PENERBIT
CITRA LESTARI

Citra Journal of Computer Science and Technology

Journal homepage:
https://citralestari.my/index.php/cjcst/index
ISSN: 3093-7108

Numerical Technique for Autonomous Path Searching in Designated Scene with Rotated AOR 9-Point Laplacian Iteration

Ling Wai Kiat[1], A'Qilah Ahmad Dahalan[1,*], Alexandre C M De Oliveira[2], Azali Saudi[3]

[1] Department of Mathematics, Centre for Foundation Defence Studies, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia
[2] Depto of Informatics, Federal university of Maranhão, Av. Dos Portugueses 1966, São Luís, MA, 65080-805, Brazil
[3] Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

ARTICLE INFO

Article history:
Received 6 July 2025
Received in revised form 20 August 2025
Accepted 25 August 2025
Available online 29 August 2025

Keywords:
Mobile robot path searching; collision free; nine-point Laplace operator; rotated iteration

ABSTRACT

Autonomous navigation is a field that has been under constant research throughout recent years as human yearns to develop better path searching ability for autonomous navigation. Up until now, there has been much advancement on the subject matter, however there are much room for improvement that yields better result thus pushing the limits of autonomous navigation ability further. In this study, we will attempt to improve the path searching efficiency of mobile robot by using numerical technique to solve path searching problems iteratively. By utilizing harmonic functions, the Laplace's equation can be used to generate potential function values for mobile robot's configuration space. Thus, this paper proposed the method of Half-Sweep Accelerated Overrelaxation 9-Point Laplacian (HSAOR-9P) iteration to improve the path searching ability of mobile robot in a configuration space. Through this, the experiment shows that a smooth path was able to be produced from any starting point to the goal point in the configuration space. Aside from that, the results also show that this numerical method was more efficient in solving mobile robot path searching problem compared to its predecessors.

1. Introduction

Efficiency of path searching in mobile robot navigation plays an influential role, as it enables the mobile robot to move from a starting point to a goal point while avoiding any obstacles along its way in the shortest time as possible. Solving the path searching problem in the most efficient way as possible has been essential throughout the years as it is applied in various fields of easing human activities such as in automated surveillance, transportation, animations and even robotics surgery. In this paper, the path searching problem is modelled as Laplace's equation which controls the generation of harmonic potential functions (HPF) in the configuration space. Prior to this, various methods were available to compute the value of HPF, including Jacobi, Gauss-Seidel (GS) and

[*] *Corresponding author.*
E-mail address: a.qilah@upnm.edu.my

https://doi.org/10.37934/cjcst.2.1.4455

Successive Overrelaxation (SOR) [1–4]. For this paper, we will implement the iterative method known as Half-Sweep Accelerated Overrelaxation via 9-Point Laplacian (HSAOR-9P) to compute the HPF values, which in turn will generate temperature values that will be used for the mobile robot path generation via Gradient Descent Search (GDS) method.

The application of HPF in robot path searching was first introduced by Khatib [5], where in his work it was shown that through potential function method, every obstacle in the configuration space shall exert a repelling force while the goal shall exert attraction force. In other related work, Connolly and Gruppen [6] in their early work has shown that HPF have useful properties when applied in robotic applications. Koditschek [7] has concluded that, geometrically in a certain domain, potential functions were able to guide the effector from nearly any point to any given point in the said domain. The methods mentioned above, however, were all suffering from the problem arise from the generation of local minima, which traps the robot in the unwanted configuration space other than the desired ones, hence preventing the robot from reaching its intended goal point. The works by Connolly *et al.,* [1] and Akishita *et al.,* [2] both demonstrated independent global method that generates smooth path using solutions obtained from Laplace's equation where the potential fields were computed over global manner across the entire region in the configuration space. By introducing Laplace's equation solution in the configuration space, the local minima were able to be avoided, and are guaranteed to provide trajectory path towards the goal point all of the time by following the path generated through GDS. Another notable work was by Sasaki [4] where numerical techniques were demonstrated to be able to solve path searching problem, resulting in efficient path searching in complex maze simulation. Some other more useful applications of HPF includes work by Waydo and Murray [8] which used steam function that is similar to HPF to generate motion planning for vehicles, Daily and Bevly [9] used HPF for path searching for high speed vehicles, Szulczynski *et al.,* [10] used HPF to generate real time obstacle avoidance path searching, Yang and Ariyur [11] utilized Laplacian method of path searching in avoiding moving obstacles and Liang *et al.,* [12] which applied 3D potential path searching method or unmanned aerial vehicle (UAV) motion in complex environments.

2. Methodology

In order to perform this experiment, we have applied the Laplacian potential numerical technique into the process. The Laplace's equation in this experiment is also known as steady state equation [13]. The solutions to the Laplace's equation reflect the temperature value in the configuration space, which in turn be used to generate path for mobile robot through the GDS technique. To solve the Laplace's equation, we implemented numerical technique, HPF, which offers many advantages for mobile robot path searching, other than having many useful properties for applications relating to robotics field [6], it also offers a complete path searching algorithm. The paths generated through HPF are generally smooth [6]. Moreover, through implementation of HPF into the solution, we can avoid any occurrence of spurious local minima [1], which is something that traps the mobile robot preventing the mobile robot from reaching its intended goal point.

For the designated scene, which is the configuration space the autonomous robot operates in, we used 4 different layouts to perform the experiments, each with different number of obstacles [14]. In the designated areas, there consists of the starting point, interior and exterior boundary walls, obstacles, and the goal point. The starting and goal points vary from each scene as they were not fixed, in fact the starting point and goal point could be anywhere within the configuration space so long as they are within the inner boundary walls, and do not intercept with obstacles. Since mobile robot path searching problem can be treated as heat transfer problem, the inner and outer boundary walls, and obstacles were fixed with constant temperature, and were treated as heat source, whereas the goal point was assigned with the lowest potential value and acted as a heat sink. Following the principal of heat conduction, where heat flows from the area with higher temperature into the area with lower temperature, the goal point acted as a heat sink pulling heat into it. This occurrence of flowing heat represented by the Laplacian potential creates a heat flux line which in turn was to be used by the mobile robot to navigate its way towards the goal point. Through implementing HPF into the solution, the mobile robot was able to be guided towards the goal point while avoiding obstacles along its way, and as shown by Connolly *et al.,* [1], any occurrence of local minima which traps the robot were able to be avoided.

To better understand the concept mathematically, consider the Laplace's equation in Eq. (1):

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

Where is the i-th Cartesian coordinate and is the dimension.

A harmonic function in the domain is a function that satisfies Laplace's equation in Eq. (1). Under the circumstances of mobile robot path searching, the domain comprises of the outer and inner boundary walls, obstacles, starting point and the goal point. Applying HPF into the solution means that the occurrence of local minima can be avoided, since HPF satisfies the min-max principle [15]. The HPF restricts the number of functions produced in the configuration space, in doing so avoids local minima, thus resulting in a smooth and effective path being generated from the starting point to goal point due to the complete path searching algorithm it provided.

There were various ways in solving the Laplace's equation as presented in past related works, the solutions include conventional numerical technique of GS, Jacobi, SOR and other methods [16-18]. This paper will implement the HSAOR-9P, also known as Rotated Accelerated Overrelaxation 9-Point Laplacian iteration method, due to the 'rotated' style of scanning of the nodal sets in the configuration space instead of the standard linear approach. Other iteration methods from previous related work [19] will also be used, where the results will be presented for the aim of comparison. The iteration methods include, Full-Sweep SOR 5-Point Laplacian (FSSOR-5P), Full-Sweep SOR 9-Point Laplacian (FSSOR-9P), Half-Sweep SOR 5-Point Laplacian (HSSOR-5P), Half-Sweep SOR 9-Point Laplacian (HSSOR-9P), Full-Sweep AOR 5-Point Laplacian (FSAOR-5P), Full-Sweep AOR 9-Point Laplacian (FSAOR-9P), and Half-Sweep AOR 5-Point Laplacian (HSSOR-5P).

2.1 The Half-Sweep Accelerated Overrelaxation 9-Point Laplacian Iterative Method

The half-sweep (HS) iteration method was first presented and used by Abdullah [20] when performing calculation through Explicit Decoupled Group (EDG) to solve the 2D Poisson equation. Before getting into understanding the HS iterative method, first we take a look into the configuration space, where spaces contained within it were divided evenly throughout the designated scene into small nodal points, which increases in number as the configuration space increases in size. These nodal points, under normal circumstances were all being considered into the calculation when solving the HPF, as shown in Figure 1 (a). This standard method of calculation technique is also called as full-sweep (FS) iteration method. In the HS iteration method, of all the nodal points present in the configuration space, only half of them were considered into the calculation process. To better understand this concept, consider the nodal points samples as shown in Figure 1.

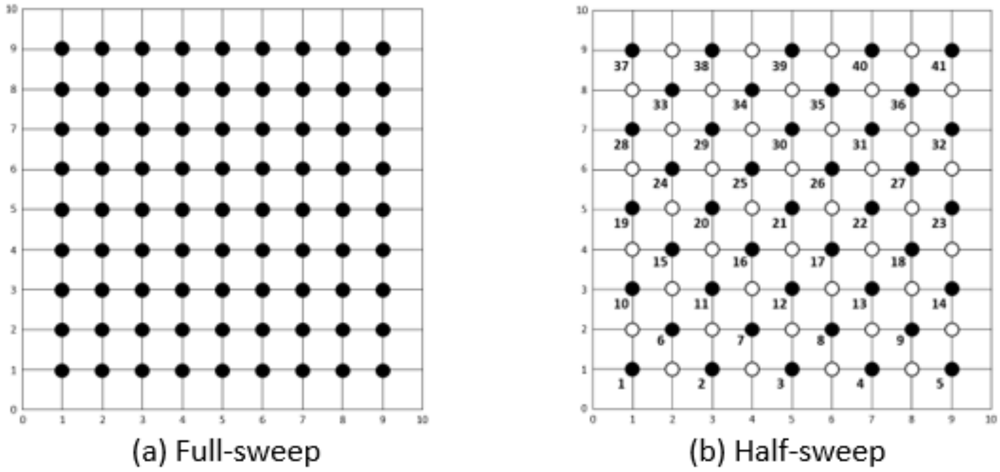


Fig. 1. Nodal points consideration pattern in configuration space

The FS method, while being thorough, consumes the maximum time required for this sort of process to complete its computation. The HS method enhances the standard technique by skipping the nodal points taken into iterative consideration. Whenever one set of the nodal points had been through the iterative computational process, the next set of points taken into consideration are the ones after skipping the nodal points neighboring the previous set of considered nodal points, creating a ‘rotating’ style of computation, thus the name ‘rotated’ iterative method. The nodal points taken into consideration are represented by the black dots, • in Figure 1(b), while the remaining white dots, ◦ are computed via direct method. This type of iteration saves much of the time needed to compute the solutions desired.

Figure 2 shows the said sets of nodal points isolated from the configuration space for better visualization, while Figure 3 shows the computational grid for the nodal sets mentioned, in the configuration space. The visualizations in Figure 2 and Figure 3 demonstrates nodal sets that were considered in a standard approach of numerical technique in obtaining solutions for Laplacian potential where 5 nodal points were considered into the computation. In this paper, we adapted the 9-point Laplacian approach. Whereas the name would suggest, 9 nodal points were considered into the computation of Laplacian potential solutions, and paired with the HS approach, results in a more accurate data in a shorter time frame.

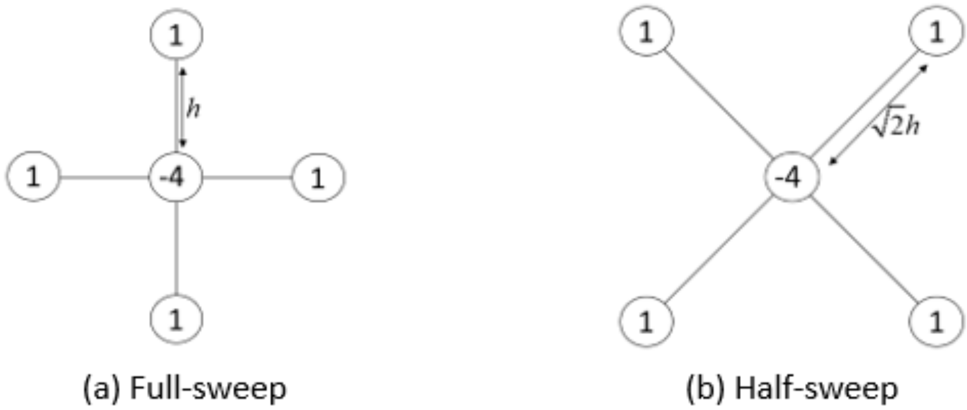
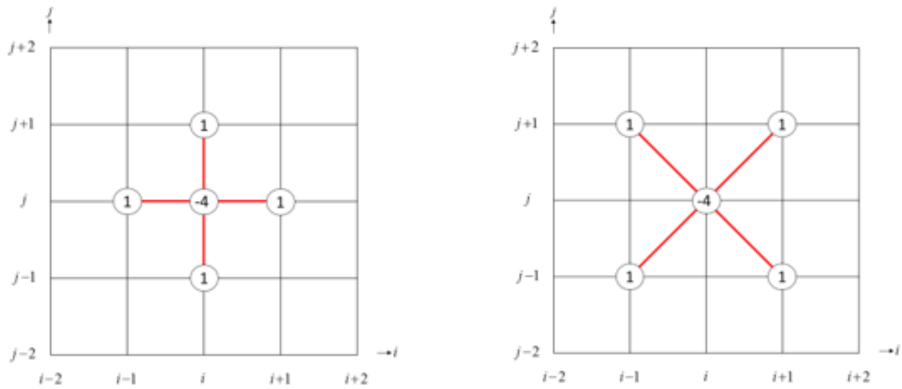


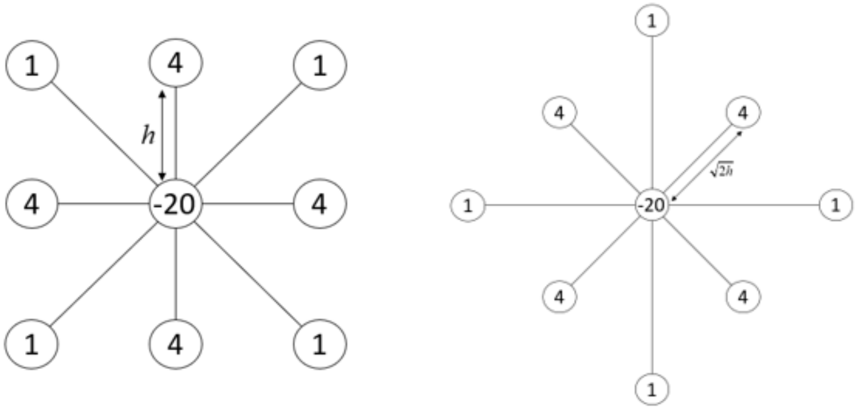
Fig. 2. The standard nodal points set (5-point) computational model for finite difference approximation



(a) Full-sweep (b) Half-sweep

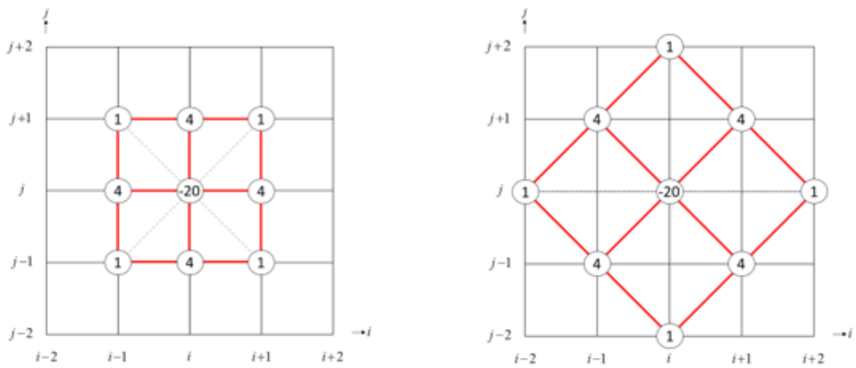
Fig. 3. Standard nodal points (5-point) set computational grid about point (i, j)

In understanding the 9-point concept visually, consider the illustration shown in Figure 4 and Figure 5. Instead of taking 5 nodal points, 9 nodal points were considered into computation, where in Figure 4 (a) and Figure 5 (a) are the nodal points set model for full-sweep approach, while in Figure 4 (b) and Figure 5 (b) are for the HS approach where it is essentially augmented from the similar 9-point stencil but rotated about the i-j plane by 45°.



(a) Full-sweep (b) Half-sweep

Fig. 4. The standard nodal points set (9-point) computational model for finite difference approximation



(a) Full-sweep (b) Half-sweep

Fig. 5. Standard nodal points (9-point) set computational grid about point

Considering the 2-dimensional Laplace's equation in Eq. (1) be defined as:

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

By applying the second-order central difference scheme into the HS 9-point standard finite difference approximation equation, Eq. (2) can then be simplified as:

$$4(U_{i+1,j-1} + U_{i+1,j+1} + U_{i-1,j-1} + U_{i-1,j+1}) + U_{i+2,j} + U_{i,j+2} + U_{i-2,j} + U_{i,j-2} - 20U_{i,j} = 0 \tag{3}$$

For standard numerical, SOR technique, one weighted parameter, ω [21-23] would be implemented into computation, the formulation of HSSOR-9P can now be viewed as:

$$U_{i,j}^{k+1} = \frac{\omega}{5}(U_{i-1,j+1}^{k} + U_{i-1,j-1}^{k+1} + U_{i+1,j-1}^{k+1} + U_{i+1,j+1}^{k}) + \frac{\omega}{20}(U_{i-2,j}^{k+1} + U_{i+2,j}^{k} + U_{i,j-2}^{k+1} + U_{i,j+2}^{k}) + (1-\omega)U_{i,j}^{k} \tag{4}$$

This paper tests one of the overrelaxation family numerical technique, called Accelerated Overrelaxation (AOR). By adding another weighted parameter, *r*, Eq. (4) can be enhanced to obtain HSAOR-9P iteration formulation. Discretizing Eq. (4) by replacing $U_{i-1,j-1}^{k+1}$, $U_{i+1,j-1}^{k+1}$, $U_{i-2,j}^{k+1}$ and $U_{i,j-2}^{k+1}$ with $U_{i-1,j-1}^{k}$, $U_{i+1,j-1}^{k}$, $U_{i-2,j}^{k}$ and $U_{i,j-2}^{k}$ respectively, and adding the terms $\frac{r(U_{i-1,j-1}^{k+1} - U_{i-1,j-1}^{k})}{5}$, $\frac{r(U_{i+1,j-1}^{k+1} - U_{i+1,j-1}^{k})}{5}$, $\frac{r(U_{i-2,j}^{k+1} - U_{i-2,j}^{k})}{20}$, and $\frac{r(U_{i,j-2}^{k+1} - U_{i,j-2}^{k})}{20}$, we get the formulation of HSAOR-9P as follows:

$$\begin{aligned} U_{i,j}^{k+1} &= \frac{r}{5}(U_{i-1,j-1}^{k+1} - U_{i-1,j-1}^{k} + U_{i+1,j-1}^{k+1} - U_{i+1,j-1}^{k}) + \frac{r}{20}(U_{i-2,j}^{k+1} - U_{i-2,j}^{k} + U_{i,j-2}^{k+1} - U_{i,j-2}^{k}) \\ &\quad + \frac{\omega}{5}(U_{i-1,j+1}^{k} + U_{i-1,j-1}^{k} + U_{i+1,j-1}^{k} + U_{i+1,j+1}^{k}) + \frac{\omega}{20}(U_{i-2,j}^{k} + U_{i+2,j}^{k} + U_{i,j-2}^{k} + U_{i,j+2}^{k}) + (1-\omega)U_{i,j}^{k} \end{aligned} \tag{5}$$

Applying Eq. (5) into the computation gives us linear systems, which will be iterated individually until the maximum error of the solutions generated falls in the specified tolerance error range, i.e., at either 8.8818 E-16 or 9.9920 E-16. The tolerance error range should be set to a minimum for avoiding excessive occurrence of flat regions in the final solutions. In studies made by Hajidimos [24], it was stated that the values of the weighted parameter r are customarily set to be as close to the value of ω for its correlative SOR, and the value of ω is determined after testing the values sequentially until an optimum performance was achieved in the simulation. For this paper, the values of ω were tested and found out to be between 1.8 and 2.0. After the iteration of the linear systems were completed, upon conducting the GDS on the iteration solutions, a path will have been generated from the starting point towards the goal point.

The algorithm of HSAOR-9P in the simulation is shown below:

Algorithm of path searching via HSAOR-9P iteration:

i. Set up the configuration space with an arbitrary position for the starting point and objective point each in the free space.
ii. Activate the starting point U, $\varepsilon \leftarrow 1.0e^{-15}$, $iteration \leftarrow 0$
iii. In all free spaces, i.e., non-occupied grids, for all accounted nodal points in HS scheme, compute

$$U_{i,j}^{k+1} \leftarrow \frac{r}{5}(U_{i-1,j-1}^{k+1} - U_{i-1,j-1}^{k} + U_{i+1,j-1}^{k+1} - U_{i+1,j-1}^{k}) + \frac{r}{20}(U_{i-2,j}^{k+1} - U_{i-2,j}^{k} + U_{i,j-2}^{k+1} - U_{i,j-2}^{k})$$
$$+ \frac{\omega}{5}(U_{i-1,j+1}^{k} + U_{i-1,j-1}^{k} + U_{i+1,j-1}^{k} + U_{i+1,j+1}^{k}) + \frac{\omega}{20}(U_{i-2,j}^{k} + U_{i+2,j}^{k} + U_{i,j-2}^{k} + U_{i,j+2}^{k}) + (1-\omega)U_{i,j}^{k}$$

iv. For the remaining nodes in the free space, plotted as $\circ$, calculate via direct method using

$$U_{i,j}^{(k+1)} = \frac{1}{5}\left[U_{i+1,j}^{(k)} + U_{i-1,j}^{(k+1)} + U_{i,j+1}^{(k)} + U_{i,j-1}^{(k+1)}\right] + \frac{1}{20}\left[U_{i-1,j-1}^{(k+1)} + U_{i+1,j+1}^{(k)} + U_{i-1,j+1}^{(k)} + U_{i+1,j-1}^{(k+1)}\right]$$

v. Examine the error values from convergence test for $\varepsilon \leftarrow 1.0e^{-15}$. If yes, proceed to (vi), else, revert to (iii).
vi. Apply GDS for the steepest descent search in potential values and generate path from the starting point to the objective point.

3. Experiments and Results

The experiment was performed on 4 different set of designated scenes, of 3 different pixel sizes, i.e. 300×300, 600×600 and 900×900, with increasing obstacles as we go forward into the environments. Each region contains boundary walls, various forms of obstacles, along with starting point and goal point. In the environment, the initial temperature of the exterior and interior boundary walls was set to the highest temperature values, meanwhile the goal point possesses the lowest temperature value. After iteration and obtaining the function values, GDS can be performed to generate a path. The software used for solving and obtaining the linear systems was the Delphi Project where in the software a pre-built platform namely Robot 2D Simulator [25] was utilized, and the computation was carried out on a computer with Intel Core i5-5200U CPU at 2.2 GHz utilizing 4 GB of RAM. The results produced were as shown in Table 1 and Table 2. As observed on Table 1 and Table 2, the iterations done through HSAOR-9P clearly have the most efficient rate of iterations.

Table 1
Execution results of the examined numerical technique in terms of number of iterations

	Methods	N x N		
		300 x 300	600 x 600	900 x 900
Environment 1	FSSOR-5P	1312	4185	7398
	FSAOR-5P	1151	3105	5417
	HSSOR-5P	1095	3050	4747
	HSAOR-5P	916	2562	3741
	FSSOR-9P	1231	3612	6541
	FSAOR-9P	1113	2805	4771
	HSSOR-9P	918	2581	4094
	HSAOR-9P	783	2191	3211
Environment 2	FSSOR-5P	2231	4868	10916
	FSAOR-5P	1726	3467	8110
	HSSOR-5P	1241	3065	6968
	HSAOR-5P	914	2329	5513
	FSSOR-9P	1878	4272	9667
	FSAOR-9P	1549	3055	7159
	HSSOR-9P	1040	2624	6034
	HSAOR-9P	848	1995	4778
Environment 3	FSSOR-5P	1544	5541	9034
	FSAOR-5P	1093	4081	7560
	HSSOR-5P	948	2362	7046
	HSAOR-5P	757	1964	5398
	FSSOR-9P	1284	4662	7722
	FSAOR-9P	908	3448	6457
	HSSOR-9P	834	1989	5909
	HSAOR-9P	661	1651	4568
Environment 4	FSSOR-5P	766	2662	4962
	FSAOR-5P	748	2129	3884
	HSSOR-5P	567	1575	4160
	HSAOR-5P	524	1059	3057
	FSSOR-9P	756	2253	4264
	FSAOR-9P	736	1798	3322
	HSSOR-9P	550	1275	3512
	HSAOR-9P	482	928	2574

Table 2
Execution results of the examined methods in terms of CPU processing time (in seconds)

	Methods	N x N		
		300 x 300	600 x 600	900 x 900
Environment 1	FSSOR-5P	1.86	35.00	168.77
	FSAOR-5P	1.86	29.05	137.24
	HSSOR-5P	0.95	18.16	72.53
	HSAOR-5P	0.88	15.20	60.66
	FSSOR-9P	1.99	32.99	168.01
	FSAOR-9P	2.28	31.47	146.46
	HSSOR-9P	0.88	16.39	67.72
	HSAOR-9P	0.91	15.53	62.64
Environment 2	FSSOR-5P	3.23	42.14	247.08
	FSAOR-5P	2.88	33.56	208.82
	HSSOR-5P	1.09	18.39	110.16
	HSAOR-5P	0.74	13.85	91.88
	FSSOR-9P	3.13	41.08	249.36
	FSAOR-9P	3.23	34.44	222.62
	HSSOR-9P	1.09	17.24	102.72
	HSAOR-9P	1.02	14.38	94.00
Environment 3	FSSOR-5P	2.11	47.33	197.27
	FSAOR-5P	1.70	36.49	185.62
	HSSOR-5P	0.77	13.50	104.06
	HSAOR-5P	0.69	11.55	82.93
	FSSOR-9P	1.98	41.28	190.95
	FSAOR-9P	1.76	36.08	188.60
	HSSOR-9P	0.78	12.30	92.22
	HSAOR-9P	0.73	11.27	82.57
Environment 4	FSSOR-5P	1.05	21.19	103.46
	FSAOR-5P	1.17	18.77	91.91
	HSSOR-5P	0.47	8.53	59.55
	HSAOR-5P	0.53	5.94	46.35
	FSSOR-9P	1.17	19.75	100.79
	FSAOR-9P	1.44	18.74	94.55
	HSSOR-9P	0.52	7.81	54.69
	HSAOR-9P	0.56	6.20	46.42

Figure 6 shows the path line samples that were generated from the simulation, where the red dot portrays the starting point, while the green rectangle depicts the goal point. As observed from Figure 6, a smooth and short path line that will be used by the mobile robot were produced for every different starting and goal points in the environment. The performance of the iteration methods in terms of CPU processing time was inconsistent in a smaller environment, i.e. in 300×300 where the CPU processing time results were less than 5 seconds, but it became more consistent in larger environments. This is due to the CPU performance itself where background processes would affect small time frame of simulation processing, yet as observed in the results, obviously the HSAOR-9P will and still gave slightly faster execution time compared to other proposed methods.

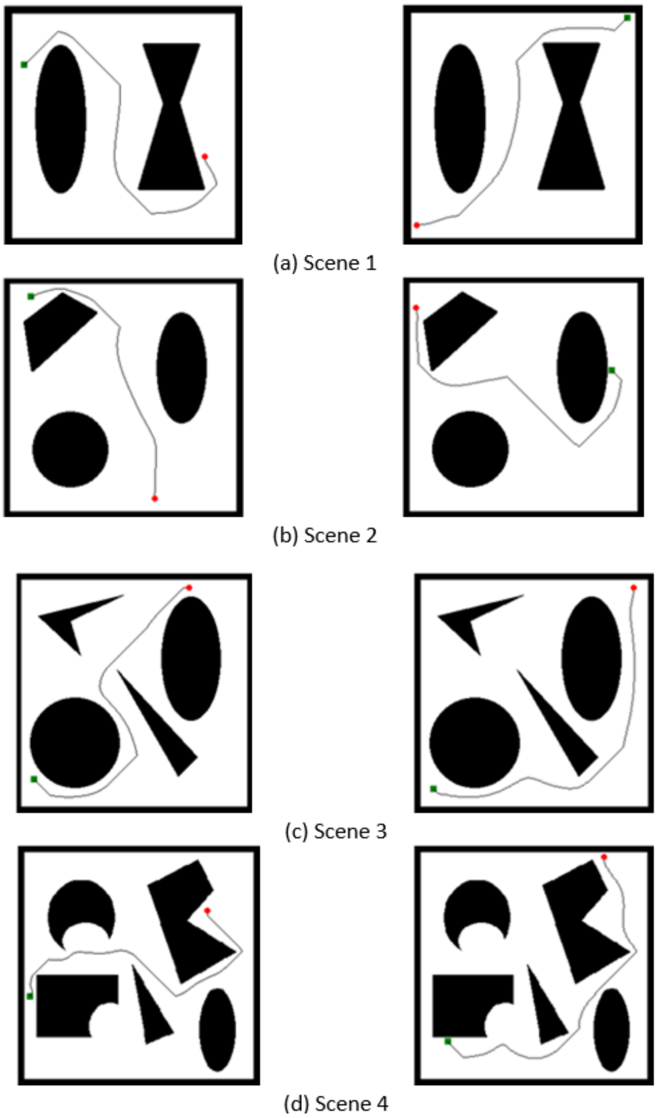


Fig. 6. Path lines generated through HSAOR-9P

4. Conclusions and Future Work

Through the results we obtained from this experiment, we were able to show that solving the Laplace's equation can be done efficiently with the correct method, and the more efficient the numerical technique applied, the better the mobile robot's path searching will be. In this case, the numerical technique HSAOR-9P was able to perform better than its predecessor's standard techniques of SOR and AOR. The number of obstacles affects the path navigation in a good and desired way as the more the obstacles present in the environment, the better the mobile robot path navigation is, this is due to the fact that more computational areas can be ignored if it were occupied. In future works, we can try improving the numerical method by introducing Quarter-Sweep (QS) technique in attempt to improve computational efficiency [26-28].

Acknowledgement
This research was not funded by any grant. The authors declare that there is no conflict of interest regarding the publication of this study.

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] Akishita, S., S. Kawamura, and K-I. Hayashi. "Laplace potential for moving obstacle avoidance and approach of a mobile robot." In *Japan-USA Symposium on flexible automation, A Pacific rim conference*, pp. 139-142. 1990. https://doi.org/10.1109/21.148426

[3] Barraquand, Jerome, Bruno Langlois, and J-C. Latombe. "Numerical potential field techniques for robot path planning." *IEEE transactions on systems, man, and cybernetics* 22, no. 2 (2002): 224-241.

[4] Sasaki, S. "A practical computational technique for mobile robot navigation." In *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, vol. 2, pp. 1323-1327. IEEE, 1998. https://doi.org/10.1109/CCA.1998.721675

[5] 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/10.1109/ROBOT.1985.1087247

[6] 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

[7] Koditschek, Daniel. "Exact robot navigation by means of potential functions: Some topological considerations." In *Proceedings. 1987 IEEE international conference on robotics and automation*, vol. 4, pp. 1-6. IEEE, 1987. https://doi.org/10.1109/ROBOT.1987.1088038

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

[9] Daily, Robert, and David M. Bevly. "Harmonic potential field path planning for high speed vehicles." In *2008 American Control Conference*, pp. 4609-4614. IEEE, 2008. https://doi.org/10.1109/ACC.2008.4587222

[10] 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.

[11] Yang, Fei, and Kartik Ariyur. "Laplacian Path Planning: Implementation And Generalizations." In *Infotech@ Aerospace 2011*, p. 1631. 2011. https://doi.org/10.2514/6.2011-1631

[12] Liang, Xiao, Honglun Wang, Dawei Li, and Chang Liu. "Three-dimensional path planning for unmanned aerial vehicles based on fluid flow." In *2014 IEEE Aerospace Conference*, pp. 1-13. IEEE, 2014. https://doi.org/10.1109/AERO.2014.6836520

[13] Evans, Lawrence C. *Partial differential equations*. Vol. 19. American mathematical society, 2022.

[14] Zelinsky, Alexander. "Mobile robot map making using sonar." *Journal of Robotic Systems* 8, no. 5 (1991): 557-577. https://doi.org/10.1002/rob.4620080502

[15] Zachmanoglou, Eleftherios C., and Dale W. Thoe. *Introduction to partial differential equations with applications*. Courier Corporation, 1986.

[16] Evans, D. J. "Group explicit iterative methods for solving large linear systems." *International Journal of Computer Mathematics* 17, no. 1 (1985): 81-108. https://doi.org/10.1080/00207168508803452

[17] Evans, D. J., and W. S. Yousif. "Explicit Group Iterative Methods for solving elliptic partial differential equations in 3-space dimensions." *International journal of computer mathematics* 18, no. 3-4 (1986): 323-340. https://doi.org/10.1080/00207168608803498

[18] Ibrahim, Arsmah. "The Study of the Iterative Solution Of Boundary Value Problem by the Finite Difference Methods." PhD diss., PhD Thesis. Universiti Kebangsaan Malaysia, 1993.

[19] Dahalan, A. A., A. Saudi, J. Sulaiman, and W. R. W. Din. "Autonomous navigation in static indoor environment via rotated Laplacian operator." In *AIP conference proceedings*, vol. 1974, no. 1, p. 020035. AIP Publishing LLC, 2018. https://doi.org/10.1063/1.5041566

[20] 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

[21] Young, David M. *Iterative solution of large linear systems*. Elsevier, 2014.

[22] Young, David M. "Second-degree iterative methods for the solution of large linear systems." *Journal of Approximation Theory* 5, no. 2 (1972): 137-148. https://doi.org/10.1016/0021-9045(72)90036-6

[23] Young, David M. *Iterative solution of linear systems arising from finite element techniques*. Center for Numerical Analysis, University of Texas at Austin, 1975.

[24] Hadjidimos, Apostolos. "Accelerated overrelaxation method." *Mathematics of computation* 32, no. 141 (1978): 149-157. https://doi.org/10.1090/S0025-5718-1978-0483340-6

[25] Saudi, Azali Bin. "Robot path planning using family of SOR iterative methods with laplacian behaviour-based control." PhD diss., Universiti Malaysia Sabah, 2015.

[26] 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

[27] Ali, L. H., J. Sulaiman, and A. Saudi. "Iterative method for solving nonlinear Fredholm integral equations using Quarter-Sweep Newton-PKSOR method." In *International Conference on Computational Science and Technology*, pp. 33-46. Singapore: Springer Nature Singapore, 2022. https://doi.org/10.1007/978-981-19-8406-8_3

[28] Dahalan, A. A., A. Saudi, and J. Sulaiman. "Development of Path Optimization Using Quarter-Sweep Modified Successive Over-Relaxation Iterative Technique." In *Proceedings of Eighth International Conference on Information System Design and Intelligent Applications*, pp. 411-425. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-97-4892-1_35