Numerical solution of 2D Helmholtz Equation by Bicubic B-spline Collocation with SOR Iteration

Claire NC Motiun[1], **Jumat Sulaiman**[1#], **Aini Janteng**[1], **Asep Kuswandi Supriatna**[2]

1 Faculty of Science and Technology, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, MALAYSIA.
2 Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang 45363, West Java, INDONESIA.
#Corresponding author. E-Mail: jumat@ums.edu.my; Tel: +608832000.

ABSTRACT Two-dimensional Helmholtz equations arise in various fields, including acoustics, electromagnetics, and fluid dynamics, in which their numerical solutions are essential for modeling wave propagation phenomena. This study presents a numerical approach based on the Bicubic B-spline collocation method combined with the Successive Overrelaxation (SOR) iterative solver to efficiently solve the 2D Helmholtz equation. The method involves constructing a system of equations by discretizing the domain using Bicubic B-spline interpolation and collocation techniques. The resulting linear system is solved iteratively, and the performance of the SOR method is compared with the classical Gauss-Seidel (GS) iteration. Numerical experiments on three test cases demonstrate that the SOR iteration significantly reduces the number of iterations and computational time compared to the GS method, highlighting the effectiveness and efficiency of the proposed approach.

KEYWORDS: SOR Iteration; Gauss Seidel Iteration; B-spline Collocation Solution; Two-Dimensional Helmholtz Equation
Received 27 October 2025 **Revised** 17 November 2025 **Accepted** 22 November 2025 **Online** 13 December 2025
© Transactions on Science and Technology
Original Article

INTRODUCTION

The Helmholtz equation is a fundamental partial differential equation (PDE) that arises in various fields of science and engineering, including acoustics, electromagnetics, vibration analysis, and fluid dynamics (Juraev *et al.*, 2024). It models steady-state wave propagation and resonance behavior, which are essential for understanding oscillatory systems. However, analytical solutions to the Helmholtz equation are often limited to simple geometries and boundary conditions. For more complex domains, numerical methods are required to obtain accurate and efficient approximations.

Over the years, a variety of numerical methods have been developed to solve the Helmholtz equation, such as the Finite Difference Method (FDM), Finite Element Method (FEM), and Boundary Element Method (BEM). Although these traditional methods are widely used, they can encounter challenges such as high computational cost, numerical dispersion, and difficulty in maintaining solution smoothness. To overcome these issues, spline based numerical techniques have been proposed as an alternative due to their inherent smoothness, flexibility, and local support properties.

Among these, the bicubic B-spline collocation method offers an effective approach for solving two-dimensional (2D) partial differential equations (PDEs), as it provides continuous first and second derivatives that suit equations involving second-order terms. This method allows for smooth and accurate function approximation without requiring complex integration procedures. However, the resulting linear system from the collocation process can be large and computationally demanding. To efficiently solve such systems, iterative solvers are often applied. While more advanced solvers, such as Preconditioned Conjugate Gradient (PCG) and Multigrid methods, can achieve faster convergence for large-scale problems, this study focuses on the well-understood SOR method to establish a robust baseline performance within the bicubic B-spline collocation framework.

The Successive Over-Relaxation (SOR) method, an improved version of the Gauss-Seidel (GS) iterative approach, introduces a relaxation factor that accelerates convergence. Combining the bicubic B-spline collocation method with the SOR iteration scheme can yield a computationally efficient and accurate numerical framework for solving the 2D Helmholtz equation. Therefore, the main objective of this study is to develop and analyze a bicubic B-spline collocation scheme integrated with the SOR iteration method for solving the 2D Helmholtz equation. The proposed approach aims to achieve high accuracy, smoothness, and faster convergence compared to conventional numerical methods.

To evaluate the performance SOR method, let us consider a 2D Helmholtz equation over the region $\Omega = [a,b] \times [a,b]$ given as (Equation (1))

$$U_{xx} + U_{yy} + \beta U = f(x,y), \tag{1}$$

with Dirichlet boundary conditions given by

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

BACKGROUND THEORY

Development of Numerical Methods

The Helmholtz equation is a key mathematical model used to describe steady-state wave propagation and oscillatory behavior in various physical systems such as acoustics, electromagnetics, and vibration analysis. It is expressed as

$$\nabla^2 W + k^2 W = f(x,y), \tag{2}$$

where ∇^2 is Laplacian, k^2 is wavenumber, W is amplitude and $f(x,y)$ represents the source term. Over the years, classical numerical methods such as FDM, FEM, and BEM have been widely used to approximate solutions of the Helmholtz equation due to their efficiency and accuracy. For example, Zhang (2010) applied the FDM to solve the generalized Helmholtz equation with a focus on the possibility of cloaking an object by bending waves around it. This highlighted the potential of FDM for visualizing the effects of cloaking devices. However, standard FDM often suffers from numerical dispersion, particularly for high wave numbers, limiting its accuracy in oscillatory solutions.

To overcome these limitations, higher-order schemes were developed. For instance, third and fourth-order compact finite difference (FD) schemes were proposed for solving the Helmholtz equation with discontinuous media along straight interfaces in 2D (Feng *et al.*, 2011). Numerical examples demonstrated that these schemes efficiently and accurately solve the Helmholtz equation with discontinuities in the wavenumber and source term. The authors highlighted that the fourth-order scheme is preferable, as the third-order scheme can lose accuracy at large wavenumbers.

Building on these developments, Christodoulou *et al.* proposed two high-order FEM for solving 2D wave problems governed by the Helmholtz equation in 2017. They compared the Partition of Unity FEM (PUFEM) and the Spectral Element Method (SEM), evaluating their accuracy, condition number and storage requirements. While both methods accurately solve 2D Helmholtz problems, PUFEM is more efficient but less stable, whereas SEM is more stable but computationally expensive. In addition, Ma *et al.* (2010) proposed the Galerkin BEM to solve the exterior problems of the 2D Helmholtz equation with arbitrary wavenumber, which can effectively handle the hypersingular integral and non-uniqueness problems. Several numerical examples demonstrated that the proposed scheme is practical and effective for the exterior problems of the 2D Helmholtz equation with arbitrary wavenumber.

Despite these advances, these traditional methods may suffer from limitations such as numerical dispersion, high memory usage, or slow convergence. These limitations have motivated alternative formulations that provide higher smoothness and local control, leading to the development of spline-based numerical methods. B-spline collocation approaches, in particular, offer smooth approximations with continuous derivatives, making them highly suitable for solving PDEs such as the Helmholtz equation.

The B-spline collocation method is a mesh-based approach in which the differential equation is satisfied at selected collocation points. This method avoids complicated integral formulations, resulting in a system of linear equations that approximates the true solution. Cubic and bicubic B-splines are particularly effective because they provide continuous first and second derivatives and produce smooth approximations across 2D domains.

Several studies have successfully applied B-spline collocation methods to elliptic and parabolic problems. A novel B-spline collocation method for the solution of the incompressible Navier-Stokes equations (NSe) is presented by Botella and Shariff in 2003. By the end of this paper, the fractional step approaches, in conjunction with local discretizations like FDM and finite volume methods, are usually regarded as the most cost-effective methods for solving the NSe. This study also shows that the effectiveness of a B-spline approach for solving differential problems is highly dependent on the choice of approximation methods (i.e. Galerkin or collocation) and the regularity of the B-spline basis. Kadalbajoo *et al.* (2011) implemented the uniform cubic B-spline collocation method to obtain the numerical solution of the generalized Black-Scholes PDE. In this approach, the time variable is discretized using a suitable time-stepping method, while the spatial variable is approximated using the cubic B-spline collocation method. The study concluded that the proposed numerical method is effective for solving the generalized Black–Scholes equation, and it is both stable and second-order accurate. However, the high continuity of B-splines leads to large systems of linear equations which increase computational cost and require efficient solution approaches. This naturally motivates the use of iterative solvers.

To address this, iterative solvers such as the Successive Over-Relaxation (SOR) method have been widely employed. Two main categories of solvers are typically used in direct and iterative methods. Direct solvers, such as Gaussian elimination, can be computationally expensive and memory intensive for large-scale problems. Therefore, iterative solvers are often preferred for their simplicity and efficiency in handling sparse systems (Ferrari, 2024).

Among iterative methods, the GS and SOR methods are widely used. The GS method improves upon the Jacobi method by using the most recently updated values during iteration, resulting in faster convergence. However, its performance can still be slow for large matrices. To accelerate convergence, the SOR method was introduced by Young (1950), which modifies the GS method through the inclusion of a relaxation parameter, ω that adjusts the weighting of the iterative update.

The choice of an optimal relaxation parameter significantly affects the speed of convergence. When $\omega = 1$, the SOR method reduces to the GS method, while $1 < \omega < 2$ typically provides accelerated convergence for well-conditioned systems. The SOR technique has been widely applied in numerical solutions of elliptic PDEs due to its efficiency and ease of implementation.

In this study, the SOR iteration is employed to solve the linear system generated by the bicubic B-spline collocation discretization of the 2D Helmholtz equation. This combination aims to achieve high accuracy and computational efficiency by integrating the smoothness of B-spline approximation with the fast convergence of the SOR method. This direction is motivated by the findings of Mohammed & Rivaie (2017), who showed that among the iterative methods, to wit, Jacobi, GS, and SOR, the SOR method yields the most efficient and superior performance.

METHODOLOGY

Formulation of Bicubic B-spline Collocation Approximation Equations

The proposed method begins with the discretization of the 2D Helmholtz equation over a rectangular domain. The domain is divided into uniform subintervals in both x and y directions, then the subinterval distance for both directions is given as $h = \frac{b-a}{m}$. All node points of Equation (1) are denoted as (x_i, y_j) with $x_i = a + ih$ and $y_j = a + jh$, $0 \le i, j \le m$. Prior to having the 2D mesh network, let B-spline basis functions of order d and degree $d-1$ be denoted as $B_{i,d}(x)$ (Marsh, 2005) and defined as

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

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

Based on the Equation (3), the cubic B-spline basis, $B_{i,4}(x)$ function is given by (Hamid *et al.*, 2015)

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

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. This cubic form provides sufficient smoothness (C^2-continuity) and accuracy, making it suitable for numerical solutions of PDEs. The basic functions can be visualized as a triangular scheme to illustrate the Cox-de Boor recursion algorithm as in Figure 1 (Ahmad, 2021).

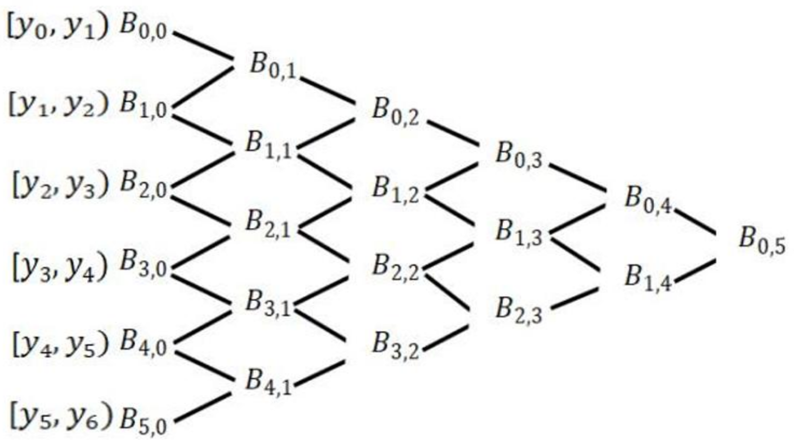


Figure 1. Triangular computation scheme for the B-spline basis function.

From the basis function in Equation (5), the bicubic B-spline approximation function could be defined as

$$S_B(x,y)=\sum_{i=-3}^{m-1}\sum_{j=-3}^{m-1}C_{i,j}B_{i,4}(x)B_{j,4}(y), \tag{6}$$

where $C_{i,j,}\ -3\le i,j\le m-1$ are unknown coefficients to be determined. The bicubic B-spline approximation function, $S_B(x,y)$ in Equation (6) is also known as a bicubic B-spline surface equation in which this equation is built from two cubic B-spline bases. By applying the simplification of the bicubic B-spline approximation function, $S_B(x,y)$ at any arbitrary node point, we have

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

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

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

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

Similar 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

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

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

Before getting the approximate solution, $u(x,y)$ over Equation (1), the values of $C_{i,j,}\ -3\le i,j\le m-1$ on the boundary conditions of the domain solution in Equation (1) can be determined by using approximate solution (6) in the boundary conditions (1) (discussed by Arshad *et al.*, 2019). Given the bottom boundary condition, $U(x,a)=f_2(x)$ gives

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

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

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

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

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

On the other hand, the right boundary condition, $U(b,y)=f_1(y)$ can be written as

$$\begin{bmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \mathrm{O} & \mathrm{O} & \mathrm{O} & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} C_{m-1,-2} \\ C_{m-1,-1} \\ C_{m-1,0} \\ \mathrm{M} \\ C_{m-1,m-1} \\ C_{m-1,m-2} \end{bmatrix} = \begin{bmatrix} 6f_1(y_0)-C_{m-1,-3} \\ 6f_1(y_1) \\ 6f_1(y_2) \\ \mathrm{M} \\ 6f_1(y_{m-1}) \\ 6f_1(y_m)-C_{m-1,m-1} \end{bmatrix}. \tag{15}$$

The tridiagonal system of Equations (12) - (15) is solved by forward and backward substitution. It means that the lower-upper (LU) matrix decomposition approach has been performed over the tridiagonal linear systems (12) – (15) to obtain the value of $C_{i,j}$ at the boundary conditions. Next, we attempt to calculate the approximate values of $C_{i,j}$ at all interior points. To do that, the discretization process over Equation (1) 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 Equation (1) becomes

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

Then substitute Equations (9) and (11) into Equation (16), Equation (1) can be approximated and simplified as a bicubic B-spline collocation approximation equation, which is given by

$$\begin{gathered}\alpha C_{i-3,j-3}+\mu C_{i-2,j-3}+\alpha C_{i-1,j-3}+\mu C_{i-3,j-2}-\vartheta C_{i-2,j-2}+\mu C_{i-1,j-2}+\alpha C_{i-3,j-1}+\mu C_{i-2,j-1}+\alpha C_{i-1,j-1} \\ =3h^2 f_{i,j}.\end{gathered} \tag{17}$$

where $\alpha=1-A,\ \ \mu=1-4A,\ \ \vartheta=8+16A,\ \ A=\dfrac{h^2\beta}{12}$ for $i,j=1,2,3,\mathrm{L},\mathrm{m}-1$. By imposing the bicubic B-spline collocation approximation Equation (17) over all node points, $(x_i,y_j), i,j=1,2,3,\mathrm{L},m-1$, a large-scale and sparse linear system generated from Equation (17) can be rewritten in a matrix form as

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

Formulation of SOR Iterative Method

By solving the linear system presented in Equation (18), this section outlines the derivation and implementation of the SOR iterative method. To assess its performance, the classical GS method is employed as a benchmark. Both iterative methods are known as a family of stationary iterative methods, meaning that each new approximation $x^{(k+1)}$ depends only on the most recent approximation, $x^{(k)}$ (Bindel, 2012; Saad, 2003; Quarteroni *et al.*, 2007; Varga, 2009; Golub & Van Loan, 2013). In 1950, the formulation of the SOR iterative method was developed by Young and Frankel (Liu & Chang, 2024). The basic idea of this method has been inspired by modifying the GS iterative method (Mayooran & Light, 2016). To establish the formulation of the SOR iterative method, 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 an upper triangular parts of A respectively. The SOR iterative approach can be formulated from the linear system (18) and shown in vector form as (Akhir *et al.*, 2009)

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

where $C^{(k+1)}$ represents the unknown vector of the $(k+1)^{th}$ iteration and relaxation parameter $\omega \in [1,2)$. Meanwhile, the SOR scheme can be expressed based on the point iteration as

$$C^{(k+1)}_{i-2,j-2} = (1-\omega)C^{(k)}_{i-2,j-2} + \frac{1}{\vartheta}\left(\begin{matrix} \alpha C^{(k+1)}_{i-3,j-3} + \mu C^{(k+1)}_{i-2,j-3} + \alpha C^{(k+1)}_{i-1,j-2} + \mu C^{(k+1)}_{i-3,j-2} \\ + \mu C^{(k)}_{i-1,j-2} + \alpha C^{(k)}_{i-3,j-1} + \mu C^{(k)}_{i-2,j-1} + \alpha C^{(k)}_{i-1,j-1} \end{matrix} - 3h^2 f_{i,j}\right). \tag{21}$$

for $i, j = 1,2,3,L,m-1$. The SOR method is similar to the GS method except the use of values relaxation parameter, ω where when taking $\omega = 1$, the formulation of the SOR iterative method can be reduced as the GS iterative method. When $0 < \omega < 1$, we get under relaxation while for $1 < \omega < 2$, we get over relaxation which can accelerate convergence. The relaxation parameter, ω in this study was determined empirically by testing values between 1.0 and 1.9 for each mesh size and selecting the value that minimized the number of iterations and computational time compared to the previous mesh size. The implementation of the SOR iterative method can be defined in Algorithm 1.

Algorithm 1. *SOR iteration*

```
I.   Initialize $C^{(0)} = 0$ and $\varepsilon = 1.0 \times 10^{-10}$.
II.  Assign the value of relaxation parameter, $\omega$.
III. For $i = 1,2,3,...,m-1$ and $j = 1,2,3,...,m-1$, calculate Equation (21).
IV.  If $\left|C^{(k+1)} - C^{(k)}\right| \le \varepsilon$ is satisfied, then go to step V. Otherwise, go back to step III.
V.   Calculate and display approximate values of $S_B(x_i, y_j)$.
```

RESULTS

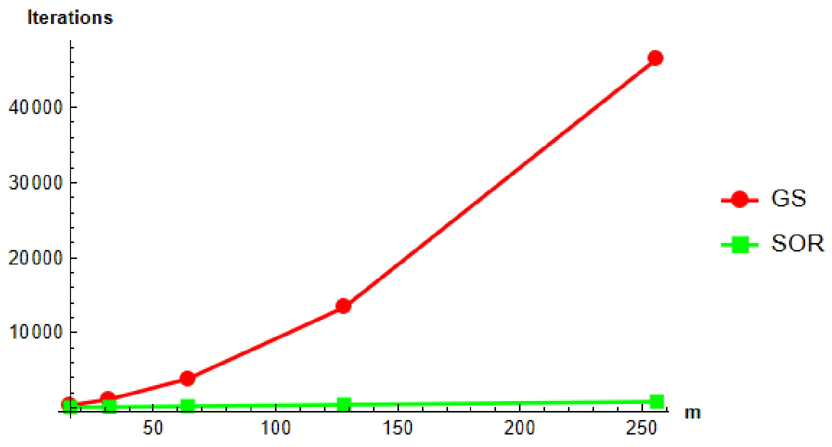
To investigate the performance of the GS and SOR iterative methods, we evaluated three examples of the 2D Helmholtz equation. The goal was to validate the efficiency of both iterative approaches based on the number of iterations (W), execution time in seconds (t) and maximum errors (L_∞-norm). For Example 1, the GS and SOR iteration efficiency is illustrated in Figure 2, where parts (a) and (b) show the behavior across different mesh sizes. The corresponding numerical comparison of the number of iterations, execution time and maximum errors is summarized in Table 1. For Example 2, the iteration efficiency results are presented in Figure 3, based on parts (c) and (d) for various mesh sizes. Table 2 provides a detailed comparison of iterations, execution time, and maximum errors. Figure 4 shows the efficiency of the GS and SOR methods for Example 3, and the numerical results are systematically summarized in Table 3. Throughout the implementation of the point iterations, a convergence test was performed by considering a tolerance error. This ensured that the iterative methods continued until the desired level of accuracy was achieved.

Example 1 (Evans, 1985)

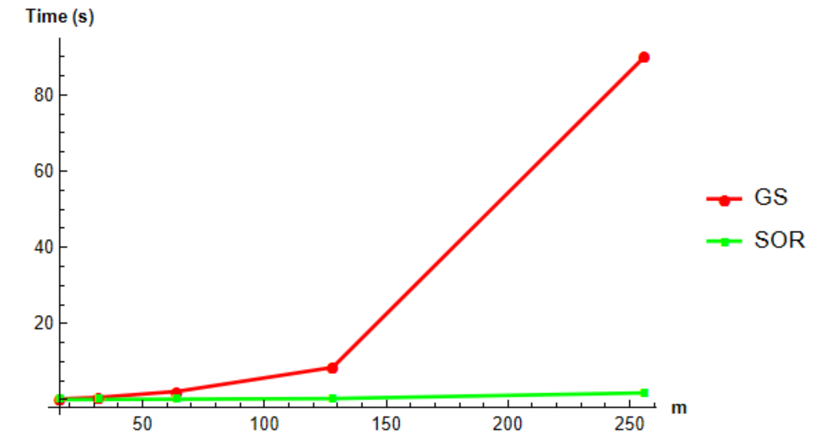
$$U_{xx} + U_{yy} - 10U = 6 - (20x^2 + 10y^2), \qquad x, y \in [0,1]. \tag{22}$$

The analytical solution for Equation (22) is given by

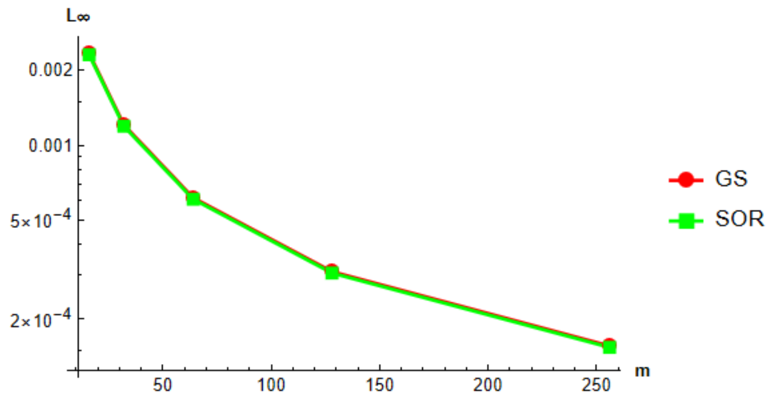
$$U(x, y) = 2x^2 + y^2, \qquad x, y \in [0,1].$$



a) Number of iterations (W) for Example 1



b) The execution time in second (t) for Example 1



c) The maximum errors (L_∞ -norm) for Example 1

Figure 2. GS and SOR (c), iteration efficiency are compared based on (a) and (b) for different mesh sizes (m) by GS and SOR for Example 1.

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

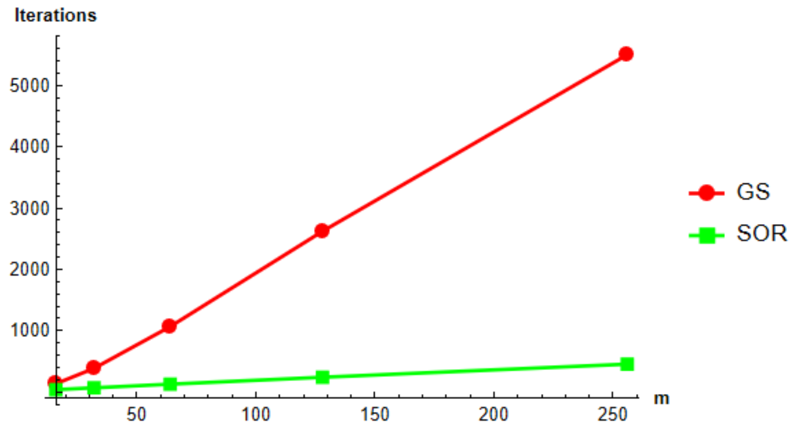
Mesh sizes (m)	Method	W	t	L_∞-norm
16	GS	343	0.21	2.338938 e-03
	SOR	61	0.03	2.310172 e-03
32	GS	1124	0.61	1.206577 e-03
	SOR	112	0.08	1.191561 e-03
64	GS	3842	2.18	6.137324 e-04
	SOR	215	0.15	6.060563 e-04
128	GS	13358	8.49	3.096344 e-04
	SOR	419	0.31	3.057502 e-04
256	GS	46467	89.95	1.555315 e-04
	SOR	811	1.80	1.535761 e-04

Example 2 (Evans *et al.*, 1985)

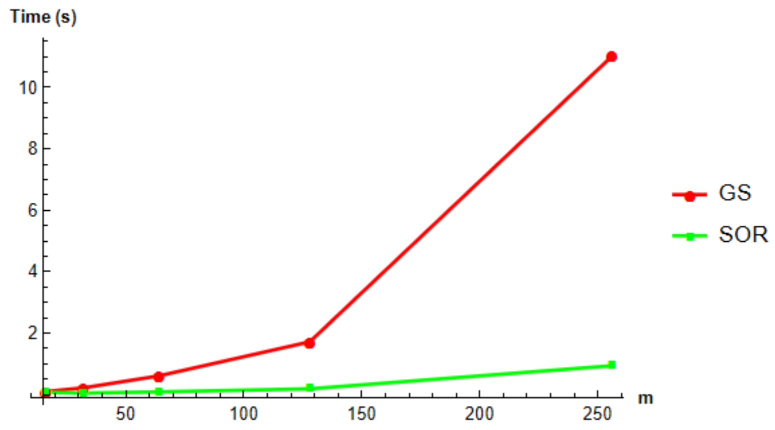
$$U_{xx} + U_{yy} - 2U = \sin y(12x^2 + 3x^4), \qquad x, y \in [0,1]. \tag{23}$$

Then, the analytical solution of Equation (23) is presented as follows.

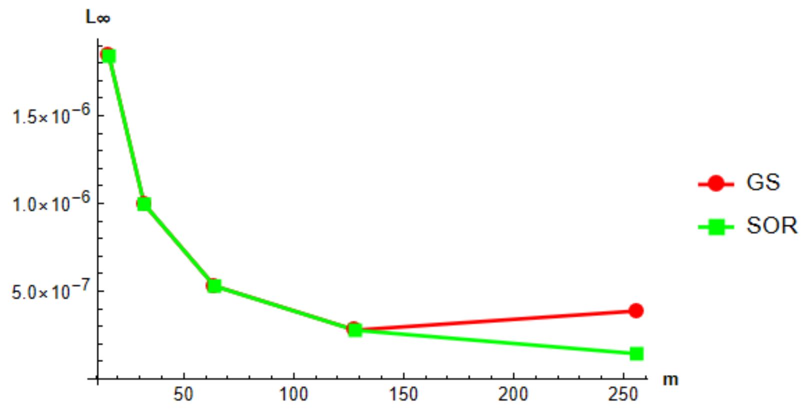
$$U(x, y) = x^4 \sin y, \qquad x, y \in [0,1].$$



a) Number of iterations (W) for Example 2



b) The execution time in second (t) for Example 2



c) The maximum errors (L_∞-norm) for Example 2

Figure 3. GS and SOR (c), iteration efficiency are shown based on (a) and (b) for different mesh sizes (m) by GS and SOR for Example 2.

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

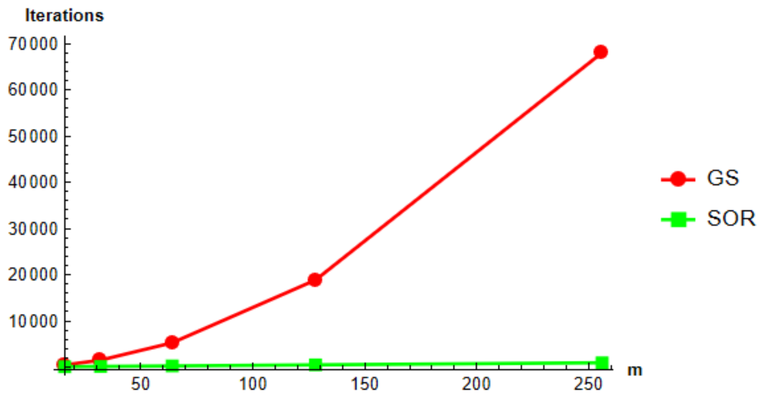
Mesh sizes (m)	**Method**	W	t	L_∞**-norm**
16	GS	134	0.10	1.839938 e-06
	SOR	38	0.08	1.838963 e-06
32	GS	385	0.22	9.950343 e-07
	SOR	68	0.05	9.954634 e-07
64	GS	1066	0.61	5.293511 e-07
	SOR	127	0.09	5.287697 e-07
128	GS	2621	1.72	2.777447 e-07
	SOR	240	0.19	2.770963 e-07
256	GS	5501	11.01	3.867561 e-07
	SOR	454	0.94	1.433375 e-07

Example 3 (Dasari & Parikh, 2023)

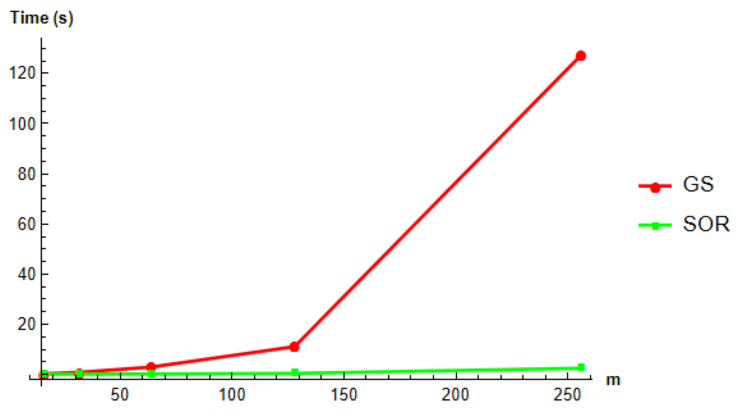
$$U_{xx} + U_{yy} - \beta^2 U = (\sinh(x) + \cosh(y))(1-\beta^2), \qquad x, y \in [0,1]. \tag{24}$$

Then, the analytical solution of Equation (24) is stated as follows.

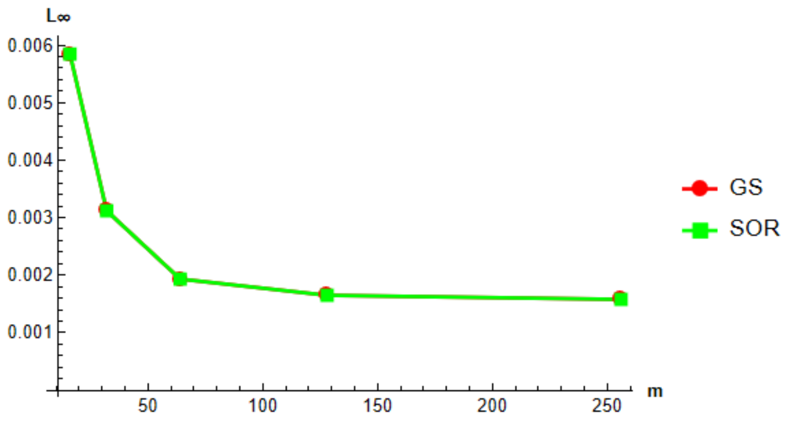
$$U(x,y) = \sinh(x) + \cosh(y), \qquad x, y \in [0,1].$$



a) Number of iterations (W) for Example 3



b) The execution time in second (t) for Example 3



c) The maximum errors (L_∞-norm) for Example 3

Figure 4. GS and SOR (c), iteration efficiency are evaluated based on (a) and (b) for different mesh sizes (m) by GS and SOR for Example 3.

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

Mesh sizes (m)	**Method**	W	t	L_∞**-norm**
16	GS	448	0.24	5.841906 e-03
	SOR	70	0.06	5.841906 e-03
32	GS	1499	0.81	3.117555 e-03
	SOR	131	0.09	3.117556 e-03
64	GS	5255	2.92	1.931084 e-03
	SOR	250	0.16	1.931104 e-03
128	GS	18841	11.07	1.648264 e-03
	SOR	478	0.39	1.648368 e-03
256	GS	68051	127.29	1.576925 e-03
	SOR	930	2.44	1.577366 e-03

DISCUSSION

The numerical results presented in Tables 1-3 clearly demonstrate the superior performance of the proposed SOR iterative method compared to the GS approach. To compute the performance improvement of the SOR method, the percentage reduction relative to the GS method was calculated using Equation (25)

$$\text{Reduction (\%)} = \frac{GSvalue - SORvalue}{GSvalue} \times 100 \tag{25}$$

This formula was applied to both the number of iterations and the computational time. The resulting percentage reductions for all three problems are summarized in Table 4.

Table 4. Reduction Percentage of the number of iterations (W**)** and execution time in seconds (t**)** for the SOR compared with GS iterative method.

Problem	Iteration Reduction ($W\%$)	Time Reduction ($t\%$)
Problem 1	82.22 - 98.25	85.71 - 98.00
Problem 2	71.64 - 91.75	20.00 - 91.46
Problem 3	84.38 - 98.63	75.00 - 98.08

Across three problems, the SOR iterative method significantly reduces the number of iterations (71.64% to 98.63%) and computational time (20.00% to 98.08%). These improvements highlight the effectiveness of the SOR method in accelerating convergence and reducing computational effort.

Since both the GS and SOR methods are applied to the same system of linear equations, they produce nearly identical the value of L_∞-norm errors, as shown in Figure 2(c), 3(c) and 4(c). The difference between them lies only in the rate of convergence, where the SOR method reaches the same level of accuracy in fewer iterations due to the inclusion of the relaxation parameter, ω.

The enhanced performance of the SOR method can be directly related to the theoretical formulation presented in the section of background theory where SOR incorporates a relaxation parameter, ω which allows the iterative updates to move further toward the approximate solution. When $\omega = 1$, SOR reduces to the standard GS method. For $0 < \omega < 1$, the method is under-relaxed, which improves stability at the expense of slower convergence, while for $1 < \omega < 2$, the method is over-relaxed, which accelerates convergence by reducing the number of iterations needed to reach the desired accuracy. This theoretical mechanism explains the consistently faster convergence observed in all numerical experiments.

The results further show that as the mesh size increases, the number of iterations required by the GS method grows approximately at the rate of $O(m^2)$, while the SOR method exhibits a much more efficient convergence behavior, roughly proportional to $O(m)$. Here, m denotes the number of interior grid points along one spatial direction. The term $O(m^2)$ indicates that the required number of iterations grows quadratically with m, for instance, doubling m roughly quadruples the iterations, whereas $O(m)$ reflects a linear growth rate. As a result, the GS method becomes increasingly inefficient on finer grids, while the SOR method scales more favorably especially when an optimal relaxation parameter is used. Big-O notation known as Landau symbols provides a standard framework for describing such growth behaviours and computational complexity (Shakil, 2022; Hackbusch, 1995). This trend confirms that the introduction of the relaxation parameter significantly improves the scalability and efficiency of the iterative process for larger grid sizes.

Furthermore, the accuracy and robustness of the method are enhanced by the bicubic B-spline collocation methodology. The L_∞-norm results indicate that its high-order continuity and local support guarantee smooth approximations and robust convergence over a range of mesh sizes.

Overall, these results confirm that the combination of SOR iteration and bicubic B-spline collocation provides a highly efficient and reliable solver for large-scale elliptic boundary value problems. Both the iteration count and execution time results conclusively show that the SOR method outperforms the conventional GS iterative technique.

CONCLUSION

The numerical techniques for solving the 2D Helmholtz equation using the bicubic B-spline collocation solution were compared to the GS iterative approach. The numerical findings show that the SOR iterative solver performs better than GS, requiring a notably smaller number of iterations to reach convergence and offering lower computational time as the mesh resolution increases. These outcomes confirm that SOR, when coupled with the bicubic B-spline formulation, provides an efficient and reliable numerical strategy for approximating the solution of the 2D Helmholtz equation. For future work, this study intends to extend the investigation to the Explicit Group (EG) iteration family combined with the SOR method for solving the 2D Helmholtz equation.

ACKNOWLEDGEMENTS

This research is supported by UMSGreat research grant for a postgraduate student no GUG0674-1/2024, Universiti Malaysia Sabah.

REFERENCES

[1] Abd Hamid, N. N. 2016. *Splines for Two-Dimensional Partial Differential Equations.* PhD Thesis, Universiti Sains Malaysia.

[2] Ahmad, A. 2021. *Fourth-order spline methods for solving nonlinear Schrodinger equation.* PhD Thesis, Universiti Sains Malaysia.

[3] Akhir, M. K., Othman, M. & Sulaiman, J. 2009. Solving two-dimensional steady Navier-Stokes equations by modified SOR iterative method. *Proceedings of 4th Conference on Research and Education in Mathematics.* 21-23 October, Kuala Lumpur, Malaysia.

[4] Arshad, A., Mittal, R.C. & Sharma, S. 2019. An efficient scheme for solution of two-dimensional Laplace's equation. *Proceedings of International Conference on Applied Mathematics and Computational Sciences.* 17-19 October, Dehradun, India. pp 20-33.

[5] Bindel, D. 2012. *Matrix Computations (CS 6210)* (https://www.cs.cornell.edu/~bindel/class/cs6210-f12/notes/lec31.pdf). Last accessed on 22 October 2025.

[6] Botella, O. & Shariff, K. 2003. B-spline Methods in Fluid Dynamics. *International Journal of Computational Fluid Dynamics*, 17, 133 - 149.

[7] Christodoulou, K., Laghrouche, O., Mohamed, M.S. & Trevelyan, J. 2017. High-order finite elements for the solution of Helmholtz problems. *Computers and Structures*, 191, 129-139.

[8] Dasari, S. & Parikh, A. 2023. Numerical solution of 2D Inhomogeneous Helmholtz Equation using the Meshless Radial Basis Function Method. *International Journal of Innovative Science, Engineering & Technology*, 10 (1), 133-139.

[9] Evans, D. J. 1985. Group Explicit Iterative Methods for solving large linear systems. *International Journal of Computer Mathematics*, 17 (1), 81-108.

[10] Evans, D.J., Ergut, M. & Bulut, H. 2003. The numerical solution of multidimensional partial differential equations by decomposition method. *International Journal of Computer Mathematics*, 80 (9), 1189-1198.

[11] Feng, X., Li, F. & Qiao, Z. 2011. High order compact finite difference schemes for the Helmholtz equation with discontinuous coefficients. *Journal of Computational Mathematics,* 29(3), 324-340.

[12] Ferrari, M. 2024. *A comparison of sparse solvers for severely Ill-conditioned linear system in geophysical Marker-In-Cell simulations.* (https://doi.org/10.48550/arxiv.2409.11515). Last accessed on 22 October 2025.

[13] Golub, G.H. & Loan, C.F.V. 2009. *Matrix Computations* (4th edition). Baltimore, Maryland: The Johns Hopkins University Press.

[14] Hackbusch, W. 1995. *Iterative solution of large sparse systems of equations* (2nd edition). New York: Springer.

[15] Juraev, D.A., Agarwal, P., Elsayed, E.E. & Targyn, N. 2024. Helmholtz equations and their applications in solving physical problems. A*dvanced Engineering Science,* 4, 54-64.

[16] Kadalbajoo, M.K., Tripathi, L.P. 2011. A cubic B-spline collocation method for a numerical solution of the generalized Black-Scholes equation. *Mathematical and Computer Modelling,* 55(3-4), 1483-1505.

[17] Liu, C.S. & Chang, C.W. 2024. The SOR and AOR Methods with Stepwise Optimized Values of Parameters for the Iterative Solutions of Linear Systems. *Contemporary Mathematics*, 5(3), 4013–4028.

[18] Ma, J., Zhu, J. & Li, M. 2010. The Galerkin boundary element method for exterior problems of 2D Helmholtz equation with arbitrary wavenumber. *Engineering Analysis with Boundary Elements,* 34, 1058-1063.

[19] Marsh, M. 2005. *Applied Geometry for Computer Graphics and CAD* (2nd edition). London: Springer.

[20] Mayooran, T. & Light, E. 2016. Applying the Successive Over-relaxation Method to a Real World Problems. *American Journal of Applied Mathematics and Statistics*, 4(4), 113-117.

[21] Mohammed, F. D. & Rivaie, M. 2017. Jacobi-Davidson, Gauss-Seidel and Successive Over-Relaxation for solving systems of linear equations. *Applied Mathematics and Computational Intelligence,* 6, 41-52.

[22] Quarteroni, A., Sacco, R. & Saleri, F. 2007. *Numerical Mathematics* (2nd edition). Berlin Heidelberg New York: Springer.

[23] Saad, Y. 2003. *Iteratieve Methods for Sparse Linear Systems* (2nd edition). Society for Industrial and Applied Mathematics.

[24] Shakil, M. 2022. *Landau "big-O" and "small-o" symbols - A Historical Introduction, with some Applicability* (https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.researchgate.net/publication/364209922_Landau_big-O_and_small-o_symbols_-A_Historical_Introduction_with_Some_Applicability&ved=2ahUKEwjNobvStrqRAxVSbmwGHQH_LdIQFnoECBsQAQ&usg=AOvVaw0APlpjoSBA01oNjkZR8Dtm). Last accessed on 22 October 2025.

[25] Varga, R.S. 2009. *Matrix Iterative Analysis* (Expanded Edition). Verlag Berlin Heidelberg: Springer

[26] Young, D.M. 1950. *Iterative methods for solving partial differential equations of elliptic type.* Doctoral Thesis, Harvard University, Cambridge, MA.

[27] Zhang, L. *The Finite Difference Method for the Helmholtz Equation with Applications to Cloaking.* (https://math.missouristate.edu/_Files/li.pdf). Last accessed on 22 October 2025.