BAREKENG: Journal of Mathematics and Its Applications
December 2026 Volume 20 Issue 4 Page 3437-3456
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss4pp3437-3456

SMART INDOOR NAVIGATION FOR MOBILE ROBOTS: A HARMONIC POTENTIAL-BASED APPROACH WITH ACCELERATED OVER-RELAXATION ITERATIVE METHOD

A'Qilah Ahmad Dahalan [1*], Rupal Srivastava [2], Ruzanna Mat Jusoh [3], Azali Saudi [4], Jumat Sulaiman [5]

[1]*Pusat Kecemerlangan Penyelidikan dan Inovasi Sultan Ibrahim, Universiti Pertahanan Nasional Malaysia,*
[3]*Faculty of Defence Science and Technology, Universiti Pertahanan Nasional Malaysia*
57000 Kuala Lumpur, Malaysia

[2]*Faculty of Engineering and Built Environment, South East Technological University Waterford*
X91 WKT3 Waterford, Ireland

[4]*Faculty of Computing and Informatics, Universiti Malaysia Sabah*
[5]*Faculty of Science and Natural Resources, Universiti Malaysia Sabah*
Kota Kinabalu, Sabah, 88400, Malaysia

*Corresponding author's e-mail: * **a.qilah@upnm.edu.my***

Article Info

Article History:

Received: 28th October 2025
Revised: 04th May 2026
Accepted: 03rd June 2026
Available online: 01st July 2026

Keywords:

Accelerated method;
Collision free;
Iterative method;
Laplace's equation;
Optimal path;
Path finding;
Skewed approximation strategy.

ABSTRACT

Path planning for autonomous mobile robots remains a critical challenge, particularly in environments with obstacles and constraints. Efficient navigation requires a robust algorithm capable of generating smooth, collision-free trajectories while ensuring computational efficiency. This study addresses the problem of indoor mobile robot navigation by leveraging harmonic potential fields, a solution derived from Laplace's equation, to formulate an effective path-planning strategy. Conventional numerical methods for solving Laplace's equation, such as Successive Over-Relaxation and Accelerated Over-Relaxation, often require extensive computational resources, especially in large-scale environments. To overcome this limitation, this research introduces an improved iterative approach, the Explicit Decoupled Group Modified Accelerated Over-Relaxation (EDGMAOR) method, which enhances computational efficiency and convergence speed. The EDGMAOR method incorporates a half-sweep block approach, significantly reducing the number of computations required per iteration while maintaining accuracy. To validate the effectiveness of the proposed method, simulations were conducted in a static, enclosed environment with various configurations of obstacles. Different starting and goal positions were tested to assess the efficiency, accuracy, and computational cost of the generated paths. The results indicate that the EDGMAOR method outperforms conventional approaches by achieving faster convergence rates and reduced computational time, demonstrating its suitability for real-time robot pathfinding applications. Furthermore, the study highlights that with greater obstacles proliferation, the EDGMAOR method maintains its efficiency, as obstacle regions are automatically excluded from unnecessary computations. This characteristic makes the method particularly useful for complex indoor environments where real-time processing is crucial. In conclusion, this research establishes EDGMAOR as a practical and effective solution for solving mobile robot path-planning problems, providing a balance between computational speed, accuracy, and robustness. The findings contribute to the ongoing advancements in autonomous robotics and artificial intelligence-driven navigation systems, with potential applications in industrial automation, smart transportation, and defence sectors.

How to cite this article:

A. A. Dahalan, R. Srivastava, R. M. Jusoh, A. Saudi and J. Sulaiman., "SMART INDOOR NAVIGATION FOR MOBILE ROBOTS: A HARMONIC POTENTIAL-BASED APPROACH WITH ACCELERATED OVER-RELAXATION ITERATIVE METHOD", *BAREKENG: J. Math. & App.*, vol. 20, no. 4, pp. 3437-3456, Dec, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

The deployment of autonomous mobile robots is becoming increasingly prominent across a wide range of sectors, including industrial automation, aerospace operations, security monitoring, medical assistance, and intelligent transportation systems [1], [2]. These smart systems rely on efficient path-planning algorithms to navigate environments while avoiding obstacles, ensuring safety, and optimizing computational efficiency [3], [4]. The capability of a mobile robot to autonomously compute an optimal route from an initial position to a designated goal destination remains a fundamental requirement, particularly for applications that demand precision and operational reliability. This requirement becomes even more critical in structured indoor environments, where robots must operate within predefined maps and in the presence of static obstacles that constrain their movement and decision-making space [5], [6], [7].

Path-planning in mobile robotics can generally be categorized into global and local navigation. Global navigation relies on pre-existing knowledge of the environment and is commonly used in offline path-planning methods, whereas local navigation allows robots to react dynamically to obstacles in real-time [8], [9]. This study focuses on global path-planning in a predetermined, static environment, where a robot computes its path before execution. One of the fundamental challenges in path-planning is avoiding local minima, where conventional potential field methods may cause robots to become trapped in non-optimal solutions. Hence, this study investigates the problem of mobile robot pathfinding through an innovative approach based on heat transfer principles. The movement of heat, mathematically described by Laplace's equation, serves as an effective model for navigation planning. A key benefit of this approach is its inherent capability to escape local minima, a common issue in potential field-based navigation. By leveraging harmonic potential functions derived from Laplace's equation, a robot can be guided toward its goal with a smooth and collision-free trajectory [10], [11]. The subsequent sections offer additional clarifications. A range of solutions exists for deriving harmonic functions, though numerical approaches are the most frequently applied. This preference is owing to the accessibility of high-speed computing resources, as well as the sophistication and effectiveness of these methods in addressing the problem [10], [12], [13].

Traditionally, path generation methods utilizing harmonic potentials rely on the standard 5-point (5P) Laplacian discretization to approximate the solution. However, the method's efficiency is compromised by its slow convergence and heavy computational load, limiting its practicality for real-time applications. To address these challenges, this study investigates the use of a skewed 5P Laplacian operator, which introduces a refined algebraic formulation aimed at improving computational efficiency by utilizing mesh grid points [14], [15], [16], [17]. The application of this skewed operator results in the development of algebraic skewed approximation equations, forming the basis of an advanced computational framework. To further enhance performance, this research introduces a novel numerical technique known as Explicit Decoupled Group Modified Accelerated Over-Relaxation (EDGMAOR). The EDGMAOR method is designed to enhance convergence speed while maintaining high accuracy in solving algebraic skewed linear systems derived from harmonic potential equations. This technique is particularly effective in reducing the number of iterations required for computing harmonic functions, thereby making real-time path planning more feasible for indoor mobile robots.

Robotic path-planning has frequently adopted harmonic potential fields for their capacity to create continuous and feasible trajectories. In path planning scenarios, the computational domain is composed of various components, including boundaries, borders, obstacles, multiple starting points, and a designated target point. A key advantage of employing harmonic functions in path planning is its ability to eliminate false local minima. This property arises due to the minimum-maximum principle, ensuring a smooth and well-defined navigation path for autonomous robots. Consequently, harmonic potential fields offer a highly effective approach to robot motion planning by providing a continuous and collision-free trajectory toward the goal [14]. Laplace's equation has historically been solved using standard numerical approaches, such as the Jacobi method, Gauss-Seidel (GS), and Successive Over-Relaxation (SOR) [16], [18], [19], [20], [21]. Despite widespread application, conventional iterative solvers often exhibit slow convergence, which reduces its suitability for time-sensitive scenarios such as real-time robotic navigation. To overcome this computational bottleneck, this study introduces an accelerated iterative solver called EDGMAOR, building upon EDG [22] and MAOR frameworks [23], to efficiently solve Eq. (1).

Using the harmonic potential field approach, solving Laplace's equation allows for the generation of an optimal path that directs a point robot from the initial position to the target while ensuring obstacle avoidance. In this framework, obstacles are modeled as source points of high potential, while the target

location serves as a sink, characterized by the lowest potential value. This formulation naturally aligns with Dirichlet boundary conditions, ensuring a structured potential gradient for navigation. The Gradient Descent Search (GDS) algorithm is applied to the computed potential field, enabling the robot to continuously move toward regions of lower potential, ultimately reaching the goal position.

To validate the performance of the proposed EDGMAOR method, simulations will be carried out within a two-dimensional domain with an enclosed, static environment and varying obstacle configurations. The heat transfer analogy is employed to model the potential field, where temperature distribution and heat flow principles are used to determine potential values and produce a seamless navigation trail. The EDGMAOR scheme is utilized to solve Laplace's equation and compute the temperature distributions across the grid nodes. To benchmark the performance of EDGMAOR, comparative analyses will be carried out against several existing iterative solvers, including i.e. Explicit Group Successive Over-Relaxation (EGSOR), Explicit Group Accelerated Over-Relaxation (EGAOR), Explicit Group Modified Successive Over-Relaxation (EGMSOR), Explicit Group Modified Accelerated Over-Relaxation (EGMAOR), Explicit Decoupled Group Successive Over-Relaxation (EDGSOR), Explicit Decoupled Group Accelerated Over-Relaxation (EDGAOR), and Explicit Decoupled Group Modified Successive Over-Relaxation (EDGMSOR).

Ultimately, this research highlights the effectiveness and practicality of the EDGMAOR technique in computing harmonic functions for robot path planning. The findings suggest that this approach holds great potential for real-time applications in autonomous mobile robots, particularly in structured indoor environments where efficient navigation is crucial. By leveraging the principles of heat transfer and accelerated numerical solvers, this study contributes to the advancement of intelligent motion planning strategies that enhance the reliability and performance of autonomous robotic systems.

2. RESEARCH METHOD

The Explicit Decoupled Group (EDG) iterative scheme, also known as the half-sweep block technique, was first proposed by Abdullah [22] for efficiently solving two-dimensional Poisson equations. Since its inception, this approach has found extensive application and addressed partial differential equations (PDEs) across diverse domains in science and engineering [24], [25], [26], [27], [28]. Its computational efficiency, achieved by reducing the number of actively computed node points per iteration, has made it particularly useful for large-scale numerical simulations. Over time, modified versions of the EDG method have been developed, further optimizing its performance and extending its applications to areas such as fluid dynamics, electromagnetic simulations, and robotic path planning.

In the skewed block approach, each iteration involves the computation of only half the nodes within the mesh domain. In contrast, the Explicit Group (EG) scheme [15], which operates as a full-sweep technique, evaluates all node points in the region per iteration. Fig. 1 illustrates the EG concept, while Fig. 2 demonstrates the EDG schemes within a 10×10 mesh. The Laplacian potentials are determined by first evaluating the red nodes, followed by the black nodes, in a repetitive manner until convergence is achieved. The convergence criteria in this study are based on a predefined threshold, ensuring numerical stability. Any remaining residual nodes (white nodes in Fig. 2) are computed separately using a direct method [15], [22], [29]. By employing the skewed block technique, the computational complexity of each iterative step is effectively halved, significantly improving efficiency. Moreover, block iterative methods compute four Laplacian potentials at once per iteration, further accelerating convergence. As shown in Figs. 1 and 2, specific sets of one- and two-node points are designated for processing nodal values near the boundary, optimizing overall computational performance.

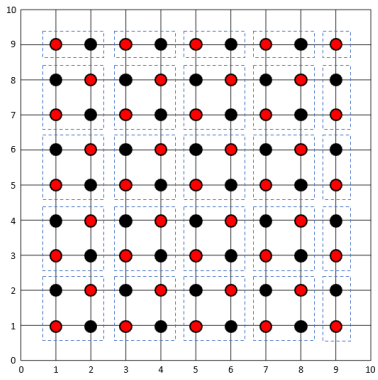


Figure 1. The Computational Particles of EDG for Full-Sweep Iterations

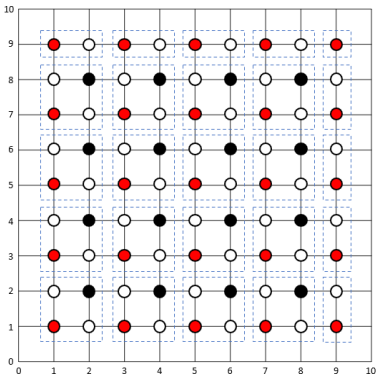


Figure 2. The Computational Particles of EDG for Half-Sweep Iterations

The half-sweep approach for robot path planning, first introduced by Saudi and Sulaiman [30], offers a major advantage by reducing the number of processed points per iteration, resulting in a simplified and computationally efficient framework. The stencil or structural patterns of both full-sweep (standard iteration) and half-sweep (rotated iteration) techniques are depicted in Figs. 3 and 4, respectively. In these diagrams, h represents the length within adjacent node points in each direction. The corresponding structural patterns/arrangements for both iteration strategies can be represented in matrix form as follows:

$$\nabla^2 U = \frac{1}{h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } \nabla^2 U = \frac{1}{2h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

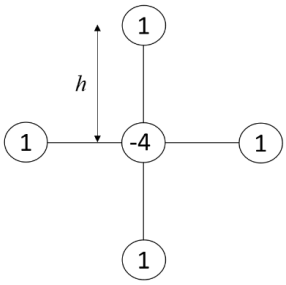


Figure 3. The Computational Patterns for Standard Cases

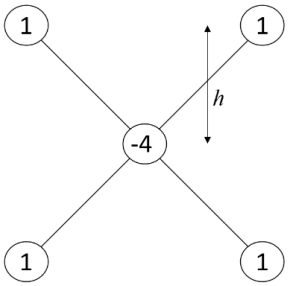


Figure 4. The Computational Patterns for Rotated Cases

2.1 Five-Point Finite Difference Approximation

In a given computational domain, a harmonic function satisfies Laplace's equation, viewed as:

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \tag{1}$$

where U represents the potential field, and the domain is defined in a Cartesian coordinate system with specified dimensions.

For the two-dimensional Laplace's equation as stated in Eq. (1), a widely adopted discretization method is the standard five-point (5P) finite difference scheme. This technique is based on the second-order central difference approximation and can be expressed as:

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j} = 0. \tag{2}$$

For Laplace's Eq. (2), the computational approach primarily involves iteratively updating each nodal value by averaging the values of its four neighbouring nodes. The nodal values at internal and external boundaries, obstacles, and the designated target point remain fixed throughout the computation to ensure proper constraints for pathfinding. Rather than applying the EDG approach to all points in the computational domain, it is instead implemented on a reduced subset of points, significantly minimizing the computational workload. As illustrated in Fig. 2, the EDG iteration process is executed exclusively on pairs of red-black

nodes per block, in contrast to the full-sweep technique shown in Fig. 1, which iterates over all red-black nodes in the domain.

The EDG approximation equation is constructed based on an intersect-alignment operator, which is effectively achieved by rotating 45° of the *i-j* plane axis in a clockwise direction. This transformation results in a rotated or skewed 5P approximation formula, mathematically represented as:

$$U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - 4U_{i,j} = 0. \tag{3}$$

Incorporating Eqs. (2) and (3) into Laplace's equation, as defined by Eq. (1), results in a linear algebraic system. This system, characterized by its large and sparse matrix structure, is expressed as:

$$Au = b. \tag{4}$$

In this context, A represents a known coefficient matrix, while b denotes a known vector. The solution vector coefficient, u, is the unknown to be determined. Given that the matrix of Eq. (4) is both large and sparse, iterative solvers are suitable for obtaining a solution, utilizing either the conventional point-based or block-based methods [22].

To introduce the block method clearly, it is helpful to first outline the fundamental idea behind the point approach. Based on the second-order central finite difference from Eqs. (2) and (3), the point-based Gauss-Seidel iterative methods under full-sweep (FS) and half-sweep (HS) conditions are reformulated and specified as follows

$$U_{i,j}^{(k+1)} = \frac{1}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k)}\right], \tag{5}$$

$$U_{i,j}^{(k+1)} = \frac{1}{4}\left[U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right]. \tag{6}$$

By incorporating a weighted parameter, the point-based Successive Over-Relaxation (SOR) iterative approach for both FS and HS scenarios is expressed separately as follows:

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}, \tag{7}$$

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}. \tag{8}$$

The optimal relaxation parameter is typically selected within the interval of $1 \leq \omega < 2$ [31]. When this parameter is set to 1, the method simplifies to the traditional Gauss-Seidel approach. While the SOR and Gauss-Seidel techniques share structural similarities, the primary distinction lies in the inclusion of a scaling factor in the SOR method to further reduce approximation errors.

The Accelerated Over-Relaxation (AOR) technique extends this concept by introducing an additional relaxation parameter, denoted as r in this study, alongside the standard SOR weighting factor. These two parameters are employed to construct iterative schemes aimed at accelerating convergence. AOR is a straightforward yet effective scheme, particularly for larger linear systems, initially proposed by Hadjidimos [31]. The point-based AOR methods for both standard and skewed formulations, respectively, are presented as:

$$U_{i,j}^{(k+1)} = \frac{r}{4}\left[U_{i-1,j}^{(k+1)} - U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right] + \frac{\omega}{4}\left[U_{i-1,j}^{(k)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} + U_{i,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}, \tag{9}$$

$$U_{i,j}^{(k+1)} = \frac{r}{4}\left[U_{i-1,j-1}^{(k+1)} - U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right] + \frac{\omega}{4}\left[U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}. \tag{10}$$

Hadjidimos suggested that the r parameter is generally selected to be close to the corresponding weighted relaxation parameter ω. It was also emphasized that the lack of precise optimal values for these parameters does not significantly impede the ability to minimize the iteration numbers essential for convergence.

2.2 Red-Black Scheme

The red–black ordering strategy operates on the fundamental principle of computing iterations layer by layer. In the formulation of full- and half-sweep cases within the mesh grid of the configuration environment, computations are performed on red nodes first, followed by black nodes. As illustrated in Figs. 1 and 2, the computational grid for the modified iterative variants adopts the red–black ordering technique across all sweep approaches.

The Modified Successive Over-Relaxation (MSOR) method is an enhancement of the standard SOR iterative technique. This modification introduces two distinct weighted parameters, denoted as ω and ω', which corresponds to the red–black ordering implementation. Specifically, the relaxation parameters ω and ω' are applied to the red and black equations, respectively. A well-optimized choice of these parameters can significantly accelerate convergence and enhance computational efficiency. Notably, when $\omega = \omega' = 1$, the MSOR method reduces to the Gauss-Seidel approach. Conversely, setting $\omega = \omega'$ aligns the MSOR scheme with the red–black ordering structure of the original SOR method.

Applying the red–black ordering strategy in full- and half-sweep SOR methods, namely, Full-Sweep MSOR (FSMSOR) and Half-Sweep MSOR (HSMSOR), can further improve convergence speed. The FSMSOR formulation is mathematically represented by the red nodes as

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j}^{(k)} - U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} - U_{i,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)},$$

while black nodes are given as

$$U_{i,j}^{(k+1)} = \frac{\omega'}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k+1)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k+1)}\right] + (1-\omega')U_{i,j}^{(k)}. \tag{11}$$

Next, the formulation of the HSMSOR technique can be broadly described for red nodes as

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}, \tag{12}$$

and for black nodes as

$$U_{i,j}^{(k+1)} = \frac{\omega'}{4}\left[U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j+1}^{(k+1)}\right] + (1-\omega')U_{i,j}^{(k)}. \tag{13}$$

As an extension of the AOR method, the Modified Accelerated Over-Relaxation (MAOR) technique incorporates various parameters that align with the row blocks of the matrix A in Eq. (4), allowing it to generalize the extrapolation used in the Jacobi or Modified SOR approaches through specific configurations of acceleration and relaxation matrices. Unlike conventional approaches, the MAOR iteration scheme employs a red–black ordering strategy, utilizing three distinct weighted parameters, i.e. r, ω and ω', each defined within a specific range [31], [32]. Optimizing the convergence behaviour of the method is significantly influenced by these parameters.

Once again, incorporating the red–black ordering strategy into the Full-Sweep and Half-Sweep AOR methods, referred to as Full-Sweep MAOR (FSMAOR) and Half-Sweep MAOR (HSMAOR), can significantly enhance convergence speed. The mathematical formulation of FSMAOR for red nodes is expressed as follows

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j}^{(k)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} + U_{i,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}, \tag{14}$$

and black nodes are given as

$$\begin{aligned} U_{i,j}^{(k+1)} = {} & \frac{r}{4}\left[U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right] + \frac{r}{4}\left[U_{i+1,j}^{(k+1)} - U_{i+1,j}^{(k)} + U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)}\right] \\ & + \frac{\omega'}{4}\left[U_{i-1,j}^{(k)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} + U_{i,j+1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}. \end{aligned} \tag{15}$$

Subsequently, the HSMAOR technique formulation can be generally described for red nodes as

$$U_{i,j}^{(k+1)} = \frac{\omega}{4}\left[U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)} + U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)}\right] + (1-\omega)U_{i,j}^{(k)}, \tag{16}$$

and for black nodes as

$$U_{i,j}^{(k+1)} = \frac{r}{4}\left[U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)} + U_{i-1,j-1}^{(k+1)} - U_{i-1,j-1}^{(k)}\right] + \frac{r}{4}\left[U_{i+1,j+1}^{(k+1)} - U_{i+1,j+1}^{(k)} + U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right] + \frac{\omega'}{4}\left[U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)} + U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)}\right] + (1-\omega')U_{i,j}^{(k)}. \tag{17}$$

2.3 Explicit Group Scheme

As previously described, the block iterative method computes four Laplacian potentials in each iteration. Fig. 1 shows that all EG-based iterative formulations simultaneously process a class of four nodes during each iteration step. Fundamentally, the EG scheme operates on a small set of points and is derived from the standard 5P finite difference approximation. By analysing Eq. (2) along with the point-based SOR method in Eq. (7), the corresponding block-based SOR formulation [17], [22] can be expressed as follows

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}, \tag{18}$$

for

$$S_1 = U_{i-1,j} + U_{i,j-1}, \qquad S_2 = U_{i+2,j} + U_{i+1,j-1}, \tag{19}$$
$$S_3 = U_{i-1,j+1} + U_{i,j+2}, \qquad S_4 = U_{i+2,j+1} + U_{i+1,j+2}.$$

This scheme similarly produces a linear system analogous to that in Eq. (4). After computing the matrix inverse, the scheme is then reformulated into its corresponding coefficient form, as shown below.

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{24} \begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix}, \tag{20}$$

with

$$S_a = 2(S_2 + S_3) + S_1 + S_4, \tag{21}$$
$$S_b = 2(S_1 + S_4) + S_2 + S_3.$$

Based upon the concept of point-based SOR through the 5P finite difference approximation, the EGSOR scheme for Eq. (20) is expressed as:

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{24} \begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix} + (1-\omega) \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{22}$$

Additionally, by analysing Eq. (2) alongside the point-based AOR approximation in Eq. (9), the block-based AOR technique can be constructed as follows [17], [23], [32]:

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \\ U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}, \tag{23}$$

with

$$S_1 = r\left(U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right) + \omega\left(U_{i-1,j}^{(k)} + U_{i,j-1}^{(k)}\right), \tag{24}$$
$$S_2 = r\left(U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right) + \omega\left(U_{i+1,j-1}^{(k)} + U_{i+2,j}^{(k)}\right),$$

$$S_3 = r\left(U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)}\right) + \omega\left(U_{i-1,j+1}^{(k)} + U_{i,j+2}^{(k)}\right),$$

$$S_4 = \omega\left(U_{i+2,j+1}^{(k)} + U_{i+1,j+2}^{(k)}\right).$$

Once again, Eq. (23) can likewise be rewritten into a linear system resembling that of Eq. (4) and represented as Eq. (20). The EGAOR iterative scheme is then can also be generally expressed as Eq. (22).

Employing the red–black ordering strategy in block-based EGSOR and EGAOR, in particular EGMSOR and EGMAOR, respectively, will significantly enhance convergence speed. The EGMSOR iterative method follows a similar formulation to the EGSOR method, with the key distinction being the inclusion of an additional relaxation parameter. By incorporating the MSOR method [33] into Eqs. (18)-(22), a group of four points (4×4) was analysed, as depicted in Fig. 1. Taking into account the approximation in Eq. (2), along with Eq. (11), the block-based EGMSOR iterative scheme can be generally formulated for black nodes as [21], [32]

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega'}{24}\begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix} + (1-\omega')\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{25}$$

where every coefficient S is referring to Eqs. (19) and (21). For red nodes, the EGSOR iterative scheme is applied, as defined in Eq. (17).

Likewise, by considering the approximation in Eq. (2) and from Eqs. (15) and (16), the block-based EGMAOR iterative scheme can be generally expressed as Eq. (22) every coefficient S is referring to Eqs. (19) and (21) for red nodes. Whereas for black nodes,

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{1}{24}\begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix} + (1-\omega')\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{26}$$

with

$$S_1 = r\left(U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right) + \omega'\left(U_{i-1,j}^{(k)} + U_{i,j-1}^{(k)}\right), \tag{27}$$

$$S_2 = r\left(U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)} + U_{i,j+2}^{(k+1)} - U_{i,j+2}^{(k)}\right) + \omega'\left(U_{i+1,j-1}^{(k)} + U_{i+2,j}^{(k)}\right),$$

$$S_3 = r\left(U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)} + U_{i,j+2}^{(k+1)} - U_{i,j+2}^{(k)}\right) + \omega'\left(U_{i-1,j+1}^{(k)} + U_{i,j+2}^{(k)}\right),$$

$$S_4 = \omega\left(U_{i+2,j+1}^{(k)} - U_{i+1,j+2}^{(k)} + U_{i+1,j+2}^{(k+1)} - U_{i+1,j+2}^{(k)}\right) + \omega'\left(U_{i+2,j+1}^{(k)} + U_{i+1,j+2}^{(k)}\right).$$

and every other coefficient S is referring to Eq. (21).

3. RESULTS AND DISCUSSION

3.1 Explicit Decoupled Group Scheme

As described hereinabove, all EDG-based iterative formulations update four nodes concurrently in each iteration, as outlined in Fig. 2. The EDG method essentially functions as much as the same of EG, operating on small node clusters and originating from the standard 5P finite difference approximation. To further the development of the block-based EDGSOR method, a set of four nodes (refer to Fig. 2) is first selected to construct a (4x4) linear system equations, expressed as Eq. (23) where

$$S_1 = U_{i-1,j-1} + U_{i-1,j+1} + U_{i+1,j-1}, \tag{28}$$

$$S_2 = U_{i,j+2} + U_{i+2,j+2} + U_{i+2,j},$$

and

$$S_3 = U_{i,j-1} + U_{i+2,j-1} + U_{i+2,j+1}, \tag{29}$$
$$S_4 = U_{i-1,j} + U_{i-1,j+2} + U_{i+1,j+2}.$$

The linear system presented in Eq. (23) can be simplified as represented in Eqs. (28) and (29), and are able to be separated into two independent (2x2) linear algebraic equations as follows

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \end{bmatrix},$$
$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_3 \\ S_4 \end{bmatrix}.$$

The corresponding block-based EDGSOR iterative schemes for these equations can then be written individually, as shown below.

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{15}\begin{bmatrix} 4S_1 + S_2 \\ S_1 + 4S_2 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{30}$$
$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{15}\begin{bmatrix} 4S_3 + S_4 \\ S_3 + 4S_4 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k)}. \tag{31}$$

Similarly, to formulate the block-based EDGMSOR method, a group of four points (refer to Fig. 2) is considered to form a (4x4) system of linear algebraic equations. With the red-black strategy, the red and black nodes are computed independently. Note that the red nodes for block-based EDGMSOR are computed using the same equations that are used to compute the red nodes for block-based EDGSOR, as provided in Eqs. (28) and (29). Meanwhile, by considering the approximation in Eq. (3) and from Eqs. (23) and (30), the block-based EDGMSOR iterative scheme for black nodes can be written independently as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega'}{15}\begin{bmatrix} 4S_1 + S_2 \\ S_1 + 4S_2 \end{bmatrix} + (1-\omega')\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{32}$$
$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega'}{15}\begin{bmatrix} 4S_3 + S_4 \\ S_3 + 4S_4 \end{bmatrix} + (1-\omega')\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k)}. \tag{33}$$

Furthermore, building upon the concept introduced in EGAOR, the AOR technique is applied to the group iterative scheme derived from the rotated five-point formula in Eq. (6), leading to the development of the EDGAOR formula, which is expressed as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{r}{15}\begin{bmatrix} 4S_2 \\ S_2 \end{bmatrix} + \frac{\omega}{15}\begin{bmatrix} 4S_1 + S_3 \\ S_1 + 4S_3 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{34}$$

where

$$S_1 = U_{i-1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)}, \tag{35}$$
$$S_2 = U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1,$$
$$S_3 = U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)} + U_{i+2,j}^{(k)}.$$

The formulation of the block-based EDGMAOR iterative method follows the same structure as the block-based EDGAOR method, with the key difference being the inclusion of an additional relaxation parameter. To illustrate, consider a 4×4 group of points, as depicted in Fig. 2. By applying the approximation in Eq. (3), the block-based EDGMAOR iterative scheme can generally express the red nodes as

$$U_{i,j}^{(k+1)} = \frac{\omega}{15}[4S_1 + S_2] + (1-\omega)U_{i,j}^{(k)}, \tag{36}$$
$$U_{i+1,j+1}^{(k+1)} = \frac{\omega}{15}[S_1 + 4S_2] + (1-\omega)U_{i+1,j+1}^{(k)},$$

where

$$S_1 = U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)},$$

$$S_2 = U_{i+2,j}^{(k)} + U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)},$$

while for black nodes, it is given by

$$U_{i,j}^{(k+1)} = \frac{1}{15}[4\omega' S_1 + 4rS_2 + \omega' S_2 + rS_3] + (1-\omega')U_{i,j}^{(k)}, \tag{37}$$

$$U_{i+1,j+1}^{(k+1)} = \frac{1}{15}[\omega' S_1 + rS_2 + 4\omega' S_2 + 4rS_3] + (1-\omega')U_{i+1,j+1}^{(k)},$$

where

$$S_1 = U_{i-1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)},$$

$$S_2 = U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1,$$

$$S_3 = U_{i+2,j}^{(k)} + U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)},$$

$$S_4 = U_{i+2,j}^{(k+1)} + U_{i,j+2}^{(k+1)} + U_{i+2,j+2}^{(k+1)} - S_3.$$

At present, a general formula for obtaining the minimum value of the optimal weighted parameters is still missing. Instead, the values of r and ω' are carefully selected to be close to the corresponding value optimal relaxation factor (ω) of the standard SOR method [31]. This careful parameter selection ensures efficient convergence and improved numerical stability, making the EDGMAOR approach a computationally viable enhancement for solving Laplace's equation in robotic path planning and other engineering applications. Nonetheless, not all parameter values guarantee convergence. The optimal relaxation parameters for each full-sweep and half-sweep case vary, as they are specifically determined for each scenario. Since these values are predefined before execution, the computational complexity associated with selecting relaxation parameters remains unchanged. Table 1 presents the optimal weighted parameter values employed throughout the experiments. These values were identified through sensitivity analysis. Although the parameters remain constant for each specific grid size (N), minor variations may occur across different grid dimensions. However, these fluctuations are negligible, typically within the range of approximately 0.0001.

Table 1. Relaxation Parameters Tuning

Methods	ω	ω'	r
EGSOR	1.82	-	-
EGAOR	1.83	-	1.82
EGMSOR	1.82	1.83	-
EGMAOR	1.83	1.81	1.82
EDGSOR	1.81	-	-
EDGAOR	1.82	-	1.84
EDGMSOR	1.82	1.81	-
EDGMAOR	1.80	1.81	1.84

Thus, **Algorithm 1** provides a detailed depiction of the red-black rotated MAOR technique, which is applied to solve the two-dimensional Laplace problem as formulated in Eq. (1), using the iterative expressions defined in Eqs. (36) and (37).

Algorithm 1: EDGMAOR scheme

Input:

- Grid domain with initial potential
- Boundary conditions (goal = 0, obstacles = 1)
- Relaxation parameters associated with EDGMAOR scheme
- Maximum iterations (k) and convergence tolerance (ε)

i. Define the simulation domain with designated start and target locations.

ii. Initialize starting location $U, \varepsilon \leftarrow 10^{-15}, iteration \leftarrow 0$.

iii. For all non-occupied red nodes, compute

$$S_1 \leftarrow U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)},$$

$$S_2 \leftarrow U_{i+2,j}^{(k)} + U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)},$$

$U_{i,j}^{(k+1)} \leftarrow \frac{\omega}{15}[4S_1 + S_2] + (1-\omega)U_{i,j}^{(k)}$,

$U_{i+1,j+1}^{(k+1)} \leftarrow \frac{\omega}{15}[S_1 + 4S_2] + (1-\omega)U_{i+1,j+1}^{(k)}$,

For all non-occupied black nodes, compute

$S_1 \leftarrow U_{i-1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)}$,

$S_2 \leftarrow U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1$,

iv. $S_3 \leftarrow U_{i+2,j}^{(k)} + U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)}$,

$S_4 \leftarrow U_{i+2,j}^{(k+1)} + U_{i,j+2}^{(k+1)} + U_{i+2,j+2}^{(k+1)} - S_3$,

$U_{i,j}^{(k+1)} \leftarrow \frac{1}{15}[4\omega' S_1 + 4rS_2 + \omega' S_2 + rS_3] + (1-\omega')U_{i,j}^{(k)}$,

$U_{i+1,j+1}^{(k+1)} \leftarrow \frac{1}{15}[\omega' S_1 + rS_2 + 4\omega' S_2 + 4rS_3] + (1-\omega')U_{i+1,j+1}^{(k)}$.

Solve boundary-adjacent nodes (comprising either one or two nodes) using a direct approach, as defined in Eqs. (12) and (13).

v. Subsequently, evaluate the remaining white nodes by applying Eq. (2)

$U_{i,j}^{(k+1)} \leftarrow \frac{1}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k)}\right]$.

vi. Check for convergence, $\varepsilon \leftarrow 10^{-15}$. If not satisfied, repeat from step (iii); otherwise, proceed.

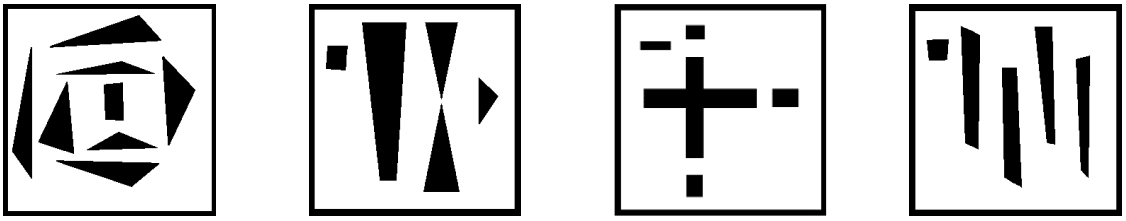
vii. Generate the final path using GDS over the computed potential field.

Output:

- Converged potential field
- Number of iterations required to achieve convergence
- CPU execution time for the iterative process
- A collision-free, smooth path from start position to target location generated via GDS on computed potential field

By leveraging the family of red-black MAOR iterative schemes, this study aims to significantly reduce computational time while ensuring efficient and accurate numerical solutions for Laplace's equation. The proposed approach presents a promising advancement in the field of computational mathematics and robot path planning, offering a more efficient alternative to traditional numerical solvers.

In line with the objectives of this research, the proposed EDGMAOR scheme was evaluated using a custom-built two-dimensional robot simulator platform [34]. The simulations were executed within a block mesh grid structure (refer to Fig. 2), where the rotated red-black node update strategy enabled computations on only half of the total grid points per iteration. This approach significantly reduces computational effort by approximately 50%. Performance analysis was conducted across four distinct configuration environments, illustrated in Fig. 5 and adapted from the layout described by Shiang [35]. Each environment was tested with five mesh grid resolutions, i.e., 300×300, 600×600, 900×900, 1200×1200, and 1800×1800. The algorithm's performance was measured using three primary metrics: number of iterations required to meet convergence criteria, CPU execution time (measured in seconds) to compute the temperature field, and the success rate in generating a valid path. CPU time and iteration counts were recorded once the solution satisfied the predefined convergence tolerance. The GDS mechanism then employed the obtained Laplacian potential values to generate a feasible trajectory from the start point to the designated goal location.



Zone 1 Zone 2 Zone 3 Zone 4

Figure 5. Four Configuration Environments are Relatively Simple to Navigate

In the designated environments, obstacles of varying sizes and shapes are strategically placed to simulate realistic navigation challenges. To model the harmonic potential field, specific thermal conditions were applied to different regions of the environment:

1. Obstacles and walls were assigned high potential values (representing maximum temperature) to simulate barriers.
2. The target point is set at a very low potential value (minimum temperature), representing an attractive goal.
3. The starting point is left without an initial temperature value, allowing the algorithm to dynamically determine its temperature distribution.
4. All remaining grid points are initialized to a neutral state with a zero-temperature value.

The simulations are planned to run on a system equipped with a 2.50 GHz processor and 8 GB RAM. The iterative process continued until the stopping criterion was met, ensuring accurate computation of temperature values at every point. The simulation terminated when there was no further change in potential values between consecutive iterations, meaning the difference between the current iteration (k) and the next iteration $(k+1)$ was extremely small, or when the values converged to a threshold of 10^{-15}. This high level of accuracy was essential to ensure the stability of the solution, preventing any critical point slopes from disrupting the descending gradient patterns, which are crucial for maintaining a smooth and reliable path-planning process.

The comparison of eight iterative schemes evaluated in this study is summarized in Table 2, which reports both the number of iterations (k) and CPU execution time (t) for each method. The results clearly indicate that the block (EG) and rotated block (EDG) schemes outperform their respective generalization techniques, with the MAOR variant exhibiting exceptional performance. Notably, the rotated block approach significantly reduces computational complexity, cutting it by nearly 50% compared to the HS process, since it computes only half of the total mesh grid nodes in each iteration. As a result, the rotated schemes consistently surpass the performance of conventional block schemes that rely on the FS approach. Both the size of the mesh grid and the spatial configuration of the region critically influence the required number of iterations and execution time. As expected, larger grid sizes entail more iterations and longer computational time.

The improvement ratio of each over-relaxation technique varies depending on the grid size, ranging from 300×300 to 1800×1800. The formula to get the percentage, in general, for each comparison approach, for example, is given as

$$\frac{EGMSOR - EGMAOR}{EGMSOR} \times 100.$$

In terms of iteration count, on average:

1. EGAOR reduced iterations by 6–12% compared to EGSOR, while EDGAOR is reduced to 19–30% compared to EDGSOR.
2. EGMAOR further decreased iterations by 6–10% relative to EGMSOR, while EDGMAOR is reduced to 13–20% compared to EDGMSOR.

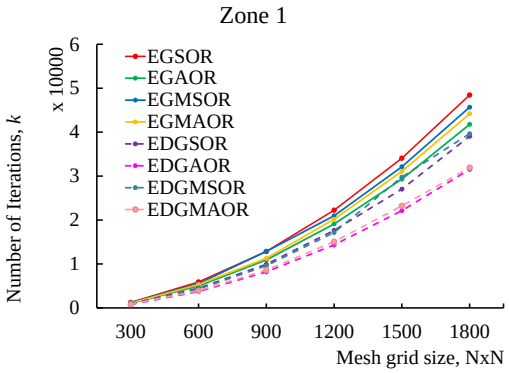
Regarding the CPU execution time:

1. EGAOR achieved a 5–8% reduction relative to EGSOR, and 14–22% compared of EDGAOR to EDGSOR.
2. EGMAOR lowered execution time by 1–5% compared to EGMSOR and 16–24% relative to EDGMSOR from EDGMAOR.

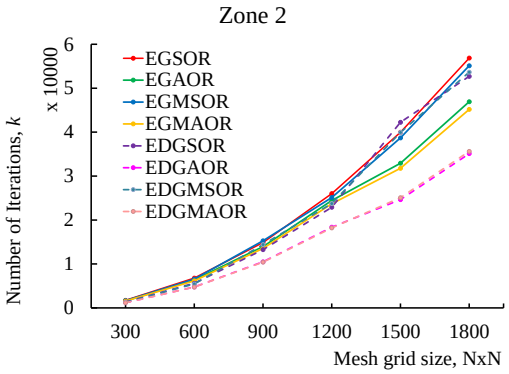
These results are noteworthy, as previous studies [35], [36], [37] primarily conducted experiments on much smaller grid sizes (e.g., 50×50 and 70×70 meshes). Figs. 6 and 7 provide a graphical representation of Table 2, illustrating the number of iterations and CPU time, respectively. The graphs further reinforce that the EDGMAOR method consistently delivers optimal performance compared to its predecessors. This finding confirms that EDGMAOR is the fastest and most efficient method among all tested approaches, making it a highly effective solution for solving pathfinding problems in autonomous robot navigation.

Table 2. Number of iterations, *k*, and CPU execution time, *t*, for the proposed iterative schemes. N × N is the size of the grid mesh, e.g., N = 300

		N x N											
	Methods	**300**		**600**		**900**		**1200**		**1500**		**1800**	
		k	***t***	***k***	***t***	***k***	***t***	***k***	***t***	***k***	***t***	***k***	***t***
Zone 1	EGSOR	1258	6.88	5899	163.72	12844	871.66	22227	2694.80	34055	6286.69	48446	12675.73
	EGAOR	1042	6.05	4994	137.87	10928	751.78	19107	2442.66	29306	5551.02	41775	10459.68
	EGMSOR	1182	6.74	5552	165.65	12871	983.56	20964	2886.75	32123	6505.53	45685	12606.78
	EGMAOR	1079	6.66	5305	169.96	11289	880.50	20190	2987.43	31069	6355.96	44178	13361.89
	EDGSOR	953	4.34	4495	110.88	9916	603.41	17653	1947.12	27037	3878.32	39060	8991.94
	EDGAOR	766	3.67	3730	95.08	8200	498.19	14338	1599.91	22141	3336.68	31517	7034.19
	EDGMSOR	898	4.73	4283	117.61	9585	631.16	17161	2105.56	29759	5011.41	39640	10315.22
	EDGMAOR	816	4.36	3921	119.39	8619	621.90	15069	2061.04	23219	4209.04	31908	9268.73
Zone 2	EGSOR	1729	7.67	6782	199.59	14874	1009.48	26007	2827.46	39968	6925.80	56858	14336.95
	EGAOR	1610	8.25	6368	185.36	13953	926.49	24429	3003.98	32926	5909.85	46923	12802.89
	EGMSOR	1657	7.99	6559	205.17	15276	1150.19	25122	3170.90	38693	7117.38	55107	15134.05
	EGMAOR	1557	8.25	6154	205.08	13543	1022.43	23692	3299.85	31778	6561.31	45179	14243.83
	EDGSOR	1324	4.70	5599	142.64	13252	787.53	22898	2386.46	42248	6170.35	52695	9876.12
	EDGAOR	1188	4.17	4767	122.85	10490	624.48	18371	1930.06	24644	3900.10	35105	7960.71
	EDGMSOR	1252	5.06	5456	154.59	14417	972.91	23756	2801.61	39926	5840.88	53654	10244.72
	EDGMAOR	1175	5.25	4733	146.83	10416	734.06	18206	2329.19	25083	4677.88	35627	9947.37
Zone 3	EGSOR	2666	13.24	11076	315.87	24519	1602.81	42897	5591.93	65977	12331.17	100842	26171.60
	EGAOR	2480	13.83	10389	301.27	22995	1633.35	40322	5261.60	62423	11975.63	89182	23616.21
	EGMSOR	2552	12.64	10709	334.98	23608	1828.16	41526	5907.46	63606	12857.61	94368	28069.16
	EGMAOR	2370	12.28	10002	327.72	21996	1801.35	38814	5831.19	59632	12342.99	85405	26471.93
	EDGSOR	2035	8.00	10243	261.74	24864	1529.74	45435	5017.92	70627	10771.29	101290	22145.88
	EDGAOR	1835	7.67	7741	203.61	17131	1072.61	30027	3437.95	46055	7612.81	65815	15280.15
	EDGMSOR	1989	8.74	10238	285.14	23681	1686.18	44270	5541.21	70173	11977.00	101844	25644.81
	EDGMAOR	1820	8.23	7663	237.03	16981	1291.76	29842	4156.98	45930	8796.85	65390	18646.99
Zone 4	EGSOR	1629	7.80	6487	187.33	14194	990.20	24913	2979.14	38195	6919.25	54508	13843.15
	EGAOR	1392	7.56	5648	167.65	12367	891.51	21724	2609.11	33518	6139.29	48120	12833.82
	EGMSOR	1568	6.94	6255	196.36	13701	1041.04	24085	3205.68	36925	7195.51	52596	14985.11
	EGMAOR	1317	6.41	5433	170.40	12863	1059.32	22541	3036.23	31067	6535.92	44404	13689.79
	EDGSOR	1238	4.33	5958	152.73	13501	828.24	23920	2527.82	38175	5948.26	54477	12553.41
	EDGAOR	1116	4.53	4223	109.96	9369	615.38	16313	1725.41	25023	3991.87	25023	4079.41
	EDGMSOR	1265	4.83	5702	162.47	13079	929.50	23922	2971.32	37454	6534.05	56018	14524.93
	EDGMAOR	1023	4.70	4214	133.39	9637	745.72	16368	2110.00	24615	4708.25	24615	4747.41



(a)



(b)

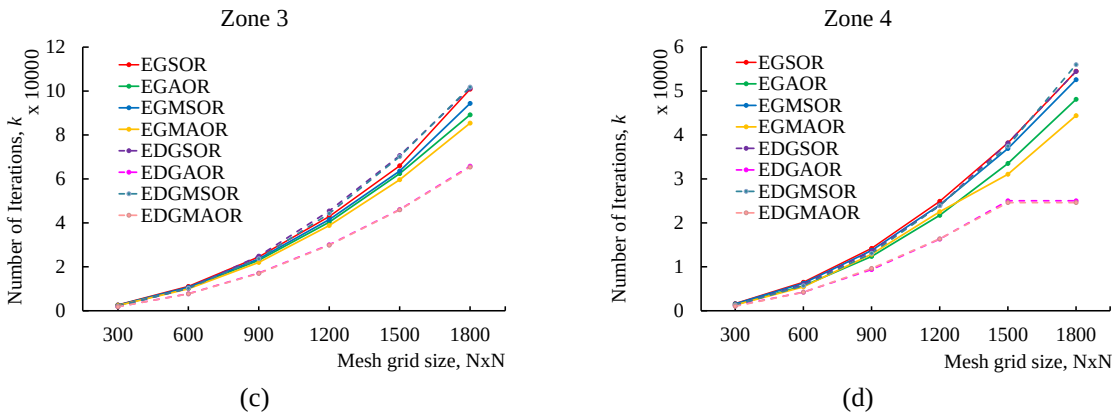


Figure 6. Graph performance with respect to the number of iterations (a) Zone 1, (b) Zone 2, (c) Zone 3, and (d) Zone 4

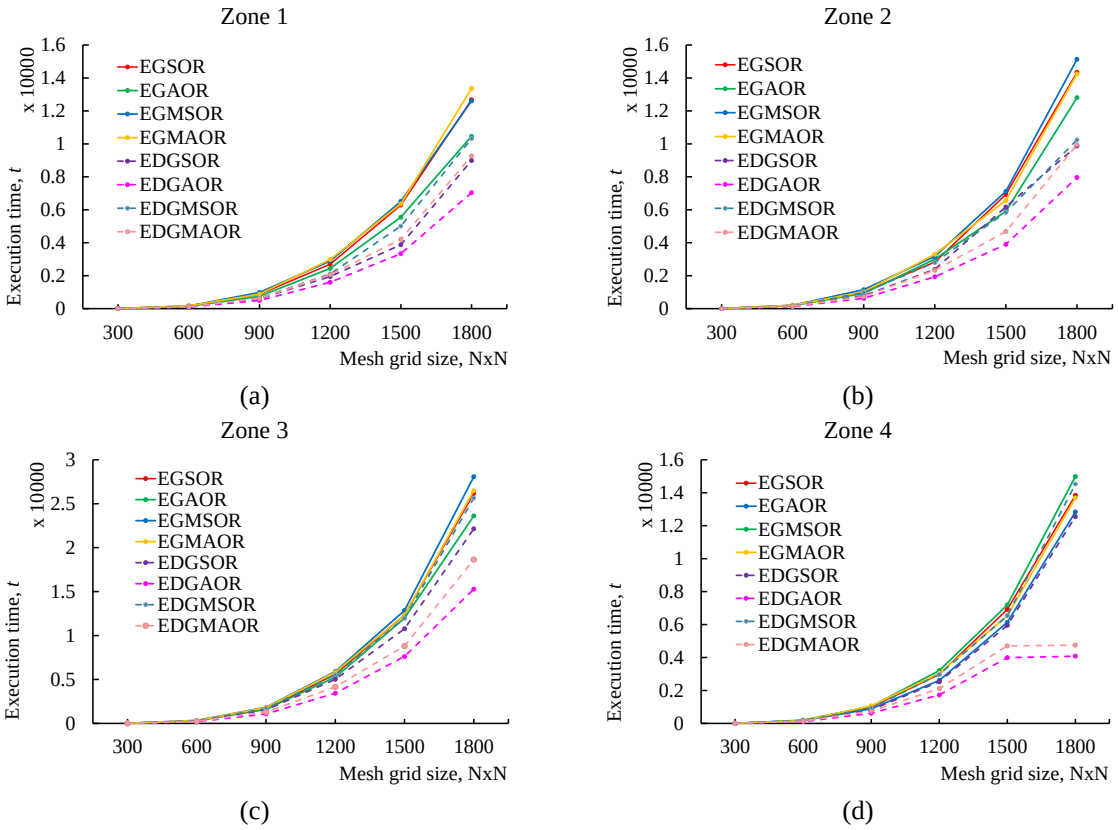


Figure 7. Graph performance with respect to the cpu execution time (a) Zone 1, (b) Zone 2, (c) Zone 3, and (d) Zone 4

The numerical results demonstrate that the proposed EDGMAOR method delivers substantial improvements over conventional iterative techniques, achieving faster convergence (computation time), lower iteration counts, and enhanced computational efficiency (overall path smoothness) while preserving accuracy and stability in path generation. As illustrated in Figs. 6 and 7, which present iteration count and CPU execution time, respectively, consistent performance trends are observed across all zones and environment configurations, indicating robustness independent of starting positions. Despite these improvements, the efficiency of the method remains influenced by the selection of relaxation parameters. In this study, optimal values were determined through sensitivity analysis and applied uniformly across grid sizes. Future work will therefore investigate adaptive parameter-tuning strategies to further enhance convergence performance, particularly for large-scale or real-time applications.

3.2 Computational Complexity

This section presents an analysis of the computational complexity associated with each iterative technique considered in this study. Each arithmetic operation (addition and multiplication) is assumed to

consume a single unit of computation time. However, the convergence test does not account for the arithmetic operations involved in the GDS algorithm's path-tracing process. A summary of the total arithmetic operations for each method examined is provided in Table 3.

In theory, a reduction in computational complexity should lead to fewer iterations, ultimately reducing CPU time. Although the AOR method family involves more arithmetic operations due to the presence of additional weighted parameters, it still achieves faster convergence compared to the SOR method family [32], [38]. Meanwhile, as the remaining points are computed in a single iteration, their contribution to the total computational complexity is considered negligible, meaning they have minimal impact on the overall computation.

Table 3. Number of Arithmetic Operations per Iteration for Block Over-Relaxation Techniques and its Modified Variants Methods

Methods	ADD/SUB	MUL/DIV
EGSOR/ EGMSOR	$18\left(\frac{N}{2}-1\right)^2+4(2N-4)+4$	$10\left(\frac{N}{2}-1\right)^2+2(2N-4)+2$
EDGSOR/ EDGMSOR	$8\left(\frac{N}{2}-1\right)^2+4(N-2)+4$	$4\left(\frac{N}{2}-1\right)^2+2(N-4)+2$
EGAOR/EGMAOR	$26\left(\frac{N}{2}-1\right)^2+4(2N-4)+4$	$17\left(\frac{N}{2}-1\right)^2+2(2N-4)+2$
EDGAOR/EDGMAOR	$13\left(\frac{N}{2}-1\right)^2+4(N-2)+4$	$6\left(\frac{N}{2}-1\right)^2+2(N-2)+2$

The path-planning process in this experiment begins with the identification of the starting point and the goal location. Subsequently, the optimal parameters are determined in accordance with the respective iterative scheme applied. Once the harmonic potential field values are computed for the configuration domain using the selected algorithm, the path construction is carried out via the steepest descent search algorithm, also referred to as the gradient descent search (GDS) technique. This method initiates from the defined starting point and proceeds by following the direction of the negative gradient, iteratively choosing adjacent nodes with decreasing potential values until it reaches the target, which holds the minimum potential. The algorithm effectively generates a smooth trajectory while dynamically avoiding any obstacles in the environment. Fig. 8 depicts the outcome of this process, illustrating the computed paths within a predefined, stationary environment. In the simulation experiment, the initial position is marked by a square (green box), and the target location is represented by a circle (red dot). As seen in Fig. 8, all start points from various regions successfully reach the target while efficiently navigating around any present walls or obstructions. The speed of path generations is notably high due to the reliance solely on gradient evaluations of the precomputed Laplacian potential [37]. Furthermore, the configuration spaces used in this study are comparatively simpler than those described in [35], allowing for more straightforward interpretation of the path trajectories. The results confirm that all starting points successfully reach the designated target point, effectively bypassing multiple obstacles within the given configuration space. This demonstrates the reliability and effectiveness of the proposed method in handling complex pathfinding challenges for autonomous mobile robots.

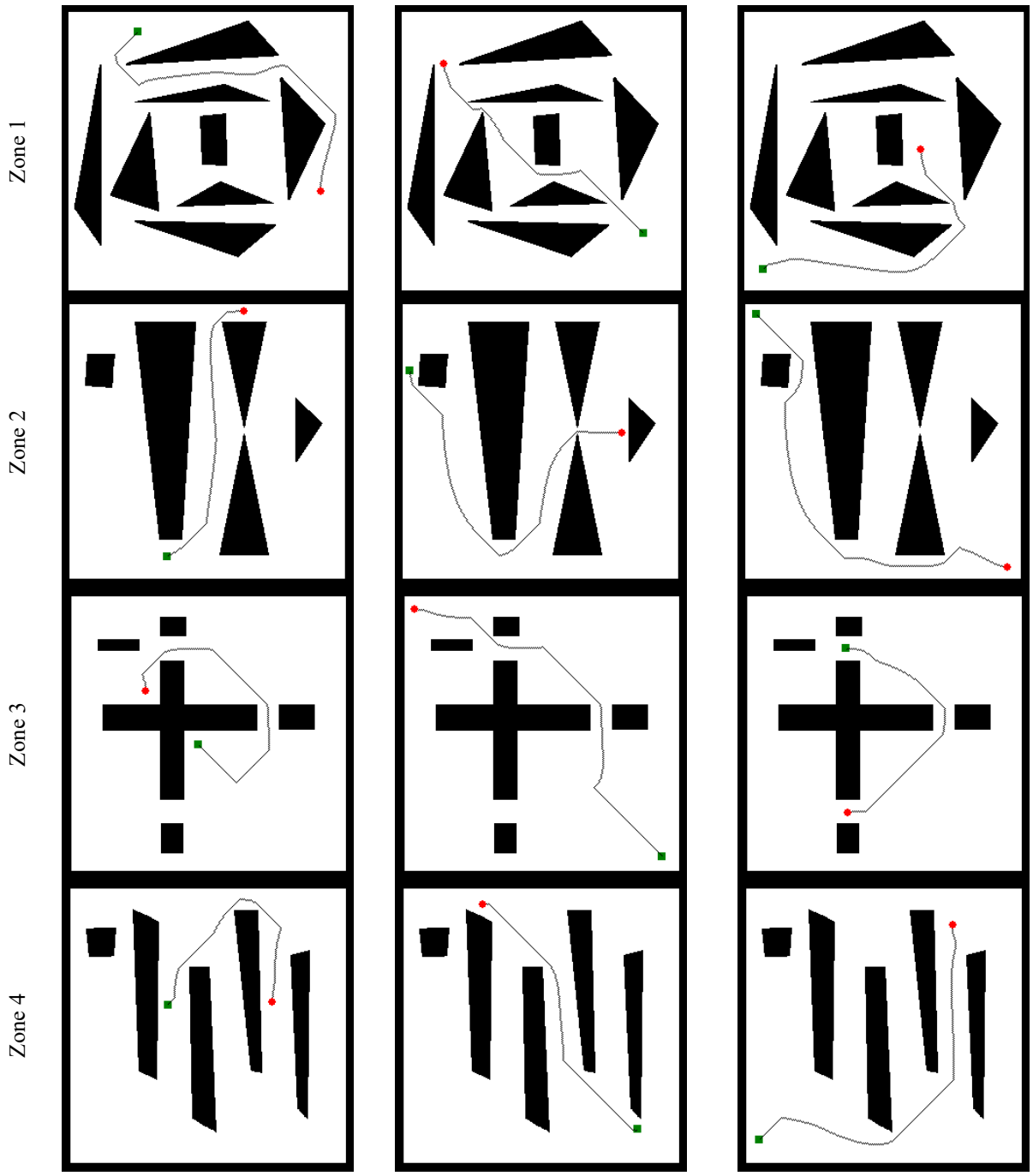


Figure 8. Structured Path Generation in a Defined Static Environment

The numerical results indicate that the proposed EDGMAOR method consistently performs better than conventional iterative techniques across all tested configurations, including environments with different obstacle densities and grid sizes. As detailed in Section 3.1, the method achieves faster convergence with fewer iterations and reduced computational time, demonstrating improved efficiency in solving Laplace-based path-planning problems. Furthermore, the generated paths, as illustrated in Fig. 8, are both accurate and smooth, reflecting the stability and reliability of the computed harmonic potential field. The reduction in computational cost is largely attributed to the rotated block framework, which limits the number of processed nodes while preserving solution accuracy. Overall, these results confirm the robustness and practicality of the EDGMAOR scheme for efficient and reliable autonomous navigation in structured indoor environments.

4. CONCLUSION

In summary, this study presents the Explicit Decoupled Group Modified Accelerated Over-Relaxation (EDGMAOR) iterative scheme as an efficient approach for solving Laplace-based path-planning problems in structured indoor environments. The proposed method integrates the advantages of the decoupled group framework with accelerated relaxation strategies to enhance convergence performance. The simulation results demonstrate that EDGMAOR consistently outperforms conventional iterative techniques, including Successive Over-Relaxation (SOR-) and Accelerated Over-Relaxation (AOR-based) methods, across various grid sizes and obstacle configurations. Quantitatively, the EDGMAOR scheme achieves an average reduction of approximately 13–20% in iteration count and 16–24% in CPU execution time, indicating significant improvements in convergence rate and computational efficiency. These gains are achieved while maintaining high accuracy and stability in the generated paths, confirming the reliability of the harmonic potential field for navigation. The consistent performance observed across different zones further highlights the robustness of the proposed method, regardless of variations in starting positions or environment configurations. Notably, the efficiency of the proposed EDGMAOR method remains unaffected by an increasing number of obstacles. In fact, the computation time is further reduced, as obstacle regions are automatically excluded from the iterative process. Overall, the findings establish EDGMAOR as a fast, efficient, and reliable technique for autonomous path planning. A major contribution of this work is the integration of the red-black AOR scheme family into mobile robot pathfinding, marking a novel application of these techniques in the field. While this study primarily focuses on full- and half-sweep approaches, future research could explore quarter-sweep techniques [29], [39], [40] to further enhance convergence speed and optimize computational efficiency. Future work will focus on incorporating adaptive parameter-tuning strategies to further enhance performance, particularly for large-scale or dynamic environments.

Author Contributions

A'Qilah Ahmad Dahalan: Conceptualization, Data Curation, Investigation, Methodology, Validation, Visualization, Writing - Original Draft. Rupal Srivastava: Data Curation, Project Administration, Writing - Review and Editing. Ruzanna Mat Jusoh: Formal Analysis, Investigation, Validation, Writing - Review and Editing. Azali Saudi: Data Curation, Methodology, Software, Supervision, Validation, Writing - Review andEditing. Jumat Sulaiman: Data Curation, Resources, Supervision. All authors discussed the results and contributed to the final manuscript.

Funding Statement

This research was financially supported by Universiti Pertahanan Nasional Malaysia scheme.

Acknowledgment

The authors would like to express their gratitude to the National Defence University of Malaysia for funding the publication of this article.

Declarations

The authors declare that there is no conflict of interest regarding the publication of this study.

Declaration of Generative AI and AI-assisted technologies

The authors declare that no generative AI or AI-assisted technologies were used in the preparation of this manuscript, including for writing, editing, data analysis, or the creation of tables and figures.

REFERENCES

[1] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "DEEP REINFORCEMENT LEARNING FOR ROBOTICS: A SURVEY OF REAL-WORLD SUCCESSES," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 8, no. 1, pp. 153–188, May 2025. doi: https://doi.org/10.1146/annurev-control-030323-022510

[2] D. Fernández Llorca *et al.*, "TESTING AUTONOMOUS VEHICLES AND AI: PERSPECTIVES AND CHALLENGES FROM CYBERSECURITY, TRANSPARENCY, ROBUSTNESS AND FAIRNESS," *European Transport Research Review*, vol. 17, no. 1, p. 38, Jul. 2025. doi: https://doi.org/10.1186/s12544-025-00732-x

[3] M. Lan, S. Lai, T. H. Lee, and B. M. Chen, "A SURVEY OF MOTION AND TASK PLANNING TECHNIQUES FOR UNMANNED MULTICOPTER SYSTEMS," *Unmanned Systems*, vol. 09, no. 02, pp. 165–198, Apr. 2021. doi: https://doi.org/10.1142/S2301385021500151

[4] L. Zhang *et al.*, "MOTION PLANNING FOR ROBOTICS: A REVIEW FOR SAMPLING-BASED PLANNERS," *Biomimetic Intelligence and Robotics*, vol. 5, no. 1, p. 100207, Mar. 2025. doi: https://doi.org/10.1016/j.birob.2024.100207

[5] B. Sundaralingam *et al.*, "CUROBO: PARALLELIZED COLLISION-FREE ROBOT MOTION GENERATION," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2023, pp. 8112–8119. doi: https://doi.org/10.1109/ICRA48891.2023.10160765

[6] W. Xia, Y. Lu, W. Xu, and X. Xu, "DEEP REINFORCEMENT LEARNING BASED PROACTIVE DYNAMIC OBSTACLE AVOIDANCE FOR SAFE HUMAN-ROBOT COLLABORATION," *Manuf. Lett.*, vol. 41, pp. 1246–1256, Oct. 2024. doi: https://doi.org/10.1016/j.mfglet.2024.09.151

[7] K. Katona, H. A. Neamah, and P. Korondi, "OBSTACLE AVOIDANCE AND PATH PLANNING METHODS FOR AUTONOMOUS NAVIGATION OF MOBILE ROBOT," *Sensors*, vol. 24, no. 11, p. 3573, Jun. 2024. doi: https://doi.org/10.3390/s24113573

[8] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "PATH PLANNING FOR AUTONOMOUS MOBILE ROBOTS: A REVIEW," *Sensors*, vol. 21, no. 23, p. 7898, Nov. 2021. doi: https://doi.org/10.3390/s21237898

[9] M. Badamasi Aremu, I. K. Kabir, G. Ahmed, and S. El-Ferik, "AUTONOMOUS MOBILE ROBOT PATH PLANNING TECHNIQUES—A REVIEW: CLASSICAL AND HEURISTIC TECHNIQUES," *IEEE Access*, vol. 13, pp. 117999–118022, 2025. doi: https://doi.org/10.1109/ACCESS.2025.3579863

[10] R. Daily and D. M. Bevly, "HARMONIC POTENTIAL FIELD PATH PLANNING FOR HIGH SPEED VEHICLES," in *2008 American Control Conference*, IEEE, Jun. 2008, pp. 4609–4614. doi: https://doi.org/10.1109/ACC.2008.4587222

[11] Y. Liu, Y. Ren, J. Wang, L. Zhao, Q. Wang, and J. Shan, "PATH PLANNING FOR MOBILE ROBOT BASED ON IMPROVED ARTIFICIAL POTENTIAL FIELD METHOD," in *2023 China Automation Congress (CAC)*, IEEE, Nov. 2023, pp. 4757–4762. doi: https://doi.org/10.1109/CAC59555.2023.10452131

[12] H. J. Chou, P. L. Kuo, and J. S. Liu, "NUMERICAL STREAMLINE PATH PLANNING BASED ON LOG-SPACE HARMONIC POTENTIAL FUNCTION: A SIMULATION STUDY," *2017 IEEE International Conference on Real-Time Computing and Robotics, RCAR 2017*, vol. 2017-July, pp. 535–542, 2017. doi: https://doi.org/10.1109/RCAR.2017.8311918

[13] Z. Guo, H. Yang, and D. Wang, "DEVELOPMENT AND APPLICATION OF POSITIONING AND QUEUE TRANSFORMATION TECHNOLOGY FOR MULTI-SMART CARS," *Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020*, pp. 3919–3924, 2020. doi: https://doi.org/10.1109/CCDC49329.2020.9164191

[14] C. I. Connolly, "APPLICATIONS OF HARMONIC FUNCTIONS TO ROBOTICS," *IEEE International Symposium on Intelligent Control - Proceedings*, vol. 199j2-Augus, no. 7, pp. 498–502, 1992. doi: https://doi.org/10.1109/ISIC.1992.225141

[15] D. J. Evans, "GROUP EXPLICIT ITERATIVE METHODS FOR SOLVING LARGE LINEAR SYSTEMS," *Int. J. Comput. Math.*, vol. 17, no. 1, pp. 81–108, 1985. doi: https://doi.org/10.1080/00207168508803452

[16] A. Sunarto, J. Sulaiman, and J. V. L. Chew, "COMPUTATIONAL HALF-SWEEP PRECONDITIONED GAUSS--SEIDEL METHOD FOR TIME-FRACTIONAL DIFFUSION EQUATIONS," in *Fractional Order Systems and Applications in Engineering*, Elsevier, 2023, pp. 285–300. doi: https://doi.org/10.1016/B978-0-32-390953-2.00024-4

[17] A. A. Dahalan and A. Saudi, "AN ITERATIVE TECHNIQUE FOR SOLVING PATH PLANNING IN IDENTIFIED ENVIRONMENTS BY USING A SKEWED BLOCK ACCELERATED ALGORITHM," *AIMS Mathematics*, vol. 8, no. 3, pp. 5725–5744, 2022. doi: https://doi.org/10.3934/math.2023288

[18] M. U. Alibubin, J. Sulaiman, F. A. Muhiddin, and A. Sunarto, "IMPLEMENTATION OF THE KSOR METHOD FOR SOLVING ONE-DIMENSIONAL TIME-FRACTIONAL PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS WITH THE CAPUTO FINITE DIFFERENCE SCHEME TITLE OF MANUSCRIPT," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 48, no. 1, pp. 168–179, 2025. doi: https://doi.org/10.37934/araset.48.1.168179

[19] J. V. L. Chew, E. Aruchunan, and J. Sulaiman, "EFFICIENCY EVALUATION OF HALF-SWEEP NEWTON-EGSOR METHOD TO SOLVE 1D NONLINEAR POROUS MEDIUM EQUATIONS," *Studies in Systems, Decision and Control*, vol. 383, pp. 399–412, 2022. doi: https://doi.org/10.1007/978-3-030-79606-8_25

[20] E. Aruchunan, Z. Siri, M. H. B. N. Aziz, M. H. B. A. Wahab, M. S. Muthuvalu, and J. Sulaiman, "SOLUTION OF PEAK JUNCTION TEMPERATURE WITH CRANK-NICOLSON AND SOR APPROACH," *Studies in Systems, Decision and Control*, vol. 444, pp. 225–234, 2022. doi: https://doi.org/10.1007/978-3-031-04028-3_15

[21] M. K. M. Akhir, M. Othman, J. Sulaiman, Z. A. Majid, and M. Suleiman, "NUMERICAL SOLUTION OF HELMHOLTZ EQUATION USING A NEW FOUR POINT EGMSOR ITERATIVE METHOD," *Applied Mathematical Sciences*, vol. 5, no. 77–80, pp. 3991–4004, 2011.

[22] A. R. Abdullah, "THE FOUR POINT EXPLICIT DECOUPLED GROUP (EDG) METHOD : A FAST POISSON SOLVER," *Int. J. Comput. Math.*, vol. 38, no. 1–2, pp. 61–70, 1991. doi: https://doi.org/10.1080/00207169108803958

[23] N. A. Syafiq, M. Othman, N. Senu, and F. Ismail, "AN EXPERIMENTAL STUDY OF THE MODIFIED ACCELERATED OVERRELAXATION (MAOR) SCHEME ON STATIONARY HELMHOLTZ EQUATION," *J. Phys. Conf. Ser.*, vol. 1366, no. 1, p. 012093, Nov. 2019. doi: https://doi.org/10.1088/1742-6596/1366/1/012093

[24] A. A. Dahalan and A. Saudi, "AN ITERATIVE TECHNIQUE FOR SOLVING PATH PLANNING IN IDENTIFIED ENVIRONMENTS BY USING A SKEWED BLOCK ACCELERATED ALGORITHM," *AIMS Mathematics*, vol. 8, no. 3, pp. 5725–5744, 2023. doi: https://doi.org/10.3934/math.2023288

[25] A. Ali, M. Abbas, and T. Akram, "NEW GROUP ITERATIVE SCHEMES FOR SOLVING THE TWO-DIMENSIONAL ANOMALOUS FRACTIONAL SUB-DIFFUSION EQUATION," *Journal of Mathematics and Computer Science*, vol. 22, no. 2, pp. 119–127, 2020. doi: https://doi.org/10.22436/jmcs.022.02.03

[26] A. A. Dahalan, A. Saudi, and J. Sulaiman, "ENHANCING AUTONOMOUS GUIDED VEHICLES WITH RED-BLACK TOR ITERATIVE METHOD," *Mathematics*, vol. 11, no. 20, pp. 1–17, 2023 doi:. https://doi.org/10.3390/math11204393

[27] P. Agarwal, A. Sunarto, J. V. L. Chew, J. Sulaiman, and S. Momani, "NEW PRECONDITIONING AND HALF-SWEEP ACCELERATED OVERRELAXATION SOLUTION FOR FRACTIONAL DIFFERENTIAL EQUATION," *J. King Saud Univ. Sci.*, vol. 35, no. 2, p. 102461, 2023. doi: https://doi.org/10.1016/j.jksus.2022.102461

[28] G. B. Ekal, J. V. L. Chew, K. Ghazali, and J. Sulaiman, "HALF-SWEEP NEWTON-KSOR WITH WAVE VARIABLE TRANSFORMATION TO SOLVE FORCED POROUS MEDIUM EQUATIONS," *International Journal of Engineering Trends and Technology*, vol. 72, no. 2, pp. 19–28, 2024. doi: https://doi.org/10.14445/22315381/IJETT-V72I2P103

[29] M. Othman and A. R. Abdullah, "EFFICIENT FOUR POINTS MODIFIED EXPLICIT GROUP POISSON SOLVER," *Int. J. Comput. Math.*, vol. 76, no. 2, pp. 203–217, 2000. doi: https://doi.org/10.1080/00207160008805020

[30] A. Saudi and J. Sulaiman, "HALF-SWEEP GAUSS-SEIDEL (HSGS) ITERATIVE METHOD FOR ROBOT PATH PLANNING.," *The 3rd Int. Conf. on Informatics and Technology (Informatics09)*, no. February 2019, pp. 27–28, 2009.

[31] A. Hadjidimos, "ACCELERATED OVERRELAXATION METHOD," *Math. Comput.*, vol. 32, no. 141, pp. 149–157, 1978. doi: https://doi.org/10.1090/S0025-5718-1978-0483340-6

[32] A. M. Saeed and N. M. AL-harbi, "GROUP SPLITTING WITH SOR/AOR METHODS FOR SOLVING BOUNDARY VALUE PROBLEMS: A COMPUTATIONAL COMPARISON," *European Journal of Pure and Applied Mathematics*, vol. 14, no. 3, pp. 905–914, 2021. doi: https://doi.org/10.29020/nybg.ejpam.v14i3.4031

[33] D. Young, "ITERATIVE METHODS FOR SOLVING PARTIAL DIFFERENCE EQUATIONS OF ELLIPTIC TYPE," 1954. doi: https://doi.org/10.1088/1751-8113/44/8/085201

[34] A. Saudi, "ROBOT PATH PLANNING USING FAMILY OF SOR ITERATIVE METHODS WITH LAPLACIAN BEHAVIOUR-BASED CONTROL," 2015.

[35] S.-F. Chen, "COLLISION-FREE PATH PLANNING," 1997.

[36] S. Sasaki, "PRACTICAL COMPUTATIONAL TECHNIQUE FOR MOBILE ROBOT NAVIGATION," *IEEE Conference on Control Applications - Proceedings*, vol. 2, no. September, pp. 1323–1327, 1998. doi: https://doi.org/10.1109/CCA.1998.721675

[37] C. I. Connolly, J. B. Burns, and R. Weiss, "PATH PLANNING USING LAPLACE'S EQUATION," pp. 2102–2106, 1990. doi: https://doi.org/10.1109/ROBOT.1990.126315

[38] A. A. Dahalan and A. Saudi, "PATHFINDING ALGORITHM BASED ON ROTATED BLOCK AOR TECHNIQUE IN STRUCTURED ENVIRONMENT," *AIMS Mathematics*, vol. 7, no. 7, pp. 11529–11550, 2022. doi: https://doi.org/10.3934/math.2022643

[39] L. H. Ali, J. Sulaiman, and A. Saudi, "ITERATIVE METHOD FOR SOLVING NONLINEAR FREDHOLM INTEGRAL EQUATIONS USING QUARTER-SWEEP NEWTON-PKSOR METHOD," *Lecture Notes in Electrical Engineering*, vol. 983 LNEE, pp. 33–46, 2023. doi: https://doi.org/10.1007/978-981-19-8406-8_3

[40] A. A. Dahalan, A. Saudi, and J. Sulaiman, "DEVELOPMENT OF PATH OPTIMIZATION USING QUARTER-SWEEP MODIFIED SUCCESSIVE OVER-RELAXATION ITERATIVE TECHNIQUE," *Lecture Notes in Networks and Systems*, vol. 1056, pp. 411–425, 2024. doi: https://doi.org/10.1007/978-981-97-4892-1_35