Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Performance Analysis of Log-Space Harmonic Potentials for Path Planning in Indoor Environment

Nowyannie Willie D. Tamsin[1,3], Jumat Sulaiman[1], Azali Saudi[2,*], Andang Sunarto[4]

[1] Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[2] Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia
[3] College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA Sabah Branch Kota Kinabalu, Sabah, Malaysia
[4] Tadris Matematika, Universitas Islam Negeri Fatmawati Sukarno Bengkulu, Indonesia

ARTICLE INFO

Article history:
Received 21 March 2024
Received in revised form 22 November 2024
Accepted 16 December 2024
Available online 31 December 2024

Keywords:
Harmonic potentials; log-space Gauss-Seidel; Laplace's equation; path planning

ABSTRACT

In recent years, there has been an increasing interest in the study of path planning algorithms for autonomous robot systems. The current study aims to examine the simulation performance of path planning using the Successive Overrelaxation (SOR) method and the log-space Gauss-Seidel (LGS) method. The simulation addresses the numerical solution of a boundary value problem for Laplace's equation, utilising Log-space mapping and adhering to Dirichlet boundary conditions within a closed region. The experiment done in both sparse and dense environments shows the benefits of the iterative methods. The paths generated by the SOR and LGS approach are compared using specific numerical examples in smooth trajectory planning. The comparative analysis suggests that the LGS iterative method effectively addresses the issue of achieving high precision in the numerical computation of the gradient of harmonic potentials for generating safe paths in dense environments with narrow corridors and difficult regions. The simulation results also indicate that the LGS employed in this study exhibits faster convergence. Although it has a slower execution time, it is successful in dense environments compared to the SOR method.

1. Introduction

The study of path planning for robots has been a critical focus in the field of robotics development over the years. A truly autonomous robot must possess the capability to navigate itself from its start location to a specified destination point. Mobile robots are frequently required to traverse their environment while averting obstacles to reach a predetermined destination from any given starting point. The task of path planning or navigation holds significant importance in autonomous mobile robotics as it directly impacts accuracy, productivity, and safety Dahalan *et al.,* [1]. Numerous research studies have been conducted about path planning across various industries. Previous studies conducted by Li *et al.,* [2] and Sheng [3] have focused on the exploration of path-planning techniques within the domain of computer animation. Several scholars, including Larsen *et al.,* [4]

* *Corresponding author.*
E-mail address: azali@ums.edu.my

https://doi.org/10.37934/ard.123.1.186197

and Zhang *et al.,* [5], have studied path planning for industrial robotics. In addition, Shamsudin *et al.,* [6] researched the delivery industry, Shi *et al.,* [7] focused on the automatic generation of ship navigation routes, and Song *et al.,* [8] studied path planning for automated surveillance.

Mobile robots have become essential components in a wide range of applications in the fast-growing field of robotics, from autonomous automobiles and warehouse logistics to healthcare and agriculture. These robots are designed to navigate complex environments, often necessitating well-calculated route planning to reach target locations efficiently. Path planning is a critical aspect of mobile robot navigation as it enables safe and quick transit between locations, even within unfamiliar and changing settings. The field of mobile robot navigation includes the study and implementation of algorithms, strategies, and techniques that optimise pathway selection, obstacle avoidance, and adaptation to dynamic environments. This process is essential for improving the usefulness of mobile robots in various settings, whether they are working indoors, outdoors, or in confined spaces.

Harmonic functions have been said to be a good method in path planning as their properties align well with the demands of robust path planning, particularly in handling potential fields with smoothness and stability. As asserted by Connolly and Roderic [9], harmonic functions exhibit several properties, such as completeness, that are crucial for robotics applications. For instance, Kim and Pradeep [10] found that when employed, harmonic functions could effectively reduce local minima, even in the presence of a congested environment. In a similar vein, Motonaka *et al.,* [11] integrated HPF with an invariant manifold to enhance motion planning for a two-wheeled drive mobile robot. Kazemi *et al.,* [12] in their study demonstrated that the integration of the Harmonic Function-based Probabilistic Roadmap (HFPRM) yields better outcomes compared to the use of each component separately, particularly in scenarios characterised by the presence of narrow passages. The work by Masoud [13] also illustrates the efficacy of the HPF planning approach in producing a properly constrained path for a robot with second-order dynamics in a cluttered environment.

This study uses numerical potential functions inspired by the heat transfer theory to model the path planning process of a point robot in the configuration space. The environment generated through this heat conduction model is free of local minima and advantageous for the robot's navigation control. Solutions to Laplace's equation, commonly called harmonic functions, are applied to simulate temperature distribution inside the configuration space for path generation. Numerical approaches have been used to determine harmonic functions because they are easy to use on high-speed computing machines and are good at finding solutions to problems.

According to Andrili and David [14], the iterative method such as the Jacobi Method, or the Gauss-Seidel Method is used to find a solution to a linear system and is known to provide one of the fastest ways to obtain the actual solution. Furthermore, Mohammedali *et al.,* [15] and Al-Jizani *et al.,* [16] argue that its variants such as the variational iterative method are one of the well-known semi-analytical methods for solving linear and nonlinear ordinary equations as well as partial differential equations. In robot path planning problems, the iterative method is commonly employed to obtain the HPF. In the current study, several experiments were conducted to determine the efficacy of the iterative method in the Log-space domain for obtaining the HPF required to generate a path for a mobile robot in an environment of varying sizes. Some of the simple geometries obstacles commonly applied in HPFs are elliptical static obstacles, which were done by Szulczynski *et al.,* [17], rectangles, used in the research of Tsaryova *et al.,* [18], and circular obstacles conducted in the study of Waydo and Richard [19] and Liu *et al.,* [20]. In the current work, the rectangular obstacle was chosen due to its simplicity and efficiency in representing structured indoor environments, which enhances the applicability of the results for real-world scenarios.

Despite the advancement in path planning methodologies, much uncertainty still exists about the efficient and precise methods for navigating sparse indoor environments. Most studies in the field of path planning algorithms have only focused on traditional iterative approaches, such as Gauss-Seidel method, Jacobian method, and SOR. These methods, however, have limitations in convergence speed and scalability, particularly when working with complex large-scale configurations. In addition, very little attention has been paid to the use alternative methods such as the LGS in path planning, particularly when using harmonic potentials for structured indoor environments. As such, a systematic understanding of how LGS contributes to the development of safe paths for robot navigation is still lacking. To address this gap, this study examines the efficacy of LGS in the computation of harmonic potential fields for robot path planning. The current work makes an important contribution by comparing the performance of LGS and Gauss-Seidel across different mesh sizes and obstacle configurations, as well as the actual use of LGS for path planning in real-world indoor situations. These findings provide an important opportunity to advance the understanding mobile robot path planning and developing algorithms for collision-free avoidance in indoor environment.

The subsequent sections of this paper are organized as follows: Section 2 provides a thorough overview of the research background, discussing the problems and algorithms of this study, and describes the algorithmic framework and methodology of Successive Over-Relaxation (SOR) and Log-space Gauss-Seidel (LGS). Section 3 details various simulation environments, performance metrics, and the comparison process of the algorithms are comprehensively described and followed by a discussion in Section 4. Finally, concluding remarks are provided in Section 5.

2. Methodology

This study presents a simulation-based approach for implementing robot vehicle movement by applying a point that goes through a known location rather than using the actual physical robot vehicle itself. The robot path planning problem can be described as an analogy to the steady-state heat conduction problem. In the context of heat conduction, the aim is to examine the target as a heat sink that attracts heat energy. Physical barriers and constraints maintained at a constant temperature are referred to as heat sources. The temperature distribution is altered because of thermal conductivity, and the design space becomes saturated with thermal fluxes that flow into the sink. The target, robots, and obstacles could all potentially communicate with other objects through this structure.

The temperature distribution throughout the field can serve as guidance for a mobile robot to navigate from the initial location to the desired destination. This can be achieved by monitoring the flow of heat from areas with higher temperatures to areas with the lowest point within the region. The harmonic function is applied to represent the environment's configuration to determine the temperature dispersion within the heat configuration region. Conolly *et al.,* [21] stated that by solving the associated partial differential equation, one can obtain a flow that optimises obstacle avoidance while moving towards a designated goal. This generates a smooth path that promotes a natural movement with no local minima. Algorithm 1 illustrates the algorithm for the fundamental process that forms the path planning construction phase of this study (Table 1). Harmonic functions are the solutions to Laplace's equation and is expressed as in Eq. (1):

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

Where x_i is the i^{th} Cartesian coordinates, and n denotes the dimension of the region $\Omega \subset \mathbb{R}^n$. The domain Ω comprises the inner and outer boundary walls, obstacles, start, and goal points. As harmonic functions are bound by the min-max rule, it maintains that there cannot be any unexpected local minima in the solution domain. In this regard, Conolly and Roderic [9] mentioned that path planning is complete when it applies Laplace's equation with certain Dirichlet Boundary Conditions (BC). Therefore, the Dirichlet BC for the Laplace's Eq. (1) can be defined as in Eq. (2):

$$(x, y) = \begin{cases} 1, \text{if } (x, y) \text{is at obstacles} \\ 0, \text{if } (x, y) \text{is at the goal point} \end{cases} \tag{2}$$

Table 1
Algorithm 1. Process of path planning

Begin
Step 1: Create a map of the robot's region (known as grid space, obtaining the goal position, obstacles, and walls).
Step 2: Initiate the formulation and modeling of the iterative schemes.
Step 3: Formulate and execute the proposed iterative schemes.
Step 4: Perform numerical simulations to obtain the solutions.
Step 5: Evaluation of the performance and algorithms complexity analysis.
End

In this framework, a robot is denoted by a point in configuration space. The region is formed according to a grid pattern, whereby each node's coordinates and corresponding function values are computed iteratively by applying a numerical method. Different initial temperature values are assigned to the obstacles and boundaries, with the starting point given the highest potential value and the goal point assigned the lowest. The application of the Dirichlet constraint, $\phi| \partial\Omega = c$, where c is constant, has been utilised in the solution of Laplace's equation. Once the harmonic function has been established based on the given boundary conditions, the appropriate path can be determined by tracking the flow of heat using the gradient descent search (GDS) applied to the computed potential values [21]. The descending search reaches the goal point, which goes to the point with the lowest potential values.

2.1 Finite Difference Approximation

Suppose that the solution of $u(x, y)$ of Laplace's equation in two dimensions, which has been set out in Eq. (1). The five-point approximation to the Laplacian can be derived by utilising the central differences approximation of Eq. (3) and (4), generally stated as:

$$\nabla^2 u = \frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} = 0 \tag{3}$$

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

Eq. (5) represents the Gauss-Seidel iteration, which considers all nodes in the mesh points and is executed on a rectangular grid. Thus, the iterative algorithms for the standard five-point finite difference approach can be expressed as:

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

This study focuses on path planning performance between the Successive Overrelaxation (SOR) and log-space harmonic potential methods. Thus, the standard iterative scheme for corresponding methods is needed to implement the SOR and LGS. Therefore, by adding the weighted parameter ω from Eq. (4), the iterative scheme for SOR is expressed as in Eq. (6):

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

Young [22] demonstrated that the value of the weighted parameter ω of the corresponding SOR is between 0 and 2, $0 < \omega < 2$.

2.2 Log Gauss-Seidel

The drawback of past studies on potential fields with harmonic functions is due to numerical precision problems in arithmetic floating-point operations. The problem arises after several iteration cycles with the Dirichlet Boundary Conditions from Eq. (2). The u values of points in the free space are very close to 1. Therefore, Wray *et al.,* [23] overcome the numerical precision problem by introducing the log-space HPF. The log-space mapping uses the features of logarithms and exponentiation, specifically the log-sum-exp algorithm, which is presented below as lse as in Eq. (7). This method is frequently used in the field of machine learning to do the multiplication of many probabilities that are close to zero. It is formally given for any values $y = [y_1, \dots, y_k]^T$:

$$lse(y) = y^* + \log \sum_{i=1}^{k} e^{y_i - y^*} \tag{7}$$

Where generally $y^* = \max\ y_i$. To use this, it is necessary to alter the numerical relaxation into this specific format. First, in the log-space Gauss-Seidel algorithm, an additional variable $v: \mathbb{R}^n \longrightarrow \mathbb{R}$ is introduced for u as in Eq. (8):

$$v(x, y) = \log((1-\delta)\big(1 - u(x, y)\big) + \delta \tag{8}$$

Where $\delta > 0$ is a relatively small positive value that stops the outflow of log 0. Therefore, in this case, u value is the opposite of Eq. (2), and is referred in Eq. (9), where:

$$u(x, y) = \begin{cases} 0, \text{if } (x, y) \text{is at obstacles} \\ 1, \text{if } (x, y) \text{is at the goal point} \end{cases} \tag{9}$$

Therefore, u in the range $[0,1]$ is mapped to v in the range $[\log \delta, 0]$, which contains a significantly greater number of floating-point numbers. In addition to resolving the issue of numerical precision, the second benefit of employing log-space HPF is the ability to directly derive the harmonic potentials from the $v(x, y)$ in log-space. This is possible because the gradient of v is parallel to that of u, but in the opposite direction. This can be seen by $\nabla v = -(1-\delta)\exp(-v)\nabla u$. Hence, the approach for robot navigation involves following the vector field $-\nabla u$ derived from the streamline u. However, we aim to utilise the gradient ascent ∇v from v derived using the log-space method in this case. The updated log-space Gauss-Seidel for t iteration is defined as in Eq. (10):

$$v^{t+1}(x) = \begin{cases} 0, & \text{if } x\epsilon G \\ \log\delta, & \text{if } x\epsilon O \\ lse(v^t) - \log 2n, & \text{otherwise} \end{cases} \tag{10}$$

Next, by applying Gauss-Seidel iteration to Eq. (8), the log-space Gauss-Seidel (LGS) implementation for a 2-dimensional grid [23] is expressed as in Eq. (11):

$$v_{i,j}^{(k+1)} = V^* + \log\begin{bmatrix} \exp\left(v_{i-1,j}^{(k+1)} - V^*\right) + \exp\left(v_{i+1,j}^{(k)} - V^*\right) + \\ \exp\left(v_{i,j+1}^{(k)} - V^*\right) + \exp\left(v_{i,j-1}^{(k+1)} - V^*\right) \end{bmatrix} - \log 2n \tag{11}$$

where,

$$V^* = \max\left(v_{i-1,j}^{(k+1)}, v_{i+1,j}^{(k)}, v_{i,j+1}^{(k)}, v_{i,j-1}^{(k+1)}\right) \tag{12}$$

and n is the dimension. Therefore, Algorithm 2 (Table 2) can represent the execution of the log-space Gauss-Seidel to solve a two-dimensional problem, as denoted by Eq. (11).

Table 2
Algorithm 2: Log-space Gauss-Seidel

1. Establish the start and goal points of the configuration space.
2. Initialising starting point U, $\varepsilon \longleftarrow MAX_{ERROR}$, iteration $\longleftarrow 0$.
3. Determine $V^* \longleftarrow \max\left(v_{i-1,j}^{(k+1)}, v_{i+1,j}^{(k)}, v_{i,j+1}^{(k)}, v_{i,j-1}^{(k+1)}\right)$
4. Compute the points close to the boundary using the direct method via Eq. (11),
$$v_{i,j}^{(k+1)} \longleftarrow V^* + \log\begin{bmatrix} \exp\left(v_{i-1,j}^{(k+1)} - V^*\right) + \exp\left(v_{i+1,j}^{(k)} - V^*\right) + \\ \exp\left(v_{i,j+1}^{(k)} - V^*\right) + \exp\left(v_{i,j-1}^{(k+1)} - V^*\right) \end{bmatrix} - \log 2n$$
5. Verify the convergence test for $\varepsilon \longleftarrow MAX\_ERROR$. If so, move to the next step. Else back to step (3).
6. Execute log GDS to generate the path from start to goal point.

3. Results

The tests were performed on a 2.80 GHz 11th Gen Intel(R) Core (TM) i7 system with 8 GB of RAM. The iterative technique of quantitatively computing the HPF at all points continues until the stopping criterion is reached. The iteration would be terminated when the difference between all points of the current computed HPF values at k^{th} iteration and the previous values at $(k-1)^{th}$ iteration, where the tolerance error $e^{SOR} = 1 \times 10^{-10}$ for SOR, while the tolerance error for LGS is set to be $e^{LGS} = 1 \times 10^{-3}$. The optimal parameter value ω in this simulation is set to 1.80, based on the preliminary result in the range of 1.70 to 1.90. In the case of the SOR method, the computation required a high degree of accuracy to avoid the occurrence of flat areas, commonly referred to as saddle points, which might stop the successful construction of the path. The LGS method required less accuracy since it is sufficient to overcome the numerical precision problem encountered by the SOR method. Tables 3 and 4 present the results of the experiment, specifically the number of iterations (denoted as k) and the execution time in seconds (denoted as t) for each numerical technique employed to compute the harmonic potentials across the entire region for sparse and dense environments, respectively.

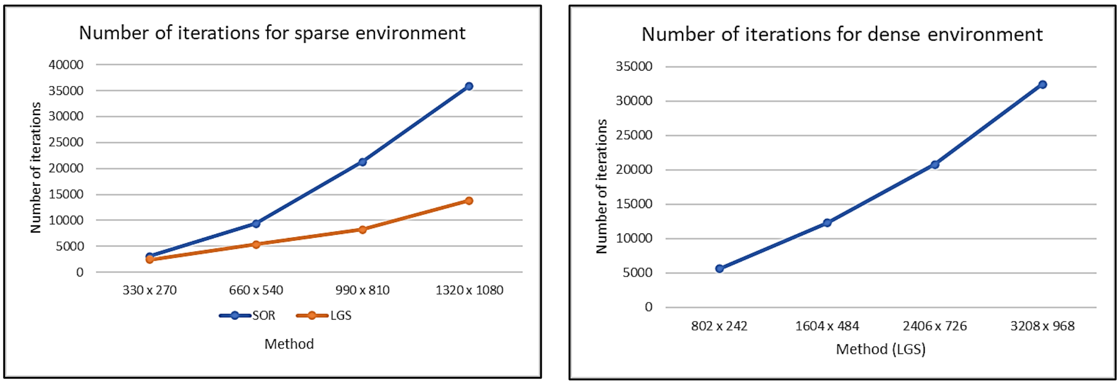
Table 3
Results of the methods evaluated based on the number of iterations and time of execution (in seconds) for the indoor environment (Sparse)

Mesh size	SOR, k	t	LGS, k	t
330 x 270	3063	1.522	2459	9.044
660 x 540	9388	20.877	5361	77.285
990 x 810	21245	104.643	8241	293.052
1320 x 1080	35867	421.546	13845	990.834

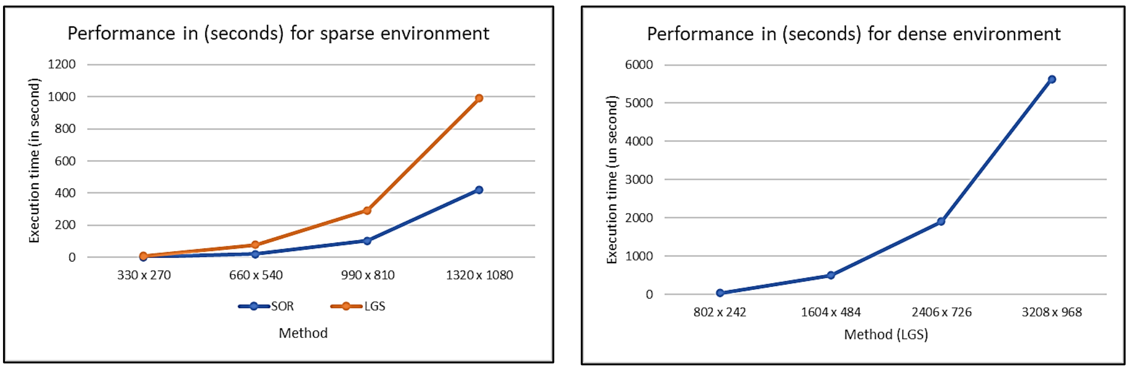
Table 4
Results of the methods evaluated based on the number of iterations and time of execution (in seconds) for the indoor environment (Dense)

Mesh size	SOR, k	t	LGS, k	t
802 x 242	Fail	Fail	5598	41.098
1604 x 484	Fail	Fail	12253	505.804
2406 x 726	Fail	Fail	23787	1909.38
3208 x 968	Fail	Fail	32417	5628.72

The graphs in Figure 1 represent the number of iterations, k, for sparse and dense indoor environments, while Figure 2 shows the execution time, t, for sparse and dense indoor environments, which indicate the output of the proposed method presented in Tables 3 and 4.



(a) (b)
Fig. 1. Number of iterations, k (a) Sparse (b) Dense environments



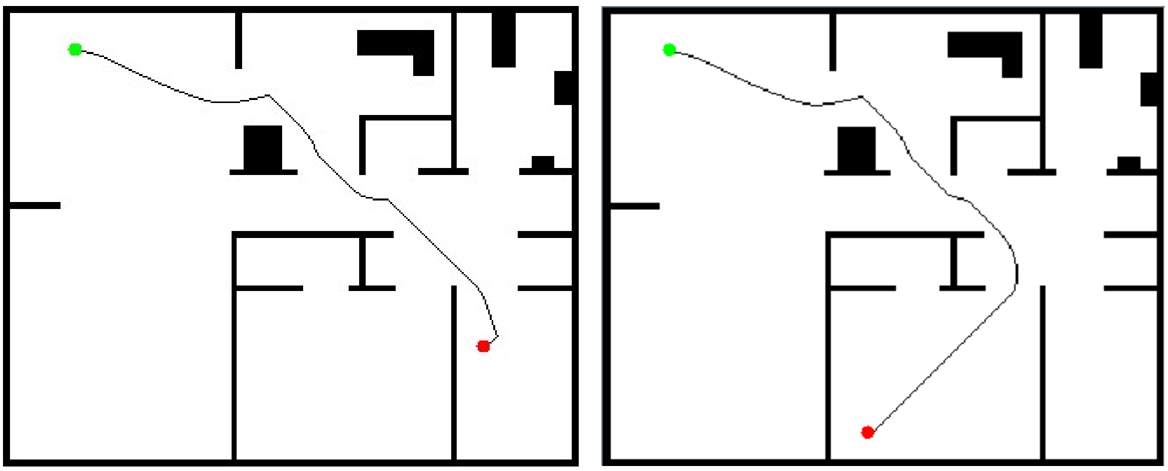
(a) (b)
Fig. 2. Performance of execution time, t (in seconds) (a) Sparse (b) Dense environments

It can be observed from Figures 1 and 2 that an increase in the number of iterations results in a corresponding increase in the duration of each execution. Both graphs for dense environments indicate that the LGS outperformed SOR in terms of iteration count and success in generating safe paths in narrow corridors and difficult regions though it had a slower execution time. Tables 3 and 4 also clearly illustrate this concept. A similar trend can be seen in Figures 1 and 2, as shown by the number of iterations and execution time The LGS iterative scheme appears to be more efficient in terms of iteration number when compared to SOR in a sparse environment, but the time required for SOR to converge is much faster. Only LGS had successfully computed the harmonic potentials for generating paths in dense environments.

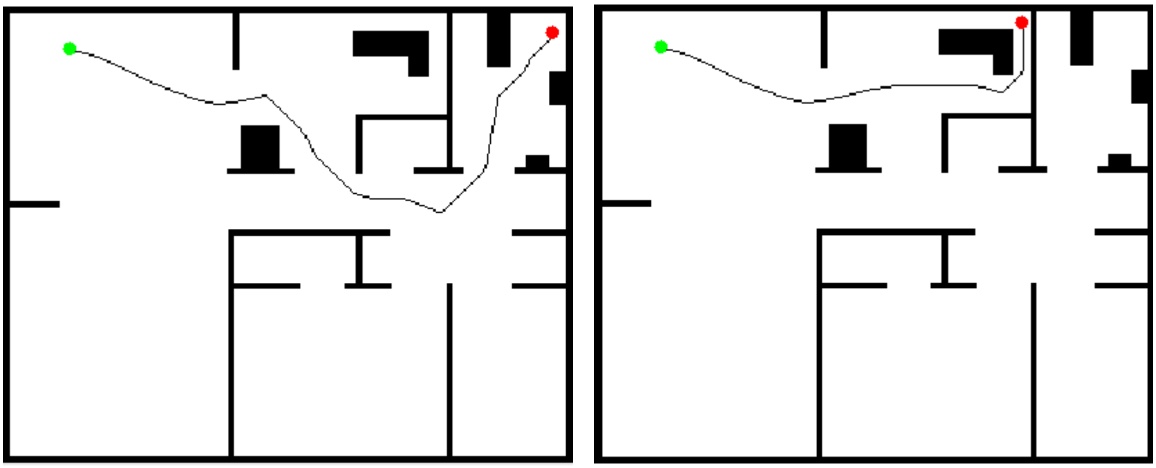
4. Discussion

In this analysis, the environment setup involves two types of maps, which are sparse and dense. The sparse map involves four different sizes: 330 x 270, 660 x 540, 990 x 810, and 1320 x 1080. As for the dense map, it also involves four different sizes: 802 x 242, 1604 x 484, 2406 x 726, and 3208 x 968. Different numbers and forms of obstacles were placed in the pathway of the sparse and dense environment. The initial setup applied the Dirichlet boundary condition, wherein the obstacles and walls were assigned high harmonic potential values, while the target point was assigned the lowest harmonic potential values.

As shown in Figure 3, both pathways for SOR and LGS are complete and yield valid paths for all locations in a sparse environment. The pathway was established through the implementation of the gradient descent search (GDS), starting from the initial point, and stopping at the goal point after obtaining the harmonic potential values. The gradient search for SOR descends because of the condition from Eq. (2), while the gradient LGS ascends based on the condition from Eq. (8). The path is generated quickly, as the algorithm continuously selects the harmonic potential value with the lowest magnitude from the neighbouring points of the current position. The iterative process continues until the desired goal is achieved.



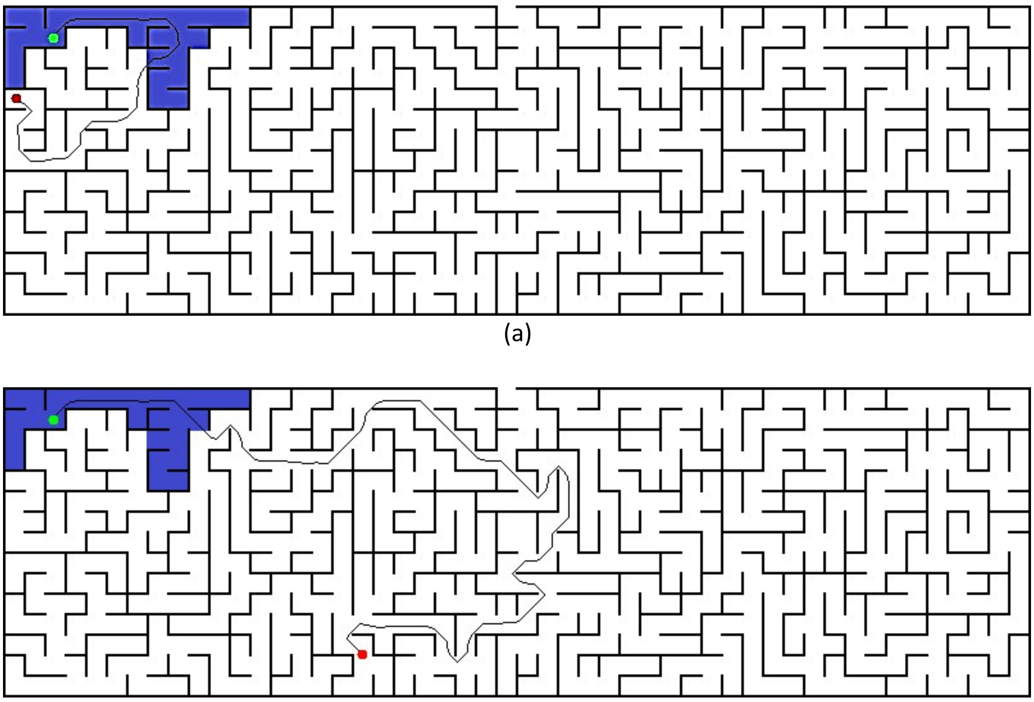
(a) (b)



(c) (d)

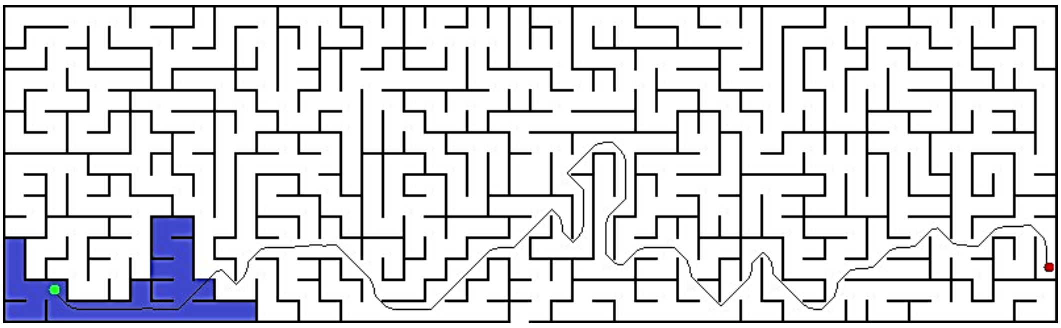
Fig. 3. Construction of paths from several starting points (round/red point) and goal points (round/green point) for sparse environments (a) 330-by-270, (b) 660-by-540, (c) 990-by-810 and (d) 1320-by-1080

As for Figure 4, which is the dense environment, all the pathways for LGS return valid paths for all locations. However, the SOR pathway is only valid within the small blue area and fails to converge properly outside this region due to numerical precision issues. This finding indicates the capability of LGS to compute harmonic potentials for a dense environment that contains challenging and narrow corridors without the occurrence of saddle points.

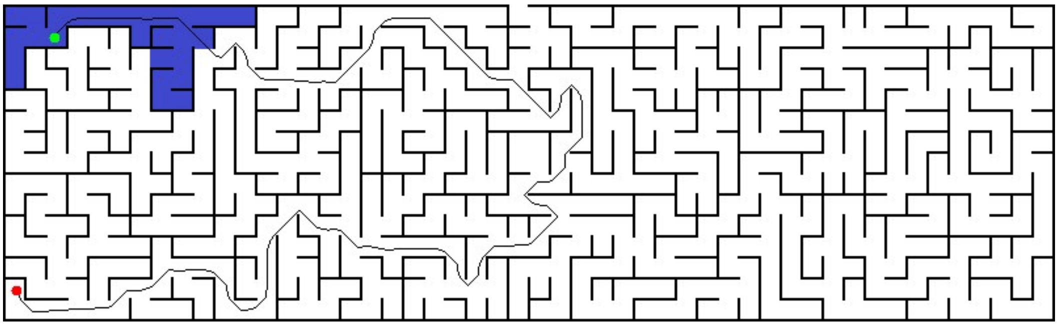


(a)

(b)



(c)



(d)

Fig. 4. Construction of paths from several starting points (round/ red point) and goal points (round/green point) for dense environment (a) 802-by-242 (b) 1604-by-484 (c) 2406-by-726 (d) 3208-by-968

The LGS iterative method demonstrates better performance in achieving the solution with less iterations compared to SOR, as evidenced by the results. The SOR method has better computation time in sparse environments but fails to maintain the same performance in dense environments. This occurs because SOR fails to find valid gradients for larger grids and can only be used in smaller ones such as 256 x 256, due to a numerical precision problem. As such, LGS is more superior than SOR in generating a valid pathway for a dense environment as its performance remains unaffected by the increasing number of obstacles. Likewise, increasing obstacles reduces computational requirements since all cells that are present with the obstacles are disregarded during the iteration. Since the LGS algorithm is free of local minima, the harmonic potentials are obtained across all regions so that the path can be generated successfully even in dense environments. This ensures a safe and smooth path from the beginning to the end goal, even in the presence of challenging obstacles.

4.1 Computational Complexity

This section analyses the computational complexity of the two iterative approaches used in this research. One unit of computational time is expected to be required for each arithmetic operation, including addition and multiplication. The computational costs for the path tracing procedure of the GDS algorithm are excluded since it requires the same amount of computation for both SOR and LGS methods. Table 5 presents the total number of arithmetic operations required by both algorithms. Theoretically, as the algorithm's computational complexity drops, the number of iterations decreases, reducing CPU time. LGS requires less iteration since the tolerance error is greater than SOR ($\mathrm{e}^{\mathrm{LGS}} > \mathrm{e}^{\mathrm{SOR}}$). However, it took a longer execution time compared to SOR due to higher number of arithmetic operations per iteration, as can be seen in Table 5.

Table 5
Number of arithmetic operations per iteration for SOR and LGS

Methods	ADD/SUB	MUL/DIV
SOR	4M	2M
LGS	9M	5M

Where $M = (H \times W) - P$; P is the number of points occupied by obstacles, while the terms H and W are the height and width of the map.

5. Conclusions

This paper shows that the simulations using harmonic potentials based on the gradient of LGS can generate nearly smooth paths, optimal, and free from local minima. This method utilises the numerical solution of a boundary value problem for Laplace's equation with Dirichlet boundary conditions. It makes it practical and highly adaptable to domains with irregular boundaries, consisting of obstacles of any shape. With regard to the aim of this study, which is to overcome the high precision issue in the numerical calculation of harmonic potentials, it was discovered that the iterative method for the LGS is efficient in path planning for complex indoor environments. This finding is consistent with Wray *et al.,* [23], which stated that LGS improves scalability vastly. Although it exhibited a longer execution time relative to alternative methods such as SOR, the LGS method proved to be more effective in producing safe paths within dense environments, especially in narrow corridors. Hence LGS method is deemed superior in precision and safety for high-complexity indoor environments. Moreover, the adaptability of the method indicates its potential relevance to higher-dimensional path planning, thereby broadening its application in advanced robotics and autonomous systems. Future researchers should explore mobile robot navigation enhancements by refining numerical methods, such as block iteration, to reduce computational costs. These could expedite the convergence rate and reduce the computational costs of the iterative process. In addition, assessing the scalability and adaptability of the method in denser and more dynamic environments may enhance its applicability in real-time robotics and other fields requiring precise navigation solutions.

References

[1] Dahalan, A'qilah A., Azali Saudi, and Jumat Sulaiman. "Pathfinding for mobile robot navigation by exerting the Quarter-Sweep Modified Accelerated Overrelaxation (QSMAOR) Iterative approach via the Laplacian Operator." *Modelling and Simulation in Engineering* 2022, no. 1 (2022): 9388146. https://doi.org/10.1155/2022/9388146

[2] Li, Yongcheng, Weimin Du, Zesheng Tang, and Bo Zhang. "Computer animation of robot motion with collision free path planning." In *Creating and Animating the Virtual World*, p. 149-161. Springer Japan, 1992. https://doi.org/10.1007/978-4-431-68186-1_10

[3] Sheng, Xuejun. "Motion planning for computer animation and virtual reality applications." In *Proceedings Computer Animation'95*, p. 56-66. IEEE, 1995. https://doi.org/10.1109/CA.1995.393547

[4] Larsen, Lars, and Jonghwa Kim. "Path planning of cooperating industrial robots using evolutionary algorithms." *Robotics and Computer-Integrated Manufacturing* 67 (2021): 102053. https://doi.org/10.1016/j.promfg.2018.10.048

[5] Zhang, Haojian, Yunkuan Wang, Jun Zheng, and Junzhi Yu. "Path planning of industrial robot based on improved RRT algorithm in complex environments." *IEEE Access* 6 (2018): 53296-53306. https://doi.org/10.1109/ACCESS.2018.2871222

[6] Shamsudin, Abu Ubaidah, Puteri Alisha Balqis Mohd Sharif, Zubair Adil Soomro, Ruzairi Abdul Rahim, Ahmad Athif Mohd Faudzi, Wan Nurshazwani Wan Zakaria, Mohamad Heerwan Peeie, and Carl John Salaan. "Autonomous Navigation Robot using Slam and Path Planning Based on a Single RP-LIDAR." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 53, no. 2 (2024): 161-169. https://doi.org/10.37934/araset.53.2.161169

[7] Shi, Chaojian, Mingming Zhang, and Jing Peng. "Harmonic potential field method for autonomous ship navigation." In *2007 7th International Conference on ITS Telecommunications*, p. 1-6. IEEE, 2007. https://doi.org/10.1109/ITST.2007.4295916

[8] Song, Changhao, Kai Wang, and Jack CP Cheng. "BIM-aided scanning path planning for autonomous surveillance UAVs with LiDAR." In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 37, p. 1195-1202. IAARC Publications, 2020. https://doi.org/10.22260/ISARC2020/0164

[9] Connolly, Christopher I., and Roderic A. Grupen. "The applications of harmonic functions to robotics." *Journal of robotic Systems* 10, no. 7 (1993): 931-946. https://doi.org/10.1002/rob.4620100704

[10] Kim, Jin-Oh, and Pradeep Khosla. "Real-time obstacle avoidance using harmonic potential functions." (1992): 1-27. https://doi.org/10.1109/ROBOT.1991.131683

[11] Motonaka, Kimiko, Tsuyoshi Goto, Keigo Watanabe, and Shoichi Maeyama. "Kinodynamic motion planning for a two-wheeled drive mobile robot using a Harmonic Potential Field." In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, p. 6140-6145. IEEE, 2016. https://doi.org/10.1109/IECON.2016.7793160

[12] Kazemi, Moslem, Mehran Mehrandezh, and Kamal Gupta. "Sensor-based robot path planning using harmonic function-based probabilistic roadmaps." In *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, p. 84-89. IEEE, 2005. https://doi.org/10.1109/ICAR.2005.1507395

[13] Masoud, Ahmad A. "Managing the dynamics of a harmonic potential field-guided robot in a cluttered environment." *IEEE Transactions on Industrial Electronics* 56, no. 2 (2008): 488-496. https://doi.org/10.1109/TIE.2008.2002720

[14] Andrilli, Stephen, and David Hecker. *Elementary Linear Algebra*. Academic Press, 2022. https://doi.org/10.1016/B978-0-12-374751-8.00014-7

[15] Mohammedali, Khalid Hammood, Noor Atinah Ahmad, and Fadhel S. Fadhel. "He's variational iteration method for solving Riccati matrix delay differential equations of variable coefficients." In *AIP Conference Proceedings*, 1830, no. 1. AIP Publishing, 2017. https://doi.org/10.1063/1.4980892

[16] Al-jiz, Khalid Hammood, Noor Atinah Ahmad, and Fadhel Subhi Fadhel. "Variational iteration method for solving Riccati matrix differential equations." *Indonesian Journal of Electrical Engineering and Computer Science* 5, no. 3 (2017). https://doi.org/10.11591/ijeecs.v5.i3.pp673-683

[17] Szulczyński, Paweł, Dariusz Pazderski, and Krzysztof Kozłowski. "Real-time obstacle avoidance using harmonic potential functions." *Journal of Automation Mobile Robotics and Intelligent Systems* 5, no. 3 (2011): 59-66.

[18] Tsaryova, O. O., V. Yu Popov, and D. N. Suzansky. "A method for constructing the trajectory for an unmanned aerial vehicle in a city." *Moscow University Physics Bulletin* 71 (2016): 161-167. https://doi.org/10.3103/S0027134916020107

[19] Waydo, Stephen, and Richard M. Murray. "Vehicle motion planning using stream functions." In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, 2, p. 2484-2491. IEEE, 2003. https://doi.org/10.1109/ROBOT.2003.1241966

[20] Liu, Chang-an, Zhenhua Wei, and Chunyang Liu. "A new algorithm for mobile robot obstacle avoidance based on hydrodynamics." In *2007 IEEE International Conference on Automation and Logistics*, p. 2310-2313. IEEE, 2007. https://doi.org/10.1109/ICAL.2007.4338962

[21] Connolly, Christopher I., J. Brian Burns, and Rich Weiss. "Path planning using Laplace's equation." In *Proceedings., IEEE International Conference on Robotics and Automation*, p. 2102-2106. IEEE, 1990. https://doi.org/10.1109/ROBOT.1990.126315

[22] Young, David. "Iterative methods for solving partial difference equations of elliptic type." *Transactions of the American Mathematical Society* 76, no. 1 (1954): 92-111. https://doi.org/10.1090/S0002-9947-1954-0059635-7

[23] Wray, Kyle Hollins, Dirk Ruiken, Roderic A. Grupen, and Shlomo Zilberstein. "Log-space harmonic function path planning." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 1511-1516. IEEE, 2016. https://doi.org/10.1109/IROS.2016.7759245