Journal of Engineering and Applied Sciences 14 (23): 8855-8861, 2019
ISSN: 1816-949X

Application of Half-Sweep MAOR Iterative Method on Autonomous Robot Path Planning in Static Indoor Environment

[1]A.A. Dahalan, [1]W.K. Ling, [2]A. Saudi and [3]J. Sulaiman
[1]Department of Mathematics, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia
[2]Faculty of Computing and Informatics,
[3]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia a.qilah@upnm.edu.my

Abstract: Mobile robots have been undergoing constant research and development to improve its capability, especially, in its ability plan its path and move to specified destination in a given environment. However, there is much room for improvement of mobile robot path planning efficiency. This study attempts to improve the path planning efficiency of mobile robots by solving the path planning problems iteratively by using numerical method. This method of solution is based on harmonic function that applies the Laplace's equation to control the generation of potential function over the regions found in the mobile robot's configuration space. This study proposed the application of Half-Sweep Modified Accelerated Over-Relaxation (HSMAOR) iterative method to solve the mobile robot path planning problem. By using approximation finite scheme, the experiment was able to produce smooth path planning for the mobile robot to move from its starting point to its goal point. Other than that, the experiment also shows that this numerical method of solving path planning problem is faster and is able to produce smoother path for the mobile robot's point to point movements.

Key words: Robot navigation, collision free, optimal path, iterative method, five-point laplacian operator, Zalf-Sweep Modified Accelerated Over-Relaxation (HSMAOR)

INTRODUCTION

In recent years, many attempts have been undergone to improve the autonomous navigation of robot for moving from its starting point to the designated goal point. The main challenge is to develop a highly efficient robot motion planning navigating through various obstacles in order to complete its journey without any collision in between. This study attempts to apply the theory of heat transfer on the robot path planning in a point to point simulation. The harmonic potential values which determines the movement of robot from one point to another in the simulation configuration space is where the heat transfer theory is applied. By applying the heat transfer model into the simulation, we can avoid the occurrence of local minima in the environment which voids the performance of robot movement in the simulation. It should also be noted that the heat transfer model in this study is represented through the Laplace's equation.

For this purpose, this study proposed the application of Half-Sweep Modified Accelerated Over-Relaxation (HSMAOR) as the iterative method in generating the robot's path from start point to designated goal point in the simulation. The results are then compared to the currently available iterative methods based on the time taken to complete the path intended and the iterations number. The results were that the HSMAOR yielded better performance compared to its predecessing methods.

Brief history of related studies: The application of potential functions in solving the path planning problem has been implemented first by Khatib (1985). It treats all the boundaries and obstacles as a repellent force omitting barriers and the goal point as point which produces attractive force. Based on the research by Connolly *et al.* (1990) and Akishita *et al.* (1993) which applied harmonic functions in generating path for autonomous navigation, has proven that the implementation of harmonic functions has improved the performance of the robot autonomous navigation in the configuration space by reducing the time taken for generating path and prevented any collisions along the path. Also, through their

Corresponding Author: A.A. Dahalan, Department of Mathematics, National Defence University of Malaysia, 57000 Kuala Lumpur, Malaysia, a.qilah@upnm.edu.my

independently developed global methods via. Laplace's equation the occurrence of local minima in the configuration space were also avoided, resulting in an efficient path line generation by the autonomous navigation. Other than that, Sasaki experimented with numerical technique in enhancing the path generation of autonomous navigation and has proposed that the applied method had resulted in efficient navigation through complex maze simulation. On the other hand, Karonava made use of Dijkstra's shortest path algorithm via. image processing where the images are compared in terms of its pixel intensity to compute the length of the shortest path between the pixels. The results showed that the algorithm was able to produce the shortest path for the robot to move from point to point in a fairly large sized maze under a short range of time. Hachour (2008) proposed the application hybrid intelligent in autonomous navigation by merging two different programming languages where one is the visual basic language, a programming language that enable the robot to avoid all obstacles while in motion, the other one is the Delphi language, a programming language that guides the robot motion through the shortest path to reach the designated goal point.

MATERIALS AND METHODS

Laplacian potential in autonomous path planning: Laplace's equation, also known as the steady-state heat equation (Evans, 1998) is used in generating the path line for autonomous navigation in this study. The process of generating path lines can be modelled as a heat transfer problem where harmonic functions are computed as the solution for the Laplace's equation. The way it works is that the harmonic functions are calculated throughout the entire region of the configuration space where in the configuration space exists the starting point, outer boundaries, inner walls, obstacles and last but not least the goal point. The outer boundaries, inner walls and obstacles are fixed with a constant temperature and were treated as heat sources whereas the goal point is assigned with the lowest potential value and is treated as a sink that pulls the heat in. Following the concept of heat conduction where heat will flow from region of higher temperature to another region with lower temperature, this phenomenon of heat distribution represented by the Laplacian potential values will produce heat flux lines that will flow to the region with lowest potential value which in this case is the sink, filling the configuration space. This way, by following the heat flux line produced, the path line for the robot to navigate through the configuration space was laid out. And because the implementation of harmonic functions as shown by Connolly *et al*. (1990) that prevents the occurrence of local minima and able to guide the robot to avoid obstacles in the configuration space, the path of the robot's navigation towards the goal point is thus certain.

In understanding the concept mathematically, a harmonic function in the domain • • R^n is a function that satisfies Laplace's Eq. 1:

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

Where:

x_i = The ith Cartesian coordinate

n = The dimension

In the case of generating robot path, the domain • consists of the outer boundaries, inner walls, obstacles, start points and goal point. Since, harmonic functions complies to the min-max principle, it allows the Laplace's Eq. 1, when applied, to limit the generation of functions in the configuration space, thus preventing the occurrence of false local minimum inside the domain •. The implementation of harmonic functions enabled a smooth and efficient robot navigation path due to the complete path planning algorithms it produces. In previous related works, Laplace's equation has been solved using various standard numerical technique (Evans, 1985; Evans and Yousif, 1986; Ibrahim, 1993) such as Jacobi method, Gauss-Seidel (GS) method and Standard Over-Relaxation (SOR) method. This study attempted to solve Eq. 1 through modified accelerated iterative method for a more efficient computation.

In this study, the Laplacian Model was implemented for generating path line for autonomous navigation. Taking into account the temperature and heat flux present which translates to potential value and path line in the configuration space, respectively, the solution for Laplace's Eq. 1 was computed. The experiment utilizes a two-dimensional domain which consists of outer boundary, inner walls and various shape of obstacles to represent the configuration space. The iterative method that was used for computing Eq. 1 in this study was HSMAOR, through this, the temperature value at each node was determined, thus the potential values. For the purpose of comparison, the performance of autonomous navigation via. various iterative method was also tested. The other iterative methods include, Full-Sweep SOR (FSSOR), Half-Sweep SOR (HSSOR), Full-Sweep Modified

SOR (FSMSOR), Half-Sweep Modified SOR (HSMSOR), Full-Sweep AOR (FSAOR), Half-Sweep AOR (HSAOR), Full-Sweep Modified AOR (FSMAOR) and Half-Sweep Modified AOR (HSMAOR).

The half-sweep modified accelerated over relaxation method: The half-sweep iterative method is where only half of the nodal points in the configuration space are considered for calculation. It was introduced by Abdullah (1991) through the Explicit Decoupled Group (EDG) that was used to solve the 2D Poisson equation. Other use of the method includes for solving partial differential equation as shown by Ibrahim and Abdullah (1995), Yousif and Evans (1995), Abdullah and Ali (1996), Dahalan *et al.* (2013, 2014, 2015), a modified version of this method was also considered to be used to solve the diffusion equation (Sulaiman *et al.*, 2004). The early usage of combination between SOR with other technique dated back to 2009 by Sulaiman. In previous resesarches, the techniques used in solving the Laplace's Eq. 1 was such as standard GS (Connolly *et al.*, 1990) and SOR (Saudi and Sulaiman, 2013). The technique used in this study is the MAOR iterative method which is a modified accelerated iteration procedure and is a faster technique in solving the Laplace's Eq. 1. Consider the 2-dimensional Laplace's equation in Eq. 1 defined as:

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad (2)$$

Equation 2 is then simplified into the five-point standard finite difference approximation equation by applying the second-order central difference scheme as:

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0 \quad (3)$$

Since, the half-sweep method is utilized, the way the Laplace's Eq. 2 was iterated was that only half of the nodal points was considered in the computation and that the iterative method was simply done by repeatedly replacing each nodal value with the average nodal value of the nodal point's four neighbours. The nodal values which was assigned to the outer boundaries, inner walls, obstacles and goal point was kept constant.

By using the half-sweep method, the computational time was able to be reduced to half, since, the method enable only half of the nodal points to be taken into account when performing calculation. To better understand the half-sweep concept, Fig. 1 shows the computational grid for both full-sweep method and half sweep method.

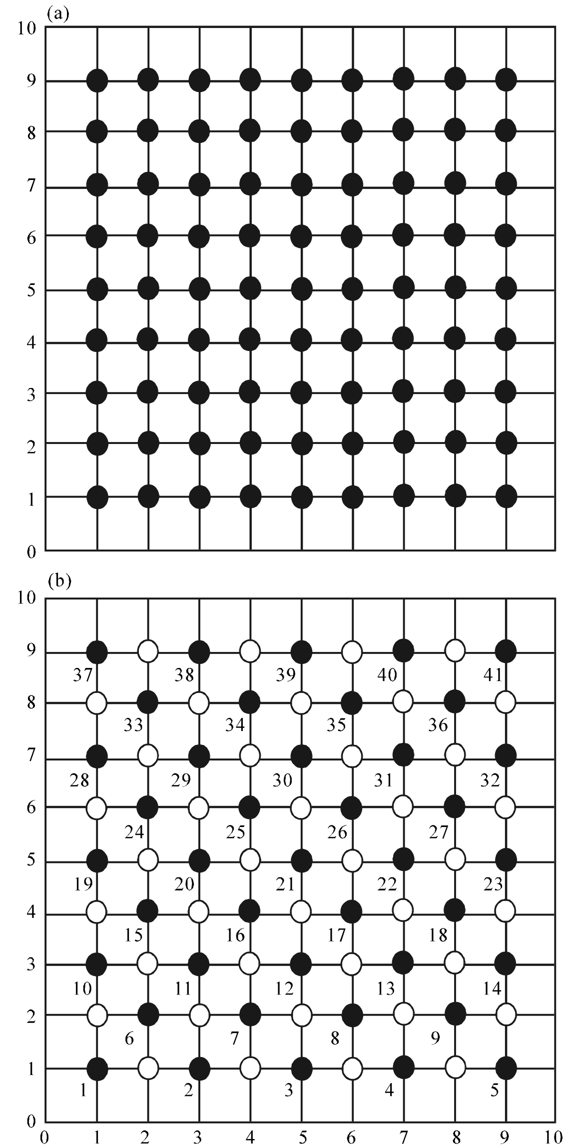


Fig. 1: The computational grids for: a) Full-sweep and b) Half-sweep

As can be seen from Fig. 1, the black dots represent the nodal point that were considered by the respective methods during computation, thus when applying full-sweep method, all of the nodal points in the configuration space were considered, whilst when applying half-sweep method only half of the nodal points were considered, therefore, reducing the computational time dramatically.

As to the computational pattern of both full-sweep and half-sweep method, Fig. 2 shows the information where the full-sweep method has a standard computational pattern for finite difference

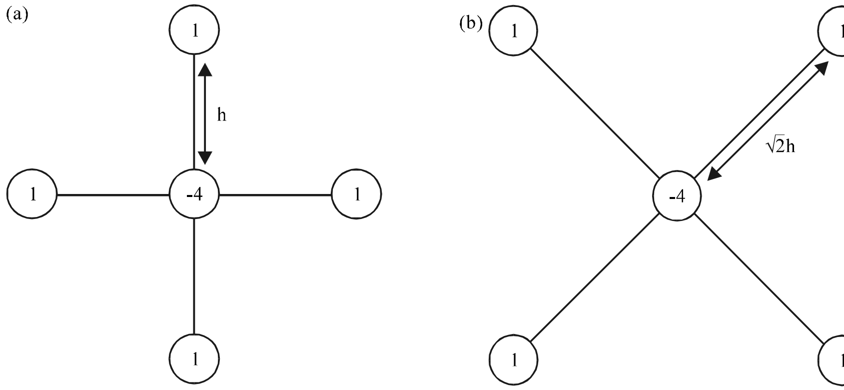


Fig. 2: The computational pattern for finite difference approximation of: a) Full-sweep (standard) and b) Half-sweep (rotated)

approximation while the half-sweep method has the rotated pattern of computation as opposed to the standard pattern.

As seen from Fig. 2, the approximation equation of half-sweep method was actually based on the cross-orientation operator where by rotating the i-j plane by 45°, the approximation equation was able to be obtained. The result was that the rotated five-point approximation formula was produced:

$$U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - 4U_{i,j} = 0 \quad (4)$$

After that SOR was implemented into Eq. (4) by adding a weighted parameter (Sulaiman *et al*., 2004, 2009), thus, the half-sweep iteration is shown as:

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

To improve the computational performance, a method known as Modified Accelerated Over-Relaxation (MAOR) method was implemented into Eq. 3. The fact that the formulation for MAOR iterative method are similar to AOR method. However, the MAOR iteration schemes involves the implementation of red-black ordering strategy through the use of three different weighted parameters r, • and • '. The formulation of family of HSMAOR methods produce 2 formulae, one for the red nodes:

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

another one for the black nodes, shown below:

$$\begin{aligned} U_{i,j}^{(k+1)} = {} & \frac{r}{4}\left[U_{i-1,j+1}^{(k+1)} - U_{i-1,j+1}^{(k)} + U_{i-1,j-1}^{(k+1)} - U_{i-1,j-1}^{(k)}\right] + \\ & \frac{r}{4}\left[U_{i+1,j+1}^{(k+1)} - U_{i+1,j+1}^{(k)} + U_{i+1,j-1}^{(k+1)} - U_{i+1,j-1}^{(k)}\right] + \\ & \frac{\omega'}{4}\left[U_{i-1,j+1}^{(k)} + U_{i+1,j+1}^{(k)} + U_{i-1,j-1}^{(k)} + U_{i+1,j-1}^{(k)}\right] + (1-\omega')U_{i,j}^{(k)} \end{aligned} \quad (7)$$

in which r, • and • ' are the optimum relaxation parameters and was defined in the range by Khatib (1985) and Connolly *et al*. (1990). There is no general formula in the effort to obtain the minimum number of iterations by determining the optimum values of r, • and • '. Based on the researches of Hadjidimos (Saudi and Sulaiman, 2013) the value r and • ' is normally chosen to be close to the value • of the corresponding SOR.

RESULTS AND DISCUSSION

In testing the performance of the proposed method, this study has set up 3 environments of different sizes, i.e., 300×300, 600×600 and 900×900 in terms of grid sizes. The elements present in the environment include, the outer boundary, the inner walls, various obstacles and the goal point. These elements were assigned with their own temperature value. By applying the Dirichlet boundary condition, the outer boundary, the inner walls and the obstacles were assigned with high temperature values, whereas the goal point was given the lowest. The temperature value of all other points was set to zero while no initial value was assigned to the starting point.

After the preparations were all set, the computational process begins, utilizing a PC running at 2.50 GHz speed with 8 GB of RAM. The computation process started and the temperature values at all points in the environment was calculated, the process continued on until stopping condition was met where when there were no more changes in temperature values and the difference of harmonic potential values between iterations k and k+1 were small enough that is around 1.0-10. When the stopping conditions were met, the loop of computation would then be terminated. The stopping conditions

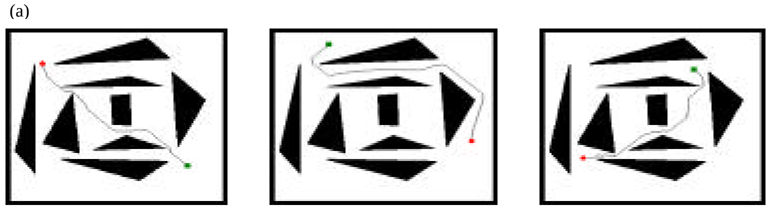
Table 1: Performance of the considered methods in terms of number of iteration

Methods	N×N 300×300	600×600	900×900
Case 1			
FSSOR	1728	8117	17831
FSAOR	1591	7529	16594
FSMSOR	1583	7557	16697
FSMAOR	1524	7311	16069
HSSOR	837	4108	9086
HSAOR	759	3803	8420
HSMSOR	747	3812	8484
HSMAOR	708	3671	8190
Case 2			
FSSOR	2228	8776	19254
FSAOR	2006	7973	17538
FSMSOR	2097	8323	18307
FSMAOR	1872	7542	16617
HSSOR	1071	4438	9813
HSAOR	944	4023	8924
HSMSOR	988	4198	9314
HSMAOR	855	3787	8435
Case 3			
FSSOR	3624	14644	33004
FSAOR	3236	13165	29680
FSMSOR	3402	13814	31194
FSMAOR	3023	12395	28037
HSSOR	1780	7445	16856
HSAOR	1568	6681	15149
HSMSOR	1659	7006	15912
HSMAOR	1448	6271	14284
Case 4			
FSSOR	2507	9868	21654
FSAOR	2288	9025	19840
FSMSOR	2395	9411	20667
FSMAOR	2169	8623	18949
HSSOR	1212	5000	11036
HSAOR	1097	4555	10098
HSMSOR	1155	4769	10526
HSMAOR	1028	4351	9643

Table 2: Performance of the considered methods in terms of CPU time (sec)

Methods	N×N 300×300	600×600	900×900
Case 1			
FSSOR	8.13	227.95	1134.25
FSAOR	8.61	230.17	1148.87
FSMSOR	6.72	240.99	1227.39
FSMAOR	7.44	247.99	1295.65
HSSOR	2.39	81.24	404.15
HSAOR	1.72	73.76	369.91
HSMSOR	2.13	73.03	373.18
HSMAOR	2.19	81.73	431.96
Case 2			
FSSOR	10.69	251.72	1270.23
FSAOR	10.27	248.24	1226.66
FSMSOR	9.36	269.68	1355.34
FSMAOR	9.30	267.18	1360.64
HSSOR	2.95	86.77	445.70
HSAOR	2.75	76.79	403.25
HSMSOR	2.64	80.20	414.86
HSMAOR	2.34	86.67	450.83
Case 3			
FSSOR	16.22	427.27	2190.45
FSAOR	18.66	418.45	2073.25
FSMSOR	14.40	462.03	2361.08
FSMAOR	15.35	450.60	2420.88
HSSOR	5.16	154.79	783.72
HSAOR	4.80	137.18	721.94
HSMSOR	4.08	140.88	739.14
HSMAOR	4.66	151.04	803.10
Case 4			
FSSOR	11.02	281.85	1441.47
FSAOR	12.52	281.78	1423.54
FSMSOR	9.78	309.74	1576.44
FSMAOR	9.83	309.98	1581.29
HSSOR	3.58	102.16	510.22
HSAOR	3.08	92.44	471.17
HSMSOR	3.28	94.51	482.17
HSMAOR	3.27	100.31	533.66

(a)



(b)

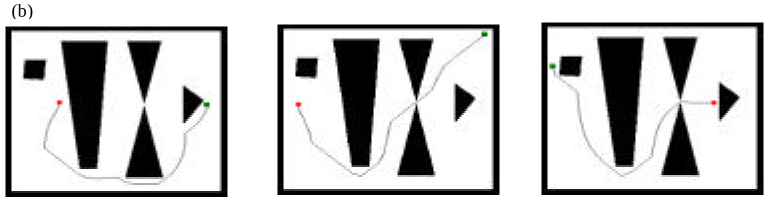


Fig. 1: Continue

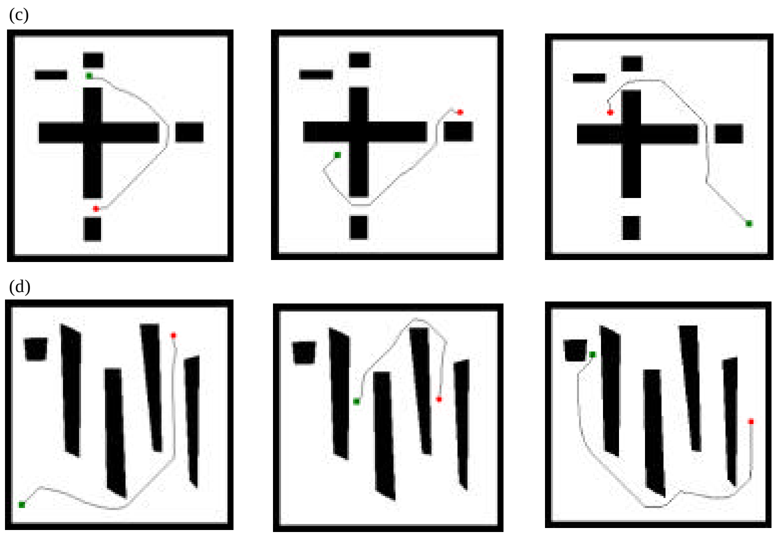


Fig. 3: The generated paths from several different start and goal positions for various environment: a) Case 1; b) Case 2; c) Case 3 and d) Case 4

regarding the small difference of harmonic potential values are very important as to prevent a condition known as 'saddle point' from occurring where flat area could occur in the solution that could fail the robot path line generation. The results were shown, where Table 1 shows the performance of the considered methods in terms of number of iteration and Table 2 shows the performance of the considered methods in terms of CPU time (sec).

As seen from results obtained, it was found out that the HSMAOR iterative method allow better performance. After succeeding in obtaining the potential values present in the environment, the path of the navigating robot can be constructed. It follows the principle of heat distribution using the steepest descent method, the algorithms will follow the descent in temperature flowing to the sink which in this case the goal point, thus, creating path line for navigation. Figure 3 shows the path line generated through HSMAOR with different starting point and goal point which follows the temperature distribution profile obtained through numerical computation. The path line generated had successfully avoided any obstacles while navigating from the start point (green dot/square dot) to the goal point (red dot/circle dot). It also goes on to show that path was able to be generated regardless of where the starting point and goal point was set to be.

CONCLUSION

The results of performance obtained in this study concludes that by numerically solving the Laplace's Eq. 1, the path planning problem's solution may not seem too large of an issue. By utilizing the current available technologies and new-found numerical techniques, the idea of having a high performance self-navigating robot may not seem too far fetch of an idea at all. In this study, it was proven that the HSMAOR iterative method is more efficient in solving path planning problem, compared to traditional methods that is SOR and AOR methods. The increase in number of obstacles in the environment the robot have to go through also does not affect the navigation performance, in fact the computation became more faster due to the fact that areas occupied by obstacles are ignored during computation. In future works, other than the concept of half-sweep iteration method, the quarter-sweep (Dahalan *et al*., 2016, 2017) iteration method can also be considered as another attempt to further speed up computational efficiency.

ACKNOWLEDGEMENT

The researcher acknowledge grant of FRGS/1/2018/ICT02/UPNM/03/1 from National Defence University of Malaysia for the funding of this study. The researchers declares that there is no conflict of interest regarding the publication of this study.

REFERENCES

Abdullah, A.R. and N.H.M.Ali, 1996. A comparative study of parallel strategies for the solution of elliptic PDE's. Parallel Algorithms Appl., 10: 93-103.

Abdullah, A.R., 1991. The four point Explicit Decoupled Group (EDG) method: A fast poisson solver. Int. J. Comput. Math., 38: 61-70.

Akishita, S., T. Hisanobu and S. Kawamura, 1993. Fast path planning available for moving obstacle avoidance by use of Laplace potential. Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93) Vol. 1, July 26-30, 1993, IEEE, Yokohama, Japan, pp: 673-678.

Connolly, C.I., J.B. Burns and R. Weiss, 1990. Path planning using Laplace's equation. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 3, May 13-18, 1990, Cincinnati, OH., pp: 2102-2106.

Dahalan, A.A., J. Sulaiman and M.S. Muthuvalu, 2014. Performance of HSAGE method with Seikkala derivative for 2-D fuzzy poisson equation. Applied Math. Sci., 8: 885-899.

Dahalan, A.A., J. Sulaiman and N.S.A. Aziz, 2017. Age iterative method applied to 2D fuzzy poisson equation (application method of age placement against poisson 2D blur equation). J. Eng. Appl. Sci., 12: 4437-4441.

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2013. Numerical solutions of two-point fuzzy boundary value problem using half-sweep alternating group explicit method. AIP Conf. Proc., 1557: 103-107.

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2015. Half-sweep successive over relaxation iterative method to solve two-point fuzzy boundary value problems. AIP Conf. Proc., Vol. 1682. 10.1063/1.4932412

Dahalan, A.A., M.S. Muthuvalu and J. Sulaiman, 2016. QSAGE iterative method applied to fuzzy parabolic equation. AIP Conf. Proc., Vol. 1705. 10.1063/1.4940272

Evans, D.J. and W.S. Yousif, 1986. Explicit group iterative methods for solving elliptic partial differential equations in 3-space dimensions. Int. J. Comput. Math., 18: 323-340.

Evans, D.J., 1985. Group explicit iterative methods for solving large linear systems. Int. J. Comput. Math., 17: 81-108.

Evans, L.C., 1998. Partial Differential Equations, Graduate Studies in Mathematics. American Mathematical Society, USA.

Hachour, O., 2008. Path planning of autonomous mobile robot. Int. J. Syst. Appl. Eng. Dev., 2: 178-190.

Ibrahim, A. and A.R. Abdullah, 1995. Solving The two dimensional diffusion equation by the four point Explicit Decoupled Group (EDG) iterative method. Int. J. Comput. Math., 58: 253-256.

Ibrahim, A., 1993. The study of the iterative solution of boundary value problem by the finite difference methods. Ph.D Thesis, National University of Malaysia, Bangi, Malaysia.

Khatib, O., 1985. Real-time obstacle avoidance for manipulators and mobile robots. Proceedings of the 1985 IEEE International Conference on Robotics and Automation Vol. 2, March 25-28, 1985, IEEE, St. Louis, Missouri, pp: 500-505.

Saudi, A. and J. Sulaiman, 2013. Robot path planning using Laplacian Behaviour-Based Control (LBBC) via half-sweep SOR. Proceedings of the 2013 International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE'13), May 9-11, 2013, IEEE, Konya, Turkey, ISBN:978-1-4673-5612-1, pp: 424-429.

Sulaiman, J., M. Othman and M.K. Hasan, 2009. Nine point-EDGSOR iterative method for the finite element solution of 2D poisson equation. Proceedings of the International Conference on Computational Science and its Applications, June 29-July 2, 2009, Seoul, Korea, pp: 764-774.

Sulaiman, J., M.K. Hasan and M. Othman, 2004. The half-sweep iterative alternating decomposition explicit (HSIADE) method for diffusion equations. Comput. Inform. Sci., 3314: 57-63.

Yousif, W.S. and D.J. Evans, 1995. Explicit de-coupled group iterative methods and their parallel implementations. Paral. Algorithms Appli., 7: 53-71.