

## **Path Planning for Indoor Mobile Robot using Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L)**

Azali Saudi<sup>1</sup>, Jumat Sulaiman<sup>2</sup>

<sup>1</sup>*School of Engineering and Information Technology*

<sup>2</sup>*School of Science and Technology University Malaysia Sabah, Kota Kinabalu, Malaysia*

---

**Abstract:** *This paper proposed fast Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L) iterative method for solving path planning problem for a mobile robot operating in indoor environment model. It is based on the use of Laplace's Equation to constraint the distribution of potential values in the environment of the robot. Fast computation with half-sweep iteration is obtained by considering only half of whole points in the configuration model. The inclusion of SOR and 9-point Laplacian into the formulation further speeds up the computation. The simulation results show that HSSOR9L performs much faster than the previous iterative methods in computing the potential values to be used for generating smooth path from a given initial point to a specified goal position.*

**Keywords** - *Robot path planning, Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L), Laplace's Equation, Harmonic functions*

---

### **I. INTRODUCTION**

The capability of moving quickly from a given initial point to a specified goal point is essential for truly autonomous mobile robot operating in indoor environment. While exploring the environment, the robot must avoid any obstacles and walls that obstructing its motion. Therefore, path planning problem pose as an important issue in building a robust indoor mobile robot. This study proposed a global numerical approach in solving this difficult path planning problem. It is based on the theory of a steady-state heat transfer, in which the configuration space is modeled with Laplace's equation. Consequently, the solutions of Laplace's equation also called harmonic functions that represent temperature values distribution in the configuration space will be used to simulate the generation of path for mobile robot motion. In the past, various approaches had been used to obtain harmonic functions, but the most common method is via numerical techniques due to the availability of fast processing machine and their elegant and efficiency in solving the problem. In this study, we investigate the performance of Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L) in computing the harmonic functions to generate path for a mobile robot operating in varying sizes of indoor environment model.

### **II. PREVIOUS WORKS**

Pioneer work by Khatib [1] utilized the use potential functions to solve path planning problem for a mobile robot, in which every obstacle produces a repelling force whereas the goal exerts an attractive force. Meanwhile, Koditschek [2] pointed out that geometrically (at least in certain types of domains) the potential functions can be employed to guide the effectors from almost any point to a specified point. These potential field methods, however, suffer from the generation of local minima that would cause the robot to get stuck without any direction to guide its next motion. After that, Connolly *et al.* [3] and Akishita *et al.* [4], both of them developed independently a global method that generates smooth path by using the solutions of Laplace's equations (also called harmonic functions) which were computed in a global manner over the entire workspace. The work by Connolly and Gruppen [5] shows that harmonic functions have a number of properties useful in robotic applications. Similarly, Waydo & Murray [6] used stream functions for vehicle motion. More recently, Szulczyński *et al.* [7] employed harmonic functions for real-time obstacle avoidance.

In the past, various approaches had been used to obtain harmonic functions. The standard methods are point iteration with Jacobi, Gauss-Seidel and SOR [8]. Alternatively, Daily and Bevly [9] employed analytical solution for arbitrarily shaped obstacles. Meanwhile, finite elements were used by Garrido *et al.* [10] to compute harmonic functions for robotic motion. Fast half-sweep iteration was introduced by Abdullah [11] to be applied on Explicit Decoupled Group (EDG) iterative method for

solving 2-D Poisson equations. This half-sweep iteration was also applied in solving partial differential equations in Ibrahim & Abdullah [12], Yousif & Evans [13], and Abdullah & Ali [14]. A modified version of half-sweep iterative method was also investigated by Sulaiman *et al.* for solving diffusion equation [15]. Furthermore, our previous work in Saudi & Sulaiman [16] introduced the application of half-sweep method in solving path planning problem for a mobile robot. In [17], we employed Half-Sweep SOR method to speed up the computation. Then in [18], Half-Sweep SOR iteration that employs red-black strategy (also known as HSSOR-RB) was applied for solving path planning problem. All of these methods are actually a family of half-sweep iteration that use standard 5-point stencil (also known as 5-point Laplacian). The 9-point Laplacian was first applied in robotics by Saudi & Sulaiman [19]. The study on the effect of SOR iteration to the 9-point Laplacian is reported in Adam *et al.* [20].

### III. HEAT TRANSFER ANALOGY AND ENVIRONMENT MODEL

Path planning problem for a mobile robot operating in indoor environment can be modeled as a well-known steady-state heat transfer problem. In this heat transfer problem, the heat sources come from the boundaries and the heat sink will pull the heat in. This heat conduction process produces a temperature distribution (or in robotics better known as potential values) and the heat flux lines that are flowing to the sink fill the workspace. In a well structured mobile robot environment setup, the goal point is treated as a heat sink whilst the boundary walls and obstacles are considered as heat sources that are fixed with constant temperate values. Once the temperature distribution in the workspace model is obtained, it will be used as a guide to generate path for mobile robot to move from the start point to the goal point. The idea is to follow the heat flux that will flow from high temperature sources to the lowest temperature point in the environment model. For computing the temperature distribution of the configuration space, harmonic function is used to model the environment setup.

Mathematically, a harmonic function on a domain  $\Omega \subset R^n$  is a function which satisfies Laplace's equation, in which  $x_i$  is the  $i$ -th Cartesian coordinate, and  $n$  is the dimension. In the case of robot path development, the domain  $\Omega$  consists of the outer boundary walls, all obstacles and inner walls in the workspace, start points and the goal point.

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

Eq. (1) can be solved efficiently using numerical method. The standards are point Jacobi and Gauss-Seidel iterative methods. Faster computation can be achieved by employing SOR into the method. In this study, the robot model is represented by a point in the configuration space which is designed in grid form. Naturally, this grid form design is readily very suitable for numerical computation. Within this workspace grid, the highest temperature is assigned to the start point whereas the goal point is assigned the lowest, meanwhile different initial temperature values are assigned to the outer boundary walls, inner walls and obstacles. All other free spaces including start points are not assigned to any initial temperature values. Then, Dirichlet boundary conditions are used to constraint the solutions of Laplace's equation. Finally, the function values associated with each node are calculated iteratively via numerical computation to satisfy Eq. (1).

### IV. FORMULATION OF HALF-SWEEP SOR VIA NINE-POINT LAPLACIAN

Previously, our work in [16, 17, 18] utilizing half-sweep iteration produced encouraging results. It was shown in [19] that iteration using 9-point formula (also known as 9-point Laplacian) performed much faster than the standard 5-point formula. Therefore, in this study we propose an improved version of [17] by employing 9-point formula into the equation. The proposed method, called Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L), would not only consider half of the whole node points in the grid but would also use 9 points in its formulation. Adding more points in the formulation would produce greater accuracy in the computation, thus leads to faster convergence rate.

As shown in Figure 1(a), full-sweep iteration involves all points in the grid. Whilst in the case of half-sweep iteration, only black points are considered, see Figure 1(b).

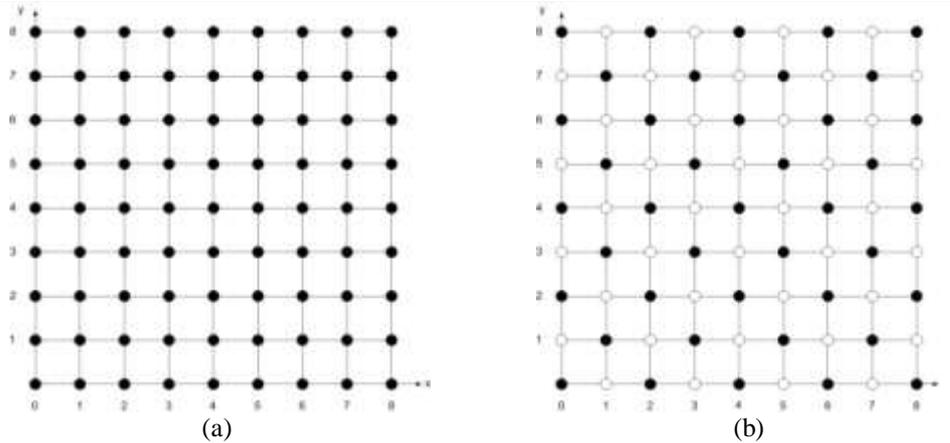


Figure 1: (a) All nodes will be considered in full-sweep iteration. (b) Only black points will be considered in half-sweep iteration.

The standard finite difference approximation of full-sweep iteration for 9-point formula uses 9-point stencil as illustrated in Figure 2(a). Meanwhile, the stencil of half-sweep iteration is essentially derived from the same 9-point stencil but rotated 45 degree as shown in Figure 2(b). This simple enhancement would lead to reduction in computational complexity since only half of the total node points are considered.

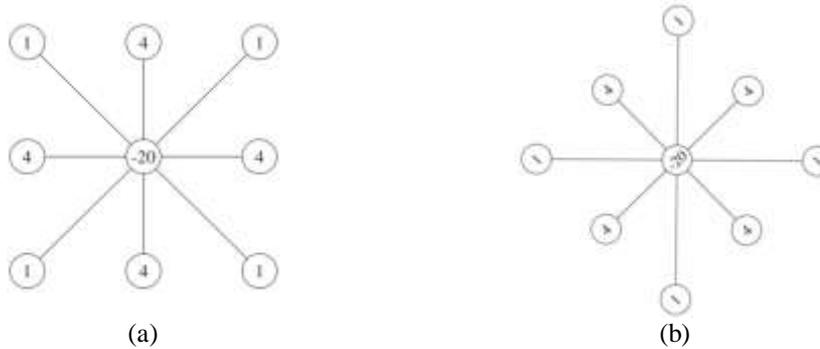


Figure 2: Stencils of 9-point formula for (a) full-sweep and (b) half-sweep iteration, respectively.

Figure 3(a) shows the five black points involve in each calculation of 5-point half-sweep iteration, whereas Figure 3(b) shows the nine black points to be considered in each calculation of HSSOR9L iterative method.

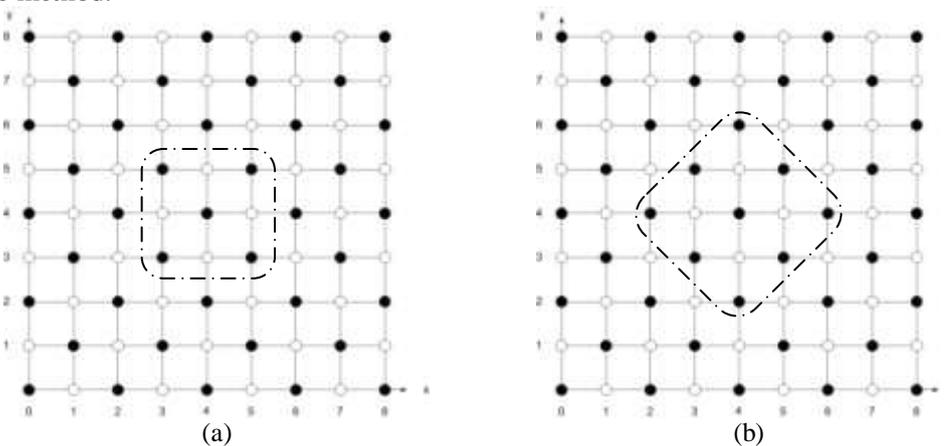


Figure 3: (a) Fives black points for each calculation of 5-point formula implementation. (b) For 9-point formula computation, nine black points are used.

Now, let us consider the 2D Laplace's equation in Eq. (1) defined as

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

Then, the discretization of Eq. (2) based on the 9-point Laplacian illustrated in Figure 2(b) and 3(b) can be defined as below

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

By adding a weighted parameter  $\omega$  (SOR) to Eq. (3), see Young [21,22,23], the formulation of HSSOR9L is now can be defined as below

$$U_{i,j}^{k+1} = \omega [0.2(U_{i+1,j-1}^k + U_{i+1,j+1}^k + U_{i-1,j+1}^k + U_{i-1,j-1}^k) + 0.05(U_{i+2,j}^k + U_{i,j+2}^k + U_{i-2,j}^k + U_{i,j-2}^k)] + (1-\omega)U_{i,j}^k \tag{4}$$

Based on the previous experiments [17], the optimal parameter value of  $\omega$  is set to 1.9. The weighted 9-point finite difference approximation equation of Eq. (4) is then used to generate linear system. Finally, the iterative process is run to compute the generated linear system until the maximum error falls into a specified tolerance error, in which the iteration stops. In order to minimize the occurrence of flat region in the final solutions, the tolerance error must be set to a very small value. Once the convergence to tolerance error is achieved and final value of all black points are computed, the approximate values of the rest of white points will be calculated using standard direct method.

### V. EXPERIMENTS AND RESULTS

For the experiments, varying sizes of static environment model, i.e. 128x128, 256x256 and 512x512, were considered. The environment model setup consists of a goal point, three starting points and varying setup of inner walls and outer boundary walls. Initially, the inner and outer walls were fixed with high temperature values, whereas the goal point was set to a very low temperature. No initial temperature value was assigned to all other free spaces. The code, called RobotPath Simulator, was written in Delphi for very fast computation, see Figure 4. In the previous works [16 - 19], MatLab was used to implement the code but it was too slow. By writing the code in Delphi, the program can now be compiled with true compiler, thus producing faster code. The experiments run on a machine with Intel Core 2 Duo processor at 3 GHz speed utilizing 1GB of RAM. With this Delphi implementation, the experiments now run 5x faster than the previous MatLab code.

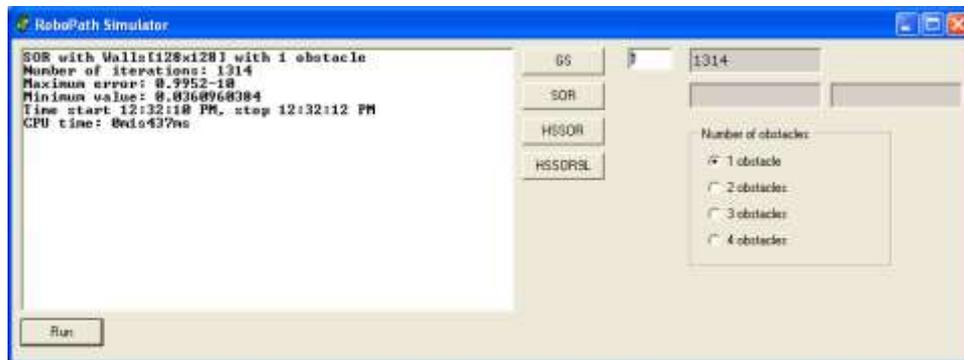


Figure 4: RoboPath Simulator was written in Delphi for very fast computation.

The iteration process stops when the computation converged to a specified value, i.e.  $1.0^{-10}$ , where there are no more significant changes in temperature values. Table 1 shows the number of iterations, maximum error and elapsed time recorded for the experiments. It is clear that HSSOR9L performs much faster than the various previous methods, see Figure 5. As shown in Table 2, it is interesting to observe that the speed of computation gets faster with the increase in number of obstacles, since nodes occupied by obstacles are simply ignored during the iteration process.

Table 1. Performance comparison of GS vs SOR vs HSSOR vs HSSOR9L in 1 obstacles setup.

	Iterative methods	Sizes of environment		
		128x128	256x256	512x512
Number of iteration	GS	21552	78522	281220
	SOR	1314	5059	18699
	HSSOR	630	2596	9756
	HSSOR9L	530	2190	8240
Maximum error	GS	$0.9999^{-10}$	$0.9999^{-10}$	$0.9999^{-10}$
	SOR	$0.9952^{-10}$	$0.9985^{-10}$	$0.9998^{-10}$
	HSSOR	$0.9945^{-10}$	$0.9952^{-10}$	$0.9991^{-10}$

	HSSOR9L	0.9734 <sup>-10</sup>	0.9940 <sup>-10</sup>	0.9988 <sup>-10</sup>
Elapsed time (m:s:ms)	GS	0:15:517	3:52:336	61:00:070
	SOR	0:00:829	0:14:782	04:16:000
	HSSOR	0:00:203	0:03:031	00:44:656
	HSSOR9L	0:00:219	0:03:312	00:49:281

GS: Gauss-Seidel; SOR: Successive Over-Relaxation; HSSOR: Half-Sweep SOR; HSSOR9L: HSSOR via Nine-Point Laplacian.

Table 2. Performance of HSSOR9L with varying number of obstacles.

	Number of obstacles	Sizes of environment		
		128x128	256x256	512x512
Number of iterations	1	530	2190	8240
	2	512	2131	8040
	3	499	2089	7903
	4	435	1883	7212
Maximum error	1	0.9734 <sup>-10</sup>	0.9940 <sup>-10</sup>	0.9988 <sup>-10</sup>
	2	0.9808 <sup>-10</sup>	0.9927 <sup>-10</sup>	0.9983 <sup>-10</sup>
	3	0.9789 <sup>-10</sup>	0.9972 <sup>-10</sup>	0.9999 <sup>-10</sup>
	4	0.9949 <sup>-10</sup>	0.9988 <sup>-10</sup>	0.9981 <sup>-10</sup>
Elapsed time in m:s:ms	1	0:0:219	0:3:312	0:49:281
	2	0:0:219	0:3:219	0:48:250
	3	0:0:218	0:3:156	0:48:375
	4	0:0:188	0:2:843	0:43:125

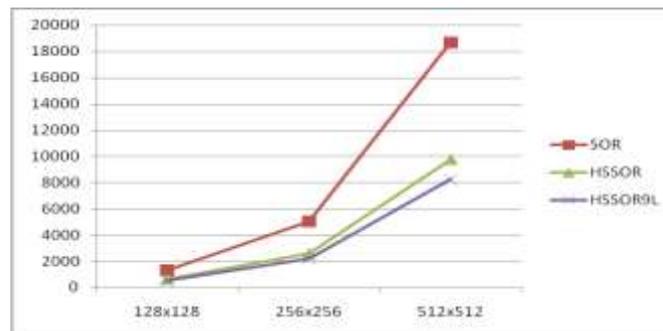


Figure 5: Graph performance of the three iterative methods. Number of iterations against sizes of environment model in 1 obstacle setup.

Once the temperature distributions were obtained, the path was generated by performing steepest descent search from the start points to the goal point. The process of generating the paths was very fast. From the current point, the algorithm simply picked the lowest temperature value from its eight neighbouring points. This process continues, until the generated path reached the goal point. Figure 6 (a) - (d) shows the generated paths in varying number of obstacles.

## VI. CONCLUSIONS

The simulation results show that solving robot path planning problem using numerical techniques are indeed very attractive and feasible. The performance of the proposed HSSOR9L iterative method was significantly faster than the previous methods [16 - 19]. Future work would consider a more difficult environment model setup. Furthermore, faster iteration including block methods (Sulaiman *et al.* [24] and Akhir *et al.* [25]) and quarter-sweep methods (Muthuvalu & Sulaiman [26, 27]) would also be considered.

## REFERENCES

- [1] Khatib, O. 1985. Real time obstacle avoidance for manipulators and mobile robots. IEEE Transactions on Robotics and Automation 1:500-505.
- [2] Koditschek, D.E. 1987. Exact robot navigation by means of potential functions: Some topological considerations. Proceedings of the IEEE International Conference on Robotics and Automation: 1-6.
- [3] Connolly, C. I., Burns, J.B. & Weiss, R. 1990. Path planning using Laplace's equation. Proceedings of the IEEE International Conference on Robotics and Automation: 2102-2106.



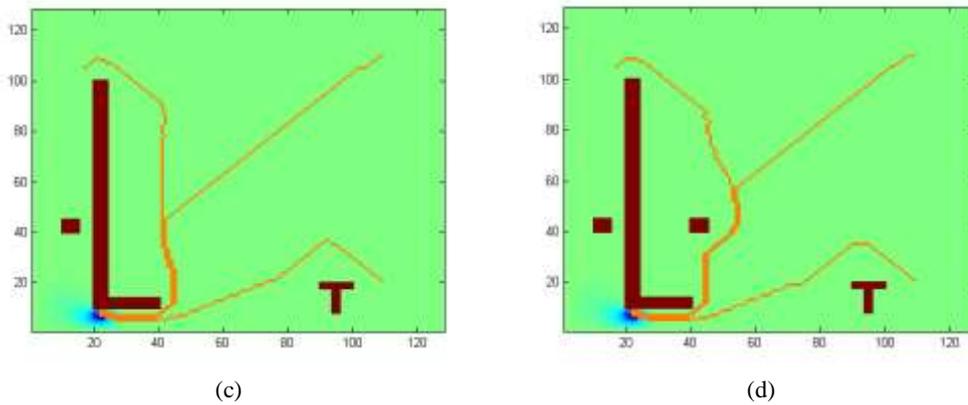


Figure 6: The paths generated with Half-Sweep SOR via Nine-Point Laplacian iterative method. (a) One obstacle; (b) Two obstacles; (c) Three obstacles; (d) Four obstacles.