Penerbit Akademia Baru

Journal of Advanced Research in Computing and Applications

Journal homepage: https://akademiabaru.com/submit/index.php/arca/index
2462-1927

Path Navigation in Designated Environment using Modified TOR

A'Qilah Ahmad Dahalan[1,*], Azali Saudi[2], Jumat Sulaiman[3]

[1] Department of Mathematics, Universiti Pertahanan Nasional Malaysia, 57000 Kuala Lumpur, Malaysia
[2] Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88450 Kota Kinabalu, Malaysia
[3] Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88450 Kota Kinabalu, Malaysia

ARTICLE INFO

Article history:
Received 16 January 2025
Received in revised form 19 February 2025
Accepted 3 March 2025
Available online 10 March 2025

Keywords:
Laplace's equation; Finite Difference Method; Over-Relaxation Iterative Techniques; path planning; obstacle avoidance

ABSTRACT

Research in path navigation has seen significant advancements, particularly focusing on generating collision-free paths for agents moving within designated environments. Despite these developments, achieving smooth and efficient navigation remains a critical challenge. This study addresses the problem by introducing the Modified Two-Parameter Over-Relaxation (MTOR), a novel numerical iterative approach designed to enhance navigation efficiency. The MTOR method is applied to solve Laplace's equations, producing harmonic functions that serve as potential fields for guiding agents. These harmonic functions are integrated into a gradient descent scheme to construct smooth and collision-free paths for agents moving through the designated environment. The study provides a comprehensive formulation of the MTOR iterative method and evaluates its performance through extensive numerical experiments and simulations. Results demonstrate that the MTOR method significantly outperforms existing approaches in terms of computational efficiency and path quality. The main contribution of this research lies in the development of the improvised iterative method, the MTOR scheme, which offers a robust and efficient solution for path navigation in designated environments.

1. Introduction

Path navigation has emerged as a critical area of research, driven by the growing demand for efficient and reliable navigation systems in various applications such as robotics, unmanned aerial vehicles, and autonomous vehicles. The ability to plan collision-free trajectories in complex, obstacle-laden environments is fundamental to achieving robust navigation performance. This has led to significant advancements in leveraging artificial intelligence, machine learning, and optimization algorithms to enhance navigation capabilities with higher accuracy and lower computational costs.

Harmonic functions, derived as solutions to Laplace's equation, have long been recognized for their advantageous properties in the applications of robotics and automation [1]. Early studies by Connolly and Gruppen [2] as well as Akishita *et al.,* [3], established the use of harmonic function solutions in solving navigational challenges. These functions have been particularly effective in

* *Corresponding author.*
E-mail address: a.qilah @upnm.edu.my (A'Qilah Ahmad Dahalan)

https://doi.org/10.37934/arca.38.1.3950

addressing boundary value problems [4] and have been extensively applied in real-time obstacle avoidance and smooth path generation [5,6].

Traditional numerical methods, such as Jacobi, Gauss-Seidel (GS), and Successive Over Relaxation (SOR), were employed to tackle path planning problems, with SOR demonstrating notably faster performance [7]. Subsequently, the finite element method has been successfully applied to solve Laplace's equation for robotic motion [8]. However, analytical solutions to harmonic functions often face limitations when dealing with arbitrarily shaped obstacles or dynamic environments [5,6].

Recent studies have focused on enhancing the navigation efficiency of numerical methods and integrating advanced algorithms and computational techniques [9,10]. For instance, the development of potential field methods has provided robust frameworks for generating smooth and collision-free paths in dynamic environments [11]. Additionally, the integration of machine learning techniques, such as reinforcement learning, has enabled systems to learn and adapt to new environments autonomously, further enhancing navigation capabilities and precision [12]. More recently, a variant of over-relaxation families known as the Two-Parameter Over Relaxation (TOR) method has emerged, and has shown promise in solving partial differential equations, offering improved performance for path-planning tasks [13,14].

Motivated by these advancements and the need to overcome the challenges associated with existing methods, this paper proposes a novel approach for autonomous agent navigation based on heat transfer theory. By utilizing a numerical potential function within the configuration area (C-area), the method models the heat transfer problem through Laplace's equation. The resulting harmonic functions, representing temperature distributions, are used to simulate smooth and collision-free trajectories inside the C-area. To enhance computational efficiency and path accuracy, the improvised version of TOR – the Modified Two-Parameter Over-Relaxation (MTOR) iterative method is introduced, leveraging a red-black block iterative scheme. This approach demonstrates significant potential for improving path planning performance in diverse environments while reducing computational overhead, as evidenced by the experimental results presented in this study, making it a valuable contribution to the field of path navigation.

2. Methodology

2.1 Path-Planning Scheme

The primary objective of path navigation is to establish a collision-free trajectory for a machine to travel from an initial point to a specified target location within a predetermined environment. Leveraging the mathematical principles underlying Laplace's solutions, a methodology for generating paths is derived, providing valuable guidance for the agent's navigation. This approach to path navigation is conceptualized through an analogy with heat distribution, as elaborated in the next section.

In simple terms, gradient descent is the process of locating the minimal function that is proportionate to the gradient's negative at the current position. As a result, the gradient descent search (GDS) path from the current position to a specific location is traced by the lower potential values' steepest fall. The gradient descent method, according to Rumelhart *et al.,* [13], is basically a searching approach that can promise the discovery of a local minimum for a specific task. While these local minima may not always be the optimal solution, it frequently satisfies the requirements. The ability to trace a path using gradient information is demonstrated by the existence of potential gradients over space using GDS. This study mainly uses the harmonic function for GDS. If the harmonic function is comprehensive, GDS works well; otherwise, it might get stuck in flat areas or local minima.

2.2 Harmonic Functions

Laplace's Eq. (1) is mathematically satisfied by the harmonic function on a domain Ω confined in region R^n, where *n* is the dimension and x_i represents the Cartesian coordinates of *i*-th, and its solution is constrained by the Dirichlet condition $\phi \mid \partial\Omega = c$, where c is constant. The domain Ω in this study composes the region boundaries, starting locations, obstacles, and target points for path construction.

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

The min-max rule is valid for harmonic functions, so it logically follows that there are no sudden appearances of local minima within the solution domain [2]. Furthermore, the Gauss Integral Theorem [16] affirms that a balance between inward and outward flow at the boundary of any volume within the solution domain (excluding barriers and the target point), ensures there is always a way to escape any situation. A harmonic function adheres to the min-max principle and possesses a gradient vector field with zero curls. This means that only saddle points can emerge as critical points. Exploring the region around such a critical point can help to find an escape route. Additionally, any path disruptions caused by these points result in a smooth path throughout.

2.3 Heat Transfer Analogy

In this model, a robot is represented by a point in the C-area, which is constructed in a grid layout. The coordinates and function values of each node are computed iteratively using a numerical method. Initial temperature values are assigned to the boundaries and obstacles, with a high potential value set for the initial location and the lowest for the target point. By following the heat flow generated by the gradient descent method and using estimated potential values, the ideal path is discovered the instant the harmonic function has been constructed within these boundary constraints. This search leads to the location with the lowest potential value, implying the target point. The descent process involves a sequence of points with decreasing potential values, and the coordinates and temperature gradients of nodes obtained from finite difference analysis provide the path's trajectory. Essentially, harmonic potentials are weighed across the C-area, containing obstacles, to map out the path for a point robot from any starting position to a specified target.

2.4 Red-Black Strategy

The modified variants of the proposed iterative scheme incorporate a red-black ordering strategy, a technique commonly employed in numerical methods for solving partial differential equations and sparse matrix problems for decades. This strategy is depicted through the computational grid and computational molecules in Figure 1 and Figure 2, respectively. The core theory of the red-black ordering strategy is to compute iterations layer by layer. This means that the iterative approach will prioritize computing the red nodes first, followed by calculating the black nodes, in the grid points within the C-area.

The red-black ordering approach has a long story, dating back to 1946 when William [17] highlighted its application in solving linear systems of equations arising from Markov chain problems. Since then, it has become a widely recognized and well-established technique in numerical computation. Recent studies [13,18,19] have further explored and refined its usage, underscoring its continued relevance and effectiveness in modern computational methods.

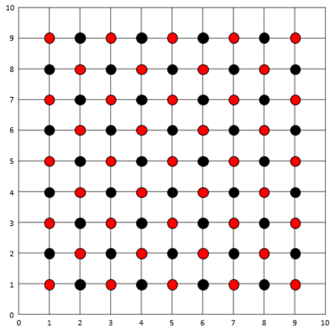
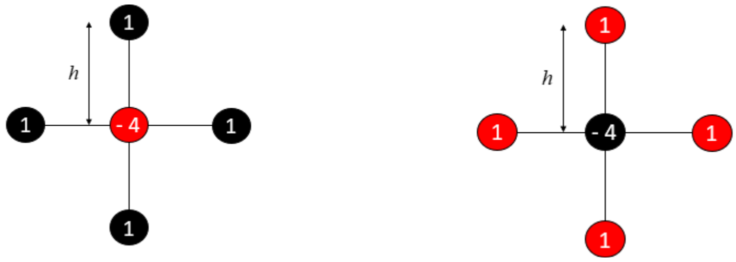


Fig. 1. Computational nodes of red-black strategy



(a) red node computation (b) black node computation

Fig. 2. Computational molecules for finite difference approximation (*h* represents the spacing between node points in *i* and *j* directions of a rectangular grid)

2.5 Formulation of MTOR Iterative Method

The modified variants of the Accelerated Over Relaxation (AOR) and TOR methods can decrease to Jacobi extrapolation or modified SOR (MSOR) through selective acceleration and relaxation matrix choices based on specific parameters corresponding to matrix A's row blocks. These modified over-relaxation methods all incorporate a red-black strategy and utilize distinct supplemental weighted relaxation parameters compared to one another.

The general formulation for the MSOR method can be stated as follows:

$$u_{i,j}^{(k+1)} = \frac{\omega}{4}\left[u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right] + \left(1-\omega\right)u_{i,j}^{(k)}, \tag{2a}$$

for red nodes, and the black nodes as

$$u_{i,j}^{(k+1)} = \frac{\omega'}{4}\left[u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k+1)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k+1)}\right] + \left(1-\omega'\right)u_{i,j}^{(k)}. \tag{2b}$$

Next, the formulation of the modified AOR (MAOR) method, can be described in red nodes as follows:

$$u_{i,j}^{(k+1)} = \frac{\omega}{4}\left[u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right] + \left(1-\omega\right)u_{i,j}^{(k)}, \tag{3a}$$

while the black nodes are written as

$$\begin{aligned} u_{i,j}^{(k+1)} &= \frac{r}{4}\left[u_{i-1,j}^{(k+1)} - u_{i-1,j}^{(k)} + u_{i,j-1}^{(k+1)} - u_{i,j-1}^{(k)}\right] + \frac{r}{4}\left[u_{i+1,j}^{(k+1)} - u_{i+1,j}^{(k)} + u_{i,j+1}^{(k+1)} - u_{i,j+1}^{(k)}\right] \\ &\quad + \frac{\omega'}{4}\left[u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right] + \left(1-\omega'\right)u_{i,j}^{(k)} \end{aligned} . \tag{3b}$$

The TOR method is an extension of the AOR method. The MTOR method, which is a modified version of TOR, introduces four weighted parameters denoted as r, r', ω and ω'. These weighted parameters have no generic calculation that can yield optimal values. Typically, r, r' and ω' are selected to closely match to the corresponding SOR's ω value. All four possible acceleration constants in this study are defined within the range of $\left[1,2\right)$ [20,21]. The formulation of the MTOR approach is expressed below.

For red nodes:

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

For black nodes:

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

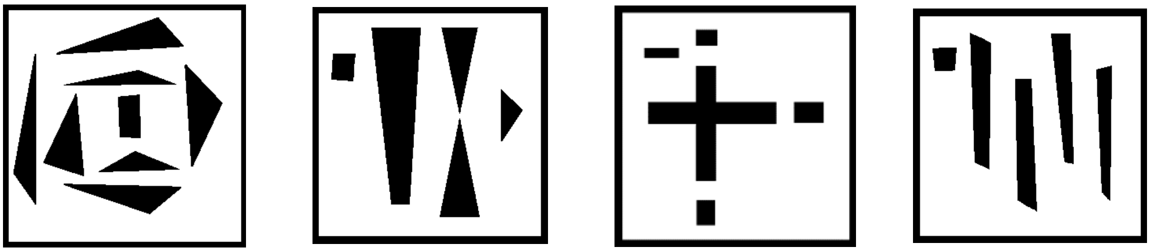
Therefore, the depiction of the red-black MTOR technique, implemented to solve the 2-dimensional Laplace's problem outlined in Eq. (1) using Eq. (4a) and (4b), is detailed in Algorithm 1.

Algorithm 1
Red-Black MTOR scheme

i. Set up the C-area with designated start and target location.
ii. Initializing starting location $u, \varepsilon \leftarrow 10^{-15}, iteration \leftarrow 0$.
iii. Compute for all non-occupied red node points
$$u_{i,j}^{(k+1)} \leftarrow \frac{\omega}{4}\left[u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right] + \left(1-\omega\right)u_{i,j}^{(k)} .$$
iv. Compute for all non-occupied black node points
$$\begin{aligned} u_{i,j}^{(k+1)} &\leftarrow \frac{\omega'-r}{4}\left[u_{i,j}^{(k)} + u_{i+1,j-1}^{(k)}\right] + \frac{r}{4}\left[u_{i,j}^{(k+1)} + u_{i+1,j-1}^{(k+1)}\right] \\ &\quad + \frac{\omega'-r'}{4}\left[u_{i+1,j}^{(k)} + u_{i+1,j+1}^{(k)}\right] + \frac{r'}{4}\left[u_{i+1,j}^{(k+1)} + u_{i+1,j+1}^{(k+1)}\right] + \left(1-\omega'\right)u_{i,j}^{(k)} . \end{aligned}$$
v. Verify the convergence test for $\left\| u^{(k+1)} - u^{(k)} \right\| \le \varepsilon$. If yes, go to (vi). Otherwise, return to (iii).
vi. Execute GDS for constructing the path.

3. Results and Discussion

The study evaluated a stationary C-area with varying obstacles across four distinct scenarios (see Figure 3), following the framework established by Shiang [22]. Each event included six different area sizes (by pixels): 300×300, 600×600, 900×900, 1200×1200, 1500×1500, and 1800×1800. The experiments were conducted using an AMD A10 machine with 8 GB of memory running at 2.50 GHz, along with a 2D robot simulator built by the author [23]. The iteration process involved numerically evaluating temperature values at all points until a stopping criterion was met. The loop continued until temperature values no longer changed significantly between iterations k and $k+1$, indicated by an extremely small difference in measurement values ($\varepsilon = 10^{-15}$). Such a high level of accuracy was crucial to prevent saddle points or flat areas from impacting the creation of pathways.



Event 1 Event 2 Event 3 Event 4

Fig. 3. Paths generation from different starting to the target points in a predetermined environment

Table 1 and Table 2 present the iteration numbers and execution times in seconds required for each method used in the experiments. While Table 3 gives a list of the optimal values used throughout the experiments. It is evident that the MTOR iterative approach demonstrated exceptional performance in comparison with the other techniques proposed. Note that $N \times N$ represents the size of the grid mesh, for example, $N = 300$.

Table 1
Iterations number for the proposed iterative schemes

		$N \times N$					
	Methods	300	600	900	1200	1500	1800
Event 1	MSOR	1583	7557	16697	29132	44800	63671
	MAOR	1524	7311	16069	28188	43396	61685
	MTOR	1593	7610	16711	24705	41644	59216
Event 2	MSOR	2097	8323	18307	31931	49131	69822
	MAOR	1872	7542	16617	28982	35351	37356
	MTOR	1765	6337	14028	21510	28655	24165
Event 3	MSOR	3402	13814	31194	54363	83604	131946
	MAOR	3023	12395	28037	48890	75154	106841
	MTOR	2623	10927	24772	43228	66498	94552
Event 4	MSOR	2395	9411	20667	36037	55428	78781
	MAOR	2169	8623	18949	33056	50864	72308
	MTOR	1933	7778	17137	29913	46048	65470

Table 2
CPU time in seconds for the proposed iterative schemes

		$N \times N$					
	Methods	300	600	900	1200	1500	1800
Event 1	MSOR	6.72	240.99	1227.39	4082.35	9588.90	19327.49
	MAOR	7.44	247.99	1295.65	4330.56	10208.13	19265.66
	MTOR	7.96	259.96	1327.50	3692.53	9545.36	17988.62
Event 2	MSOR	9.36	269.68	1355.34	4329.63	8601.34	18314.85
	MAOR	9.30	267.18	1360.64	4342.87	6977.82	11888.25
	MTOR	8.45	218.08	1140.14	3172.98	5436.88	6864.25
Event 3	MSOR	14.40	462.03	2361.08	7957.70	16036.74	41566.73
	MAOR	15.35	450.60	2420.88	7800.25	16291.28	38068.00
	MTOR	13.02	390.81	2100.51	6780.01	13870.02	32962.68
Event 4	MSOR	9.78	309.74	1576.44	5150.07	11768.10	23502.56
	MAOR	9.83	309.98	1581.29	5163.24	10231.80	23211.15
	MTOR	9.35	273.09	1441.78	4515.76	8981.92	19880.94

Table 3
Grid search of relaxation parameters values

Methods	ω	ω'	r	r'
MSOR	1.83	1.81	-	-
MAOR	1.82	1.81	1.84	-
MTOR	1.80	1.81	1.84	1.85

The results obtained from the proposed approaches, as indicated in Tables 1 and 2, are visually represented in Figure 4 for the number of iterations, and Figure 5 for the CPU time. Both figures illustrate that the execution time increases proportionally with the number of iterations. Although the iteration count and time taken for MTOR varied slightly from previous methods, in the Event 1 region area, particularly in smaller region sizes, the red-black block MTOR iterative scheme demonstrated significantly greater efficiency compared to other proposed approaches. The graphs in both figures exhibit a consistent pattern, with MTOR steadily achieving the lowest values compared to MAOR and MSOR, as reflected in the results table. Hence, these findings suggest that the MTOR iterative scheme offers significant improvement with regard to iteration numbers as well as time taken in contrast to the other suggested approaches.

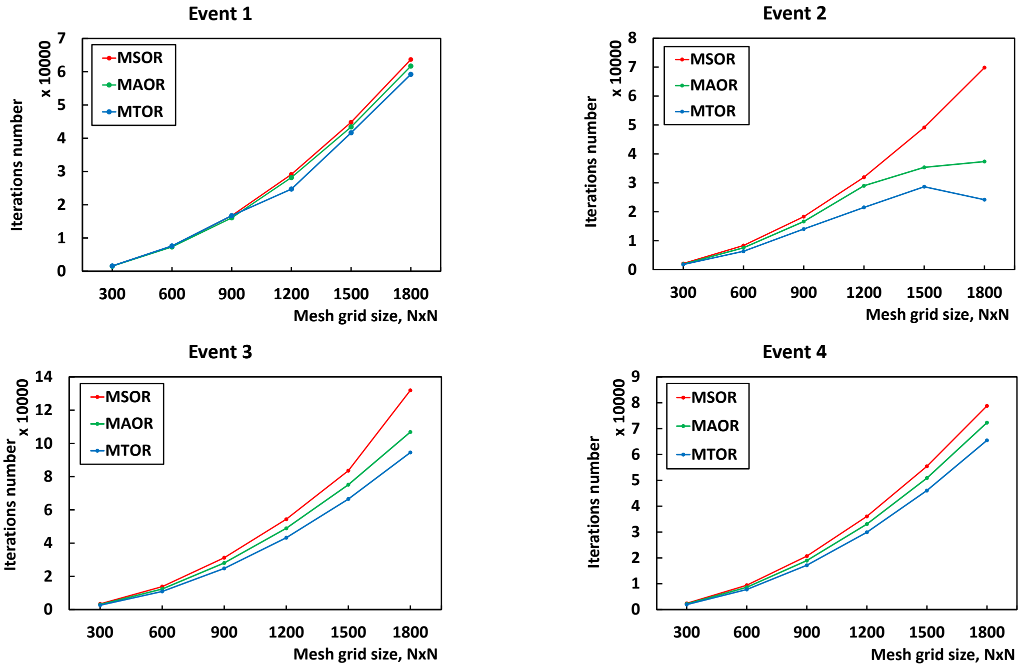


Fig. 4. Graph depicting performance relative to the number of iterations

Algorithm 1 implements an improved adaptation of potential field approaches. In this method, the target point and obstacles act as charged surfaces, creating an overall potential that exerts imaginary forces on the point robot. These imaginary forces attract the point robot toward the target while repelling it from obstacles [24]. As the point robot gets closer to its target, it follows the negative gradient to avoid obstacles. To prevent issues with local minima, this study leverages the harmonic function [2]. Additionally, Algorithm 1 significantly improves computational efficiency by using the red-black relaxation scheme, which speeds up Laplace's equation solution in answering the path navigation challenge.

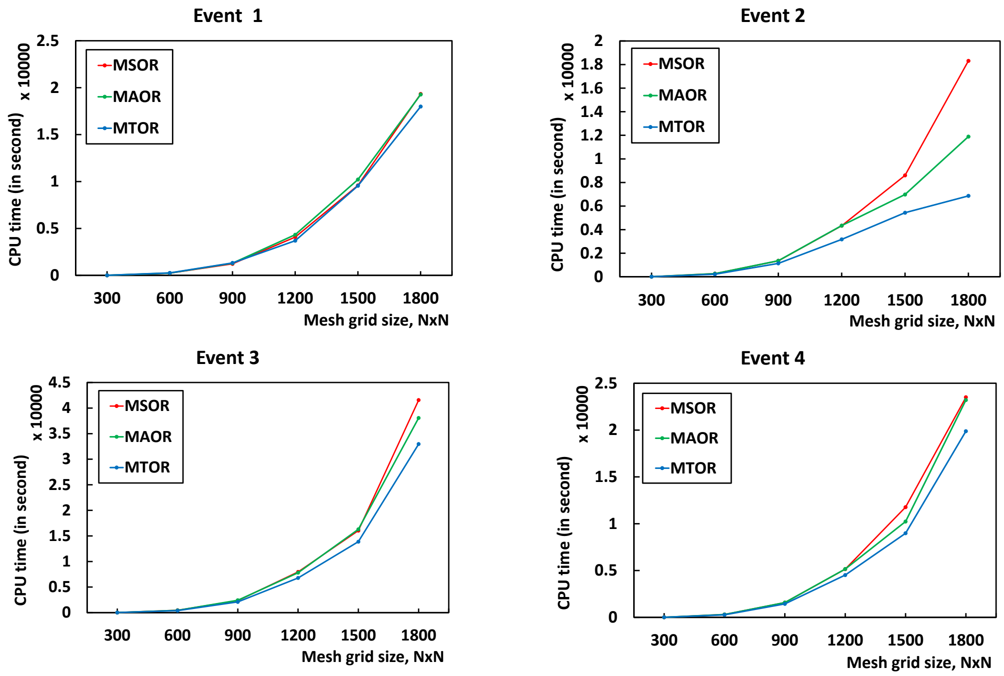
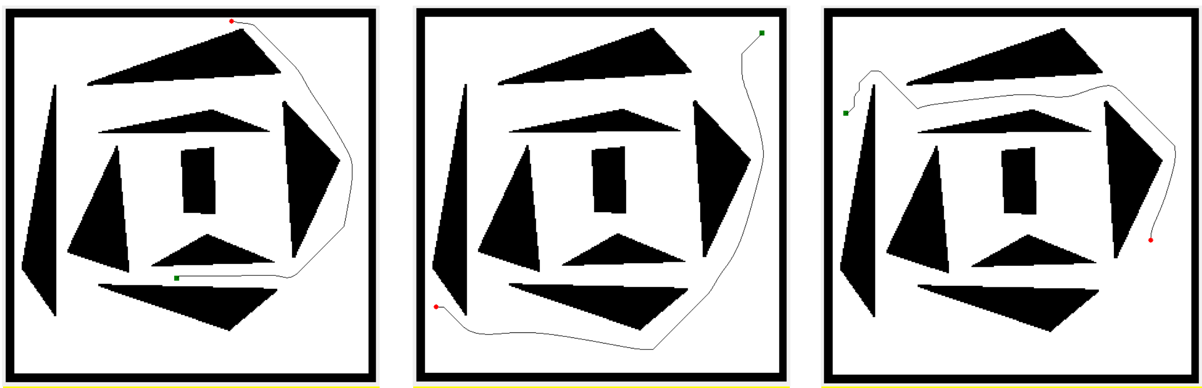


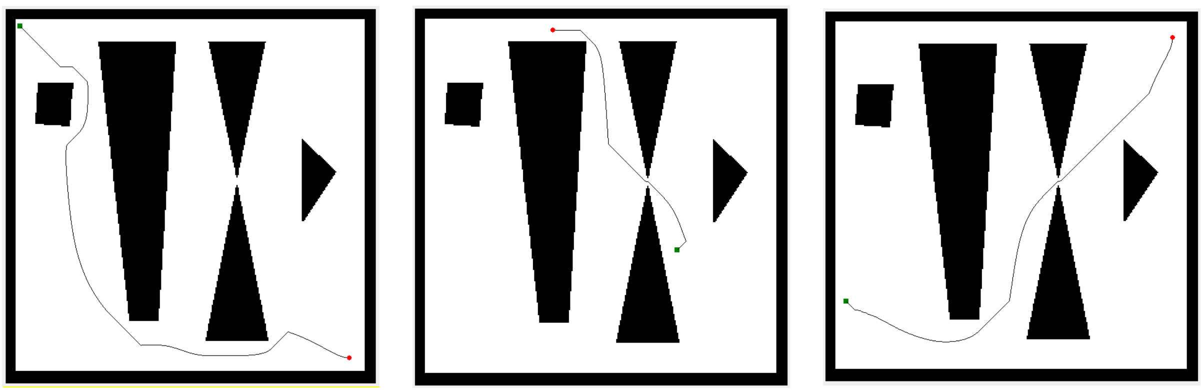
Fig. 5. Graph depicting performance relative to the CPU time in second

Recent advancements in path navigation, particularly those leveraging machine learning and advanced optimization techniques, have focused on enhancing the efficiency and accuracy of autonomous navigation systems. The integration of deep reinforcement learning (DRL) into multi-agent pathfinding has significantly improved the ability of systems to navigate complex environments by optimizing paths and minimizing collisions [9,10]. Similarly, advancements in sensor fusion, such as the use of Kalman filters and particle filters, have improved environmental perception and localization, which are critical for accurate navigation [25]. The MTOR approach, with its red-black block iterative scheme, aligns well with these advancements by providing a robust computational method for solving Laplace's equation, which is essential for generating smooth potential fields used in navigation. This method not only reduces computational overhead but also enhances path accuracy, addressing the same core goals of efficiency and precision seen in recent DRL and sensor fusion studies.

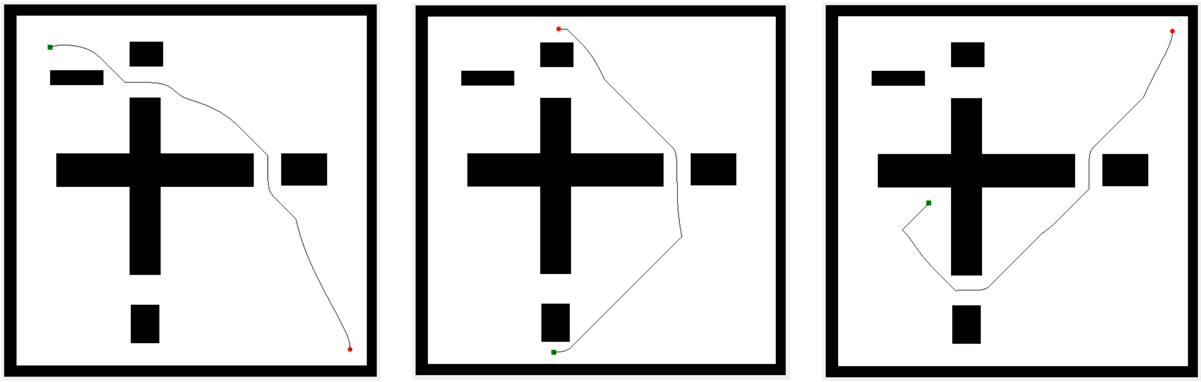
Moreover, the focus on iterative methods for path navigation, as seen in the improvements brought by MTOR over traditional TOR, echoes the broader trend in the field where iterative optimization and machine learning methods are increasingly employed to refine navigation algorithms. By demonstrating significant efficiency gains and computational savings, MTOR contributes to the ongoing efforts to make autonomous navigation systems more practical and effective in real-world applications. Overall, the MTOR approach with the red-black block iterative scheme represents a valuable enhancement in path navigation techniques, aligning with and advancing the current state-of-the-art methods in the field.



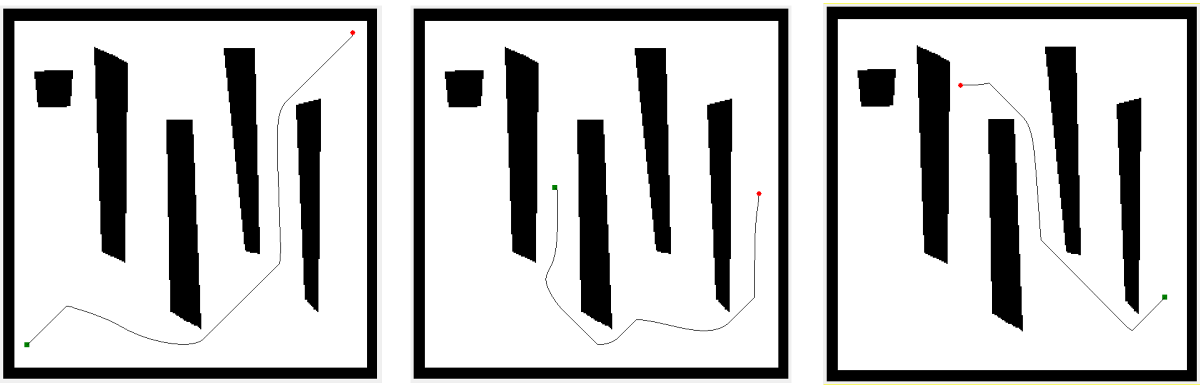
(a) Event 1



(b) Event 2



(c) Event 3



(d) Event 4

Fig. 6. Paths generation from different starting to the target points in a predetermined environment

4. Conclusions

This research makes a significant contribution to the field of robot path navigation by introducing the Modified Two-Parameter Over Relaxation (MTOR) scheme, a novel numerical approach that integrates the red-black ordering strategy into over-relaxation methods. The MTOR scheme has demonstrated substantial improvements in execution performance and computational efficiency, outperforming existing methods such as the Modified Successive Over Relaxation (MSOR) technique. It's worth noting that MTOR performs better than MSOR, reducing the iteration count by 15-25% and processing time by approximately 8-15%, as evidenced by experimental results. One of the standout contributions of this research is the innovative use of accelerated weighted parameters for respective nodes, enabling the MTOR technique to achieve faster and more efficient path computation. Moreover, the method efficiently handles environments with varying numbers of obstacles, as computational performance improves when larger areas are occupied by obstacles (regions affected by obstacles are ignored), effectively reducing the computing domain and resource requirements. The integration of the red-black scheme into the iterative process, detailed in Algorithm 1, marks another key advancement. This approach leverages the inherent strengths of red-black ordering to prioritize calculations and enhance overall computational flow, establishing it as a robust solution for solving robot path navigation problems. By demonstrating the feasibility of solving complex path navigation problems using numerical techniques and advanced iterative algorithms, this research paves the way for further exploration. Future work will delve into optimizing computational strategies, such as half- [7,26-28] and quarter-sweep methods [14,21,29], to enhance performance further. These advancements position the MTOR scheme as a transformative tool in numerical path planning, contributing significantly to the field's ongoing progress.

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

References

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

[2] 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

[3] Akishita, Sadao, Takashi Hisanobu, and Sadao Kawamura. "Fast path planning available for moving obstacle avoidance by use of Laplace potential." In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1, pp. 673-678. IEEE, 1993. https://doi.org/10.1109/IROS.1993.583188

[4] Szukiewicz, Mirosław K. "Efficient numerical method for solution of boundary value problems with additional conditions." *Brazilian Journal of Chemical Engineering* 34 (2017): 873-883. https://doi.org/10.1590/0104-6632.20170343s20150663

[5] 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.

[6] Khairudin, M., R. Refalda, S. Yatmono, H. S. Pramono, A. K. Triatmaja, and A. Shah. "The mobile robot control in obstacle avoidance using fuzzy logic controller." *Indonesian Journal of Science and Technology* 5, no. 3 (2020): 334-351. https://doi.org/10.17509/ijost.v5i3.24889

[7] Suparmin, Sumiati, and Azali Saudi. "Path planning of indoor mobile robot using harmonic potentials via half-sweep modified SOR method." In *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 46-51. IEEE, 2017. https://doi.org/10.1109/I2CACIS.2017.8239031

[8] Garrido, Santiago, Luis Moreno, Dolores Blanco, and Fernando Martín Monar. "Robotic motion using harmonic functions and finite elements." *Journal of Intelligent and Robotic Systems* 59 (2010): 57-73. https://doi.org/10.1007/s10846-009-9381-3

[9] Qin, Hongwei, Shiliang Shao, Ting Wang, Xiaotian Yu, Yi Jiang, and Zonghan Cao. "Review of autonomous path planning algorithms for mobile robots." *Drones* 7, no. 3 (2023): 211. https://doi.org/10.3390/drones7030211

[10] Nahavandi, Saeid, Roohallah Alizadehsani, Darius Nahavandi, Shady Mohamed, Navid Mohajer, Mohammad Rokonuzzaman, and Ibrahim Hossain. "A comprehensive review on autonomous navigation." *arXiv preprint arXiv:2212.12808* (2022). https://doi.org/10.48550/arXiv.2212.12808

[11] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT Press, 2011. https://doi.org/10.5860/choice.49-1492

[12] Kober, Jens, J. Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey." *The International Journal of Robotics Research* 32, no. 11 (2013): 1238-1274. http://dx.doi.org/10.1177/0278364913495721

[13] Dahalan, A'Qilah Ahmad, Azali Saudi, and Jumat Sulaiman. "Enhancing Autonomous Guided Vehicles with Red-Black TOR Iterative Method." *Mathematics* 11, no. 20 (2023): 4393. https://doi.org/10.3390/math11204393

[14] Dahalan, A'Qilah Ahmad, and Azali Saudi. "Computational Analysis of a Mobile Path-Planning via Quarter-Sweep Two-Parameter Over-Relaxation." In *International Congress on Information and Communication Technology*, pp. 297-309. Singapore: Springer Nature Singapore, 2023. https://doi.org/10.1007/978-981-99-3243-6_23

[15] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." *nature* 323, no. 6088 (1986): 533-536. https://doi.org/10.1038/323533a0

[16] Courant, Richard, and Hilbert, David. *Methods of Mathematical Physics: Partial Differential Equations*. Vol. 2. John Wiley and Sons, 1989. https://doi.org/10.1002/9783527617234

[17] Stewart, William J. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press: Princeton, 2021. https://doi.org/10.2307/j.ctv182jsw5

[18] Li, Ruitian, Liang Gong, and Minghai Xu. "A heterogeneous parallel Red–Black SOR technique and the numerical study on SIMPLE." *The Journal of Supercomputing* 76 (2020): 9585-9608. https://doi.org/10.1007/s11227-020-03221-1

[19] Fernandez, Gonzalo, Mariana Mendina, and Gabriel Usera. "Heterogeneous computing (CPU-GPU) for pollution dispersion in an urban environment." *Computation* 8, no. 3 (2020). https://doi.org/10.3390/computation8010003

[20] Hadjidimos, Apostolos. "Accelerated overrelaxation method." *Mathematics of Computation* 32, no. 141 (1978): 149-157. https://doi.org/10.1090/S0025-5718-1978-0483340-6
[21] Ali, Norhashidah Hj Mohd, and K. P. Foo. "Modified explicit group AOR methods in the solution of elliptic equations." *Applied Mathematical Sciences* 6, no. 50 (2012): 2465-2480.
[22] Chen, Shiang-Fong. *Collision-free path planning*. Iowa State University, 1997. https://doi.org/10.31274/rtd-180813-10480
[23] Saudi, Azali Bin. "Robot path planning using family of SOR iterative methods with Laplacian behaviour-based control." PhD diss., Universiti Malaysia Sabah, 2015.
[24] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500-505. IEEE, 1985. https://doi.org/10.1177/027836498600500106
[25] Chung, Jaehoon, Jamil Fayyad, Younes Al Younes, and Homayoun Najjaran. "Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding." *Artificial Intelligence Review* 57, no. 2 (2024): 41. http://dx.doi.org/10.1007/s10462-023-10670-6
[26] Dahalan, A'Qilah Ahmad, and Saudi, Azali. "An iterative technique for solving path planning in identified environments by using a skewed block accelerated algorithm." *AIMS Mathematics* 8, no. 3 (2023): 5725-5744. https://doi.org/10.3934/math.2023288
[27] Abdullah, Abdul Rahman. "The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver." *International Journal of Computer Mathematics* 38, no. 1-2 (1991): 61-70. https://doi.org/10.1080/00207169108803958
[28] Ibrahim, Azzam, and Abdul Rahman Abdullah. "Solving the two dimensional diffusion equation by the Four Point Explicit Decoupled Group (EDG) iterative method." *International Journal of Computer Mathematics* 58, no. 3-4 (1995): 253-263. https://doi.org/10.1080/00207169508804447
[29] Othman, Mohamed, and Abdul Rahman Abdullah. "An efficient four points modified explicit group poisson solver." *International Journal of Computer Mathematics* 76, no. 2 (2000): 203-217. https://doi.org/10.1080/00207160008805020