AIMS Mathematics

AIMS Mathematics, 8(3): 5725–5744.
DOI: 10.3934/math.2023288
Received: 26 August 2022
Revised: 18 October 2022
Accepted: 26 October 2022
Published: 22 December 2022

http://www.aimspress.com/journal/Math

Research article

An iterative technique for solving path planning in identified environments by using a skewed block accelerated algorithm

A'qilah Ahmad Dahalan[1,2,3,*] and Azali Saudi[4]

[1] CONFIRM Centre for SMART Manufacturing, University of Limerick, Limerick, Ireland
[2] MACSI, Department of Mathematics and Statistics, University of Limerick, Limerick, Ireland
[3] Department of Mathematics, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia
[4] Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia

* **Correspondence:** Email: A.Qilah@ul.ie.

Abstract: Currently, designing path-planning concepts for autonomous robot systems remains a topic of high interest. This work applies computational analysis through a numerical approach to deal with the path-planning problem with obstacle avoidance over a robot simulation. Based on the potential field produced by Laplace's equation, the formation of a potential function throughout the simulation configuration regions is obtained. This potential field is typically employed as a guide in the global approach of robot path-planning. An extended variant of the over-relaxation technique, namely the skewed block two-parameter over relaxation (SBTOR), otherwise known as the explicit decoupled group two-parameter over relaxation method, is presented to obtain the potential field that will be used for solving the path-planning problem. Experimental results with a robot simulator are presented to demonstrate the performance of the proposed approach on computing the harmonic potential for solving the path-planning problem. In addition to successfully validating pathways generated from various locations, it is also demonstrated that SBTOR outperforms existing over-relaxation algorithms in terms of the number of iterations, as well as the execution time.

Keywords: rotated iterative scheme; numerical analysis; Laplace's equation; accelerated method; path finding; obstacle avoidance
Mathematics Subject Classification: 35A35, 65F10, 68T40

1. Introduction

Path planning is a computational problem to find a sequence of valid configurations that move the object of interest from an arbitrary source to a specific destination. The term path planning has been used widely, mostly in the robotics area. Good path planning for a mobile robot means it should be executed while avoiding walls and dodging any obstacles in between. Autonomous robot path planning is very impactful, as it is extremely beneficial not only for transportation, but also for households, pharmaceuticals, industries, manufacturing, aerospace and others. In general, the ability of a robot to navigate autonomously is a prerequisite and foundation for the development of intelligent robots. To ensure the successful completion of the many tasks that mobile robots are designed to accomplish, path-planning algorithms must be both effective and practical.

On the whole, navigation problems are divided into four main categories, i.e., localization, path planning, motion control and cognitive mapping [1]. In essence, the localization of a robot is telling where the robot is currently located, whereas path planning reveals in which direction the robot is meant to travel and what is the best route to arrive at the destination. Robot motion control gives instructions on the manner in which the robot is able to move. Cognitive mapping concerns the previous, current and next positions of the robot, as well as to what extent the robot should remember. Hence, from all of these category's functions, it is not wrong to say that path planning is a crucial issue to be solved in achieving intelligent robots. With the aid of a path-planning algorithm, the robot should be able to select and recognize the ideal route inside the given configuration region.

It is common knowledge that there are two types of navigation, i.e., global and local approaches. This work considers global navigation, where prior knowledge of the configuration region is given, which is also known as an off-line mode for path planning. The robot's ability to represents the real world and to execute the algorithm are the two key elements of global path planning. These two elements are interconnected and significantly influence one another while determining the optimal route and minimum duration for the robot to travel across the known environment [2]. The main objective of this study is to utilize the harmonic potential in simulating point-robot path-planning in the identified environment following the resemblance of heat transfer. The model of the aforementioned heat transfer problem was designed by using Laplace's equation [3]. Harmonic functions are literally the solutions to Laplace's equation, which are also called Laplacian potentials. These Laplacian potentials portray the analogy of temperature values from heat transfer in the configuration region that will be used to simulate the path generation. Heat transfer's capacity to surpass the local minima enigma is one of its most crucial characteristics, making it very encouraging for robot navigation control. The following sections provide further explanations. Numerous approaches can be used to achieve harmonic functions, but the most prevalent approaches rely on numerical techniques due to the availability of rapid processing machines, as well as on their elegance and competence in solving the problem [4–6].

Furthermore, the iterative approach could be implemented to produce the Laplacian potential by computationally solving Laplace's equation and discretizing the environment through the use of mesh grid points [7–10]. The cost of computation for iterative techniques is substantially higher than analytical techniques since the discretization process produces a sparse linear system. However, the discretization resolution determines the computational complexity. Using a finite difference approximation, the Laplace equation can be translated to a linear system. When this linear system is documented in matrix notation, the majority of the elements are zero, and the matrix size of the end product is usually large and sparse. Therefore, the iterative technique is employed to solve the linear system to prevent the higher requirement for memory storage of those large and sparse matrices. Several experiments were executed in this work to measure the competency of the skewed block accelerated iterative approach in terms of the computation of the Laplacian potential when generating robot-point paths. Fundamentally, a complete process of the path-planning construction phase in this study consists of the following steps:

```
Begin
    Step 1:  Mapping of the robot's configuration region.
    Step 2:  Formulation and modeling of the finite-difference approximation of the proposed
             iterative schemes.
    Step 3:  Algorithms of the proposed iterative schemes.
    Step 4:  Numerical simulations.
    Step 5:  Evaluation and analysis.
End.
```

2. Materials and methods

2.1. Harmonic function

Consider Laplace's partial differential equation in two dimensions:

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

where $U(x,y)$ is some unknown function of two variables. A harmonic function in the region $\Omega\subset\Re^n$ should be able to satisfy the generalization of Eq (1), as below.

$$\nabla^2 U=\sum_{i=1}^{n}\frac{\partial^2 U}{\partial x_i^2}=0, \tag{2}$$

in which ∇^2 is the n-dimensional Laplace operator (or Laplacian as it is often called), the x_i is the *i*-th coordinate in a Cartesian system and n is the dimension of the region. This work only focuses on two-dimensional problems with $x-y$ notation represented as $i-j$. The region boundary of Ω comprises internal and external walls, boundaries of obstaclesand the goal position. According to [7], the existence of the spontaneous formation of local minima in the solution region cannot arise because harmonic functions obey the min-max principle. Thus, the only critical points that can happen are saddle points. The escape from such a critical point would then be found by searching the neighborhood. Any path deviation or disturbance from the critical point will eventually cause a smooth path throughout the region. For this reason, harmonic potential provides a highly beneficial option in navigation, as it offers a complete path-planning algorithm. The equation of Laplace can be solved with a range of numerical techniques, i.e., Jacobi, Gauss-Seidel and successive over-relaxation (SOR) are the standard approaches [11,12]. The development of the SOR iterative method has sparked interest in examining and resolving several issues. Since its establishment, the generalization of over-relaxation iterative approaches has been the subject of extensive research, and it has had promising outcomes. Therefore, this work will solve Eq (2) by exploiting the combination of skewed block accelerated iterative approach with the accelerated over-relaxation (AOR) and two-parameter over-relaxation (TOR) methods, for rapid computation.

In the configuration region for this model, the robot is described as a node point and the designated area is defined in the mesh grid pattern. As stated before, the off-line mode approach is used to compute the Laplacian operator of the point-robot designated area by utilizing the analogy of heat transfer. All wall boundaries and obstacles (with the highest potential value) stand for heat sources, while the goal position (with the lowest potential value) represents the heat sink. This heat transfer activity is modeled from Laplace's equation and then numerically solved to gain the heat distribution, which represents the harmonic potential for each nodal point in the mesh grid. By making use of the heat distribution property which flows from higher to lower temperatures, a gradient search can be used to generate the path from any starting position with a high potential value to the goal which has the lowest potential value. The path-planning algorithm utilizes the gradient descent search (GDS) to determine a feasible route for the robot to travel around the environment safely from the start point to reach the goal position. From the current point, the GDS method examines the potential values of its eight neighboring points on the finite-difference grid and simply picks the node with the lowest Laplacian potential value. This procedure is repeated until it reaches the target point [7,13–15].

This work aims to imitate the aforesaid paradigm for path planning, describing the solution of Laplace's Eq (2) over the analogy of temperature (for the potential) and heat flow (for the pathway). The experiment was carried out in a two-dimensional domain with walls and various forms of obstacles. A new technique called the skewed block two-parameter over-relaxation (SBTOR) iterative approach is proposed to solve Eq (2) and obtain the potential values for each node. For the purpose of performance comparison, the existing block successive over-relaxation (BSOR), block accelerated over-relaxation (BAOR), block two-parameter over-relaxation (BTOR), skewed block successive over-relaxation (SBSOR), and skewed block accelerated over-relaxation (SBAOR) techniques are also investigated.

2.2. Skewed block technique

The pioneering work of the skewed block technique (originally known as explicit decoupled group, EDG) arose from Abdullah's [16] focus on solving two-dimensional Poisson equations. The EDG iterative scheme essentially engaged the half-sweep technique in a block frame on the rotated mesh grid. By virtue of the proficiency of this technique, it has been widely used to solve numerous differential equations [17–25]. In the robotics literature, the standard Gauss-Seidel [18,19] and SOR [3,20] schemes have been employed to solve Eq (2). This work, however, generalizes a reliable numerical solver, when addressing the solutions of Laplace's equation, by utilizing a skewed block technique.

Fundamentally, merely half of the node points in the mesh region were computed while using a skewed block approach. In contrast, the explicit group (EG) scheme is basically a full-sweep (FS) technique in a block frame, evaluated every node point in the mesh region. Figure 1 provides an example of each EG and EDG scheme in a 10×10 mesh region. To obtain the Laplacian potentials in the configuration region, only the black node points (see Figure 1) will be computed throughout the iteration process, at least until the convergence condition is satisfied. The convergence criteria in this work are denoted as ε, and the stopping condition is subject to $\left\| u^{(k+1)} - u^{(k)} \right\| \le \varepsilon$. The rest of the residual nodes, or the white node points (see Figure 1(b)), will be calculated independently using a direct technique [16,26,27]. Moreover, the block iterative techniques involve four Laplacian potentials simultaneously per calculation, thus indirectly speeding up the computation. As shown in Figure 1, a set of one-node point and two-node points can be earmarked to measure groups of nodal points that are adjacent to the boundary.

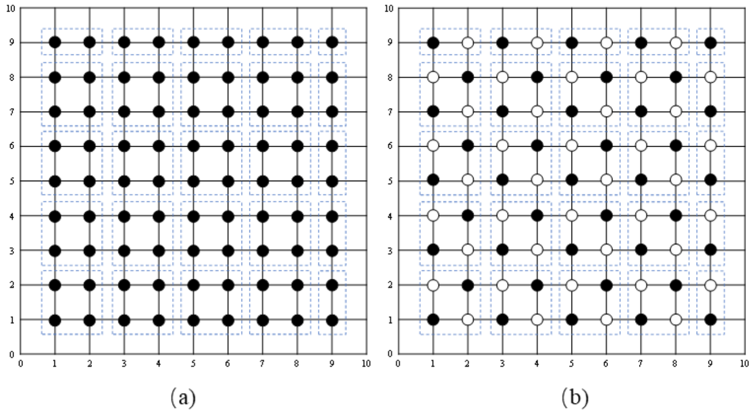

Figure 1. Interior mesh nodes for the (a) block (EG) and (b) skewed block (EDG) techniques.

From Figure 1(a), each and every formulation that practices the EG iterative approaches shall compute a set of four nodes in a group all at once throughout the iteration process (excluding the unique set adjacent to the boundary). Meanwhile, the EDG is essentially derived from a skewed 5-point finite-difference approximation (5-FDA) [16]. As illustrated in Figure 1(b), the configuration domain for the EDG technique is distributed with two different types of nodal points, i.e., the black node (●) and the white node (○). By pairing the identical node, the solutions of each group could be executed independently for each pair. On account of this independency, roughly half of the iterations across the solution domain are to be executed using either type of node, reducing performance time as well as computation complexity. Additionally, to compute the unique sets adjacent to the boundary, the direct method [27] is computed by Eq (1). One of EDG's main advantages is that these approaches diminish the computational complexity by measuring only half of all node points. The FS and half-sweep (HS) iterative approaches' computational stencils are shown in Figure 2, where h is the distance between node points for each direction. These molecules can be displayed in matrix form for FS and HS cases, respectively, as

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

and

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

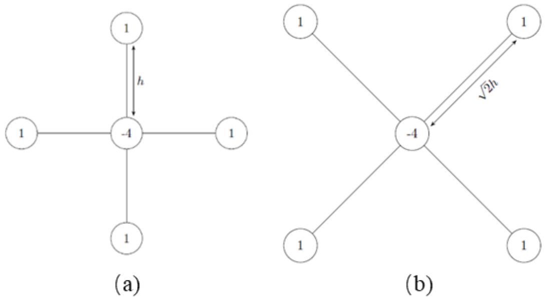

Figure 2. 5-point stencil for (a) FS/standard and (b) HS/skewed FDA.

2.3. 5-point finite-difference approximation

Laplace's equation is a special case of Poisson's equation $\nabla^2 U = f$, in which the function f is equal to zero. Hence, a two-dimensional Laplace equation (2D-Le) could be written using Eqs (1) and (2) altogether.

The system must be discretized using the finite-difference approach in order to solve the 2D-Le before it can be computed effectively using the numerical technique. The simplest second derivative of 5-FDA (also known as an FS iteration) is

$$\nabla^2 U(x,y) \approx \frac{1}{h^2}\left[u(x+h,y)+u(x-h,y)+u(x,y+h)+u(x,y-h)-4u(x,y)\right],$$

in which U is a function to satisfy Laplace's equation, while u is the potential node at the point (x,y). In addition, the $x-y$ plane of the configuration region is rotated by 45° clockwise to provide the approximation that is based on the cross-orientation operator [28,29]. As a result, a skewed 5-point approximation, commonly referred to as the HS iteration, is produced and denoted by

$$\nabla^2 U(x,y) \approx \frac{1}{2h^2}\begin{bmatrix} u(x-h,y-h)+u(x+h,y-h)+u(x-h,y+h) \\ \qquad\qquad +u(x+h,y+h)-4u(x,y) \end{bmatrix}.$$

Using the notation U_{ij} to represent the solution of the potential node for 2D-Le through the mesh grid point (x_i, y_j), the Laplace equation for the FS case is discretized using a standard 5-point stencil, as follows:

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

whereas the discretization of a rotated case via the 5-point stencil is

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

A linear system is then generated by applying Eqs (3) and (4) to the Laplace's problem subject to the 2D-Le. Matrix notation can be used to build this linear system, which results in large and sparse matrices, and is represented as

$$Au = b, \tag{5}$$

in which A represents known coefficients that are expressed in matrix form and b is a known vector, while u is the unknown vector. The expansion of each coefficient for both the FS and HS cases is discussed in depth in [25]. Since the linear system of Eq (5) creates large and sparse matrices, iterative approaches can be used to solve this problem using conventional point or block techniques [16]. In order to elucidate the block technique, the fundamental concept behind the point technique must first be described. Thus, from the second-order central finite difference of Eqs (3) and (4), the point Gauss-Seidel iterative schemes for FS and HS can be reformulated and designated respectively as below.

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

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

By integrating a weighted parameter [30], the point SOR iterative technique for FS and HS cases is given respectively as

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

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

Normally, the optimal weighted parameter ω is determined in the range of $1 \le \omega < 2$ [31]. If the parameter value is equal to 1, the approach is essentially simplified to the conventional Gauss-Seidel method. The SOR and Gauss-Seidel methods are actually rather comparable, with the exception that SOR uses a scaling factor to reduce the approximation error.

The AOR contributes a supplementary relaxation parameter that, in this work is denoted as r, along with the weighted parameter ω from the SOR technique. Both parameters are used to generate iterative schemes that are able to accelerate the convergence rates. AOR is a simple yet powerful scheme, due to the fact that two parameters are involved, for the larger linear system first introduced by Hadjidimos [32]. The point AOR schemes for standard and skewed approaches are shown respectively as

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

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

According to Hadjidimos, the r value is typically selected to be close to the corresponding ω value. Additionally, he claimed that the uncertain optimal values of these two parameters did not impose any limitations on the ability in to obtain the minimum iterations number.

The TOR iterative method is basically a generalization of the AOR method with another additional parameter, r'. The TOR scheme reduces to the AOR iterative scheme provided the value of r' is equal to r [33]. As stated in [33], the additional parameter r' could provide faster convergence if an appropriate value is chosen. The point TOR iterative schemes for FS and HS are stated respectively as

$$U_{i,j}^{(k+1)} = \frac{r}{4}U_{i,j-1}^{(k+1)} + \frac{r'}{4}U_{i-1,j}^{(k+1)} + \frac{\omega}{4}\left(U_{i,j+1}^{(k)} + U_{i+1,j}^{(k)}\right) + \left(\frac{\omega - r}{4}\right)U_{i,j-1}^{(k)} + \left(\frac{\omega - r'}{4}\right)U_{i-1,j}^{(k)} + (1-\omega)U_{i,j}^{(k)}, \quad (12)$$

$$\begin{aligned} U_{i,j}^{(k+1)} &= \frac{r}{4}U_{i+1,j-1}^{(k+1)} + \frac{r'}{4}U_{i-1,j-2}^{(k+1)} + \frac{\omega}{4}\left(U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)}\right) \\ &\quad + \left(\frac{\omega - r}{4}\right)U_{i+1,j-1}^{(k)} + \left(\frac{\omega - r'}{4}\right)U_{i-1,j-1}^{(k)} + (1-\omega)U_{i,j}^{(k)}. \end{aligned} \quad (13)$$

Similar to the AOR approach, all of the parameters are in the range of $[1,2)$ and they are chosen to be near to the value of the corresponding SOR parameters [32,34]. In accordance with the 2D-Le, all of the proposed iterative schemes simply substitute each node's value with the average of its four neighbors' values. In this work, the node values that represent the walls, obstacles and target position remain constant.

2.4. Explicit group two-parameter over-relaxation scheme

As discussed earlier, the block iterative technique obtains four Laplacian potentials per computation. As illustrated in Figure 1(a), all formulations using EG iterative methods compute a group of four nodes at once during the iteration process. Basically, the EG iteration is based on a group of a small number of points, and it is derived using the standard 5- FDA. By analyzing Eq (3) and the point SOR iterative scheme in Eq (8), the block SOR scheme [16,28,31] can be written as

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \quad (14)$$

for

$$\begin{aligned} S_1 &= U_{i-1,j} + U_{i,j-1}, \quad & S_3 &= U_{i-1,j+1} + U_{i,j+2}, \\ S_2 &= U_{i+2,j} + U_{i+1,j-1}, \quad & S_4 &= U_{i+2,j+1} + U_{i+1,j+2}. \end{aligned}$$

For this scheme, it is also possible to generate a linear system of the same form as Eq (5). The scheme can later be translated to the coefficient A after determining the inverse of its matrix, as represented below.

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{24} \begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix}, \tag{15}$$

with

$$S_a = 2\left(S_2 + S_3\right) + S_1 + S_4,$$
$$S_b = 2\left(S_1 + S_4\right) + S_2 + S_3.$$

Based on the point SOR concept using 5-FDA, the block SOR (BSOR) iterative scheme for Eq (15) is now denoted as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{24} \begin{bmatrix} 6S_1 + S_a \\ 6S_2 + S_b \\ 6S_3 + S_b \\ 6S_4 + S_a \end{bmatrix} + \left(1-\omega\right) \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}. \tag{16}$$

Besides, by examining Eq (3) and the point AOR approximation in Eq (10), the formulation of the block AOR method is stated as [35]

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \\ U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix}, \tag{17}$$

with

$$S_1 = r\left(U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)} + U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right) + \omega\left(U_{i-1,j}^{(k)} + U_{i,j-1}^{(k)}\right),$$
$$S_2 = r\left(U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right) + \omega\left(U_{i+1,j-1}^{(k)} + U_{i+2,j}^{(k)}\right),$$
$$S_3 = r\left(U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)}\right) + \omega\left(U_{i-1,j+1}^{(k)} + U_{i,j+2}^{(k)}\right),$$
$$S_4 = \omega\left(U_{i+2,j+1}^{(k)} + U_{i+1,j+2}^{(k)}\right).$$

Again, Eq (17) may also be transformed into a linear system in the form of Eq (5) and translated as in Eq (15). Hence, the block AOR (BAOR) iterative scheme can also be expressed as Eq (16).

The general expression of the block TOR (BTOR) iterative scheme, considering Eq (3) and the point TOR iterative method in Eq (12), can be written as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{1}{24}\begin{bmatrix} 6S_1+S_a \\ 6S_2+S_b \\ 6S_3+S_b \\ 6S_4+S_a \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{18}$$

where

$$\begin{aligned}
S_1 &= r\left(U_{i-1,j}^{(k+1)} - U_{i-1,j}^{(k)}\right) + r'\left(U_{i,j-1}^{(k+1)} - U_{i,j-1}^{(k)}\right) + \omega\left(U_{i-1,j}^{(k)} + U_{i,j-1}^{(k)}\right), \\
S_2 &= r'\left(U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right) + \omega\left(U_{i+1,j-1}^{(k)} + U_{i+2,j}^{(k)}\right), \\
S_3 &= r\left(U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)}\right) + \omega\left(U_{i-1,j+1}^{(k)} + U_{i,j+2}^{(k)}\right), \\
S_4 &= \omega\left(U_{i+2,j+1}^{(k)} + U_{i+1,j+2}^{(k)}\right), \\
S_a &= 2\left(S_2+S_3\right) + S_1 + S_4, \\
S_b &= 2\left(S_1+S_4\right) + S_2 + S_3.
\end{aligned}$$

2.5. Explicit decoupled group two-parameter over-relaxation scheme

As noted in the previous section, the EDG is derived from skewed 5-FDA [16]. The solution domain for the EDG technique is shown in Figure 1(b). From the solution domain, the block EDGSOR formulation may be expressed in matrix form, as Eq (14), with

$$\begin{aligned} S_1 &= U_{i-1,j-1} + U_{i-1,j+1} + U_{i+1,j-1}, \\ S_2 &= U_{i,j+2} + U_{i+2,j+2} + U_{i+2,j}, \end{aligned} \tag{19}$$

and

$$\begin{aligned} S_3 &= U_{i,j-1} + U_{i+2,j-1} + U_{i+2,j+1}, \\ S_4 &= U_{i-1,j} + U_{i-1,j+2} + U_{i+1,j+2}. \end{aligned} \tag{20}$$

Both Eqs (19) and (20) are deduced from the system of linear equations in Eq (14). These equations can be solved independently, as shown below.

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \tag{21}$$

$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_3 \\ S_4 \end{bmatrix}; \tag{22}$$

hence, it can be defined as follows (respectively):

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{15}\begin{bmatrix} 4S_1+S_2 \\ S_1+4S_2 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \tag{23}$$

$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k+1)} = \frac{\omega}{15}\begin{bmatrix} 4S_3+S_4 \\ S_3+4S_4 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k)}. \qquad (24)$$

The algorithm for SBSOR scheme can be enforced by using either Eq (23) or (24). Both equations lead to an equivalent solution.

Similarly, adopting the skewed AOR formula from Eqs (7) and (11) in the block scheme yields a new formula known as SBAOR [36]. As a result, the SBAOR iteration scheme is described as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{r}{15}\begin{bmatrix} 4S_2 \\ S_2 \end{bmatrix} + \frac{\omega}{15}\begin{bmatrix} 4S_1+S_3 \\ S_1+4S_3 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \qquad (25)$$

where

$$\begin{aligned} S_1 &= U_{i-1,j-1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k)}, \\ S_2 &= U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1, \\ S_3 &= U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)} + U_{i+2,j}^{(k)}. \end{aligned}$$

The same goes for the rotated block TOR, which applies the skewed TOR formula from Eq (7), as well as Eq (13), to the block scheme, and it provides the SBTOR iterative scheme as follows:

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{r}{15}\begin{bmatrix} 4S_3 \\ S_3 \end{bmatrix} + \frac{r'}{15}\begin{bmatrix} 4S_4 \\ S_4 \end{bmatrix} + \frac{\omega}{15}\begin{bmatrix} 4S_1+4S_2+S_5 \\ S_1+S_2+4S_5 \end{bmatrix} + (1-\omega)\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k)}, \qquad (26)$$

in which

$$\begin{aligned} S_1 &= U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)}, \\ S_2 &= U_{i-1,j+1}^{(k)}, \\ S_3 &= U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1, \\ S_4 &= U_{i-1,j+1}^{(k+1)} - S_2, \\ S_5 &= U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)} + U_{i+2,j}^{(k)}. \end{aligned}$$

As stated before, there is no generic formula for defining the optimal value for all weighted parameters, which results in a minimum number of iterations. Consistent with [32], the values of r and r' are commonly appointed as close to the corresponding ω value. Note that not every parameter value leads to convergence. The optimal values for all three relaxation parameters for each FS and HS case are different. Due to each relaxation value being predetermined before execution, the complexity related to determining the relaxation values for the overall computation is unaffected. Table 1 provides a list of the optimal relaxation values used throughout the experiments. Sensitivity analysis has been used to specify each relaxation parameter's value. The values are the same for every N-size grid, although they differ from other grid sizes and are not significant enough to be revealed; they vary by roughly 0.0001.

Table 1. Grid search of relaxation parameter values.

Methods	ω	r	r'
BSOR	1.82	-	-
BAOR	1.83	1.82	-
BTOR	1.83	1.86	1.89
SBSOR	1.81	-	-
SBAOR	1.82	1.84	-
SBTOR	1.82	1.87	1.88

Algorithm 1 describes the implementation of the SBTOR scheme in accordance with Eq (26) to solve the two-dimensional Laplace problem.

Algorithm 1: SBTOR scheme

```
i.    Setup the configuration region with start and goal positions.
ii.   Initialize the starting point $U, \varepsilon \leftarrow 10^{-15}, iteration \leftarrow 0$ .
iii.  Set the variables
      $S_1 \leftarrow U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)},$
      $S_2 \leftarrow U_{i-1,j+1}^{(k)},$
      $S_3 \leftarrow U_{i-1,j-1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - S_1,$
      $S_4 \leftarrow U_{i-1,j+1}^{(k+1)} - S_2,$
      $S_5 \leftarrow U_{i,j+2}^{(k)} + U_{i+2,j+2}^{(k)} + U_{i+2,j}^{(k)}.$
iv.   For each non-occupied black nodes (●), calculate
      $U_{i,j}^{(k+1)} \leftarrow \frac{r}{15} 4S_3 + \frac{r'}{15} 4S_4 + \frac{\omega}{15}\left[4S_1 + 4S_2 + S_5\right] + (1-\omega) U_{i,j}^{(k)},$
      $U_{i+1,j+1}^{(k+1)} \leftarrow \frac{r}{15} S_3 + \frac{r'}{15} S_4 + \frac{\omega}{15}\left[S_1 + S_2 + 4S_5\right] + (1-\omega) U_{i+1,j+1}^{(k)}.$
v.    Compute the unique sets adjacent to the boundary via the direct method using Eq (13).
      Then, evaluate the remaining white nodes (○) by using
      $U_{i,j}^{(k+1)} \leftarrow \frac{1}{4}\left[U_{i-1,j}^{(k+1)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k+1)} + U_{i,j+1}^{(k)}\right].$
vi.   Encounter the convergence test $\varepsilon \leftarrow 10^{-15}$ , go to (vii). Else, back to (iii).
vii.  Run GDS for path construction.
```

3. Results and discussion

Following the aim of this study, an experiment has been conducted by using a robot 2D simulator [37] built by the authors to analyze the competency of the proposed algorithm, i.e., SBTOR scheme, as a tool to solve the path-planning problem. In the block mesh grid domain (see Figure 1) with the execution of the skewed block iterative method, approximately half of the group of node points are computed throughout the iteration process. This will logically result in a substantial reduction of the computational complexity by nearly 50% of the entire operation. The simulations are carried out on four different configuration spaces with five different grid sizes in order to assess the performance of the proposed algorithms. The aforementioned performances were analyzed with three criteria: (i) number of iterations, (ii) CPU time in seconds and (iii) success of path planning. The amount of CPU time and iteration count required by every algorithm is recorded once the tolerance rate for the iteration process is reached. The GDS approach then makes use of the computed Laplacian potentials to guarantee that a path from the starting point to the goal position is successfully constructed.

In the designated areas, several obstacles with different sizes and shapes are positioned. The Dirichlet boundary condition was utilized in the initial configuration, in which the walls and obstacles are set at the high potential value (the highest temperature). The start position had no initial value specified, whereas the goal location was set at the low potential value (viewed as the lowest temperature). The remainder of mesh grid points were initialized to zero. A machine running a 2.50-GHz processor with 8 GB of RAM has been used to carry out the simulation experiments. Numerical computation was performed iteratively until the stopping condition was met. Provided the Laplacian potential values give no further changes and the difference of current iterations (k) with the next iterations (k+1) is extremely small, e.g., 1.0^{-15}, the loop should be stopped. This accuracy level was necessary to prevent any point on the surface which has critical point slopes that result in an abrupt absence of descending gradient patterns.

Each iterative scheme compared in the experiments is shown in Table 2, along with the iteration counts (*k*) and the execution time (*t*). Both the block (EG) and skewed block (EDG) schemes evidently show that the TOR iterative approach surpassed its generalization techniques. The computational complexity of the skewed block iterative approach will decrease by approximately half of that relative to the HS process. Since only half of the total nodes in the mesh grid are examined, the skewed schemes surpass the standard block schemes that utilize the FS process. The size of the mesh grid and pattern of the region plays a significant role in producing the outputs of *k* and *t*. Obviously, the larger grid size will require more iterations and a longer time to be executed. The improvement ratio between each over-relaxation technique for the smallest grid size (300) to the largest (1500) varies. In terms of the iteration count, on average, the BTOR efficiently decreased by 3–6% compared to BAOR, and 12–20% next to BSOR, whereas SBTOR reduced it by more than 3–8% compared to SBAOR, and roughly 19–32% of SBSOR. However, in terms of execution time, BTOR is between 3–7% lower relative to BAOR, and 14–18% in comparison with BSOR, while SBTOR effectively depreciated approximately 1–4% from SBAOR, and 15–23% in comparison with SBSOR. This study still contributes good results since previous studies [3,13,38] only did the experiments on extremely small grid meshes, i.e., 50 by 50 and 70 by 70 grid. Figures 3 and 4 show the results of Table 2 graphically for the number of iterations and CPU time, respectively. From the graphs, again, the skewed block TOR undoubtedly offers the optimum performance in contrast to its predecessor.

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

	Techni q-ues	**N × N**									
		300		**600**		**900**		**1200**		**1500**	
		k	*t*	*k*	*t*	*k*	*t*	*k*	*t*	*k*	*t*
Region 1	BSOR	1258	6.88	5899	163.72	12844	871.66	22227	2694.80	34055	6286.69
	BAOR	1042	6.05	4994	137.87	10928	751.78	19107	2442.66	29306	5551.02
	BTOR	997	5.05	4812	133.00	10581	720.29	18549	2394.62	28445	5404.33

Continued on next page

	Techniques	N × N									
		300		**600**		**900**		**1200**		**1500**	
		k	*t*	*k*	*t*	*k*	*t*	*k*	*t*	*k*	*t*
Region 1	SBSOR	953	4.34	4495	110.88	9916	603.41	17653	1947.12	27037	3878.32
	SBAOR	766	3.67	3730	95.08	8200	498.19	14338	1599.91	22141	3336.68
	SBTOR	743	3.69	3649	91.03	8038	490.77	14077	1626.65	21676	3303.54
Region 2	BSOR	1729	7.67	6782	199.59	14874	1009.48	26007	2827.46	39968	6925.80
	BAOR	1610	8.25	6368	185.36	13953	926.49	24429	3003.98	32926	5909.85
	BTOR	1489	7.64	5957	169.39	13062	867.20	22905	2787.69	31552	5700.19
	SBSOR	1324	4.70	5599	142.64	13252	787.53	22898	2386.46	42248	6170.35
	SBAOR	1188	4.17	4767	122.85	10490	624.48	18371	1930.06	24644	3900.10
	SBTOR	1131	4.63	4561	119.23	10032	603.66	17542	1881.31	24001	3815.88
Region 3	BSOR	2666	13.24	11076	315.87	24519	1602.81	42897	5591.93	65977	12331.17
	BAOR	2480	13.83	10389	301.27	22995	1633.35	40322	5261.60	62423	11975.63
	BTOR	2371	11.66	9977	296.46	22111	1883.36	38917	5094.93	59912	10921.11
	SBSOR	2035	8.00	10243	261.74	24864	1529.74	45435	5017.92	70627	10771.29
	SBAOR	1835	7.67	7741	203.61	17131	1072.61	30027	3437.95	46055	7612.81
	SBTOR	1784	6.69	7557	202.95	16707	1251.76	29270	3398.37	45091	7337.51
Region 4	BSOR	1629	7.80	6487	187.33	14194	990.20	24913	2979.14	38195	6919.25
	BAOR	1392	7.56	5648	167.65	12367	891.51	21724	2609.11	33518	6139.29
	BTOR	1328	7.10	5428	163.21	11907	850.26	20963	2573.12	34842	6494.66
	SBSOR	1238	4.33	5958	152.73	13501	828.24	23920	2527.82	38175	5948.26
	SBAOR	1116	4.53	4223	109.96	9369	615.38	16313	1725.41	25023	3991.87
	SBTOR	1125	5.02	4106	107.90	9149	605.69	15908	1774.41	26360	4380.35

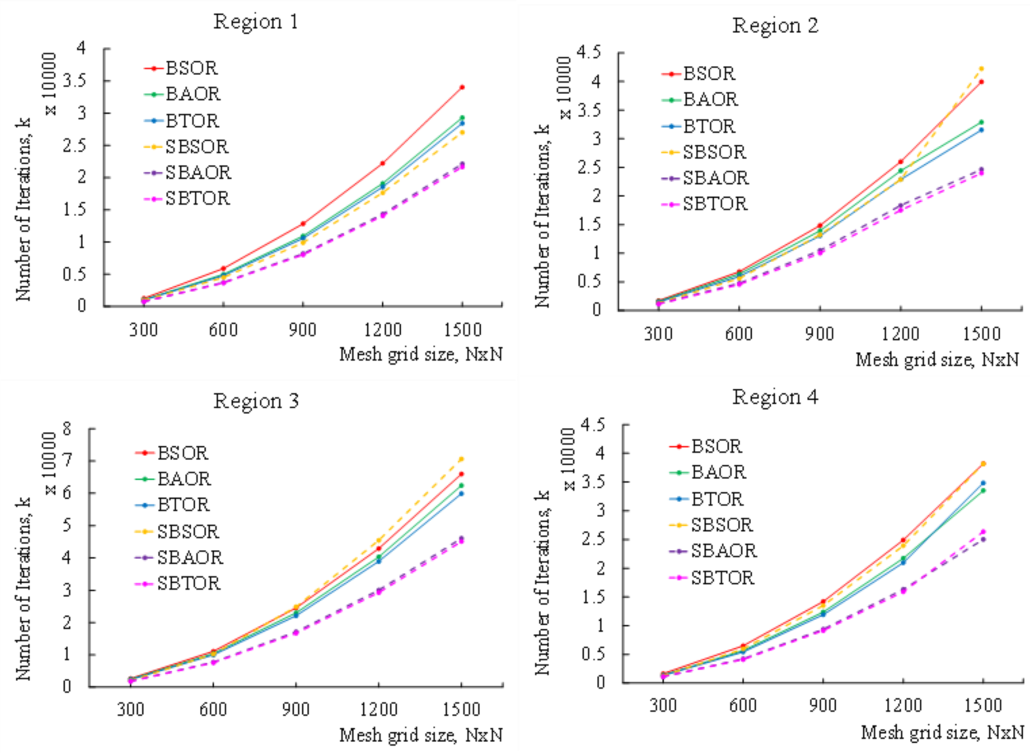

Figure 3. Performance graph in relation to the number of iterations.

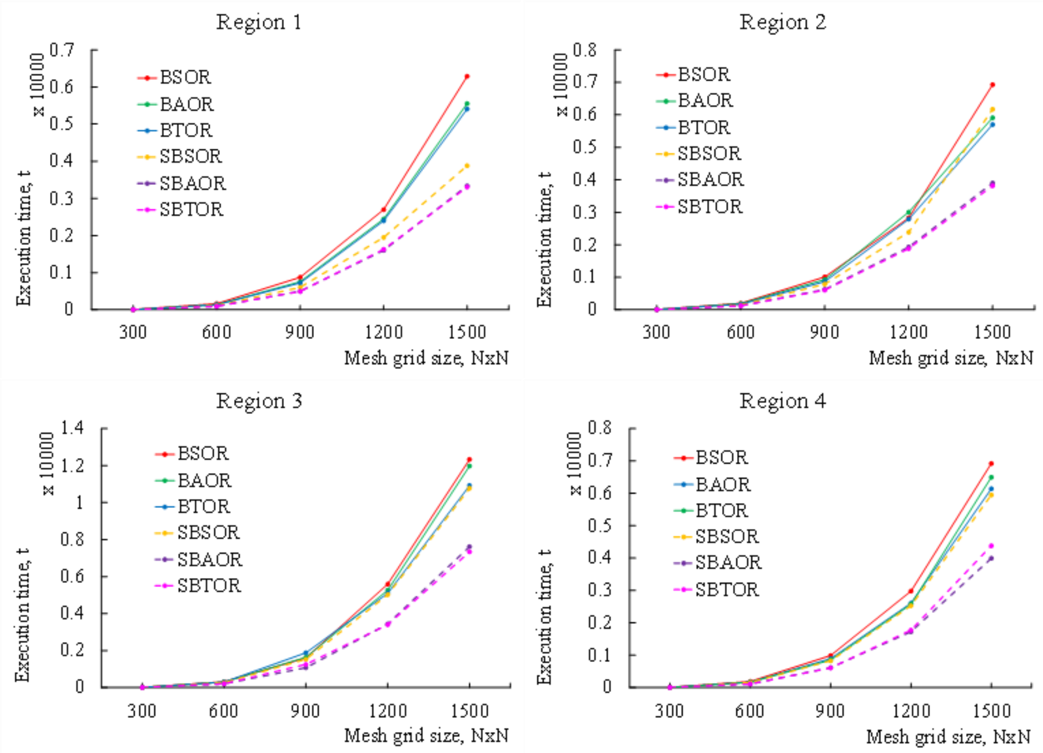

Figure 4. Performance graph in relation to the execution time.

The ideology of path-planning flow within this experiment starts with establishing the initial start position and the target location. Next, we identify the optimal parameter according to the corresponding iterative schemes. Once the harmonic potential is generated from the selected algorithm, a complete smooth path is developed through the use of the GDS technique. This impression could describe the iterative scheme that tracks the descending gradient from its starting point to the next consecutive points with lower potentials from previous points, up to the target location (with the lowest potential value). These successful trails can be observed as demonstrated in Figure 5. In this simulation, the starting point is indicated by a green square point, while a red circle point is allocated for the target. As shown in Figure 5, each start location from every region has efficiently accomplished the route by arriving at the target position along while escaping any walls and obstacles in between (if any). The path trajectories can be really fast because they only involve the gradient evaluation of the precomputed Laplacian potential [13]. All four configuration regions in this experiment are relatively simple relative to those in [38].

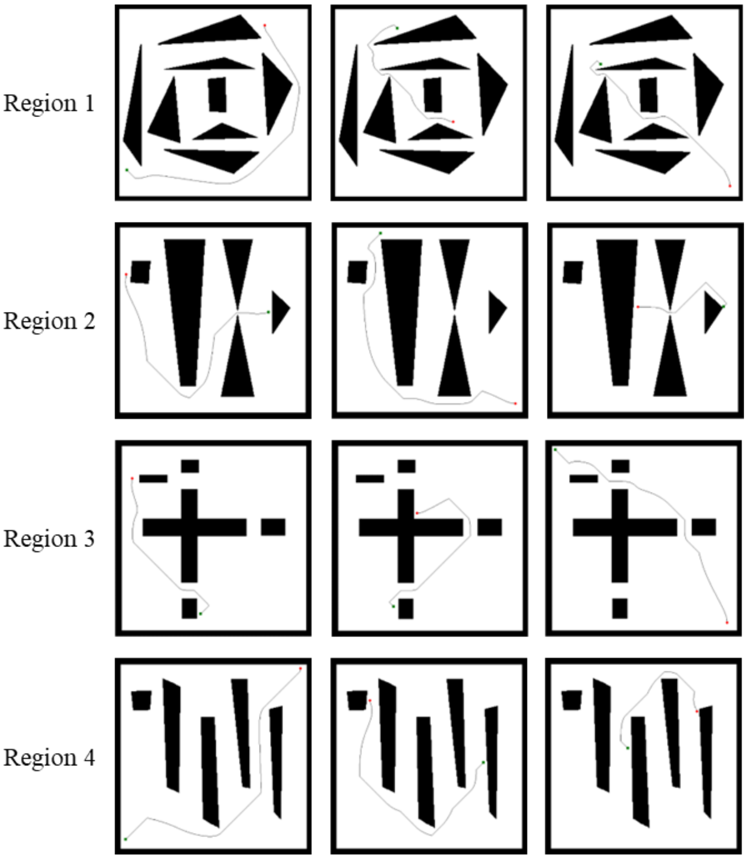

Figure 5. Path construction in an identified stationary environment.

There are various techniques to solve navigation problems. According to Patle et al. [1], these techniques can be divided into two main groups, i.e., classical approaches and reactive approaches. Due to the lack of artificial intelligence tools at the time, the classical approach was initially quite common as a tool to address robot navigational issues. When a task is performed utilizing a classical approach, it can be noticed that either a result will be acquired, or it will be confirmed that there is no result. This method is less suitable for real-time implementation due to its significant disadvantages, including its high processing cost and inability to respond to the uncertainty existing in the environment. Reactive techniques have surpassed traditional ways in recent years as the preferred method for mobile robot navigation. They are extremely adept at managing the environment's uncertainty. Meanwhile, Sanchez-Ibanez et al. [39] classified path planning into four categories with two major subcategories (see Figure 6). There are characteristics shared by two adjacent subcategories from distinct categories. The schematic also shows how some subcategories lean more toward global planning or local planning than others. From the reviewed papers [1,39], it is safe to say the proposed algorithm in this study has its own edge and drawback. Algorithm 1 has clearly implemented the improved version of potential field approaches. In essence, the goal and obstacles operate as charged surfaces, and the overall potential creates the imaginary forces on the robot. The robot is drawn toward the goal and kept clear of the obstacles by this imaginary force [8]. Later, the robot will travel along the negative gradient to avoid obstacles and arrive at its destination point. To prevent a local minimum problem, this work utilizes the harmonic function [7]. Moreover, Algorithm 1 exhibits superior computing execution when the skewed accelerated relaxation technique is implemented, as it performs much faster to obtain the solution of Laplace’s equation to solve the path-planning problem.

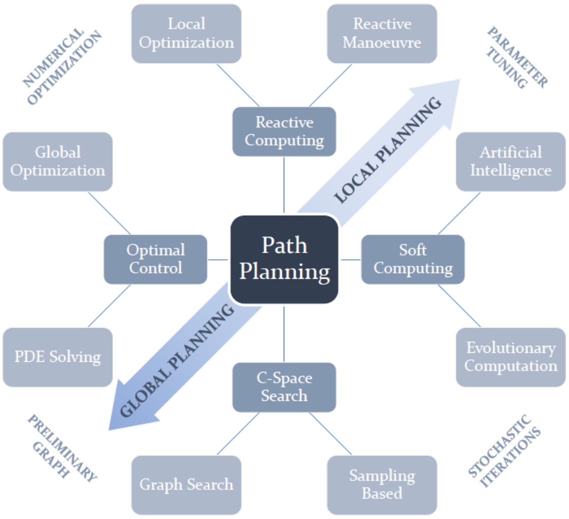

Figure 6. Classification of existing path-planning techniques [39].

4. Conclusions

The iterative approaches investigated in this work are from over-relaxation families including the implementation of the block and skewed block schemes to enhance the execution performance as well as reduce the execution time. The introduction of accelerated weighted parameters to respective skewed nodes further improves the overall performance of the newly proposed SBTOR iterative schemes, yielding promising results. The experiments evidently demonstrate that the solution to the robot path-planning problem is feasibly solved by using numerical approaches owing to advanced algorithms and the accessibility of fast machines these days. From the table of results, the SBTOR scheme was found to significantly outperform its predecessor’s techniques in terms of both the number of iterations and the time taken. The computational performance is unaffected by increasing the number of obstacles; in fact, it will complete more quickly because the calculation disregards the regions engaged by the obstacles. It can be said that the larger the area of obstacles taken, the less the required calculations and storage, or, in other words, the obstacles reduce computational complexity. Once again, the authors would like to highlight that the SBTOR outperformed the SBAOR (approximately by 3–8%) and SBSOR (by about 19–32%) in terms of the iteration count, while the SBTOR saved roughly 1–4% over SBAOR and 15–23% over SBSOR in terms of the processing time. The application of the skewed block TOR scheme families in robot pathfinding and in Algorithm 1 is another aspect of this study's originality or novelty. The authors intend to investigate the red and black strategy [23,24,40–42] by applying the proposed approaches in future work. It is believed that the aforementioned strategy will enhance and refine the overall computation.

Acknowledgments

This research was financially supported by Universiti Pertahanan Nasional Malaysia. The authors also acknowledge support from Science Foundation Ireland (SFI) under Grant Number

SFI/16/RC/3918 (Confirm), and the Marie Skłodowska-Curie grant agreement No. 847577, co-funded by the European Regional Development Fund.

Conflict of interest

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

References

1. B. Patle, B. Ganesh, A. Pandey, D. Parhi, A. Jagadeesh, A review: On path planning strategies for navigation of mobile robot, *Def. Technol.*, **15** (2019), 582–606. https://doi.org/10.1016/j.dt.2019.04.011
2. N. Buniyamin, N. Sariff, W. Wan Ngah, Z. Mohamad, Robot global path planning overview and a variation of ant colony system algorithm, *International Journal of Mathematics and Computers in Simulation*, **5** (2011), 9–16.
3. S. Sasaki, A practical computational technique for mobile robot navigation, *Proceedings of the IEEE International Conference on Control Applications*, 1998, 1323–1327. https://doi.org/10.1109/CCA.1998.721675
4. H. Chou, P. Kuo, J. Liu, Numerical streamline path planning based on log-space harmonic potential function: a simulation study, *Proceedings of IEEE International Conference on Real-time Computing and Robotics*, 2017, 535–542. https://doi.org/10.1109/RCAR.2017.8311918
5. Y. An, S. Wang, C. Xu, L. Xie, 3D path planning of quadrotor aerial robots using numerical optimization, *Proceedings of the 32nd Chinese Control Conference*, 2013, 2305–2310.
6. M. Silva, W. Silva, R. Romero, Performance analysis of path planning techniques based on potential fields, *Proceedings of Latin American Robotics Symposium and Intelligent Robotics Meeting*, 2010, 115–119. https://doi.org/10.1109/LARS.2010.33
7. C. Connolly, R. Gruppen, The applications of harmonic functions to robotics, *Journal of Robotic Systems*, **10** (1993), 931–946. https://doi.org/10.1002/rob.4620100704
8. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Proceedings of IEEE International Conference on Robotics and Automation*, 1985, 500–505. https://doi.org/10.1109/ROBOT.1985.1087247
9. H. Ghassemi, S. Panahi, A. Kohansal, Solving the Laplace's equation by the FDM and BEM using mixed boundary conditions, *American Journal of Applied Mathematics and Statistics*, **4** (2016), 37–42. https://doi.org/10.12691/ajams-4-2-2
10. A. Dahalan, A. Saudi, Pathfinding algorithm based on rotated block AOR technique in structured environment, *AIMS Mathematics*, **7** (2022), 11529–11550. https://doi.org/10.3934/math.2022643
11. K. Al-Khaled, Numerical solutions of the Laplace's equation, *Appl. Math. Comput.*, **170** (2005), 1271–1283. https://doi.org/10.1016/j.amc.2005.01.018
12. K. Shivaram, H. Jyothi, Finite element approach for numerical integration over family of eight node linear quadrilateral element for solving Laplace equation, *Mater. Today: Proc.*, **46** (2021), 4336–4340. https://doi.org/10.1016/j.matpr.2021.03.437
13. C. Connolly, J. Burns, R. Weiss, Path planning using Laplace's equation, *Proceedings of IEEE International Conference on Robotics and Automation*, 1990, 2102–2106. https://doi.org/10.1109/ROBOT.1990.126315

14. J. Barraquand, B. Langlois, J. Latombe, Numerical potential field techniques for robot path planning, *IEEE T. Syst. Man. Cy.*, **22** (1992), 224–241. https://doi.org/10.1109/21.148426
15. M. Karnik, B. Dasgupta, V. Eswaran, A comparative study of Dirichlet and Neumann conditions for path planning through harmonic functions, In: *Lecture notes in computer science*, Berlin: Springer, 2002, 442–451. https://doi.org/10.1007/3-540-46080-2_46
16. A. Abdullah, The four point explicit decoupled group (EDG) method: a fast Poison solver, *Int. J. Comp. Math.*, **38** (1991), 61–70. https://doi.org/10.1080/00207169108803958
17. J. Sulaiman, M. Hassan, M. Othman, The half-sweep iterative alternating decomposition explicit (HSIADE) method for diffusion equation, In: *Computational and information science*, Berlin: Springer, 2004, 57–63. https://doi.org/10.1007/978-3-540-30497-5_10
18. A. Saudi, J. Sulaiman, Block iterative method for robot path planning, *Proceedings of the 2nd Seminar on Engineering and Technology*, 2009, 1–4.
19. A. Saudi, J. Sulaiman, Half-Sweep Gauss-Seidel (HSGS) iterative method for robot path planning, *Proceedings of the 3rd International Conference on Informatics and Technology*, 2009, 34–39.
20. A. Saudi, J. Sulaiman, Robot path planning using Laplacian Behaviour-Based Control (LBBC) via half-sweep SOR, *Proceedings of the International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*, 2013, 424–429. https://doi.org/10.1109/TAEECE.2013.6557312
21. S. Matsui, H. Nagahara, R. Taniguchi, Half-sweep imaging for depth from defocus, *Image Vision Comput.*, **32** (2014), 954–964. https://doi.org/10.1016/j.imavis.2014.09.001
22. J. Chew, J. Sulaiman, A. Sunarto, Solving one-dimensional porous medium equation using unconditionally stable half-sweep finite difference and SOR method, *Mathematics and Statistics*, **9** (2021), 166–171. https://doi.org/10.13189/ms.2021.090211
23. F. Musli, J. Sulaiman, A. Saudi, Numerical simulations of agent navigation via half-sweep modified two-parameter over-relaxation (HSMTOR), *Journal of Physics Conference Series*, **1988** (2021), 012035. https://doi.org/10.1088/1742-6596/1988/1/012035
24. A. Saudi, A. Dahalan, An efficient red-black skewed modified accelerated arithmetic mean iterative method for solving two-dimensional Poisson equation, *Symmetry*, **14** (2022), 993. https://doi.org/10.3390/sym14050993
25. A. Dahalan, A. Saudi, Pathfinding algorithm based on rotated block AOR technique in structured environment, *AIMS Mathematics*, **7** (2022), 11529–11550. https://doi.org/10.3934/math.2022643
26. M. Othman, A. Abdullah, An efficient four points modified explicit group Poisson solver, *Int. J. Comput. Math.*, **76** (2000), 203–217. https://doi.org/10.1080/00207160008805020
27. D. Evans, Group explicit iterative methods for solving large linear systems, *Int. J. Comput. Math.*, **17** (1985), 81–108. https://doi.org/10.1080/00207168508803452
28. G. Dahlquist, Å. Björck, *Numerical Methods*, New Jersey: Prentice Hall, 1974.
29. W. Yousif, D. Evans, Explicit group over-relaxation methods for solving elliptic partial differential equations, *Math. Comput. Simulat.*, **28** (1986), 453–466. https://doi.org/10.1016/0378-4754(86)90040-6
30. D. Young, Iterative methods for solving partial difference equations of elliptic type, *T. Am. Math. Soc.*, **76** (1954), 92–111. https://doi.org/10.2307/1990745
31. D. Young, Iterative methods for solving partial difference equations of elliptic type, Ph. D. Thesis, Harvard University, 1950.

32. A. Hadjidimos, Accelerated overrelaxation method, *Math. Comput.*, **32** (1978), 149–157. https://doi.org/10.2307/20006264
33. J. Kuang, J. Ji, A survey of AOR and TOR methods, *J. Comput. Appl. Math.*, **24** (1988), 3–12. https://doi.org/10.1016/0377-0427(88)90340-8
34. N. Ali, F. Pin, Modified explicit group AOR methods in the solution of elliptic equations, *Applied Mathematical Sciences*, **6** (2012), 2465–2480.
35. M. Martins, W. Yousif, D. Evans, Explicit group AOR method for solving elliptic partial differential equations, *Neural, Parallel and Scientific Computations*, **10** (2002), 411–422.
36. N. Ali, L. Chong, Group accelerated OverRelaxation methods on rotated grid, *Appl. Math. Comput.*, **191** (2007), 533–542. https://doi.org/10.1016/j.amc.2007.02.131
37. A. Saudi, Robot path planning using family of SOR iterative methods with Laplacian behavior-based control, Ph. D. Thesis, University Malaysia Sabah, 2015.
38. S. Chen, Collision-free path planning, Ph. D. Thesis, Iowa State University, 1997. https://doi.org/10.31274/rtd-180813-10480
39. J. Sanchez-Ibanez, C. Perez-del-Pulgar, A. Garcia-Cerezo, A path planning for autonomous mobile robots: a review, *Sensors*, **21** (2021), 7898. https://doi.org/10.3390/s21237898
40. T. Iwashita, M. Shimasaki, Block red-black ordering: a new ordering strategy for parallelization of ICCG method, *Int. J. Parallel Prog.*, **31** (2003), 55–75. https://doi.org/10.1023/A:1021738303840
41. N. Saad, A. Sunarto, A. Saudi, Accelerated red-black strategy for image composition using Laplacian operator, *IJCDS*, **10** (2021), 1085–1095. https://doi.org/10.12785/ijcds/100198
42. A. Dahalan, A. Saudi, J. Sulaiman, Pathfinding for mobile robot navigation by exerting the quarter-sweep modified accelerated overrelaxation (QSMAOR) iterative approach via the Laplacian operator, *Mod. Simul. Eng.*, **2022** (2022), 9388146. https://doi.org/10.1155/2022/9388146