International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:13, No:4, 2019

# Autonomous Vehicle Navigation Using Harmonic Functions via Modified Arithmetic Mean Iterative Method

Azali Saudi, Jumat Sulaiman

*Abstract*—Harmonic functions are solutions to Laplace's equation that are known to have an advantage as a global approach in providing the potential values for autonomous vehicle navigation. However, the computation for obtaining harmonic functions is often too slow particularly when it involves very large environment. This paper presents a two-stage iterative method namely Modified Arithmetic Mean (MAM) method for solving 2D Laplace's equation. Once the harmonic functions are obtained, the standard Gradient Descent Search (GDS) is performed for path finding of an autonomous vehicle from arbitrary initial position to the specified goal position. Details of the MAM method are discussed. Several simulations of vehicle navigation with path planning in a static known indoor environment were conducted to verify the efficiency of the MAM method. The generated paths obtained from the simulations are presented. The performance of the MAM method in computing harmonic functions in 2D environment to solve path planning problem for an autonomous vehicle navigation is also provided.

*Keywords*—Modified Arithmetic Mean method, Harmonic functions, Laplace's equation, path planning.

## I. Introduction

MANY scientific problems often require immense amount of computing resources for solving large linear system. It is also well known that iterative methods are suitable for such large scale computations of linear system problem. In the literature, the existing two-stage iterative method namely Arithmetic Mean (AM) method and its variants have been extensively applied for solving various types of linear systems. The AM method was first introduced by Galligani and Ruggiero [10] for solving linear system on a vector computer. After that, Sulaiman et al. [11] developed a new variant of AM method known as Half-Sweep Arithmetic Mean (HSAM) method. Later, another AM variant namely Quarter-Sweep Arithmetic Mean (QSAM) method was developed to solve diffusion equations [12]. In [14], HSAM was successfully applied for solving linear Fredholm integral equations. Furthermore, a block variant of AM method was also applied for solving first kind linear Fredholm integral equations [15]. In this paper, we propose a variant of AM method known as MAM method for computing the solutions of Laplace's equation. Then, its application in autonomous vehicle navigation in indoor environment is demonstrated to verify its efficiency to solve path planning problem. The performance of the proposed MAM method will be compared with the standard Gauss-Seidel (GS) and existing AM iterative methods.

The details of harmonic functions are described in Section II. The finite difference method and the proposed MAM method is explained in Sections III and IV, respectively. Section V discusses the path planning algorithm. Results are presented in Section VI. Finally, the conclusion are discussed in Section VII.

A. Saudi is with the Cloud Computing and Intelligent Robotics Lab, Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia (e-mail: azali@ums.edu.my).

J. Sulaiman is with Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Kota Kinabalu (e-mail: jumat@ums.edu.my).

## II. Harmonic Functions

A harmonic function on a domain $\Omega \subset R^n$ is a function which satisfies Laplace's equation

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

where $x_i$ is the $i$-th Cartesian coordinate and $n$ is the dimension. In the case of path construction for an autonomous vehicle navigation, the boundary of $\Omega$ (denoted by $\partial\,\Omega$) consists of all obstacles and goals in a configuration space representation. The solutions to Laplace's equation are computed with Dirichlet boundary conditions:

$$\partial\,\Omega = c \tag{2}$$

where $c$ is constant. Harmonic functions satisfy the min-max principle [5], therefore spontaneous creation of a false local minimum inside the region is avoided if Laplace's equation is imposed as a constraint on the functions used.

Harmonic functions are known to have a number of properties useful in robotics [6]. They offer a complete path planning algorithm and paths derived from them are generally smooth. When applied to path planning of robots, they have the advantage over simple potential field based approach, as they exhibit no spurious local minima. The use of potential functions for robot path planning, as introduced by Khatib [16], views every obstacle to be exerting a repelling force on an end effector, while the goal exerts an attractive force. Koditschek [7], using geometrical arguments, showed that, at least in certain types of domains there exists potential functions which can guide the effector from almost any point to a given point. The usual formulation of potential fields for path planning does not prevent the spontaneous creation of local minima other than the goal. This may cause the robot

to terminate its path at such a minimum and achieve a stable configuration short of goal.

Connolly et al. [5] and Akishita et al. [19] independently developed a global method using solutions to Laplace's equations for path planning to generate a smooth, collision-free path. After that, several studied were conducted using similar idea. Garrido et al. [20] used harmonic functions obtained through finite elements method for robotic motion. Szulczynski et al. [17] demonstrated the application of harmonic potential functions for real-time obstacle avoidance. Saudi and Sulaiman [1], [2] applied block iteration procedure to compute the harmonic functions for solving path planning problem. Similar approaches were also successfully employed to behaviour-based robot [3], [4]. Harmonic functions via potential flow were also used for marine vessel path planning [13]. Also, 3D path planning for Unmanned Aerial Vehicles (UAV) based on fluid flow of harmonic functions was reported in [21].

Essentially, in the above approaches, the potential field is computed in a global manner and the harmonic solutions to Laplace's equation are used to find the path lines for an autonomous vehicle to move from the start point to the goal point. Obstacles are considered as current sources and the goal is considered to be the sink, with the lowest assigned potential value. This amounts to using Dirichlet boundary conditions. Then, by performing the GDS, a succession of points with lower potential values leading to the point with the least potential value (i.e. goal point) is found out. It was observed by Connolly et al. [5] that this process guarantees a path to the goal without encountering local minima and successfully avoiding any obstacle, as a harmonic function cannot possess an extremum value except at the domain boundary.

This study follows the above paradigm for path planning, by using the analogy of temperature and heat flux for the potential and path line, respectively. The experiments are carried out on two-dimensional domains having various shapes of obstacles and boundary walls.

## III. Finite Difference Method

Numerical solutions for Laplace's equation are readily obtained from finite difference methods. Based on (1), the 2D Laplace's equation can be stated as

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \tag{3}$$

Although this system can be solved using direct method, the more efficient iterative methods are used to compute the solutions, since its application in path planning problem often resulting in large linear system with sparse coefficient matrix. The main advantage of iterative solution is that the storing of large matrices is unnecessary. However, one of the disadvantages of iterative methods compared with direct methods is slow convergence or even divergence. Thus, iterative method in practice requires an appropriate stopping criterion. The simplest finite difference formula to approximate (2) is the five-point difference approximation:

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

Essentially, for Laplace's equation (3), this iterative method simply consists of repeatedly replacing each node value with the average of its four neighbours. Those node values which represent the inner and outer boundaries, obstacles and goal point are held fixed.

## IV. The MAM Method

### A. Formulation of MAM Method

Application of finite difference approximation (4) to problem (3) will result in a large and sparse linear system that can be stated in matrix form as

$$Au = b \tag{5}$$

where both matrix $A$ and the column vector $b$ are known and the column vector $u$ is unknown. The matrix $A$ has the form

$$A = \begin{bmatrix} T & I & & & & \\ I & T & I & & & \\ & I & T & & & \\ & & \ddots & \ddots & \ddots & \\ & & & I & T & I \\ & & & & I & T \end{bmatrix}_{(N-1)\times(N-1)}. \tag{6}$$

In which $A$ is an $(N-1)\times(N-1)$ block tridiagonal matrix, where each block $T$ is an $(N-1)\times(N-1)$ matrix

$$T = \begin{bmatrix} -4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ & 1 & -4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -4 & 1 \\ & & & & 1 & -4 \end{bmatrix}_{(N-1)\times(N-1)} \tag{7}$$

and each block $I$ is the $(N-1)\times(N-1)$ identity matrix

$$I = \begin{bmatrix} 1 & 0 & & & & \\ 0 & 1 & 0 & & & \\ & 0 & 1 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 0 & 1 & 0 \\ & & & & 0 & 1 \end{bmatrix}_{(N-1)\times(N-1)}. \tag{8}$$

Matrix $u$ and $b$ may be defined as

$$u = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{N-2,N-1} \\ u_{N-1,N-1} \end{bmatrix} \tag{9}$$

and

$$b = \begin{bmatrix} b_{1,1} \\ b_{2,1} \\ \vdots \\ b_{N-2,N-1} \\ b_{N-1,N-1} \end{bmatrix}. \tag{10}$$

As stated in the previous section, AM method is a two-stage iterative method and its iterative process involves of solving two independent systems such as $u^{(1)}$ and $u^{(2)}$. Now, let the coefficient matrix $A$ be decomposed into

$$A = D - L - T \tag{11}$$

where $D$, $L$ and $T$ are diagonal, strictly lower and strictly upper triangular matrices, respectively. Thus, by adding positive acceleration parameter, $\omega$, the general iterative scheme for AM method can be defined as [10]

$$\left.\begin{aligned}(D-\omega L)u^{(1)} &= ((1-\omega)D+\omega T)u^{(k)}+\omega b\\ (D-\omega T)u^{(2)} &= ((1-\omega)D+\omega L)u^{(k)}+\omega b\\ u^{(k+1)} &= \frac{1}{2}\Big(u^{(1)}+u^{(2)}\Big)\end{aligned}\right\} \quad (12)$$

where the optimal weighted parameter is in the range $1 \leq \omega < 2$ as given in [18], [8]. The AM method requires a slight additional computational effort of the sum of two matrices at each iteration $k$, but its rate of convergence is relatively insensitive to the exact choice of the parameter $\omega$ [10]. Several runs of simulations are required to be carried out to find the optimal value of $\omega$, where it give the smallest number of iterations. The general conditions which guarantee the convergence of AM method (12) are described in [18], [8].

The proposed MAM method is very much inspired by the work of Kincaid and Young [9] for their study on Modified Successive Overrelaxation (MSOR) method, where two acceleration parameters $\omega$ and $r$ were employed. By following similar idea, two parameters $\omega$ and $r$ are imposed into the first and second part of the original AM [10] equation (12), respectively. Thus, the MAM method can be defined as

$$\left.\begin{aligned}(D-\omega L)u^{(1)} &= ((1-\omega)D+\omega T)u^{(k)}+\omega b\\ (D-rT)u^{(2)} &= ((1-r)D+rL)u^{(k)}+rb\\ u^{(k+1)} &= \frac{1}{2}\Big(u^{(1)}+u^{(2)}\Big)\end{aligned}\right\} \quad (13)$$

where the optimal value of both weighted parameters are in the range $1 \leq \omega < 2$ and $1 \leq r < 2$ as given in [9]. The exact optimal parameter values are determined by running several simulations using different values of $\omega$ and $r$ until they give the least number of iterations. In the case of $\omega = r$, the MAM method simplifies to the standard AM method. From (13), the MAM method can be stated as follows

$$u^{(k+1)} = S_M u^{(k)} + c_M f \quad (14)$$

whereas

$$S_M = \frac{1}{2}\big(u^{(1)}+u^{(2)}\big) \quad (15)$$

where

$$u^{(1)} = (D-\omega L)^{-1}((1-\omega)D-\omega T)$$
$$u^{(2)} = (D-rT)^{-1}((1-r)D-rL)$$

and

$$c_M = \frac{1}{2}\big[\omega(D-\omega L)^{-1}+r(D-rT)^{-1}\big]. \quad (16)$$

The general conditions which guarantee the convergence of MAM method (13) are described in the following theorems:

**Theorem 1.** *Let $S_M$ be $(N-1)\times(N-1)$ matrix and the successive approximation (14) for $k = 0, 1, 2, ...$ converges for each $c_M \subseteq \Re^{(N-1)}$. Each $u^{(0)} \in C^{(N-1)}$ if and only if the spectral radius of the iteration matrix i.e. $S_M$ is less than 1, that is $\rho(S_M) < 1$.*

**Theorem 2.** *The necessary conditions for the MAM method to be convergent are that $0 < \omega < 2$ and $0 < r < 2$. The standard proof is given in [18], [8].*

By determining values of matrices $D$, $L$ and $T$ (11), the algorithm for MAM iterative method to solve problem (3) is given below:

INPUT:
Set value for parameters $\omega$ and $r$
Set value for convergence criterion, $\epsilon$

COMPUTATION:
$k = 0$
$t_1 =$ startclock
**repeat**
  Level 1
  Compute all non-occupied nodes with 1st part (13)
  **for** $i, j = 0, 1, 2, ..., N-1, N$ **do**
  $u_{i,j}^{(1)} \leftarrow \frac{\omega}{4}\big(u_{i-1,j}^{(1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(1)} + u_{i,j+1}^{(k)}\big) + (1-\omega)u_{i,j}^{(k)}$

  Level 2
  Compute all non-occupied nodes with 2nd part (13)
  **for** $i, j = N, N-1, N-2, ..., 1, 0$ **do**
  $u_{i,j}^{(2)} \leftarrow \frac{r}{4}\big(u_{i-1,j}^{(k)} + u_{i+1,j}^{(2)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(2)}\big) + (1-r)u_{i,j}^{(k)}$

  Level 3
  Compute all non-occupied nodes at $(k+1)$-th iteration
  using the third part of Eq. (13)
    **for** $i, j = 0, 1, 2, ..., N-1, N$ **do**
      $u_{i,j}^{(k+1)} \leftarrow \frac{1}{2}\big(u_{i,j}^{(1)} + u_{i,j}^{(2)}\big)$

**until** $||u^{(k+1)} - u^{(k)}|| < \epsilon$
$t_2 =$ stopclock

OUTPUT:
Capture the updated matrix, $u^{(k+1)}$
Capture number of iterations, $k$
Capture elapsed time, $t_e = t_2 - t_1$

The algorithm is explicitly performed until the convergence criterion is satisfied, where it is set to a very small value to avoid the occurrence of flat areas as further discussed in Section 5.

### *B. Optimal Parameter Value of MAM Method*

In the previous study [10], it was shown that the AM method converged with parameter value in the range $0 < \omega < 2$. However, the optimal value of this acceleration parameter, $\omega$ was in the range $1 \leq \omega < 2$ [18], [8]. Similarly, the MAM method also converged with the two parameter values in the range $0 < \omega < 2$ and $0 < r < 2$. The optimal values for both acceleration parameters, however, were in the range $1 \leq \omega < 2$ and $1 \leq r < 2$, as described in [9], [18], [8]. In order to determine the exact optimal parameter values, several runs of simulation are carried out to find the optimal values of $\omega$ and $r$ that give the least number of iterations. Based on previous studies [18], [8], the good acceleration parameter candidates were in the range 1.5 to 2. Hence, these ranges are used in computing the solutions to Laplace's equation (1) to obtain harmonic functions of the environment.

## V. Path Planning

Once the harmonic functions under the boundary conditions are established using Algorithm 1, the required path can be traced by the standard GDS. Starting from initial position, the

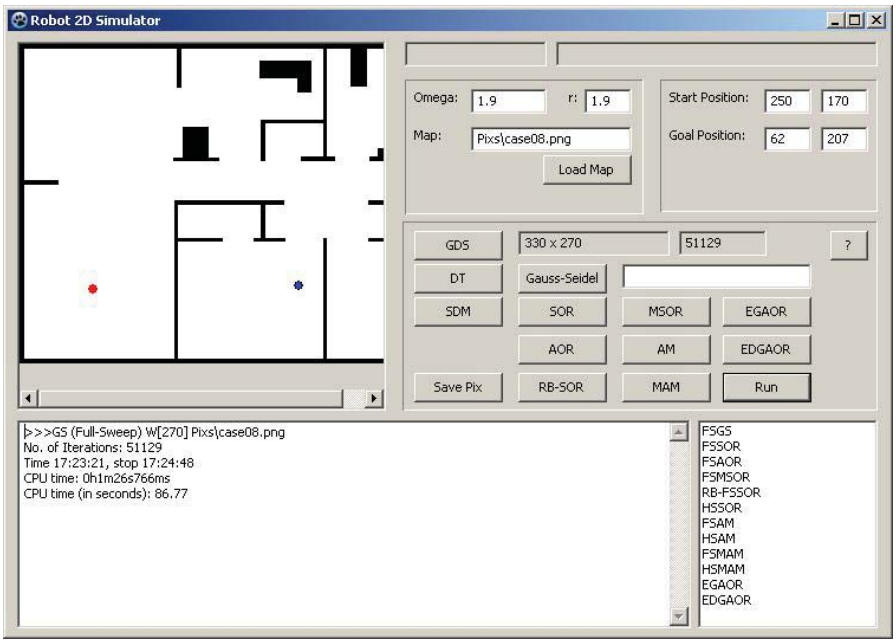


Fig. 1 The self developed Robot 2D Simulator

path searching algorithm through GDS simply moves to the next lowest neighbourhood point. This step continues until the lowest point that represents the goal point is found.

The simulations of an autonomous vehicle navigation by employing the path planning algorithm are run in a static known indoor environment. In the initial setup, the obstacles, inner and outer walls are fixed with high potential values, whilst the goal point is set to a fixed lowest potential value, and no initial values are assigned to all other free spaces. The computations are carried out on Intel machine running at 3.4 GHz speed with 16GB RAM. The codes are written in Pascal, and the generation of paths are simulated in the self-developed software namely Robot 2D Simulator, as shown in Figure 1. The simulator is developed in Lazarus.

The numerical representation used for these experiments is important. The iteration is terminated when the convergence criterion is satisfied. A very high precision is required, thus the implementation uses an 8 bytes variable storage of type Double for storing each potential value. The range value for Double is $5.0 \times 10^{-324}$ to $1.7 \times 10^{-308}$, and it can store up to 15 significant digits. The convergence criterion is set to a very small error tolerance i.e. $1.0^{-15}$, since lower precision is not sufficient to avoid flat areas in the resulting potential values. The path planning algorithm is described below:

INPUT:
Load map of the environment
Setup matrices $u^{(k)}, u^{(k+1)}, u^{(1)}, u^{(2)}$
Set the goal position
Set potential values for nodes occupied by boundaries and obstacles
Set potential values for nodes occupied by goal

COMPUTATION:
Compute harmonic functions using GS, AM or MAM (refer Algorithm 1) methods

Perform GDS on the obtained harmonic functions to find path from initial position to goal position
Draw the generated path

OUTPUT:
Save the generated path

## VI. Results

The path planning simulations were carried out in area of $330 \times 270$ with several different start points and goal points. In all of those simulations, the path planning algorithm had successfully generated smooth path from start point to the specified goal point. Figs. 2 and 3 illustrate the generated paths obtained from the path planning simulations. The solid square in green colour denotes start point, whilst the solid circle in red colour denotes goal point. In Figs. 2 (a) and (b), the autonomous vehicle has successfully moves from different start position to the same goal position. Figs. 3 (a)-(f) demonstrate the effectiveness of the path planning algorithm, where the vehicle has successfully moves from several different start and goal positions.

The performance of the considered iterative methods in terms of number of iterations and execution time are tabulated in Table I. As stated in the previous study [8], the good candidate value for the weighted parameter was in the range $1 \leq \omega < 2$, where value greater than 1.5 gave better performance. Hence, the tested weighted parameter, $\omega$ for the AM method was in the range 1.50 to 1.99, where it was found that the optimal value for the parameter was in the range $1.95 \leq \omega \leq 1.975$ (see Table I). Consequently, this optimal value (i.e. $1.95 \leq \omega \leq 1.975$) obtained in the AM method were then used to find the optimal parameter values of $\omega$ and

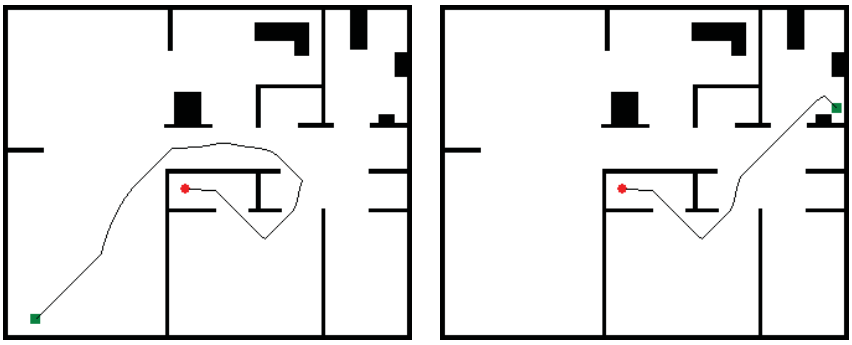


Fig. 2 The generated paths from two different start positions and same goal position

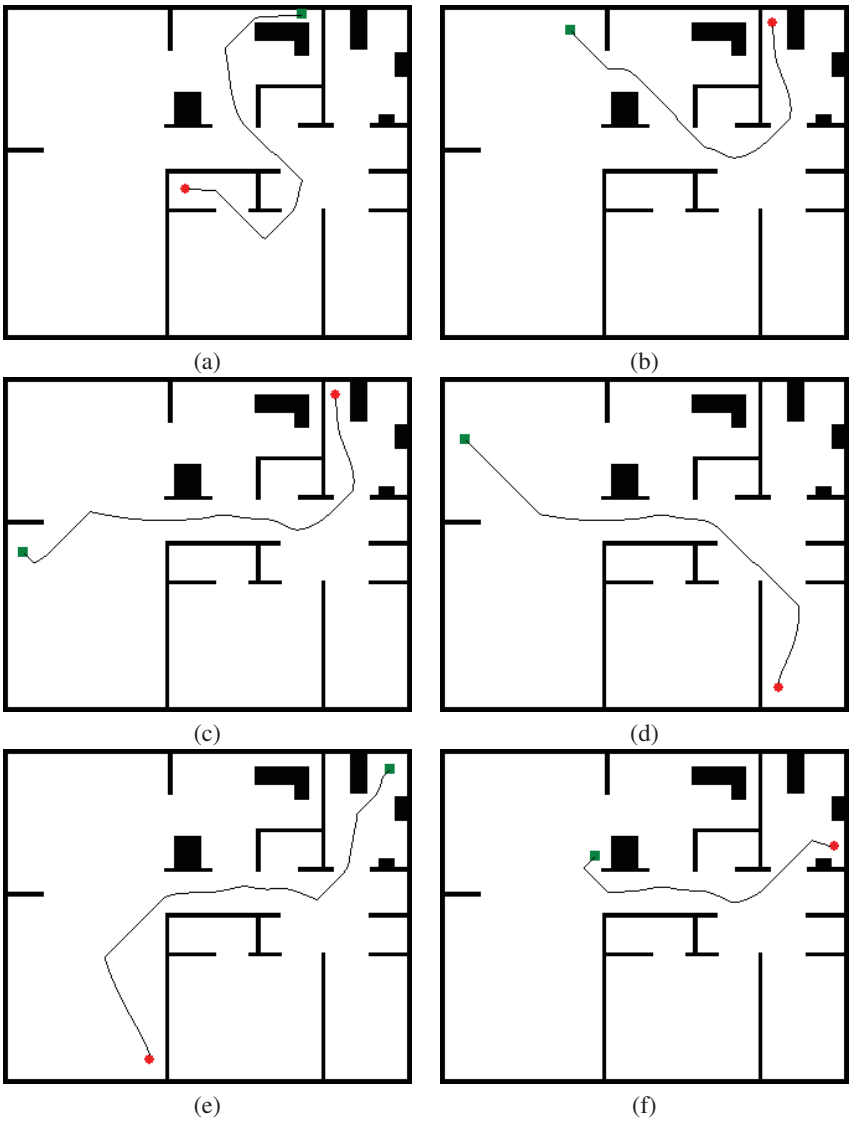


Fig. 3 The generated paths from several different start and goal positions

$r$ for the MAM method. Hence, based on the finding shown in Table I (i.e. the optimum value of $r$ was closed to $\omega$), the tested weighted parameters of $\omega$ and $r$ for the MAM method were set in the range $1.95 \leq \omega \leq 1.975$ and $1.91 \leq r \leq 1.99$,

TABLE I
NUMBER OF ITERATIONS ($k$) AND EXECUTION RIME ($t$) FOR THE CONSIDERED METHODS ON GRID SIZE OF $330 \times 270$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **GS** | $k$ | 51454 | | | | | | | | |
| | $t$ | 77.12 | | | | | | | | |
| **AM** | $\omega$ | 1.50 | 1.55 | 1.60 | 1.65 | 1.70 | 1.75 | 1.80 | 1.85 | 1.90 |
| | $k$ | 18020 | 15784 | 13696 | 11715 | 9838 | 8059 | 6366 | 4771 | 3263 |
| | $t$ | 68.95 | 61.37 | 53.59 | 45.42 | 38.23 | 31.09 | 25.00 | 18.72 | 13.01 |
| | $\omega$ | 1.91 | 1.92 | 1.93 | 1.94 | **1.95** | **1.955** | **1.96** | **1.965** | **1.97** |
| | $k$ | 2987 | 2709 | 2444 | 2192 | **1972** | **1873** | **1794** | **1743** | **1728** |
| | $t$ | 11.56 | 10.52 | 9.58 | 8.74 | **7.84** | **7.53** | **7.00** | **6.70** | **6.70** |
| | $\omega$ | **1.975** | 1.98 | 1.985 | 1.99 | | | | | |
| | $k$ | **1936** | 2445 | 3302 | 4904 | | | | | |
| | $t$ | **7.61** | 9.55 | 13.03 | 19.08 | | | | | |
| **MAM** | $\omega$ | 1.50 | 1.55 | 1.60 | 1.65 | 1.70 | 1.75 | 1.80 | 1.85 | 1.90 |
| | $r$ | 1.55 | 1.60 | 1.65 | 1.70 | 1.75 | 1.80 | 1.85 | 1.90 | 1.95 |
| | $k$ | 16830 | 14656 | 12617 | 10687 | 8846 | 7105 | 5427 | 3824 | 2220 |
| | $t$ | 67.81 | 57.45 | 49.39 | 42.59 | 35.50 | 27.86 | 21.23 | 15.16 | 8.78 |

TABLE II
NUMBER OF ITERATIONS ($k$) AND EXECUTION TIME ($t$) FOR MAM METHOD ON GRID SIZE OF $330 \times 270$ WITH OPTIMAL VALUE OF $1.95 \leq \omega \leq 1.975$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **MAM** | $\omega$ | 1.95 | 1.95 | 1.95 | 1.95 | 1.95 | 1.95 | 1.95 | 1.95 | 1.95 |
| | $r$ | 1.91 | 1.92 | 1.93 | 1.94 | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 2211 | 2175 | 2125 | 2061 | 1972 | 1836 | 1513 | 1513 | 1713 |
| | $t$ | 8.75 | 8.67 | 8.37 | 8.12 | 7.86 | 7.28 | 5.97 | 6.64 | 6.84 |
| | $\omega$ | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 |
| | $r$ | 1.91 | 1.92 | 1.93 | 1.94 | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 2059 | 2045 | 2015 | 1970 | 1971 | 1819 | 1589 | 1584 | 1752 |
| | $t$ | 8.03 | 8.58 | 7.89 | 7.89 | 7.97 | 7.17 | 6.24 | 6.84 | 6.92 |
| | $\omega$ | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 |
| | $r$ | 1.91 | 1.92 | 1.93 | 1.94 | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 1878 | 1881 | 1879 | 1871 | 1846 | 1794 | 1648 | 1626 | 1919 |
| | $t$ | 7.45 | 7.39 | 7.37 | 7.31 | 7.20 | 7.05 | 6.50 | 6.34 | 7.52 |
| | $\omega$ | 1.965 | 1.965 | 1.965 | 1.965 | 1.965 | 1.965 | 1.965 | 1.965 | 1.965 |
| | $r$ | 1.91 | 1.92 | 1.93 | 1.94 | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 1604 | 1655 | 1695 | 1729 | 1754 | 1756 | 1697 | 1763 | 2149 |
| | $t$ | 6.30 | 6.42 | 6.59 | 6.83 | 6.89 | 6.89 | 6.64 | 6.90 | 8.28 |
| | $\omega$ | 1.97 | **1.97** | **1.97** | **1.97** | 1.97 | 1.97 | 1.97 | 1.97 | 1.97 |
| | $r$ | 1.91 | **1.92** | **1.93** | **1.94** | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 1469 | **1437** | **1492** | **1430** | 1595 | 1686 | 1729 | 1939 | 2417 |
| | $t$ | 5.78 | **5.70** | **5.86** | **5.33** | 6.17 | 6.50 | 6.61 | 7.47 | 9.41 |
| | $\omega$ | 1.975 | **1.975** | **1.975** | **1.975** | 1.975 | 1.975 | 1.975 | 1.975 | 1.975 |
| | $r$ | 1.91 | **1.92** | **1.93** | **1.94** | 1.95 | 1.96 | 1.97 | 1.98 | 1.99 |
| | $k$ | 1508 | **1429** | **1482** | **1480** | 1529 | 1500 | 1766 | 2160 | 2784 |
| | $t$ | 5.89 | **5.58** | **5.84** | **5.75** | 5.94 | 5.86 | 6.91 | 8.45 | 2784 |

respectively.

Based on Table II, the optimal value of the MAM method for parameters $\omega$ and $r$ were in the range $1.97 \leq \omega 1.975$ and $1.92 \leq r \leq 1.94$, respectively. With these optimal parameter values, the number of iterations ($k$) and execution time ($t$) for the MAM method were less than $1500$ and $6$ seconds, respectively. Thus, the selection of parameter values for MAM method was much wider than AM method. In comparison to the standard GS method, both AM and MAM methods drastically reduced the number of iterations. Both AM and MAM methods also clearly outperformed GS method in terms of execution time. Overall, the proposed MAM method gave the best performance among the considered methods.

Table III shows the reduction percentage in terms of number of iterations and execution time between the currently suggested method and the previous methods. Compared to the standard GS method, AM method reduced the iteration numbers and execution time by 64.99% to 96.64% and 10.59% to 91.31%, respectively. Against the standard GS, the optimal AM (with $\omega = 1.97$) reduced the number of iterations and execution time approximately by 96% and 91%, respectively. Against AM method (with $1.50 \leq \omega \leq 1.90$), the proposed MAM method gave better performance, where the iteration numbers and execution time were further reduced by 6.60% to 31.96% and 1.65% to 32.51%, respectively. Hence, the proposed MAM method gave the best overall performance among the considered iterative methods.

## VII. CONCLUSION

The effectiveness of computing the harmonic functions using the proposed MAM method was demonstrated in this study, where it significantly improved the overall performance

TABLE III
REDUCTION PERCENTAGES OF NUMBER OF ITERATIONS AND EXECUTION TIME

| Methods | Number of iterations % | Execution time % |
|---|---|---|
| **AM against GS** | 64.99 - 96.64 | 10.59 - 91.31 |
| **MAM against AM** | 6.60 - 31.96 | 1.65 - 32.51 |

of the path planning algorithm. The calculations at Level 1 and Level 2 of (13) and (14) can be carried out independently. Therefore, AM and MAM methods are very suitable for parallel implementation.

For future work, the application of the proposed MAM method in space of higher dimensions may be examined. The combination of half-sweep iteration concepts with the existing AM and the proposed MAM methods, as demonstrated in Sulaiman et al. [11] and Muthuvalu and Sulaiman [14], are also an interesting ideas to explore for solving navigation problem of an autonomous vehicle. Also, investigations in more difficult domains such as cluttered and dynamic environments may be carried out.

## REFERENCES

[1] A. Saudi and J. Sulaiman. Block Iterative Method for Robot Path Planning. The 2nd Seminar on Engineering and Information Technology, Kota Kinabalu, July 7 8, 2009.
[2] A. Saudi and J. Sulaiman. Numerical Technique for Robot Path Planning using Four Point-EG Iterative Method. In proc. of the Int. Symposium on Information Technology, 2010, pp. 831 836.
[3] A. Saudi and J. Sulaiman. Indoor Path Planning for Mobile Robot using LBBC-EG. *Int. J. of Imaging and Robotics*, 11(3), 2013, pp. 37-45.
[4] A. Saudi and J. Sulaiman. Hybrid Path Planning for Indoor Robot with Laplacian Behaviour-Based Control via Four Point-Explicit Group. *Int. J. of Imaging and Robotics*, 12(1), 2014, pp. 12-21.
[5] C. I. Connolly, J. Burns, and R. Weiss. Path planning using Laplace's equation. In Proceedings of the IEEE International Conference on Robotics and Automation, May 13-18, 1990, Cincinnati, USA, pp. 2102-2106.
[6] C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7), 1993, pp. 931-946.
[7] D. E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In Proceedings of the IEEE International Conference on Robotics and Automation, March 31 - April 3, 1987, Raleigh, USA, vol. 4, 1987, pp. 1-6.
[8] D. M. Young. Iterative Methods for Solving Partial Difference Equations of Elliptic Type. PhD Thesis. Harvard University, 1950.
[9] D. R. Kincaid and D. M. Young. The modified successive overrelaxation method with fixed parameters. *Mathematics and Computations*, 119, 1972, pp. 705-717.
[10] I. Galligani and V. Ruggiero. The Arithmetic Mean method for solving essentially positive systems on a vector computer. *Int. J. Computer Math.*, 32, 1990. pp. 113121.
[11] J. Sulaiman, M. Othman and 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, 2004, pp. 169-173.
[12] J. Sulaiman, M. Othman and M. K. Hasan. A new Quarter Sweep Arithmetic Mean (QSAM) method to solve diffusion equations. *Chamchuri Journal of Mathematics*, 1(2), 2009, pp. 93-103.
[13] M. D. Pedersen and 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, Sep 19-21, 2012, Arenzano, Italy, pp. 188-193.
[14] M. S. Muthuvalu and J. Sulaiman. Half-Sweep Arithmetic Mean method with composite trapezoidal scheme for solving linear Fredholm integral equations. *Applied Mathematics and Computation*, 217(12), 2011, pp. 5442-5448.
[15] M. S. Muthuvalu and J. Sulaiman. An implementation of the 2-point block arithmetic mean iterative method for first kind linear Fredholm integral equations. *World Journal of Modelling and Simulation*, 8(4), 2012, pp. 293-301.
[16] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Mar 25-28, 1985, St. Louis, USA, vol. 2, 1985, pp. 500-505.
[17] P. Szulczynski, D. Pazderski and K. Kozlowski. Real-time obstacle avoidance using harmonic potential functions. *Journal of Automation Mobile Robotics and Intelligent Systems*, 5: 59-66.
[18] R. Kress. Numerical Analysis. New York: Springer-Verlag, 1998.
[19] S. Akishita, S. Kawamura, and K. Hayashi. Laplace potential for moving obstacle avoidance and approach of a mobile robot. In Japan-USA Symposium on Flexible Automation, July 9-13, 1990, Kyoto, Japan, pp. 139-142.
[20] S. Garrido, L. Moreno, D. Blanco, and M. F. Martin. Robotic motion using harmonic functions and finite elements. *Journal of Intelligent and Robotic Systems*, 59(1), 2010, pp. 57-73.
[21] X. Liang, H. Wang, D. Li, and C. Liu. Three-dimensional path planning for unmanned aerial vehicles based on fluid flow. In Proceedings of the IEEE Aerospace Conference, March 1-8, 2014, Big Sky, USA, pp. 1-13.