Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11, 710-725
2025
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v9i11.10962

A bicubic B-spline approximation and application of C-BSEG methods for the solution of 2D elliptic partial differential equations

Claire NC Motiun[1*], Jumat Sulaiman[2], Aini Janteng[3], Asep K. Supriatna[4]
[1,2,3]Faculty of Science and Technology, Universiti Malaysia Sabah, Malaysia; claire_motiun_ds23@iluv.ums.edu.my (C.N.M.) jumat@ums.edu.my (J.S.) aini_jg@ums.edu.my (A.J.).
[4]Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang 45363, West Java, Indonesia; ak_supriatna@unpad.ac.id (A.K.S.).

Abstract: This paper presents a two-level implicit bicubic B-spline discretization scheme with the collocation approach, particularly known as a bicubic B-spline collocation approach for the solution of two-dimensional elliptic partial differential equations. Then, a system of bicubic B-spline collocation approximation equations generated from the discretization process of the proposed scheme with the collocation approach is normally large-scale, with a sparse matrix. To solve this linear system, a new Bicubic B-spline Explicit Group (C-BSEG) iteration approach has been shown to enhance its convergence rate in solving any linear system. In addition, the capability of the C-BSEG iteration family, such as 2 Point-C-BSEG and 4 Point-C-BSEG methods, has been investigated in solving two-dimensional elliptic partial differential equations. Moreover, the formulation and implementation of both block iterative methods are also presented and used to solve the linear system iteratively. Numerical experiments demonstrate that the 4 Point-C-BSEG iteration combined with the bicubic B-spline collocation approach achieves superior performance compared with existing point and block iterative schemes. Hence, the proposed bicubic B-spline collocation framework provides a reliable and efficient numerical tool with a wide range of applications in fields such as physics, engineering, and applied mathematics.
Keywords: *Bicubic B-spline block iteration, B-spline collocation, Collocation approach, Two-dimensional Poisson equations.*

1. Introduction

Numerical methods for solving elliptic partial differential equations (PDEs) have been widely studied by researchers to obtain approximate solutions, which play a fundamental role in addressing various scientific and engineering challenges. For instance, one example of PDEs is the Poisson equation, which is referred to as the generalization of the well-known Laplace's equation. The aforementioned differential equation is commonly employed in theoretical physics and is one example of elliptic PDEs. Based on these equations, there are several mathematical models that can be used to govern numerous scientific and engineering fields, including astronomy, fluid mechanics, electromagnetics, heat transfer, electrostatics, and many more.

Aligned with the concept of numerical methods, numerical techniques have been developed to solve two-dimensional (2D) elliptic PDEs. In the late 1960s, the spline interpolation approach was first used to solve differential equations by Bickley [1]. In his work, cubic splines are utilized experimentally to approximate the solution of a basic two-point boundary value problem for a linear ordinary differential equation. Fyfe [2] examined and described the cubic spline method suggested by Bickley and the error predictions of Curtis and Powell [3]. Fyfe [2] concluded that, because the spline can obtain approximate solutions at any point in an interval, the spline method is better than the usual finite difference method (FDM). Due to the effectiveness of this method, numerous authors have been

History: Received: 12 September 2025; Revised: 17 October 2025; Accepted: 22 October 2025; Published: 11 November 2025
* Correspondence: claire_motiun_ds23@iluv.ums.edu.my

interested in solving differential equations using spline approaches. To mention a few, Dağ et al. [4] solved the one-dimensional (1D) Burgers' equation using the cubic B-spline collocation method (CuBsCM) over finite elements in 2005, where Burgers' equation was introduced by Akour et al. [5]. The accuracy of the proposed method was demonstrated through three test problems. According to these problems, CuBsCM is capable of solving Burgers' equation accurately, as evidenced by the comparison of the calculations with the analytical solution. In the study of Demir and Bildik [6], the numerical solution of the heat problem using cubic B-spline (CuBs) was discussed. In this study, the method used to solve the 1D heat problem and the solution were compared with the exact solution. Fourier stability method is considered as an analyzer for the stability of this method, and the result is stable for $\theta = \left[\frac{1}{2}, 1\right]$. In 2021, Ware and Ashine [7] solved a boundary value problem of an ordinary differential equation by using CuBs and FDM. The resulting system of equations has been solved by a tri-diagonal solver. There are two examples examined, and the results are compared with FDM. The results show that the CuBs method is more efficient and feasible. Beyond that, recent advancements have pushed spline-based methods even further. Du and Sun [8] developed a bicubic B-spline finite element method for solving fourth-order semilinear parabolic optimal control problems in 2D. Their numerical results, which are based on two benchmark problems, demonstrate how effective and practical the suggested strategy is in comparison to classical elements. Likewise, Salama et al. [9] proposed hybrid group iterative methods for solving 2D time-fractional cable equations. From the numerical experiments in this study, these 4-point schemes were proven to be unconditionally stable and outperformed traditional iterative solvers in terms of speed and memory usage. In 2022, Salama et al. [10] further refined their approach by introducing a modified hybrid explicit group (MHEG) method, which improved convergence for 2D diffusion problems. In their study, two benchmark problems were examined to evaluate the proposed method. The results showed that the MHEG iterative method achieved significantly faster simulations without substantially compromising accuracy compared to the hybrid standard point (HSP) iterative method [11].

Besides numerical techniques, the discovery of the iterative method for PDEs modeled issues began in the early 20th century [12]. Through the discretization process of numerical techniques for 2D elliptic PDEs, it can be observed that the coefficient matrix of the resulting linear system was large-scale and sparse. Based on the previous studies by Young [13], Hackbusch [14], and Saad [15], the findings of their studies stated that the iterative methods are the best linear solvers for solving any linear system. Recently, the explicit group (EG) iterative methods, which were introduced by Evans and Abdullah [16] that have been used successfully to solve numerical problems involving parabolic and hyperbolic PDEs [17]. A small group of 2, 4, 9, 16, and 25 points was generated in the iterative processes for solving Laplace's equation using the EG iterative approach [18]. The numerical findings demonstrate that the EG method takes less storage and is easier to implement than the block (line) iterative approaches. This approach, however, was developed only utilizing the conventional standard finite difference discretization, which limits the solutions to certain locations within the solution domain. However, the study by Mohanty [19] already introduced and applied a two-level implicit cubic spline approach together with the cubic spline alternating group explicit (C-SPLAGE) iterative method for solving 1D nonlinear parabolic PDEs. He concluded that the proposed iterative method is superior to the successive over-relaxation (SOR) method.

From the development of the spline function and the use of EG and C-SPLAGE iterative methods, as mentioned in previous paragraphs, and dealing with numerical solutions of 2D elliptic PDEs, the application of the concept of the bicubic B-spline collocation approach and a family of block iteration methods has attracted attention for obtaining numerical solutions. Based on Mohanty's [19] findings, the author focused on constructing the spline approximation equation and the AGE method for solving 1D problems. There have been limited studies concerned with a bicubic collocation approach being applied to solve 2D elliptic PDEs via the block iteration family. Therefore, this research intends to construct the derivation of the bicubic B-spline collocation approximation equation, in which this approximate equation leads us to construct a system of bicubic B-spline collocation approximation equations. Therefore, the objectives of this research are to investigate the applicability of the C-BSEG iteration family, such as 2 Point-C-BSEG and 4 Point-C-BSEG methods, based on the established bicubic B-spline collocation approximation equation for solving 2D elliptic PDEs.

2. Formulation of Bicubic B-spline Collocation Approximation Equations

Before starting to perform the bicubic B-spline collocation discretization process and implementing the proposed iteration family, let us consider the general form of the aforementioned PDEs as Ali et al. [20].

$$\nabla^2 u = -\frac{\rho}{\varepsilon}, \tag{1}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplace operator, u is the potential difference, ρ is the volume charge density, and ε represents the permittivity of the medium, respectively. Problem (1) can be extended specifically to the Poisson equation over the region, $\Omega = [a,b] \times [a,b]$ as follows

$$U_{xx} + U_{yy} = f(x,y), \qquad x, y \in [a,b], \tag{2}$$

with Dirichlet boundary conditions,

$$\begin{aligned} &U(a,y) = f_0(x), \quad U(b,y) = f_1(y), \quad a \le y \le b, \\ &U(x,a) = f_2(x), \quad U(x,b) = f_3(x), \quad a \le x \le b. \end{aligned}$$

Now, let us start the process of discretization over the proposed Problem (2) in constructing a bicubic B-spline collocation approximation equation for generating a system of linear equations. To do this, the solution domain $\Omega = [a,b] \times [a,b]$ needs to be partitioned for x and y directions, and their sub-interval distance for both directions is given as $h = \frac{b-a}{m}$. All node points of Problem (1) are denoted as (x_i, y_j) with $x_i = a + ih$ and $y_j = a + jh$, $0 \le i, j \le m$ as indicated in Figure 1. Prior to having the 2D mesh network, let B-spline basis functions of order d and degree $d-1$ denoted as $B_{i,d}(x)$ [21] and defined as

$$B_{0,d}(x) = \begin{cases} 1, & x \in [x_i, x_{i+1}], \\ 0, & otherwise. \end{cases} \tag{3}$$

$$B_{i,d}(x) = \left(\frac{x - x_i}{x_{i+d} - x_i} \right) B_{i,d-1}(x) + \left(\frac{x_{i+d} - x}{x_{i+d} - x_{i+1}} \right) B_{i+1,d-1}(x). \tag{4}$$

The cubic B-spline is defined as a piecewise function [22],

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

$$B_{i,4}(x)=\frac{1}{6h^3}\begin{cases}(x-x_i)^3, & x\in[x_i,x_{i+1}]\\ h^3+3h^2(x-x_{i+1})+3h(x-x_{i+1})^2-3(x-x_i)^3, & x\in[x_{i+1},x_{i+2}]\\ h^3+3h^2(x_{i+3}-x)+3h(x_{i+3}-x)^2-3(x_{i+3}-x)^3, & x\in[x_{i+2},x_{i+3}]\\ (x_{i+4}-x)^3, & x\in[x_{i+3},x_{i+4}]\end{cases} \quad (5)$$

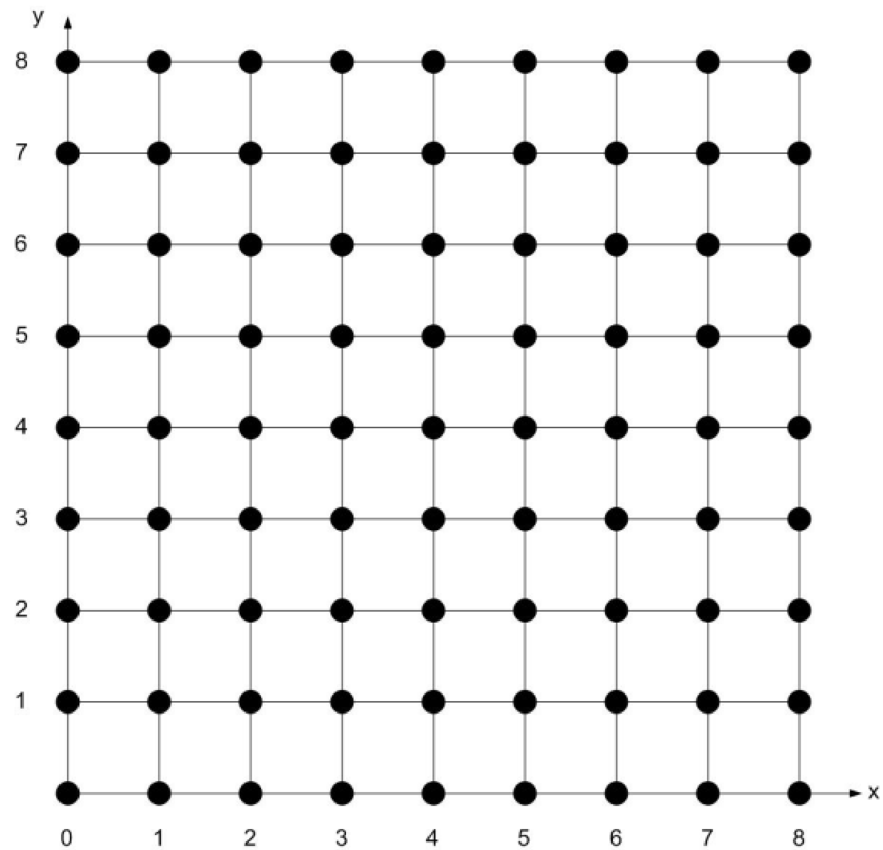

Figure 1.
Distribution of uniform node points over the solution domain for m = 8.

Clearly, it can be stated that the cubic B-spline basis, $B_{i,4}(x)$ Function (5), can be known as a piecewise polynomial of degree 3. The bicubic B-spline approximation function, derived from the basis function in Equation (5), defines a surface known as the bicubic B-spline surface because it is constructed using two cubic B-spline bases and could be defined as,

$$S_B(x,y)=\sum_{i=-3}^{m-1}\sum_{j=-3}^{m-1}C_{i,j}B_{i,4}(x)B_{j,4}(y), \qquad x\in[x_0,x_m], \quad y\in[y_0,y_m]. \quad (6)$$

where $C_{i,j}, -3\le i,j\le m-1$ are unknown coefficients are to be determined. By applying the simplification of the bicubic B-spline approximation function, $S_B(x,y)$ at any arbitrary node point,

$$S_B\left(x_i, y_j\right) = \frac{1}{36}\begin{pmatrix} C_{i-3,j-1} + 4C_{i-2,j-1} + C_{i-1,j-1} \\ +4C_{i-3,j-2} + 16C_{i-2,j-2} + 4C_{i-1,j-2} \\ +C_{i-3,j-3} + 4C_{i-2,j-3} + C_{i-1,j-3} \end{pmatrix} \tag{7}$$

Then, the first and second derivatives of the bicubic B-spline approximation function, $S_B(x,y)$ with respect to x could be simplified at any point as

$$\frac{\partial}{\partial x}S_B(x_i, y_j) = \frac{1}{12h}\begin{pmatrix} -C_{i-3,j-1} + C_{i-1,j-1} \\ -4C_{i-3,j-2} + 4C_{i-1,j-2} \\ -C_{i-3,j-3} + C_{i-1,j-3} \end{pmatrix}, \tag{8}$$

$$\frac{\partial^2}{\partial x^2}S_B(x_i, y_j) = \frac{1}{6h^2}\begin{pmatrix} C_{i-3,j-1} - 2C_{i-2,j-1} + C_{i-1,j-1} \\ +4C_{i-3,j-2} - 8C_{i-2,j-2} + 4C_{i-1,j-2} \\ +C_{i-3,j-3} - 2C_{i-2,j-3} + C_{i-1,j-3} \end{pmatrix}. \tag{9}$$

Similarly to derive Equations (8) and (9), the first and second derivatives of the bicubic B-spline approximation function, $S_B(x,y)$ with respect to y could be simplified at any point as

$$\frac{\partial}{\partial y}S_B(x_i, y_j) = \frac{1}{12h}\begin{pmatrix} C_{i-3,j-1} + 4C_{i-2,j-1} + C_{i-1,j-1} \\ -C_{i-3,j-3} - 4C_{i-2,j-3} - C_{i-1,j-3} \end{pmatrix}, \tag{10}$$

$$\frac{\partial^2}{\partial y^2}S_B(x_i, y_j) = \frac{1}{6h^2}\begin{pmatrix} C_{i-3,j-1} + 4C_{i-2,j-1} + C_{i-1,j-1} \\ -2C_{i-3,j-2} - 8C_{i-2,j-2} - 2C_{i-1,j-2} \\ +C_{i-3,j-3} + 4C_{i-2,j-3} + C_{i-1,j-3} \end{pmatrix}. \tag{11}$$

Before getting the approximate solution $U(x,y)$ over Problem (2), the values of $C_{i,j}, -3 \le i, j \le m-1$ on the boundary conditions of the domain solution in Problem (2) can be determined by using the approximate solution (6) in the boundary conditions (2) [23]. Based on the bottom boundary condition, $U(x,a) = f_2(x)$ gives

$$\begin{bmatrix} 4 & 2 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 2 & 4 \end{bmatrix}\begin{bmatrix} C_{-3,-3} \\ C_{-2,-3} \\ C_{-1,-3} \\ \vdots \\ C_{m-2,-3} \\ C_{m-1,-3} \end{bmatrix} = \begin{bmatrix} 6f_2\left(x_0\right) + 2hf_2'\left(x_0\right) \\ 6f_2\left(x_1\right) \\ 6f_2\left(x_2\right) \\ \vdots \\ 6f_2\left(x_{m-1}\right) \\ 6f_2\left(x_m\right) - 2hf_2'\left(x_m\right) \end{bmatrix} \tag{12}$$

The top boundary condition, $U(x,b) = f_3(x)$ yields

$$\begin{bmatrix} 4 & 2 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 2 & 4 \end{bmatrix} \begin{bmatrix} C_{-3,m-1} \\ C_{-2,m-1} \\ C_{-1,m-1} \\ \vdots \\ C_{m-2,m-1} \\ C_{m-1,m-1} \end{bmatrix} = \begin{bmatrix} 6f_3(x_0)+2hf_3'(x_0) \\ 6f_3(x_1) \\ 6f_3(x_2) \\ \vdots \\ 6f_3(x_{m-1}) \\ 6f_3(x_m)-2hf_3'(x_m) \end{bmatrix} \quad (13)$$

The left boundary condition, $U(a,y)=f_0(y)$ leads to

$$\begin{bmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} C_{-3,-2} \\ C_{-3,-1} \\ C_{-3,0} \\ \vdots \\ C_{-3,m-1} \\ C_{-3,m-2} \end{bmatrix} = \begin{bmatrix} 6f_1(y_0)-C_{-3,-3} \\ 6f_0(y_1) \\ 6f_0(y_2) \\ \vdots \\ 6f_0(y_{m-1}) \\ 6f_0(y_m)-C_{-3,m-1} \end{bmatrix} \quad (14)$$

The right boundary condition, $U(b,y)=f_1(y)$ leads to

$$\begin{bmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} C_{m-1,-2} \\ C_{m-1,-1} \\ C_{m-1,0} \\ \vdots \\ C_{m-1,m-1} \\ C_{m-1,m-2} \end{bmatrix} = \begin{bmatrix} 6f_1(y_0)-C_{m-1,-3} \\ 6f_1(y_1) \\ 6f_1(y_2) \\ \vdots \\ 6f_1(y_{m-1}) \\ 6f_1(y_m)-C_{m-1,m-1} \end{bmatrix} \quad (15)$$

The tridiagonal system of Equations (12) - (15) was solved by forward and backward substitution. It means that the LU decomposition approach has been performed over the tridiagonal linear systems (12) – (15) to obtain the value of $C_{i,j}$ on the boundary conditions. Next, we attempt to calculate the approximate values of $C_{i,j}$ at all interior points over the solution domain Ω. To do that, the discretization process over Problem (2) needs to be conducted. Firstly, let $U_{i,j}$ and $f_{i,j}$ represent $U(x_i,y_j)$ and $f(x_i,y_j)$ respectively. Then Problem (2) becomes

$$\frac{\partial^2}{\partial x^2}S_B(x_i,y_j)+\frac{\partial^2}{\partial y^2}S_B(x_i,y_j)=f(x_i,y_j) \quad (16)$$

Then, substitute Equations (9) and (11) into Equation (16). Problem (2) can be approximated and simplified as a bicubic B-spline collocation approximation equation, which is given by

$$C_{i-3,j-3}+C_{i-2,j-3}+C_{i-1,j-3}+C_{i-3,j-2}-8C_{i-2,j-2}+C_{i-1,j-2}+C_{i-3,j-1}+C_{i-2,j-1}+C_{i-1,j-1} \\ =3h^2 f_{i,j} \quad (17)$$

ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

for $i, j = 1, 2, 3, \cdots, m-1$. By imposing the bicubic B-spline collocation approximation Equation (17) over all interior node points, $\left(x_i, y_j\right), i, j = 1, 2, 3, \cdots, m-1$, a large-scale and sparse linear system generated from Equation (17) can be rewritten in a matrix form as

$$AC = f. \tag{18}$$

3. Formulation of Bicubic B-spline Explicit Group Iteration Family

As stated in the linear system (18), the main characteristics of the coefficient matrix, A (18), are that it is large-scale and sparse. In this paper, 2 Point-C-BSEG and 4 Point-C-BSEG iterative methods will be applied to solve linear systems (18) generated from the approximation Equation (16) through the discretization of Problem (2). As mentioned in the second section, this paper aims to demonstrate the effectiveness of the family of iterative methods, specifically the 2-Point and 4-Point C-BSEG methods, for solving Problem (2). These methods are based on bicubic B-spline interpolation in conjunction with the collocation methodology. To establish the formulation of the proposed method, first, let us decompose the coefficient matrix, A, in Equation (18) as

$$A = D + L + V \tag{19}$$

where D is a diagonal matrix, L is a lower triangular and V is the upper triangular parts of A respectively. From the linear system (18), the general scheme for full sweep Gauss-Seidel iterative methods can be written as,

$$C^{(k+1)} = \omega(D+L)^{-1}(f - VC^{(k)}). \tag{20}$$

3.1. Formulation of 2-Point Explicit Group Iterative Methods

The formulation and implementation of the 2-Point EG method to solve the 2D Poisson equation will be presented in this section. By referring to algebraic Equation (17), consider any point of the two points, $C_{i,j}$ and $C_{i+1,j}$ that are used simultaneously to compute the value of $C^{(k+1)}$. Thus, at a point $C_{i,j}$, the solution is approximated by

$$C_{i-1,j-1}+C_{i,j-1}+C_{i+1,j-1}+C_{i-1,j}-8C_{i,j}+C_{i+1,j}+C_{i-1,j+1}+C_{i,j+1}+C_{i+1,j+1}=f_{i,j}, \tag{21}$$

where at point $C_{i,j+1}$, the solution is given by

$$C_{i-1,j-1}+C_{i,j-1}+C_{i+1,j-1}+C_{i-1,j}-8C_{i,j}+C_{i+1,j}+C_{i-1,j+1}+C_{i,j+1}+C_{i+1,j+1}=f_{i,j+1}. \tag{22}$$

Now, the Equations (21) and (22) can be written simultaneously in the matrix form as follows,

$$\begin{bmatrix} 8 & -1 \\ -1 & 8 \end{bmatrix} \begin{bmatrix} C_{i,j} \\ C_{i,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{23}$$

where,

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

$$S_1 = C_{i-1,j+1} + C_{i+1,j+1} + C_{i-1,j-1} + C_{i,j-1} + C_{i+1,j-1} + C_{i-1,j} + C_{i+1,j} - 3h^2 f_{i,j},$$
$$S_2 = C_{i-1,j} + C_{i+1,j} + C_{i-1,j+1} + C_{i+1,j+1} + C_{i-1,j+2} + C_{i,j+2} + C_{i+1,j+2} - 3h^2 f_{i,j+1}.$$

The above, Equation (23), can be inverted to result in a two-point explicit form,

$$\begin{bmatrix} C_{i,j} \\ C_{i,j+1} \end{bmatrix} = \frac{1}{63}\begin{bmatrix} -8 & -1 \\ -1 & -8 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{24}$$

whose individual explicit equations are given by,

$$C_{i,j} = \frac{1}{63}[8S_1 + S_2],$$
$$C_{i,j+1} = \frac{1}{63}[S_1 + 8S_2]. \tag{25}$$

For the case of an ungrouped (single) point, the following iteration formula will be applied to compute the point, $C_{i,j}$ where $i = m-1, j = n-1,$

$$C_{m-1,n-1} = \frac{1}{8}\begin{bmatrix} C_{i-1,j+1} + C_{i,j+1} + C_{i+1,j+1} + C_{i-1,j-1} + C_{i,j-1} \\ C_{i+1,j-1} + C_{i-1,j} + C_{i+1,j} - 3h^2 f_{i,j} \end{bmatrix}. \tag{26}$$

By considering Equations (25) and (26), the algorithm of the 2EG method for both cases—complete grouped (Case 1) and incomplete grouped (with one point ungrouped) (Case 2)—is illustrated in Algorithm 1, respectively.

Algorithm 1 : 2EG iteration

I. Initialize $C^{(0)} = 0$ and $\varepsilon = 1.0\times10^{-10}$.

II. For $i = m-1, j = n-1,$ calculate Equation (26).

For $i = m-1, j = 1,3,5,\ldots,m-3$ and $i = 1,3,5,\ldots,m-3, j = m-1,$ calculate Equation (25).

III. If $|C^{(k+1)} - C^{(k)}| \le \varepsilon$ is satisfied, then proceed to Step IV. Otherwise, go back to Step II.

IV. Calculate and display approximate values of $S_B(x_i, y_j)$.

3.2. Formulation of 4-Point Explicit Group Iterative Methods

From Figure 2, let us consider that the solution at any group of four points on the solution domain can be obtained using Equation (18). This leads to a (4 × 4) linear system as follows,

$$\begin{bmatrix} 8 & -1 & -1 & -1 \\ -1 & 8 & -1 & -1 \\ -1 & -1 & 8 & -1 \\ -1 & -1 & -1 & 8 \end{bmatrix}\begin{bmatrix} C_{i,j} \\ C_{i+1,j} \\ C_{i,j+1} \\ C_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{27}$$

where,

$$
\begin{aligned}
S_1 &= C_{i-1,j-1} + C_{i,j-1} + C_{i+1,j-1} + C_{i-1,j} + C_{i-1,j+1} - 3h^2 f_{i,j} \\
S_2 &= C_{i,j-1} + C_{i+1,j-1} + C_{i+2,j-1} + C_{i+2,j} + C_{i+2,j+1} - 3h^2 f_{i+1,j} \\
S_3 &= C_{i-1,j} + C_{i-1,j+1} + C_{i-1,j+2} + C_{i,j+2} + C_{i+1,j+2} - 3h^2 f_{i,j+1} \\
S_4 &= C_{i+2,j} + C_{i+2,j+1} + C_{i,j+2} + C_{i+1,j+2} + C_{i+2,j+2} - 3h^2 f_{i+1,j+1}
\end{aligned} \tag{28}
$$

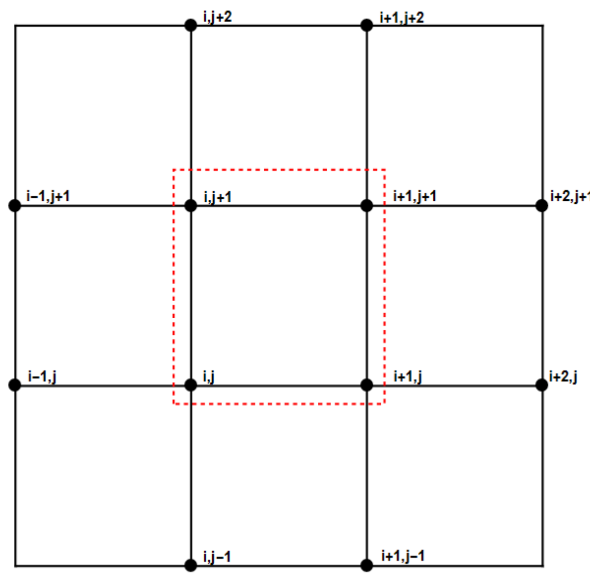

Figure 2.
Computational Molecule Equation (27).

The above Equation (27) can be inverted to result in a 4-Point equation,

$$
\begin{bmatrix} C_{i,j} \\ C_{i+1,j} \\ C_{i,j+1} \\ C_{i+1,j+1} \end{bmatrix} = \frac{1}{3645} \begin{bmatrix} 486 & 81 & 81 & 81 \\ 81 & 486 & 81 & 81 \\ 81 & 81 & 486 & 81 \\ 81 & 81 & 81 & 486 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{29}
$$

whose individual explicit equations are given by,

DOI: 10.55214/2576-8484.v9i11.10962

$$C_{i,j} = \frac{1}{3645}\left[486S_1 + T_1\right]$$

$$C_{i+1,j} = \frac{1}{3645}\left[T_2 + 486S_2\right] \qquad (30)$$

$$C_{i,j+1} = \frac{1}{3645}\left[T_3 + 486S_3\right]$$

$$C_{i+1,j+1} = \frac{1}{3645}\left[T_4 + 486S_4\right]$$

where,

$$T_1 = 81(S_2 + S_3 + S_4)$$
$$T_2 = 81(S_1 + S_3 + S_4) \qquad (31)$$
$$T_3 = 81(S_1 + S_2 + S_4)$$
$$T_4 = 81(S_1 + S_2 + S_3)$$

Algorithm 2: 4EG iteration

I. Initialize $C^{(0)} = 0$ and $\varepsilon = 1.0\times 10^{-10}$.

II. For $i = 1,3,5,\ldots, m-3$ and $j = 1,3,5,\ldots, m-3,$ calculate Equation (30).

For $i = m-1,\ j = 1,3,5,\ldots, m-3$ and $i = 1,3,5,\ldots, m-3,\ j = m-1,$ calculate Equation (25).

III. If $|C^{(k+1)} - C^{(k)}| \le \varepsilon$ is satisfied, then proceed to Step IV. Otherwise, go back to Step II.

IV. Calculate and display approximate values of $S_B(x_i, y_j)$.

4. Numerical Experiments

To investigate the performance of the cubic B-spline Gauss-Seidel (C-BSGS), 2 Point-C-BSEG, and 4 Point-C-BSEG iterative methods, we evaluated three examples of the 2D Poisson equation. The goal was to validate the efficiency of both iterative approaches based on the number of iterations (W), execution time in seconds (t) and maximum errors $(L_\infty - \text{norm})$. Throughout the implementation of the point iterations, a convergence test was performed by considering a tolerance error, $\varepsilon = 1.0\times 10^{-10}$. This ensured that the iterative methods continued until the desired level of accuracy was achieved.

Example 1: Elsherbeny, et al. [24]

$$U_{xx} + U_{yy} = f(x,y), \quad x, y \in [0,1] \qquad (32)$$

with $f(x,y) = \sin(\pi x)\sin(\pi y)$. Then, the analytical solution of Problem (32) is obtained as follows

$$U(x,y) = -\frac{1}{2\pi^2}(\sin(\pi x)\sin(\pi y)), \quad x, y \in [0,1]. \qquad (33)$$

Example 2: Roslan and Hoe [25]

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

$$U_{xx} + U_{yy} = f(x,y), \quad x, y \in [0,1] \tag{34}$$

with $f(x,y) = (x^2 - 5x + 4)(y-1)(ye^{-x-2y}) + 2(x-1)x(2y^2 - 6y + 3)(e^{-x-2y})$. Then, the analytical solution of Problem (34) is given as follows

$$U(x,y) = e^{-x-2y} x(1-x) y(1-y), \quad x, y \in [0,1] \tag{35}$$

Example 3: Stonko, et al. [26]

$$U_{xx} + U_{yy} = f(x,y), \quad x, y \in [0,1] \tag{36}$$

with $f(x,y) = -2\pi^2 \cos(2\pi x)\sin^2(\pi y) - 2\pi^2 \sin^2(\pi x)\cos(2\pi y)$. Then, the analytical solution of Problem (36) is stated as follows,

$$U(x,y) = \sin^2(\pi x)\sin^2(\pi y), \quad x, y \in [0,1] \;. \tag{37}$$

5. Discussion

The numerical results presented in Table 1 indicate that the 4-C-BSEG iterative method achieves a reduction in the number of iterations by approximately 29.73% to 33.23% compared to the C-BSGS method. Furthermore, the 4-C-BSEG method demonstrates faster computational performance, reducing execution time by 25.00% to 44.66%. This implies that the 4-C-BSEG iterative method requires fewer iterations and is more efficient in terms of computational time than both the C-BSGS and 2-C-BSEG iterative methods.

Similarly, as shown in Table 2, it can be concluded that the 4-C-BSEG iterative method has fewer iterations by 34.17% - 35.18% compared to the C-BSGS method. Additionally, in terms of computational time, the 4-C-BSEG iterative method is faster than the C-BSGS iterative method, with a range of 36.84% - 51.21%. This indicates that the 4-C-BSEG iterative method is significantly more efficient than both the C-BSGS and 2-C-BSEG iterative methods for solving the second problem of 2D Poisson equations.

From the numerical results recorded in Table 3, it can be observed that the 4-C-BSEG iterative method has a lesser number of iterations by 33.58% - 34.49% compared to C-BSGS. In terms of computational time, the implementation of the 4-C-BSEG iterative method is faster by 25.00% - 47.02% than the C-BSGS iterative method. This indicates that the 4-C-BSEG iterative method requires fewer iterations and is more efficient in computational time than both the C-BSGS and 2-C-BSEG iterative methods.

Table 1.
Comparison of the number of iterations (W), execution time in seconds (t) and maximum errors (L_∞-norm) using Example 1

m	Method	Number of iterations (W)	Execution time (t)	Maximum Errors (L_∞-norm)
16	C-BSGS	207	0.14	2.656698e-05
	2-C-BSEG	184	0.12	2.656718e-05
	4-C-BSEG	139	0.09	2.709432e-05
32	C-BSGS	626	0.32	2.354121e-05
	2-C-BSEG	557	0.28	2.354195e-05
	4-C-BSEG	418	0.24	2.379541e-05
64	C-BSGS	1943	1.03	2.203216e-05
	2-C-BSEG	1734	0.77	2.203508e-05
	4-C-BSEG	1306	0.57	2.216327e-05
128	C-BSGS	5942	3.03	2.122272e-05
	2-C-BSEG	5333	2.73	2.123435e-05
	4-C-BSEG	4057	2.00	2.131874e-05
256	C-BSGS	17152	26.85	2.058207e-05
	2-C-BSEG	15533	24.32	2.062849e-05
	4-C-BSEG	12052	16.49	2.075144e-05

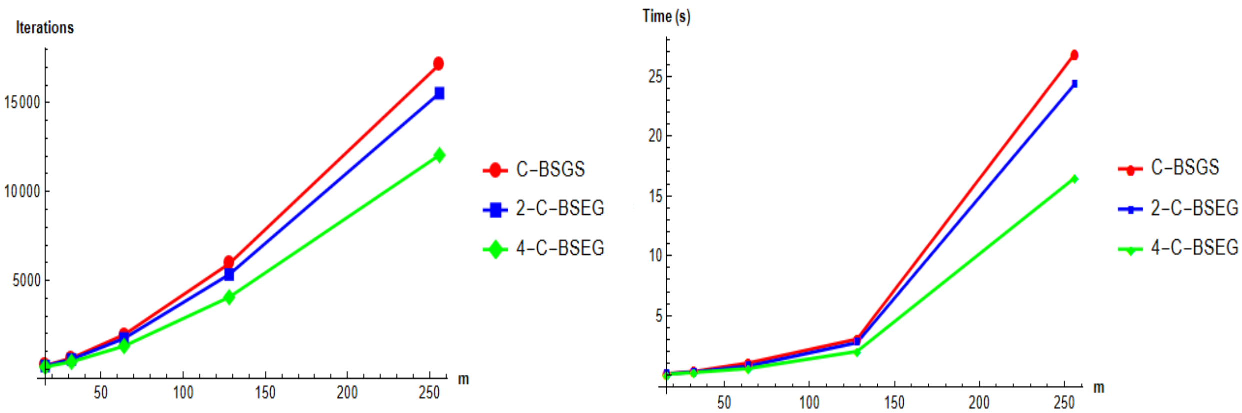

Figure 3.
The number of iterations and execution time in seconds for different mesh sizes (m) by C-BSGS, 2-C-BSEG, and 4-C-BSEG for Example 1.

Table 2.
Comparison of the number of iterations (W), execution time in seconds (t), and maximum errors (L_∞-norm) using Example 2

m	Method	Number of iterations (W)	Execution time (t)	Maximum Errors (L_∞ -norm)
16	C-BSGS	313	0.19	3.432516e-04
	2-C-BSEG	277	0.14	3.432516e-04
	4-C-BSEG	203	0.12	3.432517e-04
32	C-BSGS	1012	0.61	1.938521e-04
	2-C-BSEG	894	0.45	1.938522e-04
	4-C-BSEG	656	0.32	1.938523e-04
64	C-BSGS	3407	2.07	1.052995e-04
	2-C-BSEG	3016	1.34	1.052995e-04
	4-C-BSEG	2217	1.01	1.052996e-04
128	C-BSGS	11646	7.10	5.576506e-05
	2-C-BSEG	10323	5.58	5.576514e-05
	4-C-BSEG	7615	3.98	5.576532e-05
256	C-BSGS	39653	65.21	2.901369e-05
	2-C-BSEG	35221	57.65	2.901381e-05
	4-C-BSEG	26105	37.89	2.901405e-05

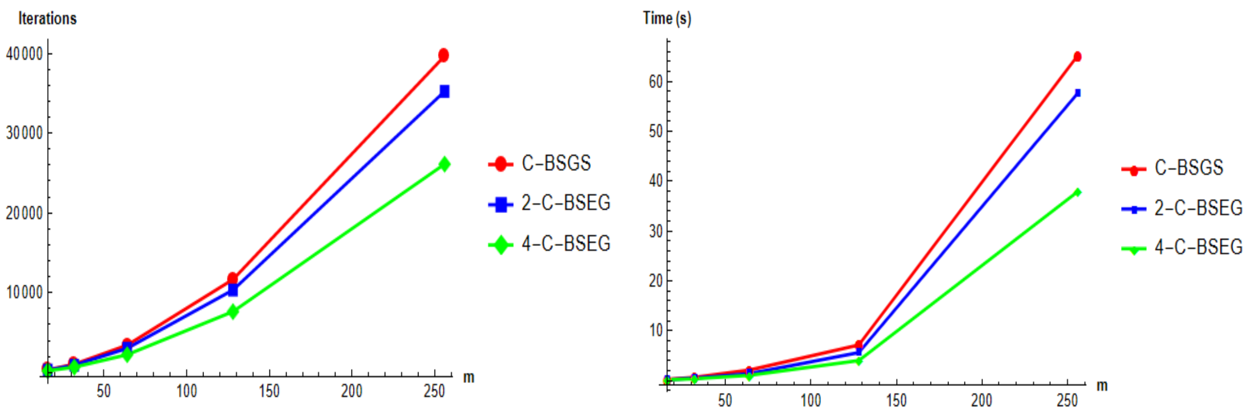

Figure 4.
The number of iterations and execution time in seconds for different mesh sizes (m) by C-BSGS, 2-C-BSEG, and 4-C-BSEG for Example 2.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

Table 3.
Comparison of the number of iterations (W), execution time in seconds (t), and maximum errors (L_∞-norm) using Example 3.

m	Method	Number of iterations (W)	Execution time (t)	Maximum Errors (L_∞-norm)
16	C-BSGS	290	0.16	1.105950e-03
	2-C-BSEG	257	0.14	1.105950e-03
	4-C-BSEG	192	0.12	1.125276e-03
32	C-BSGS	922	0.44	9.689167e-04
	2-C-BSEG	816	0.43	9.689175e-04
	4-C-BSEG	604	0.29	9.786855e-04
64	C-BSGS	3059	1.68	9.040932e-04
	2-C-BSEG	2711	1.21	9.040964e-04
	4-C-BSEG	2005	0.89	9.088428e-04
128	C-BSGS	10269	5.34	8.723361e-04
	2-C-BSEG	9119	4.60	8.723487e-04
	4-C-BSEG	6765	3.53	8.747496e-04
256	C-BSGS	34187	53.12	8.563620e-04
	2-C-BSEG	30439	45.97	8.564117e-04
	4-C-BSEG	22707	31.78	8.577004e-04

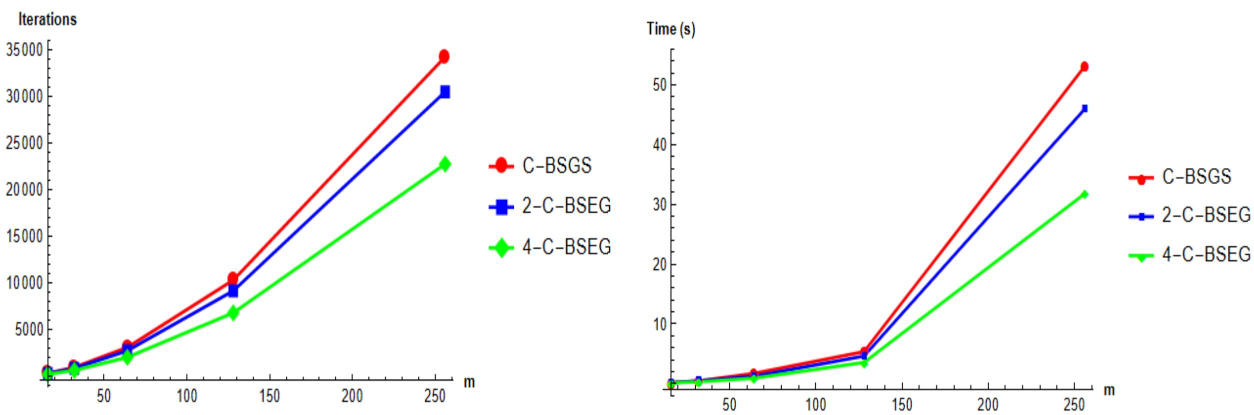

Figure 5.
The number of iterations and execution time in seconds for different mesh sizes (m) by C-BSGS, 2-C-BSEG, and 4-C-BSEG for Example 3.

6. Conclusion

A comparison of the numerical method using two methods is considered with all measured parameters for five different mesh sizes, as in Tables 1, 2, and 3, respectively. The numerical results for three examples indicate that the 4-C-BSEG iterative method outperforms the C-BSGS and 2-C-BSEG iterative methods, which require significantly fewer iterations to converge compared to the C-BSGS and 2-C-BSEG iterative methods. This demonstrates that the proposed iterative method exhibits a faster convergence rate, particularly for larger grids.

As the grid size increases, the execution time to approximate the exact solution for each example was shorter than with the C-BSGS and 2-C-BSEG methods. The observations in Figures 3, 4, and 5 illustrate how the 4-C-BSEG iteration method can be used to demonstrate the effectiveness of the bicubic B-spline collocation strategy in solving 2D Poisson equations. It can be inferred that their bicubic B-spline collocation, using both iterative approaches, may converge to their known exact solution effectively in terms of the infinity norm across various mesh sizes. For future work, this study will continue to investigate the capability of the explicit group (EG) iteration family in conjunction with the Successive Over-Relaxation (SOR) iteration approach, as introduced by Young [13] and Young

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 11: 710-725, 2025
DOI: 10.55214/2576-8484.v9i11.10962

[27]. This method belongs to the family of point-iterative techniques. The motivation for using SOR in this study comes from the work of Mohammed and Rivaie [28], who found that among the three indirect methods, namely Jacobi-Davidson, GS, and SOR, the SOR method is the most efficient and performs best.

Funding:
This research is supported by a UMSGreat research grant for a postgraduate student (Grant Number: GUG0674-1/2024), Universiti Malaysia Sabah.

Transparency:
The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References
[1] W. Bickley, "Piecewise cubic interpolation and two-point boundary problems," *The Computer Journal*, vol. 11, no. 2, pp. 206-208, 1968. https://doi.org/10.1093/comjnl/11.2.206
[2] D. Fyfe, "The use of cubic splines in the solution of two-point boundary value problems," *The Computer Journal*, vol. 12, no. 2, pp. 188-192, 1969. https://doi.org/10.1093/comjnl/12.2.188
[3] A. Curtis and M. Powell, "Using cubic splines to approximate functions of one variable to prescribed accuracy," AERE Harwell Report No. AERE-R5602(HMSO), 1967.
[4] İ. Dağ, D. Irk, and B. Saka, "A numerical solution of the Burgers' equation using cubic B-splines," *Applied Mathematics and Computation*, vol. 163, no. 1, pp. 199-211, 2005. https://doi.org/10.1016/j.amc.2004.01.028
[5] A. N. Akour, E. K. Jaradat, A. A. Mahadeen, and O. K. Jaradat, "Describing Bateman-Burgers' equation in one and two dimensions using Homotopy perturbation method," *Journal of Interdisciplinary Mathematics*, vol. 26, no. 2, pp. 271-283, 2023. https://doi.org/10.47974/JIM-1474
[6] D. D. Demir and N. Bildik, "The numerical solution of heat problem using cubic B-splines," *Applied Mathematics*, vol. 2, no. 4, pp. 131-135, 2012. https://doi.org/10.5923/j.am.20120204.06
[7] A. N. Ware and A. B. Ashine, "Cubic spline and finite difference method for solving boundary value problems of ordinary differential equation," *Asian Journal of Advances in Research*, vol. 4, no. 1, pp. 750-792, 2021.
[8] F. Du and T. Sun, "A bicubic B-spline finite element method for fourth-order semilinear parabolic optimal control problems," *Acta Mathematica Scientia*, vol. 44, pp. 2411-2421, 2024. https://doi.org/10.1007/s10473-024-0619-8
[9] F. M. Salama, N. Hj. Mohd. Ali, and N. N. Abd Hamid, "Efficient hybrid group iterative methods in the solution of two-dimensional time fractional cable equation," *Advances in Difference Equations*, vol. 2020, p. 257, 2020. https://doi.org/10.1186/s13662-020-02717-7
[10] F. M. Salama, N. N. Abd Hamid, N. H. M. Ali, and U. Ali, "An efficient modified hybrid explicit group iterative method for the time-fractional diffusion equation in two space dimensions," *AIMS Math*, vol. 7, no. 2, pp. 2370-2392, 2022. https://doi.org/10.3934/math.2022134
[11] F. M. Salama, N. H. M. Ali, and N. N. Abd Hamid, "Fast O (N) hybrid Laplace transform-finite difference method in solving 2D time fractional diffusion equation," *Journal of Mathematics and Computer Science*, vol. 23, pp. 110-123, 2021. https://doi.org/10.22436/jmcs.023.02.04
[12] Y. Saad and H. A. Van Der Vorst, "Iterative solution of linear systems in the 20th century," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1-2, pp. 1-33, 2000. https://doi.org/10.1016/S0377-0427(00)00412-X
[13] D. M. Young, *Iterative solution of large linear systems*. London, U.K: Academic Press, 1971.
[14] W. Hackbusch, *Iterative solution of large sparse systems of equations*. New York, U.S.A: Springer, 1995.
[15] Y. Saad, *Iterative methods for sparse linear systems*. Boston, MA, U.S.A: International Thomas Publishing, 1996.
[16] D. Evans and A. Abdullah, "Group explicit methods for parabolic equations," *International Journal of Computer Mathematics*, vol. 14, no. 1, pp. 73-105, 1983. https://doi.org/10.1080/00207168308803377
[17] D. Evans and M. Sahimi, "Group explicit methods for hyperbolic equations," *Computers & Mathematics with Applications*, vol. 15, no. 6-8, pp. 659-697, 1988. https://doi.org/10.1016/0898-1221(88)90288-X

[18] W. Yousif and D. Evans, "Explicit group over-relaxation methods for solving elliptic partial differential equations," *Mathematics and Computers in Simulation*, vol. 28, no. 6, pp. 453-466, 1986. https://doi.org/10.1016/0378-4754(86)90040-6
[19] R. K. Mohanty, "A variable mesh C-SPLAGE method of accuracy O(k2hl-1+khl+hl3) for 1D nonlinear parabolic equations," *Applied Mathematics and Computation*, vol. 213, no. 1, pp. 79-91, 2009. https://doi.org/10.1016/j.amc.2009.03.001
[20] A. Ali, Abdullah, and A. Ahmad, "The solution of poisson partial differential equations via double laplace transform method," *Partial Differential Equations in Applied Mathematics*, vol. 4, p. 100058, 2021. https://doi.org/10.1016/j.padiff.2021.100058
[21] D. Marsh, *Applied geometry for computer graphics and CAD*. Cambridgeshire, U.K: Springer, 2005.
[22] K. Sayevand, A. Yazdani, and F. Arjang, "Cubic B-spline collocation method and its application for anomalous fractional diffusion equations in transport dynamic systems," *Journal of Vibration and Control*, vol. 22, no. 9, pp. 2173-2186, 2016. https://doi.org/10.1177/1077546316636282
[23] A. Arshad, R. Mittal, and S. Sharma, "An efficient scheme for solution of two-dimensional Laplace's equation," in *Conference on Applied Mathematics & Computational Sciences"(ICAMCS-2019)*, 2019.
[24] A. M. Elsherbeny, R. M. I. El-hassani, H. El-badry, and M. I. Abdallah, "Solving 2D-poisson equation using modified cubic B-spline differential quadrature method," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2879-2885, 2018. https://doi.org/10.1016/j.asej.2017.12.001
[25] S. N. Roslan and Y. S. Hoe, "Numerical solutions of 2D poisson equation using finite element method and finite difference method," *Proceedings of Science and Mathematics*, vol. 23, pp. 99-107, 2024.
[26] D. Stonko, S. Khuvis, and M. K. Gobbert, *Numerical methods to solve 2-D and 3-D elliptic partial differential equations using Matlab on the cluster maya*. Baltimore: UMBC Faculty Collection, 2014.
[27] D. M. Young, "Second-degree iterative methods for the solution of large linear systems," *Journal of Approximation Theory*, vol. 5, no. 2, pp. 137-148, 1972. https://doi.org/10.1016/0021-9045(72)90036-6
[28] F. D. Mohammed and M. Rivaie, "Jacobi-davidson, gauss-seidel and successive over-relaxation for solving systems of linear equations," *Applied Mathematics and Computational Intelligence*, vol. 6, pp. 41-52, 2017.