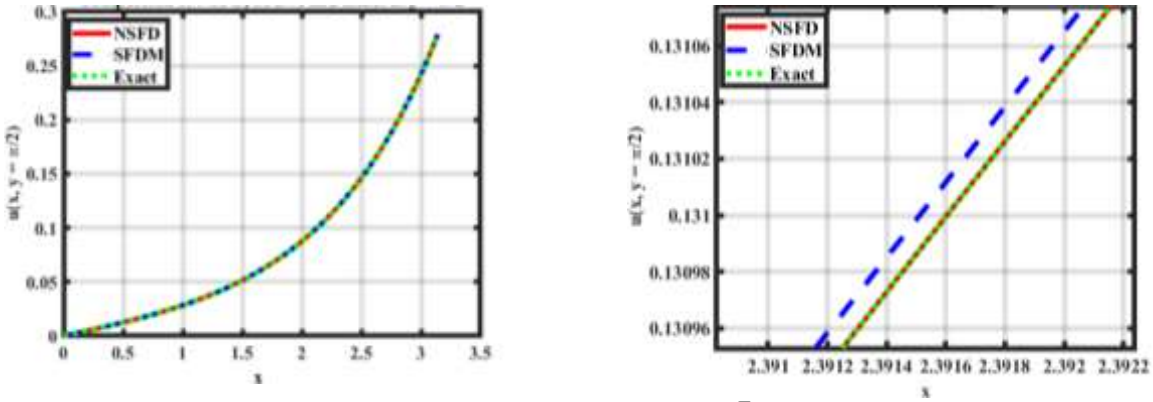

Fig 6. Comparison of NSFD, SFD, and exact solution at $y = \frac{\pi}{2}$; (a) full domain, (b) zoomed-in view showing accuracy differences.

Example2:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad , 0 < x < \pi, \quad 0 < y < \pi. \tag{20}$$

And the boundary conditions are:

$$U(x,0) = 0,\ U(x,\pi) = 0,$$
$$U(0,y) = siny, U(\pi,y) = \cosh(\pi)\sin(y). \tag{21}$$

The exact solution is given by

$$U(x,y) = \cosh(x)\sin(y). \tag{22}$$

In this example we follow the same technique of the above example and using the denominator functions φ_1 and φ_2 in (19).

TABLE 3. Computational results for Example 2.

N	SFD-MG				NSFD-MG				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N. iter	CPU	
16	1.009×10^{-1}	1.184×10^{-2}	3	0.041	1.453×10^{-13}	1.865×10^{-14}	3	0.029	168
32	5.211×10^{-2}	3.164×10^{-3}	5	0.128	1.222×10^{-14}	7.55×10^{-14}	5	0.037	640
64	2.647×10^{-2}	8.17×10^{-4}	6	0.245	5.735×10^{-12}	1.75×10^{-13}	6	0.107	2488
128	1.334×10^{-2}	2.075×10^{-4}	7	0.879	5.736×10^{-11}	8.971×10^{-13}	7	0.32	9806
256	6.697×10^{-3}	5.229×10^{-5}	7	1.526	4.470×10^{-10}	3.513×10^{-12}	7	1.01	38,926
512	3.355×10^{-3}	1.312×10^{-5}	7	5.49	5.097×10^{-10}	2.99×10^{-12}	7	8.14	16×10^4

Table 3 presents the computational results for Example 2 using SFD-MG and NSFD-MG methods. The NSFD-MG method consistently produces much smaller errors than SFD-MG across all grid sizes Both methods require a similar number of iterations, but NSFD-MG achieves significantly higher accuracy with lower CPU time, especially on coarse grids.

TABLE 4. Computational results for Example 2

N	NSFD-MG				NSFD-BicgStab				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N .iter	CPU	
16	1.453×10^{-13}	1.865×10^{-14}	3	0.029	1.64×10^{-13}	3.02×10^{-14}	22	0.037	168
32	1.222×10^{-14}	7.55×10^{-14}	5	0.037	6.88×10^{-12}	4.761×10^{-13}	48	0.046	640
64	5.735×10^{-12}	1.75×10^{-13}	6	0.107	1.10×10^{-10}	3.859×10^{-12}	95	0.038	2488
128	5.736×10^{-11}	8.97×10^{-13}	7	0.32	2.379×10^{-9}	4.137×10^{-11}	189	0.112	9806
256	4.470×10^{-10}	3.51×10^{-12}	7	1.01	1.009×10^{-9}	8.79×10^{-12}	384	1.32	38,926
512	5.097×10^{-10}	2.99×10^{-12}	7	8.14	4.698×10^{-8}	2.059×10^{-10}	789	12.01	16×10^4

Table 4 compares the performance of NSFD-MG and NSFD-BiCGStab for Example 2. NSFD-MG achieves significantly better accuracy with much fewer iterations across all grid sizes. For example, at $N = 512$, NSFD-MG converges in 7 iterations with an L_2 error of 5.097×10^{-10}, while NSFD-BiCGStab requires 789 iterations and produces a larger error. This confirms that the multigrid method with a GMRES smoother offers faster convergence and higher precision compared to BiCGStab.

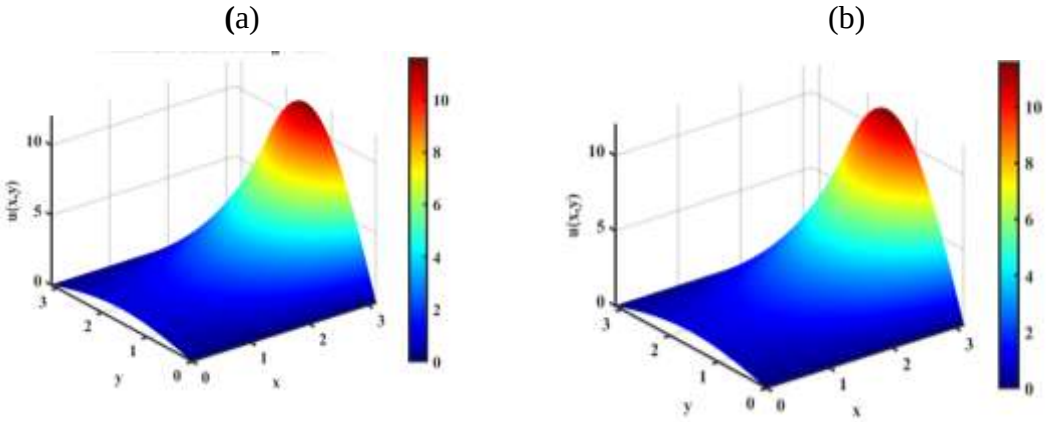

Fig. 7. (a) Approximate solution obtained using MG-GMRES for $N = 64$. (b) The corresponding exact solution for comparison.

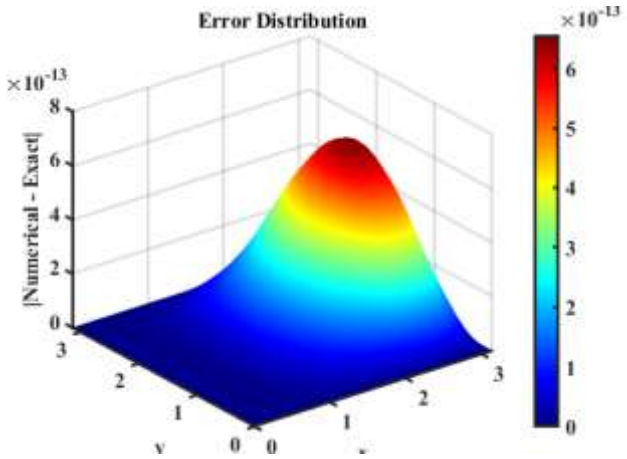

Fig. 8. Error surface for $N = 64$ using MG-GMRES, demonstrating a smooth distribution of very small errors across the domain, highlighting the high accuracy of the method.

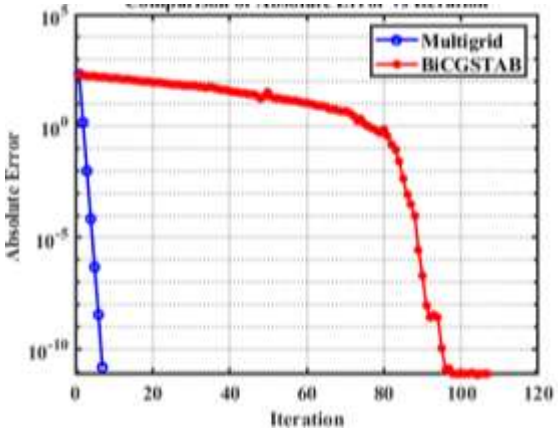

Fig.9. Absolute error versus iteration for $N = 64$, illustrating the faster convergence of the MG method in comparison to BiCGStab.

(a) (b)

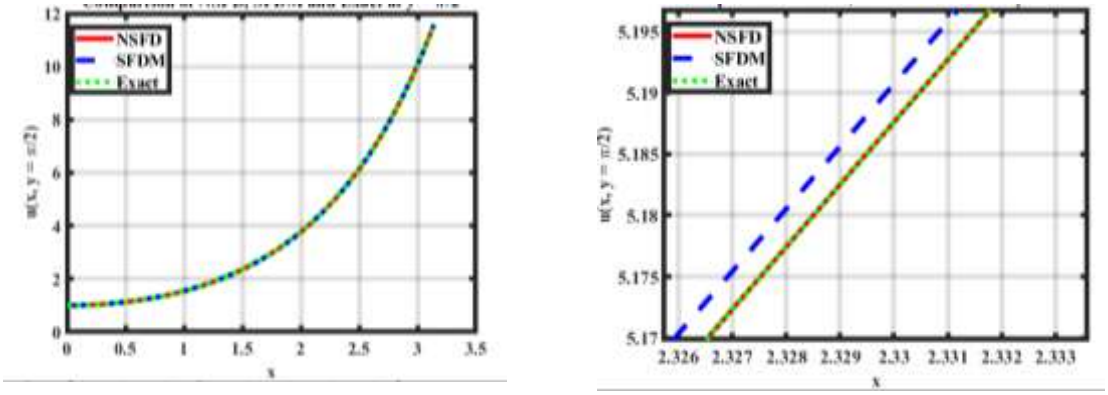

Fig.10. A comparison of the NSFD, SFD, and exact solutions at $y = \frac{\pi}{2}$: (a) entire domain, (b) close-up view highlighting the accuracy differences.

Example 3

Consider the following two-dimensional Poisson equation

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x,y), \tag{23}$$

$$f(x,y) = -2\pi^2 \sin(\pi x)\cos(y\pi) \tag{24}$$

with boundary conditions

$$U(x,0) = \sin(\pi x)\ , U(x,1) = -\sin(\pi y)$$
$$U(0,y) = 0, U(1,y) = 0 \tag{25}$$

and the exact solution is given by

$$U(x,y) = \sin(\pi x)\cos(y\pi) \tag{26}$$

In the NSFD scheme, the denominator functions and the term $\varphi_1$and φ_2 are defined as follows:

$$\varphi_1 = \frac{(\Delta x)^2}{1+\alpha^2(\Delta x)^2},\ \varphi_2 = \frac{(\Delta y)^2}{1+\alpha^2(\Delta y)^2}. \tag{27}$$

TABLE 5. Computational results for Example 3 at $\alpha = 0.91$.

N	SFD-MG				NSFD-MG				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N. iter	CPU	
16	8.275×10^{-3}	9.48×10^{-4}	3	0.03	4.233×10^{-5}	4.85×10^{-6}	3	0.04	168
32	4.263×10^{-3}	2.544×10^{-4}	4	0.036	2.723×10^{-5}	1.625×10^{-6}	4	0.05	640
64	2.164×10^{-3}	6.563×10^{-5}	5	0.09	1.457×10^{-5}	4.417×10^{-7}	5	0.17	2488
128	1.091×10^{-3}	1.666×10^{-5}	6	0.3	7.437×10^{-6}	1.136×10^{-7}	6	0.5	9806
256	5.474×10^{-4}	4.2×10^{-6}	6	0.98	3.75×10^{-6}	2.873×10^{-8}	6	0.99	38,926
512	2.742×10^{-4}	1.054×10^{-6}	6	10	1.878×10^{-6}	7.217×10^{-9}	6	13	16×10^4

In the previous table, the NSFD-MG method consistently yields lower L_2 and L∞ errors compared to SFD-MG across all grid sizes. For instance, at $N = 512$, NSFD-MG achieves an L_2 error of 1.878×10^{-6}, significantly better than SFD-MG's 2.742×10^{-4}. Both methods require the same number of iterations, but NSFD-MG provides much higher accuracy with comparable CPU time. This demonstrates the effectiveness of the NSFD scheme at $\alpha = 0.91$, especially in reducing error as the grid is refined.

TABLE 6. Computational results for Example 3 at $\alpha = 0.91$.

N	NSFD-MG				NSFD-Bicgstab				COND
	L_2	L∞	N. iter	CPU	L_2	L∞	N. iter	CPU	
16	4.233×10^{-5}	4.85×10^{-6}	3	0.04	6.057×10^{-5}	6.939×10^{-6}	9	0.047	168
32	2.723×10^{-5}	1.625×10^{-6}	4	0.05	3.666×10^{-5}	2.188×10^{-6}	22	0.07	640
64	1.457×10^{-5}	4.417×10^{-7}	5	0.17	1.936×10^{-5}	5.87×10^{-7}	47	0.21	2488
128	7.437×10^{-6}	1.136×10^{-7}	6	0.5	9.85×10^{-6}	1.505×10^{-7}	93	0.12	9806
256	3.75×10^{-6}	2.873×10^{-8}	6	0.99	4.95×10^{-6}	3.80×10^{-8}	190	0.4	38,926
512	1.878×10^{-6}	7.217×10^{-9}	6	13	2.485×10^{-6}	9.551×10^{-9}	398	15	16×10^4

Table 6 compares NSFD-MG and NSFD-BiCGStab for Example 3 at $\alpha = 0.91$. The NSFD-MG method achieves lower error values across all grid sizes with significantly fewer iterations. For instance, at $N = 512$, NSFD-MG reaches the desired accuracy in only 6 iterations, while BiCGStab requires 398. CPU times are also lower or comparable, especially for larger grids. These results confirm that multigrid with a GMRES smoother remains more efficient and accurate than BiCGStab, even when the NSFD discretization is used.

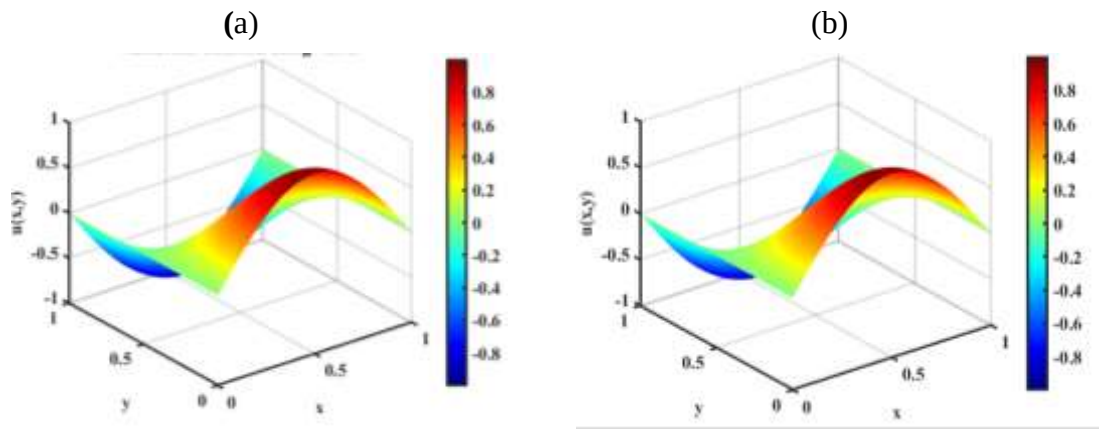

Fig.11. (a) Approximate solution obtained using MG-GMRES for $N = 64$. (b) The corresponding exact solution for comparison.

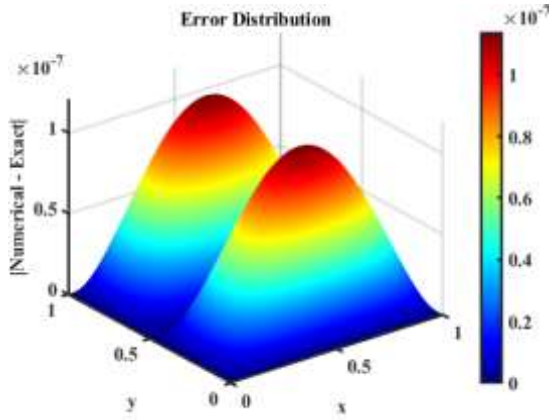

Fig.12. Error surface for $N = 128$ using MG-GMRES, demonstrating a smooth distribution of very small errors across the domain, highlighting the high accuracy of the method.

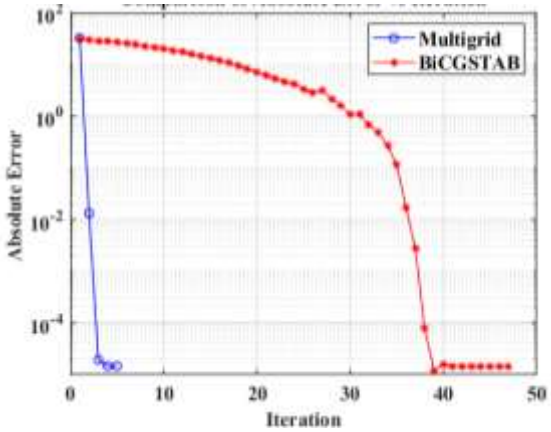

Fig.13. Absolute error versus iteration for $N = 64$, illustrating the faster convergence of the MG method in comparison to BiCGStab**.**

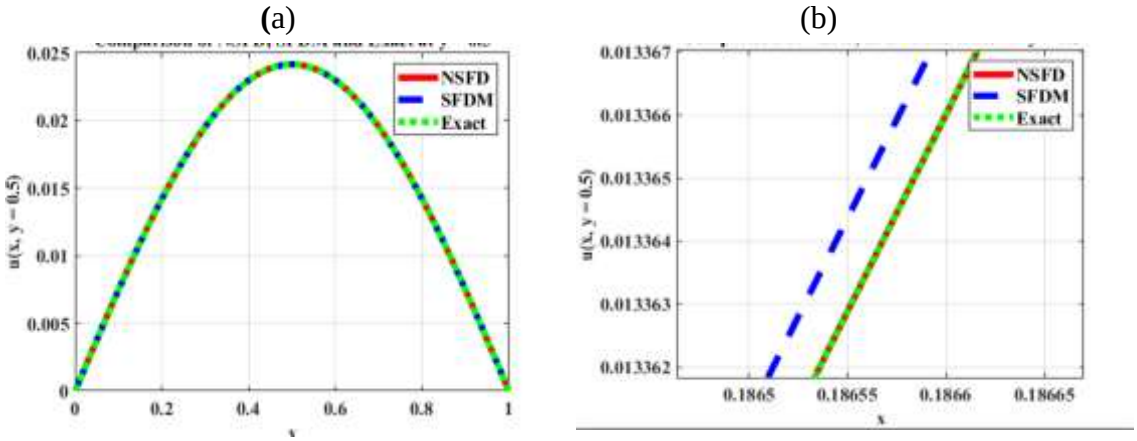

Fig.14. A comparison of the NSFD, SFD, and exact solutions at $y = \frac{1}{2}$: (a) entire domain, (b) close-up view highlighting the accuracy differences.

7. Conclusion

The comparison between NSFD and SFD methods shows that NSFD method offers superior accuracy when solving the two-dimensional Poisson equation with Dirichlet boundary conditions. This is evidenced by consistently lower error values in both the L_2 and $L\infty$ norms, indicating that NSFD provides a more precise representation of the underlying solution. Thus, NSFD proves to be a more reliable discretization method for this type of problem.

After discretizing the two-dimensional Poisson equation using NSFD method, the resulting linear system was solved using two iterative solvers: MG with GMRES as a smoothing technique, and BiCGStab. Numerical experiments show that MG outperforms BiCGStab in terms of convergence rates, demonstrating better efficiency in both CPU time and the number of iterations needed for convergence. Additionally, MG consistently achieves a lower error across several tests, showcasing its superior computational performance. These findings emphasize the effectiveness of combining NSFDM with MG, offering a robust and precise framework for solving partial differential equations efficiently. The NSFD-MG combination improves both solution accuracy and computational efficiency, making it a preferred choice for large-scale simulations.

8. REFERENCES

[1] M. A. Zaman, "Numerical solution of the Poisson equation using finite difference matrix operators," Electronics, vol. 11, no. 15, p. 2365, 2022.

[2] N. H. Sweilam, W. Mahmoud, and E. K. Rawy, "Numerical study of some systems of linear algebraic equations with noise related to boundary-value problems for Laplace's equation in rectangular domains," Studies in Nonlinear Sciences, vol. 2, no. 2, pp. 60–69, 2011.

[3] X.-J. Yang, N. H. Sweilam, and M. Bayram, "Special solutions for the Laplace and diffusion equations associated with the algebraic number field," Thermal Science, vol. 27, no. 1, pp. 477–481, 2023.

[4] E. Bierstone, P. D. Milman, and A. Parusiński, "On the Laplace equation on bounded subanalytic manifolds," Calculus of Variations and Partial Differential Equations, 2024.

[5] J. H. Tang, Partial differential equations and boundary value problems (in Chinese). Tsang Hai Publishing Company, 2017.

[6] J. Kevorkian, Partial differential equations: Analytical solution techniques, 2nd ed. Springer, 2010.

[7] L. C. Evans, Partial differential equations, 2nd ed. American Mathematical Society, 2010.

[8] R. Shiromani, V. Shanthi, and J. Vigo-Aguiar, "A finite difference method for singularly perturbed 2D elliptic convection–diffusion PDEs on Shishkin-type meshes with non-smooth convection and source terms," Mathematical Methods in the Applied Sciences, vol. 46, no. 5, pp. 5915–5936, 2023.

[9] J. Song, D. Sheen, X. Feng, and Y. He, "A difference finite element method based on nonconforming finite element methods for 3D elliptic problems," Advances in Computational Mathematics, vol. 51, Article 7, 2025.

[10] M. A. Mohye, J. B. Munyakazi, and T. G. Dinka, "A nonstandard fitted operator finite difference method for two-parameter singularly perturbed time-delay parabolic problems," Frontiers in Applied Mathematics and Statistics, vol. 9, 2023.

[11] T. A. Dao, K. Mattsson, and M. Nazarov, "Energy stable and accurate coupling of finite element methods and finite difference methods," Journal of Computational Physics, vol. 420, p. 109724, 2021.

[12] A. Radhakrishnan, M. Xu, S. Shahane, and S. P. Vanka, "A non-nested multilevel method for meshless solution of the Poisson equation in heat transfer and fluid flow," arXiv, 2021.

[13] A. Sadighi and D. D. Ganji, "Exact solutions of Laplace equation by homotopy perturbation and Adomian decomposition methods," Physics Letters A, vol. 367, no. 1–2, pp. 83–87, 2007.

[14] C.-K. Chen and Y.-H. Chang, "An analytic solution for 2D heat conduction problems with general Dirichlet boundary conditions," Axioms, vol. 12, no. 5, p. 416, 2023.

[15] G.-D. Tcaciuc and A. Popescu, "Finite difference method for solving the two-dimensional Laplace equation in curvilinear coordinates," Bulletin of the Polytechnic Institute of Iași, vol. 69, no. 3, pp. 43–56, 2023.

[16] R. E. Mickens, Advances in the applications of nonstandard finite difference schemes. World Scientific, 2005.

[17] R. E. Mickens, "Calculation of denominator functions for nonstandard finite difference schemes for differential equations satisfying a positivity condition," Numerical Methods for Partial Differential Equations, vol. 23, no. 3, pp. 672–691, 2006.

[18] R. E. Mickens, "Determination of denominator functions for a NSFD scheme for the Fisher PDE with linear advection," Mathematics and Computers in Simulation, vol. 74, no. 3, pp. 190–195, 2007.

[19] K. Moaddy, S. Momani, and I. Hashim, "The non-standard finite difference scheme for linear fractional PDEs in fluid mechanics," Computers and Mathematics with Applications, vol. 61, no. 4, pp. 1209–1216, 2011.

[20] N. H. Sweilam, S. M. Al-Mekhlafi, and D. Baleanu, "Nonstandard finite difference method for solving complex-order fractional Burgers' equations," Journal of Advanced Research, vol. 25, pp. 19–29, 2020.

[21] O. O. Kehinde, J. B. Munyakazi, and A. R. Appadu, "A NSFD discretization of two-dimensional singularly perturbed semilinear convection-diffusion problems," Frontiers in Applied Mathematics and Statistics, vol. 8, p. 861276, 2022.

[22] M. Aydın, "Finite difference methods for the numerical solution of the Laplace equation," Journal of Computational Physics, vol. 452, pp. 1–20, 2022.

[23] D. M. Young, Iterative solution of large linear systems. Elsevier, 2014.

[24] S. Thomas, A. Carr, P. Mullowney, R. Li, and K. Świrydowicz, "Neumann series in GMRES and algebraic multigrid smoothers," arXiv preprint, arXiv:2112.14681, 2021.

[25] H. Bouyghf, "An enhancement of the accuracy of the BiCGStab method for the standard, global, and block cases," Mathematical Methods in the Applied Sciences, vol. 46, no. 16, pp. 11723–11739, 2023.

[26] R. J. LeVeque, Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems. Society for Industrial and Applied Mathematics, 2007.

[27] R. E. Mickens, Nonstandard finite difference models of differential equations. World Scientific, 1994.

[28] P. M. Manning and G. F. Margrave, "Introduction to non-standard finite-difference modelling," CREWES Research Report, vol. 18, pp. 1–10, 2006.

[29] P. Sonneveld, "CGS: A fast Lanczos-type solver for nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 10, no. 1, pp. 36–52, 1989.

[30] H. A. Van der Vorst, *Iterative Krylov Methods for Large Linear Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2009.