Journal of Computer Science & Computational Mathematics, Volume 9, Issue 2, June 2019
DOI: 10.20967/jcscm.2019.02.002

Harmonic Path Planning using Two-Stage Half-Sweep Arithmetic Mean Method

Azali Saudi[1*], Jumat Sulaiman[2]

[1]Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia
[2]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia
[*]Corresponding author email: azali@ums.edu.my

Abstract: This paper presents the application of a two-stage Half-Sweep Arithmetic Mean (HSAM) iterative method for computing the solution of Laplace's equation (also known as harmonic functions) in two-dimensional space to solve the path planning problem in indoor environment. Several path planning simulations in a known indoor environment were conducted to examine the effectiveness of the proposed method. It is shown that the designed path planning algorithm is capable of generating smooth paths from various start and goal positions. Also, numerical results show that the proposed HSAM method converges much faster than the existing iterative methods, thus it drastically improves the overall performance of the path planning algorithm.

Keywords: path planning, palf-sweep arithmetic mean method, Laplace's equation, Harmonic functions.

1. Introduction

Path planning algorithm attempts to find a smooth and safe path from start point to a specified destination point. This paper describes the application of a mathematical model for solving path planning problem by utilizing the solutions of Laplace's equation (also known as harmonic functions) to generate a virtual floor gradient that can be used for automated vehicle navigation.

In [1], it was demonstrated that harmonic functions very useful for unmanned vehicle navigation, since they offer complete search algorithm. Such complete algorithm guarantees a solution if it exists. In [2], a global approach to path planning problem was developed by employing harmonic functions which does not suffer from local minima problem. Similar studies were conducted using the same global approach to solve various robot motion and vehicle navigation problems. In [3], harmonic functions were obtained through finite elements method to solve robot motion problem. Pedersen and Fossen [4] developed the path planning of a machine vessel by utilizing harmonic functions through potential flow. In [5, 6], harmonic functions were applied successfully for behaviour-based robot motion. Harmonic functions were also applied successfully for 3D path planning of Unmanned Aerial Vehicles [7].

2. Related Works

The common global approach to solve path planning problem utilizes harmonic functions for automated vehicle navigation. In this approach, obstacles act as current sources that produce high potentials, whereas the goal acts as a sink that poses the lowest potential value. This setup can be model using Dirichlet boundary conditions. Once the potential values in each point of the environment are obtained, the standard Gradient Descent Search (GDS) can be executed for path finding by descending the gradient of the harmonic potentials from higher values (start point) to the lowest potential value (goal point) [2].

In the previous studies, the harmonic potentials were obtained by employing the classical Gauss-Seidel (GS) [2] and standard Successive Over-relaxation (SOR) [8] iterative methods, where it was shown that SOR clearly outperformed GS method. Later, Saudi and Sulaiman [9] developed faster technique by combining SOR method with half-sweep [5, 6] red-black strategy.

The main challenge faced in the previous works was the immense amount of computational requirement when very large environment was involved. In this paper, we present the application of the two-stage HSAM method to compute the harmonic potentials. HSAM method is very suitable for parallel implementation since it involves the computation of two independent systems. It improves the execution time of the existing Arithmetic Mean (AM) method [10, 11] by drastically reducing the amount of computation by employing half-sweep iteration approach. Previously, HSAM method was successfully applied on one-dimensional space problem to solve Poisson equation [12], linear Fredholm integral equation [13] and Composite 6-Point Closed Newton-Cotes Quadrature Algebraic Equation [14]. In this study, the proposed HSAM considered two-dimensional space problem. For performance comparison purposes, the classical GS and the full-sweep variant of AM method were also considered.

3. Arithmetic Mean Methods

Given 2D Laplace's equation defined as

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \tag{1}$$

with Dirichlet boundary conditions. Obstacles(including inner and outer walls) and the goal point are set with constant high potentials and lowest potential values, respectively. When Eq. (1) is applied to model the distribution of potential values in the environment for path planning problem, it often results in large linear system with sparse coefficient matrix. Most often, iterative method is employed to solve such large and sparse linear system. The standard five-point finite difference formula to approximate Eq. (1) is given as

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0. \tag{2}$$

In [15], the environment grid is rotated by 45°, thus the rotated five-point finite difference formula is obtained and defined as

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

Accordingly, these two approximation equations (2) and (3) can be used to develop the full-sweep and half-sweep iterative schemes, respectively. The corresponding portion of computational grids about point (i, j) for full-sweep and half-sweep iteration are shown in Figure 1 (a) and (b).

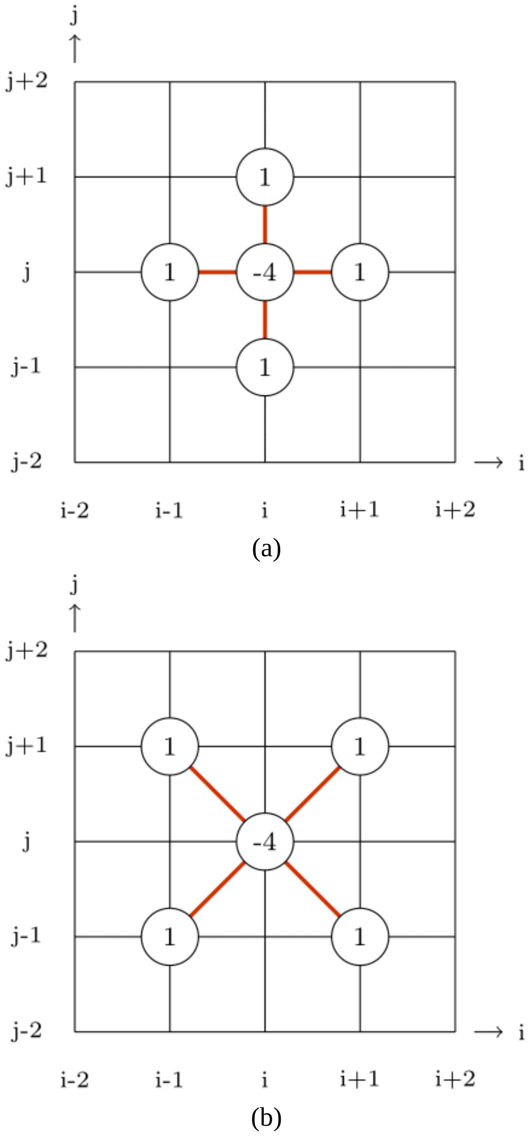

Figure 1. Portion of the computational grid about point (i, j) for (a) full-sweep, and (b) half-sweep iteration, respectively

The Gauss-Seidel (GS) iterative scheme for Eq. (2) can be written as [16]

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

The above Eq. (4) is used to develop GS iterative method.

To accelerate the computation, a weighted parameter $w$is used to obtain the iterative schemes for full-sweep and half-sweep, and defined as [17]

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

$$u_{i,j}^{(k+1)} = \frac{w}{4}\left(u_{i-1,j-1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{(i-1,j+1)}^{(k)} + u_{(i+1,j+1)}^{(k)}\right) + (1-\omega)u_{i,j}^{(k)}. \qquad (6)$$

The two equations (5) and (6) are then used to develop the AM and HSAM methods, respectively. When these finite difference approximation equations are applied to Eq. (1), it will result in a large and sparse linear system that can be stated in matrix form as

$$Au = b \qquad (7)$$

Where A and b are known, and u is unknown. We can express the matrix A = (a_{ij}) as the matrix sum

$$A = L + D + U \qquad (8)$$

Where matrix D = diag{a_{11}, a_{22}, ..., a_{nn}},L and U are strictly lower and upper triangular matrices, respectively.

AM is a two-stage iterative method and its iterative process involves of solving two independent systems such as $\hat{u}^{(1)}$and $\hat{u}^{(2)}$. With a weighted parameter w, the general iterative scheme for AM method is obtained and defined as

$$\left.\begin{aligned} (D+wL)\hat{u}^{(1)} &= \big((1-w)D - wU\big)u^{(k)} + wb, \\ (D+wL)\hat{u}^{(2)} &= \big((1-w)D - wL\big)u^{(k)} + wb, \\ u^{(k+1)} &= \tfrac{1}{2}\big(\hat{u}^{(1)} + \hat{u}^{(2)}\big). \end{aligned}\right\} \qquad (9)$$

This iterative method (9) is convergent for 0 <w< 2. An identical proof is provided in [10]. The optimal value of$w$ can be obtained by conducting several experiments to find the least iterations to converge. We assume that the values of matrices D, L and $U$are determined, as illustrated in Figure 1. The implementations of AM and HSAM methods are described in Algorithms 1 and 2, respectively. In both algorithms, the two-step iteration procedures at Level 1 and 2 and matrix sum at Level 3 are repeatedly carried out until the specified convergence requirement $\left\|u^{(k+1)} - u^{(k)}\right\| < \epsilon$ is satisfied, where ϵ is the convergence criterion. The iteration processes in Level 1 and 2 are execution in forward sweep and reverse order, respectively. During the iteration process, only non-occupied nodes are computed. All other occupied nodes (by obstacles or inner and outer walls) are skipped. To obtain the harmonic functions, the algorithm is performed explicitly until the convergence criterion is satisfied. With AM method, all nodes in the problem domain will be considered, see Figure 2 (a). Whereas for HSAM method, only black nodes are considered during the iteration process, the remaining white nodes are calculated after the convergence using direct method [15], as shown in Figure 2 (b). Therefore, the amount of computation for HSAM method is approximately 50% less than AM method.

Algorithm 1: AM method

Level 1 (Sweep forward)
For $i, j = 0,1,2, \ldots, N-1, N$ **do**
Compute

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

Level 2 (Sweep backward)
For $i, j = N, N-1, N-2, \ldots, 1, 0$ **do**
Compute

$$u_{i,j}^{(2)} = \frac{w}{4}\left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(2)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(2)}\right) + (1-w)u_{i,j}^{(k)}$$

Level 3
For $i, j = 0,1,2, \ldots, N-1, N$ **do**
Compute

$$u_{i,j}^{(k+1)} = \frac{1}{2}\left(u_{i,j}^{(1)} + u_{i,j}^{(2)}\right)$$

Algorithm 2: HSAM method

Level 1 (Sweep forward)
For firstnode **to** lastnode **do**
Compute

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

Level 2 (Sweep backward)
For lastnode **to** firstnode **do**
Compute

$$u_{i,j}^{(2)} = \frac{w}{4}\left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(2)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(2)}\right) + (1-w)u_{i,j}^{(k)}$$

Level 3
For firstnode **to** lastnode **do**
Compute

$$u_{i,j}^{(k+1)} = \frac{1}{2}\left(u_{i,j}^{(1)} + u_{i,j}^{(2)}\right)$$

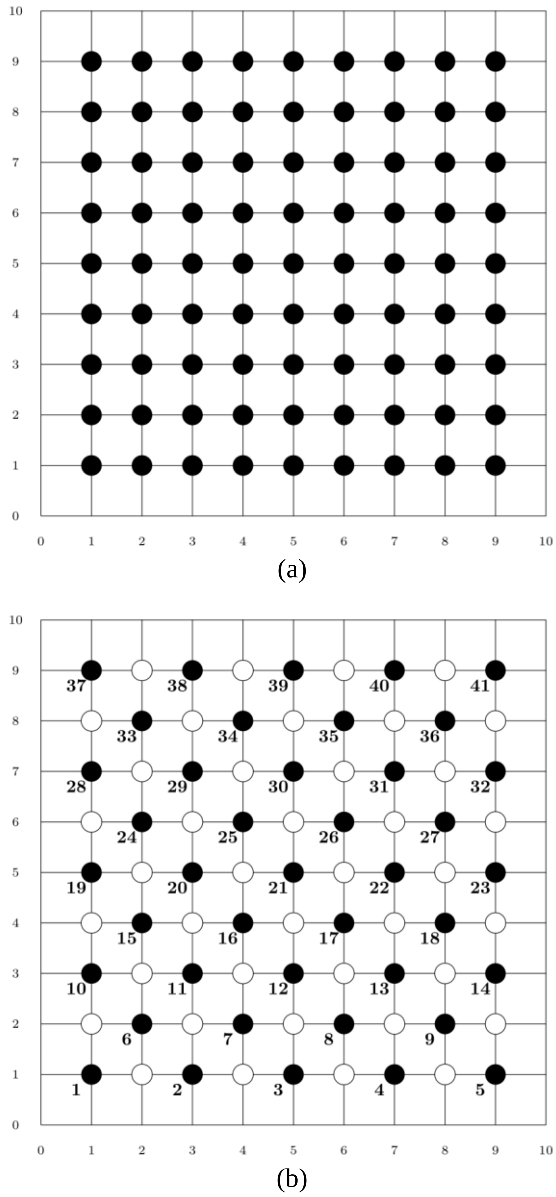

Figure 2. The computational grid for (a) full-sweep and (b) half-sweep cases, respectively

4. The Path Planning Simulation

The path planning simulation is designed in virtual static environment using the considered algorithms. Two static indoor maps that cover an area of 270 x 270 and 290 x 290 are used to represent the virtual environment of Case 1 and Case 2, respectively. The two maps contain various shapes of obstacles including inner and outer boundary walls.

Different initial potential values are assigned to the obstacles, goal point and non-occupied nodes. Nodes occupied by obstacles (including inner and outer walls) and goal point are set high potential values and lowest potentials, respectively. Random values are assigned to all non-occupied nodes. The experiments are conducted on a machine with an Intel i3 2100 CPU running at 3GHz clock.

The simulation begins by using the considered methods to compute the harmonic potentials of the virtual environment. For AM variants, the optimal weighted value of *w* is obtained by conducting several runs of experiments until it gives the least number of iterations. The iteration processes for computing the harmonic potentials continue until the convergence criterion is satisfied. The convergence criterion must be set to a very small error tolerance, i.e. $\epsilon = 1.0^{-15}$, since lower precision is not sufficient to avoid flat area in the resulting harmonic potential values. After the harmonic potentials of the environment are obtained, the required path from start to the specified goal point can be traced using the standard GDS. The GDS simply repeatedly moves to the next node with lower potential value until the goal point with the lowest potential value is found out. The implementation of path planning algorithm is described in Algorithm 3.

Algorithm 3: Path Planning Algorithm

1. Load the map of the environment
2. Setup matrices to store potential values of the environment
3. Set potential values of all boundary nodes
4. Set potential values of goal point
5. Initialize potential values for other free spaces
6. Compute the harmonic functions using the considered methods (e.g. Algorithm 1 and 2 for AM and HSAM methods, respectively).
7. Perform GDS on the obtained harmonic functions to generate path from start to goal point
8. Save the generated path

5. Results and Discussion

Table 1 shows the performance of the considered methods in computing the harmonic potentials, in terms of iteration counts and CPU time. Based on this table, it is shown that the proposed HSAM method clearly outperforms the GS and AM methods. The AM method performs much better than the standard GS method, where the iteration counts and CPU time are reduced by approximately 95% and 87%, respectively. The proposed HSAM method gives the best performance. In comparison to AM method, it reduces the iteration counts and CPU time by approximately 32% and 56%, respectively.

After the convergence of the considered methods, the obtained harmonic potential values are then utilized to find path from start to the specified goal point using GDS. Figures 3 and 4 illustrate the resulting generated paths for Case 1 and 2, respectively. The corresponding GREEN and RED dots are used to represent the start and goal points. From these figures, it is proven that the path planning algorithm is capable of generating smooth and safe paths from several different start and goal points. The algorithm is also very robust, since it capable of navigating the narrow corridor and small area in the corner.

6. Conclusion

In this paper, the great potential of the two-stage HSAM iterative method in computing the harmonic functions for application in path planning problem is presented. The proposed HSAM method had significantly improved the overall performance of the path planning algorithm in terms of execution time and its capability in generating paths in open space, narrow corridor and small corner. Both AM variants are suitable for parallel implementation since they possess two systems that can be calculated at Level 1 and 2 independently.

For future work, the path planning algorithm will be tested in higher dimensional space and dynamic environment. More advanced iterative methods such as the combination of AM variants with block iteration and quarter-sweep concept will also be examined.

Acknowledgment

We thank the Ministry of Education (Malaysia) under Grant FRGS0434-ICT-1/2016 for the financial support.

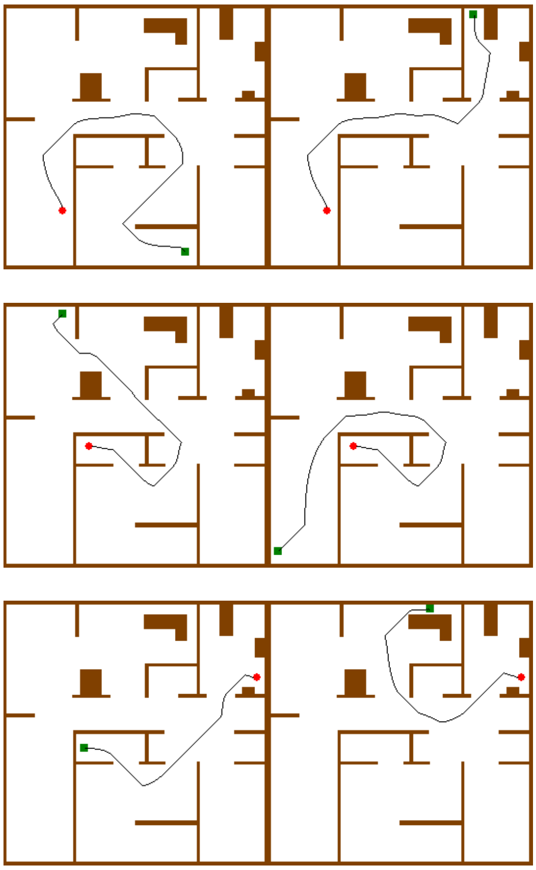

Figure 3. The generated paths for Case 1 covering an area of 270 x 270

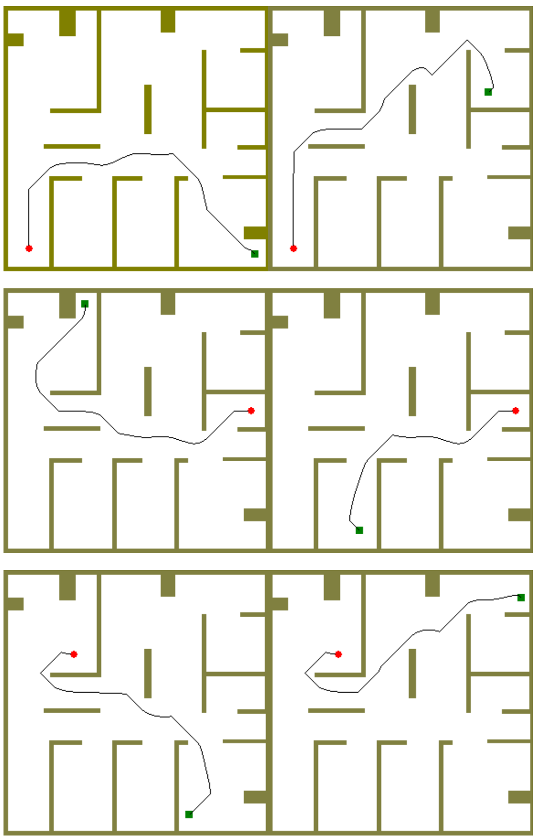

Figure 4. The generated paths for Case 2 covering an area of 290 x 290

Table 1. Iteration, *k* and CPU time (in seconds), *t* of the considered methods

Case 1:

Methods	***k***	***t***
GS	20309	28.16
AM	1407	4.16
HSAM	760	1.45

Case 2:

Methods	***k***	***t***
GS	20302	27.67
AM	1423	4.28
HSAM	803	1.55

References

[1] C. I. Connolly, R. A. Grupen, “The applications of harmonic functions to robotics”, *Journal of Robotic Systems*, vol. 10, no. 7, pp. 931-946, 1993.

[2] C. I. Connolly, J. B. Burns, R. Weiss, “Path planning using Laplace's equation”, In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2102-2106, 1990.

[3] S. Garrido, L. Moreno, D. Blanco, F. Martin Monar, “Robotic motion using harmonic functions and finite elements”, *Journal of Intelligent and Robotic Systems*, vol. 59, no. 1, pp. 57-73, 2010.

[4] M. D. Pedersen, T. I. Fossen, “Marine vessel path planning and guidance using potential flow”, In *Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft*, pp. 188-193, 2012.

[5] A. Saudi, J. Sulaiman, “Robot path planning using Laplacian Behaviour-Based Control via Half-Sweep Gauss-Seidel (LBBC-HSGS) iterative method”, *International Journal of Applied Mathematics and Statistics*, vol. 36, no. 6, pp. 68-76, 2013a.

[6] A. Saudi, J. Sulaiman, “Robot path planning using Laplacian Behaviour-Based Control (LBBC) via Half-Sweep SOR”, In *Proceedings of the International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*, pp. 424-429, 2013b.

[7] X. Liang, H. Wang, D. Li, C. Liu, “Three-dimensional path planning for unmanned aerial vehicles based on fluid flow”, In *Proceedings of the IEEE Aerospace Conference*, pp. 1-13, 2014.

[8] S. Sasaki, “A practical computational technique for mobile robot navigation”, In *Proceedings of the IEEE International Conference on Control Applications*, vol. 2, pp. 1323-1327, 1998.

[9] A. Saudi, J. Sulaiman, “Red-black strategy for mobile robot path planning”, In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, pp. 2215-2220, 2010.

[10] V. Ruggiero, E. Galligani, “An iterative method for large sparse linear systems on a vector computer”, *Computers & Mathematics with Applications*, vol. 20, no. 1, pp. 25-28, 1990.

[11] J. Sulaiman, M. Othman, M. K. Hasan, “”A new Half-Sweep Arithmetic Mean (HSAM) algorithm for two-point boundary value problems”, In *Proceedings of the International Conference on Statistics and Mathematics and its Application in the Development of Science and Technology*, pp. 169-173, 2004.

[12] J. Sulaiman, M. Othman, M. K. Hasan, “A fourth-order finite difference solver for Poisson's equations via the Half-Sweep Arithmetic Mean method”, In *Proceedings of the First IMT-GT Regional Conference on Mathematics, Statistics and Their Applications*, pp. 139-146, 2005.

[13] M. S. Muthuvalu, J. Sulaiman, “Half-sweep arithmetic mean method with composite trapezoidal scheme for solving linear fredholm integral equations”, *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5442-5448, 2011.

[14] M. S. Muthuvalu, E. Galligani, M. K. M. Ali, J. Sulaiman, R. S. Lebelo, “Performance Analysis of Arithmetic Mean Method for Solving Composite 6-Point Closed Newton-Cotes Quadrature Algebraic Equation”, *Asian-European Journal of Mathematics*, 2018.

[15] A. R. Abdullah, “The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver”, *International Journal of Computer Mathematics*, vol. 38, no. 1-2, pp. 61-70, 1991.

[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003, ISBN 0898715342.

[17] D. M. Young, *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*. PhD thesis, PhD Thesis. Harvard University, 1950.