THE SOLUTION OF 2D ELLIPTIC EQUATION USING MODIFIED GEOMETRIC MEAN METHOD ON SKEWED GRID WITH RED-BLACK ORDERING

Shaliza Hayati A. Wahab¹, A’qilah Ahmad Dahalan², Azali Saudi³§
1,3Intelligent Robotics Lab
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Kota Kinabalu, MALAYSIA
²Department of Mathematics
Centre for Defence Foundation Studies
National Defence University of Malaysia
Kuala Lumpur, MALAYSIA

Abstract: This paper presents the development of a new variant of two-stage Geometric Mean (GM) method for solving 2D elliptic equation. The proposed iterative scheme, called the Skewed Modified Geometric Mean (SkMGM), is derived from finite difference approximation discretized on a skewed grid with red-black ordering. The skewed finite difference scheme combined with red-black ordering is shown to be suitable for parallel implementation. The developed SkMGM scheme is compared with the other methods on the standard grid to confirm the effectiveness of the proposed method in terms of computational complexity and execution time. It is shown that the new proposed method gives the least number of iterations and fastest execution time compared to the other tested methods.

AMS Subject Classification: 45A05, 45B05, 65D05, 65D30, 65F10, 65Y20
Key Words: Modified Geometric Mean method; partial differential equation; elliptic equation; red-black ordering; skewed finite difference method

Received: May 1, 2020
© 2020 Academic Publications
§Correspondence author
1. Introduction

The numerical iterative method based on the two-stage approach had been widely used for solving system of linear algebraic equations. In [5], the Alternating Group Explicit (AGE) was shown to be most suitable for solving linear systems arising from the discretization of elliptic partial differential equations (PDEs) on parallel computers. After that, the more accurate Iterative Alternating Decomposition Explicit (IADE) [11] than the AGE scheme was developed to solve parabolic problem. In [13] and [14], the more efficient methods than the standard IADE known as the Half- and Quarter-Sweep IADE methods were developed to solve one-dimensional diffusion problems. Allahviranloo et al. [4] studied the Block two-stage iterative method to find the approximate solution of fuzzy system. Another two-stage method is the Arithmetic Mean (AM) method which was used for solving large linear system and very suitable for parallel implementation [6, 8]. The more recent extended version of two-stage method known as the Half-Sweep Geometric Mean (HSGM) iterative method was employed for solving third order Newton-Cotes quadrature system [9].

Previous works demonstrated that the iteration procedure on skewed grid drastically reduced the computational complexity and speed up the convergence rate of the iterative methods [1, 3, 12, 16, 2, 10].

In this paper, we present a variant of Geometric Mean (GM) method known as Skewed Modified Geometric Mean (SkMGM) method for solving elliptic partial differential equation in two dimensional space. The performance of the proposed SkMGM method will be compared with the existing Gauss-Seidel (GS) iterative methods on standard and skewed grids with single color ordering. The modified GS and GM iterative methods on standard grid with red-black ordering are also implemented for comparison purposes.

The existing GM method is described in Section 2. The proposed Modified Geometric Mean (MGM) method on standard grid and the SkMGM on skewed grid are presented in Section 3. The computational complexity of the tested methods are explained in Section 4. Numerical results obtained from the experiments conducted on the proposed and existing methods are presented and summarized in Section 5. Section 6 presents the concluding remarks on the performance of the tested methods.
2. The Geometric Mean (GM) method

Consider a typical elliptic boundary value problem as follows:

\[ \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega \]  

(1)

with Dirichlet boundary conditions

\[ u(x, y) = g(x, y), \quad (x, y) \in \partial \Omega, \]

where \( \Omega \) is a bounded region in \( \mathbb{R}^2 \). Equation (1) is known as the Poisson equation which is used to model fluid dynamics phenomena and heat conduction problems. The simplest finite difference formula to approximate Equation (1) is the standard five-point difference approximation formula:

\[ u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = h^2 f_{i,j}. \]

(2)

Here, we assume a rectangular grid with \( h \) spacing in both directions. Another type of finite difference approximation is based on the cross orientation operator which can be obtained by rotating the \( i-j \) plane axis clockwise by 45° [1]. This will result in the skewed five-point approximation formula:

\[ u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} - 4u_{i,j} = 2h^2 f_{i,j}. \]

(3)

Essentially, the finite difference approximations in Equations (2) and (3) represent the standard and skewed grids, respectively. Figure 1 shows the computational mesh for the standard and skewed grids. Hence, the computational molecules for these approximations are depicted in Figure 2 (a) and (b).

The iterative schemes for the standard five-point difference formula can be written as below [17]:

\[ u_{i,j}^{(k+1)} = \frac{1}{4} \left( u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} - h^2 f_{i,j} \right). \]

(4)

While the iterative schemes for the skewed finite difference formula are given below:

\[ u_{i,j}^{(k+1)} = \frac{1}{4} \left( u_{i-1,j-1}^{(k+1)} + u_{i+1,j-1}^{(k)} + u_{i-1,j+1}^{(k)} + u_{i+1,j+1}^{(k)} - 2h^2 f_{i,j} \right). \]

(5)

For skewed grid cases, only black nodes are considered during the iteration process. Hence, the remaining white nodes are computed using the standard difference formula Equation (4), as depicted in Figure 2 (c).
Applying these finite difference approximations to problem (1) will result in a large and sparse linear system that can be stated in matrix form as

\[ Au = b, \]  

where \( A \) and \( b \) are known, and \( u \) is unknown. For simplification, let the coefficient matrix \( A \) be decomposed into

\[ A = D - L - T, \]  

where \( D \), \( L \), and \( T \) are matrices representing different operations.
where $D$, $L$ and $T$ are diagonal, strictly lower and strictly upper triangular matrices, respectively. The three matrices are expressed as follows:

$$
D = \begin{bmatrix}
\rho_{1,1} & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & \rho_{2,2} & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & \rho_{3,3} & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \rho_{N-3,N-3} & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & \rho_{N-2,N-2} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & \rho_{N-1,N-1}
\end{bmatrix}, \quad (8)
$$

$$
L = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\phi_{2,1} & 0 & 0 & \ldots & 0 & 0 & 0 \\
\chi_{3,1} & \phi_{3,2} & 0 & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\tau_{N-3,1} & \tau_{N-3,2} & \ldots & 0 & 0 & 0 & 0 \\
\tau_{N-2,1} & \tau_{N-2,2} & \ldots & \phi_{N-2,N-3} & 0 & 0 & 0 \\
\tau_{N-1,1} & \tau_{N-1,2} & \ldots & \chi_{N-1,N-3} & \phi_{N-1,N-2} & 0 & 0
\end{bmatrix}, \quad (9)
$$

and

$$
T = \begin{bmatrix}
0 & \phi_{1,2} & \chi_{1,3} & \ldots & \tau_{1,N-3} & \tau_{1,N-2} & \tau_{1,N-1} \\
0 & 0 & \phi_{2,3} & \ldots & \tau_{2,N-3} & \tau_{2,N-2} & \tau_{2,N-1} \\
0 & 0 & 0 & \ldots & \tau_{3,N-3} & \tau_{3,N-2} & \tau_{3,N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 0 & \phi_{N-3,N-2} & \chi_{N-3,N-1} \\
0 & 0 & 0 & \ldots & 0 & 0 & \phi_{N-2,N-1} \\
0 & 0 & 0 & \ldots & 0 & 0 & 0
\end{bmatrix}. \quad (10)
$$

The general iterative scheme in matrix form can be written as

$$
(D - L)u^{(k+1)} = (D + T)u^{(k)} + b. \quad (11)
$$

As stated in the previous section, GM is a two-stage iterative method and its iterative process involves of solving two independent systems such as $\hat{u}^{(1)}$ and $\hat{u}^{(2)}$ which are calculated at Level 1 and 2, respectively. The square root of the product of the two systems is computed at Level 3. Hence, the general iterative scheme for GM method can be defined as [15]

$$
\begin{align*}
(D - \alpha L)\hat{u}^{(1)} &= ((1 - \alpha)D + \alpha T)u^{(k)} + \alpha b, \\
(D - \alpha T)\hat{u}^{(2)} &= ((1 - \alpha)D + \alpha L)u^{(k)} + \alpha b,
\end{align*}
$$

$$
\begin{align*}
u^{(k+1)} &= \sqrt{\hat{u}^{(1)}\hat{u}^{(2)}}.
\end{align*}
$$

(12)
The identical conditions which guarantee the convergence of (12) is described in [6]. The weighted parameter is in the range $1 \leq \alpha < 2$, where the optimal value of $\alpha$ can be obtained by conducting several experiments until it gives acceptable minimum number of iterations. The portion of computational grid for calculation at Level 1 and 2 of the standard and skewed cases are depicted in Figures 3 and 4, respectively. Thus, based on Equation (12), the implementations of GM and SkGM methods are described in Algorithms 1 and 2, respectively.

**Algorithm 1: GM algorithm**

**Result:** Updated matrix $V$

Initialize original matrix $U$, $f$; matrix $A$; matrix $B$; $h$, $\epsilon$, $\alpha$;

while error $> \epsilon$ do

// Level 1
for $i, j = 0,1,2,\ldots,n$ do

$A[i,j] = \frac{\alpha}{4}(A[i-1,j] + U[i+1,j] + A[i,j-1] + U[i,j+1] - h^2f[i,j]) + (1-\alpha)U[i,j];$

end

// Level 2
for $i, j = n,n-1,n-2,\ldots,0$ do

$B[i,j] = \frac{\alpha}{4}(U[i-1,j] + B[i+1,j] + U[i,j-1] + B[i,j+1] - h^2f[i,j]) + (1-\alpha)U[i,j];$

end

// Level 3
for $i, j = 0,1,2,\ldots,n$ do

$V[i,j] = \sqrt{A[i,j].B[i,j]}$

end
end
Algorithm 2: SkGM algorithm

Result: Updated matrix $V$

Initialize original matrix $U, f$; matrix $A$; matrix $B$; $h, \epsilon, \alpha$;

while error $> \epsilon$ do

for $i,j = 0,1,2,...,n$ do // Level 1

if $i + j$ is even then

$$A[i,j] = \frac{1}{4}(A[i-1,j-1] + A[i+1,j-1] + U[i-1,j+1] + U[i+1,j+1] - 2h^2 f[i,j]) + (1 - \alpha)U[i,j];$$

end

end

for $i,j = n,n-1,n-2,...,0$ do // Level 2

if $i + j$ is even then

$$B[i,j] = \frac{1}{4}(U[i-1,j-1] + U[i+1,j-1] + B[i-1,j+1] + B[i+1,j+1] - 2h^2 f[i,j]) + (1 - \alpha)U[i,j];$$

end

end

for $i,j = 0,1,2,...,n$ do // Level 3

if $i + j$ is even then

$$V[i,j] = \sqrt{A[i,j].B[i,j]};$$

end

end

// Compute the remaining WHITE nodes

for $i,j = 0,1,2,...,n$ do

if $i + j$ is odd then

$$V[i,j] = \frac{1}{4}(U[i-1,j] + U[i+1,j] + U[i,j-1] + U[i,j+1] - h^2 f[i,j]);$$

end

end
3. The Modified Geometric Mean (MGM) method

The formulation of MGM method is very much inspired by the work of Kincaid and Young [7]. With standard grid iteration, the MGM employs red-black strategy where the red and black cells are computed alternately, see Figure 5 (a). In the case of skewed grid iteration, only red and black cells are computed alternately during the iteration, whereas the white cells are computed only once after the convergence criterion is satisfied, see Figure 5 (b). Note that the computation of red cells rely on the updated values of black cells and vice versa, as depicted in Figures 6 and 7 for the standard and skewed grids, respectively.
In MGM, two acceleration parameters $\alpha$ and $\beta$ are employed. These two parameters, $\alpha$ and $\beta$, are employed for red and black nodes, respectively. The general formulation of MGM method is written as follows

\[
\begin{align*}
(D - \alpha L)\hat{u}^{(1)} &= ((1 - \alpha)D + \alpha T)u^{(k)} + \alpha b, \\
(D - \beta T)\hat{u}^{(2)} &= ((1 - \beta)D + \beta L)u^{(k)} + \beta b, \\
u^{(k+1)} &= \sqrt{\hat{u}^{(1)}.\hat{u}^{(2)}},
\end{align*}
\]

where both parameters are in the range $1 \leq \alpha < 2$ and $1 \leq \beta < 2$. Note that if $\alpha = \beta$, the MGM method simplifies to the original GM method.

The MGM method involves two levels of independent computations; Level 1 and Level 2. At Level 1, the red and black nodes are calculated alternately, where the updated values of red nodes are used to calculate black nodes and vice versa. The same procedure is implemented in the calculation at Level 2. In Level 3, the square root of multiplication of intermediate values ($\hat{u}^{(1)}$ and $\hat{u}^{(2)}$) obtained from Level 1 and Level 2 is computed to obtain the updated value of $u^{(k+1)}$.

By employing the red-black ordering, the implementations of MGM and SkMGM methods are described in Algorithms 3 and 4.
Figure 5: The computational mesh of size $N = 10$ with red-black ordering using (a) standard and (b) skewed grids.
Figure 6: The computational molecules in red-black ordering on standard grid.

Figure 7: The computational molecules in red-black ordering on skewed grid.
Algorithm 3: MGM algorithm

Result: Updated matrix $V$

Initialize original matrix $U, f$; matrix $A$; matrix $B$; $h, \epsilon, \alpha$;

\begin{algorithm}
\begin{algorithmic}
\State \textbf{while} error $> \epsilon$ \textbf{do}
\State \Comment{Level 1}
\For{all RED nodes}
\If{$i + j$ is even}
\State $A[i, j] = \text{calcLevelOne}(i, j, A, U, h, f, \alpha)$;
\EndIf
\EndFor
\For{all BLACK nodes}
\If{$i + j$ is odd}
\State $A[i, j] = \text{calcLevelOne}(i, j, A, U, h, f, \beta)$;
\EndIf
\EndFor
\Comment{Level 2}
\For{all RED nodes}
\If{$i + j$ is even}
\State $B[i, j] = \text{calcLevelTwo}(i, j, B, U, h, f, \alpha)$;
\EndIf
\EndFor
\For{all BLACK nodes}
\If{$i + j$ is odd}
\State $B[i, j] = \text{calcLevelTwo}(i, j, B, U, h, f, \beta)$;
\EndIf
\EndFor
\Comment{Level 3}
\For{all RED and BLACK nodes}
\State $V[i, j] = \sqrt{A[i, j] \cdot B[i, j]}$;
\EndFor
\EndWhile
\end{algorithmic}
\end{algorithm}
Algorithm 4: SkMGM algorithm

Result: Updated matrix $V$

Initialize original matrix $U, f$; matrices $A, B; h, \epsilon, \alpha, \beta$;

while error $> \epsilon$ do

for all RED nodes do // Level 1

if $i$ and $j$ are odd then

$A[i, j] = \text{calcLevelOne}(i, j, A, U, h, f, \alpha)$;

end

end

for all BLACK nodes do // Level 1

if $i$ and $j$ are even then

$A[i, j] = \text{calcLevelOne}(i, j, A, U, h, f, \beta)$;

end

end

for all RED nodes do // Level 2

if $i$ and $j$ are odd then

$B[i, j] = \text{calcLevelTwo}(i, j, B, U, h, f, \alpha)$;

end

end

for all BLACK nodes do // Level 2

if $i$ and $j$ are even then

$B[i, j] = \text{calcLevelTwo}(i, j, B, U, h, f, \beta)$;

end

end

for all RED and BLACK nodes do // Level 3

if $i + j$ is even then

$V[i, j] = \sqrt{A[i, j] \cdot B[i, j]}$;

end

end

for all WHITE nodes do

if $i + j$ is odd then

$V[i, j] = \text{calcWhiteNodes}(i, j, V, h, f)$;

end

end

end
4. Numerical results

In this section we present some numerical examples in order to compare the considered iterative methods which were described in the previous sections. Several numerical experiment were carried out on the following Poisson problem,

\[ \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \pi^2 (x^2 + y^2) \sin(\pi xy), \]  

(14)

\[ u(x, 0) = 0.1, \quad u(0, y) = 0.1, \]
\[ u(x, 1) = 1.0, \quad u(1, y) = 1.0, \]

defined in the unit square. The exact solution of this problem is \( u(x, y) = \sin(\pi xy) \). This problem was solved using the standard grid methods and their corresponding skewed grid variants; the standard GS and SkGS methods, and the GM variants of GM and SkGM methods; the modified MGM and SkMGM methods. The tolerance error rate for the convergence test is set to \( \epsilon = 1.0 \times 10^{-7} \). For experimental consistency, the chosen optimal values of \( \alpha \) and \( \beta \) are 1.80 and 1.85, respectively.

The numerical results for iterative methods on the standard and skewed grids with different values of \( N \) are shown in Tables 1 and 2, respectively. Figures 8 and 9 illustrate the graph performance of the tested methods in terms of number of iterations and cpu time, respectively.

5. Computational complexity analysis

For all variants of iterative methods on standard grid, all nodes are considered during the iteration process, as depicted in Figure 1 (a). While for the iterative methods on skewed grid, the iterations to approximate the solution compute black nodes only, therefore only half of the total nodes are considered. This skewed grid concept was introduced by Abdullah [1] via the explicit decoupled group iterative method to solve 2D Poisson equations. The method was shown to be very much faster compared to the standard grid version but obtained comparable accuracy without significant differences in the final solutions.

As depicted in Figure 1 (b), for computational grid of size \( N = 10 \), the iterative methods on skewed grid consider black nodes only. For the skewed grid variants of GM method, the computation at Level 1 starts from the first node \((1, 1)\) until the last node \((n - 1, n - 1)\). While, the computation at Level 2 is in reverse order starts from node \((n - 1, n - 1)\) down to node \((1, 1)\). In
Table 1: Number of iterations and cpu time for the standard grid

| N   | GS     | GM     | MGM    |
|-----|--------|--------|--------|
|     | k      | t      | k      | t      | k      | t      |
| 32  | 907    | 0.0124 | 67     | 0.0016 | 50     | 0.0014 |
| 64  | 3177   | 0.1299 | 301    | 0.0292 | 260    | 0.0230 |
| 96  | 6472   | 0.6244 | 646    | 0.1332 | 564    | 0.1144 |
| 128 | 10618  | 1.8139 | 1093   | 0.3999 | 957    | 0.3450 |
| 160 | 15492  | 4.1475 | 1633   | 0.9268 | 1433   | 0.8174 |
| 192 | 21000  | 8.1198 | 2261   | 1.9008 | 1987   | 1.6180 |
| 224 | 27065  | 14.0701| 2969   | 3.3856 | 2612   | 2.9239 |
| 256 | 33623  | 23.3200| 3754   | 5.5295 | 3306   | 4.8572 |
| 288 | 40618  | 35.8640| 4611   | 8.3708 | 4063   | 7.5404 |
| 320 | 48001  | 51.3474| 5537   | 12.4116| 4883   | 11.9815|
| 352 | 55728  | 73.7239| 6529   | 18.7565| 5760   | 16.2016|
| 384 | 63758  | 98.7446| 7583   | 25.4465| 6694   | 22.6850|
| 416 | 72057  | 131.3896| 8696 | 34.5372 | 7681 | 30.5425 |
| 448 | 80590  | 170.8649| 9868 | 44.0957 | 8720 | 39.8261 |
| 480 | 89327  | 219.3697| 11094 | 58.1631 | 9809 | 51.5436 |
| 512 | 98240  | 275.9132| 12373 | 75.7737 | 10945 | 66.1663 |

general, the computational complexities of the skewed grid variants of the considered iterative methods are reduced approximately by 50% compared to their corresponding standard grid variants.

The iteration process continues until the convergence criterion is satisfied, i.e. \( ||u^{(k+1)} - u^{(k)}|| < \epsilon \). For all implementations of iterative methods on skewed grid, after the convergence is achieved, the remaining white nodes are computed once using the standard five-point difference formula. Note that the skewed five-point formula cannot be used in this final iteration, since only black nodes are computed during the iteration process. Hence, the white nodes in skewed grid rely on their neighbouring black nodes to obtain their own final values which requires the standard five-point formula for the calculation, see Figure 2 (c).

The modified variants, MGM and SkMGM methods, employ red-black strategy. Since red and black nodes are calculated independently, the approach is very suitable for parallel implementation. The procedure for the these modified variants begins at Level 1 with weighted parameter, \( \alpha \), by calculating all red nodes first. The updated red values are then used to calculate all black nodes. Level 2 utilizes weighted parameter, \( \beta \), in which the same procedure is
executed, where red nodes are calculated followed by the calculation of black nodes. At Level 3, all red and black nodes are calculated by using the updated values obtained during the iteration procedures at Level 1 and Level 2. Once the convergence is achieved, the remaining white nodes are calculated using direct method (5).

From Tables 1 and 2, it can be seen that the number of iterations of all methods on skewed grid are approximately 50% less than their corresponding standard grid versions. Consequently, the cpu time is drastically reduced approximately by 10% to 60%, where the improvement is more significant for larger dimensions. Both standard GS methods are extremely slow compared to the weighted GM methods and their modified MGM variants. The modified MGM variants are slightly faster than their corresponding standard GM methods, where both iteration counts and cpu time are reduced approximately by 13% to 15%. It can also be noted that for small grid, \( N \), the difference is not significant.

| \( N \) | \( N \) | \( N \) |
|-------|-------|-------|
| \( k \) | \( t \) | \( k \) | \( t \) | \( k \) | \( t \) |
| 32    | 479   | 0.0038 | 42    | 0.0009 | 47    | 0.0012 |
| 64    | 1693  | 0.0535 | 151   | 0.0118 | 127   | 0.0086 |
| 96    | 3472  | 0.2534 | 335   | 0.0503 | 290   | 0.0466 |
| 128   | 5731  | 0.7480 | 572   | 0.1545 | 499   | 0.1351 |
| 160   | 8406  | 1.7041 | 861   | 0.3627 | 753   | 0.3184 |
| 192   | 11452 | 3.5284 | 1196  | 0.7290 | 1049  | 0.6385 |
| 224   | 14830 | 5.9118 | 1577  | 1.3026 | 1384  | 1.1478 |
| 256   | 18507 | 9.0728 | 1999  | 2.1689 | 1756  | 1.8951 |
| 288   | 22457 | 14.9698| 2461  | 3.3775 | 2164  | 2.9795 |
| 320   | 26654 | 21.8605| 2962  | 5.0620 | 2606  | 4.4324 |
| 352   | 31076 | 31.0942| 3499  | 7.2754 | 3081  | 6.3783 |
| 384   | 35703 | 43.3401| 4071  | 10.1207| 3587  | 8.8834 |
| 416   | 40518 | 57.7020| 4678  | 13.6735| 4124  | 12.0552|
6. Conclusion

From the numerical results as shown in Table 1, it can be seen that the GM clearly outperformed the standard GS method. The MGM performed slightly better than the GM method. Table 2 shows the performance of the skewed variants of the iterative methods. It is clearly shown that the application of iterative method on skewed grid greatly reduced the computational complexity approximately by 50%.

All iterative methods on skewed grid are superior to their corresponding variants that are implemented on standard grid, since only half of the total nodes of the computational grid are considered during the iteration process. The reduction percentage of the execution time for iterative methods on skewed grid over their corresponding standard grid variants is in the range between 10% and 60%. The proposed SkMGM method gives the best performance, in which
the cpu time is reduced approximately by 13% to 15% compared to the existing SkGM method. It can also be seen that the modified variants of MGM and SkMGM methods are more flexible since by employing two acceleration parameters, $\alpha$ and $\beta$, both methods allow wider choice of parameter tuning values. With red-black ordering, the red and black nodes are computed independently, therefore the modified variant are very suitable for parallel processing. In conclusion, the proposed method is capable of demonstrating its efficiency in terms of number of iterations and cpu time compared with the existing iterative methods. In the future work, the potential of the proposed method in the applications to solve problems in robot path planning, image processing, heat transfer, fluid computing and other problems would be examined.
Acknowledgment

This work is supported financially by the Universiti Malaysia Sabah under Grant No. SGA0016-2019.

References

[1] A.R. Abdullah, The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver, *Intern. J. of Computer Mathematics*, **38**, No 1-2 (1991), 61-70; doi: 10.1080/00207169108803958.

[2] N.H.M. Ali, A. Ali, On skewed grid point iterative method for solving 2D hyperbolic telegraph fractional differential equation, *Advances in Difference Equations*, **2019** (2019), Arthttps://doi.org/. 303; doi: 10.1186/s13662-019-2238-6.

[3] N.H.M. Ali, L.S. Chong, Group accelerated OverRelaxation methods on rotated grid, *Applied Mathematics and Computation*, **191**, No 2 (2007), 533-542; doi: 10.1016/j.amc.2007.02.131.

[4] T. Allahviranloo, E. Ahmady, N. Ahmady, K.S. Alketaby, Block Jacobi two-stage method with Gauss-Seidel inner iterations for fuzzy system of linear equations, *Applied Mathematics and Computation*, **175**, (2006) 1217-1228; doi: 10.1016/j.amc.2005.08.047.

[5] D.J. Evans, M.S. Sahimi, The alternating group explicit (age) iterative method for solving parabolic equations i: 2-dimensional problems, *Intern. J. of Computer Mathematics*, **24**, No 3-4 (1988), 311-341; doi: 10.1080/00207168808803651.

[6] I. Galligani, V. Ruggiero, The arithmetic mean method for solving essentially positive systems on a vector computer, *Intern. J. of Computer Mathematics*, **32**, No 1-2 (1990), 113-121; doi: 10.1080/00207169008803819.

[7] D.R. Kincaid, D.M. Young, The modified successive overrelaxation method with fixed parameters, *Mathematics of Computation*, **26**, No 119 (1972), 705-717; doi: 10.1090/S0025-5718-1972-0331746-2.

[8] 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 J. of Mathematics, 12, No 04, 1950061 (2019); doi: 10.1142/S179355711950061X.

[9] M.S. Muthuvalu, J. Sulaiman, Numerical performance of Half-Sweep Geometric Mean (HSGM) iterative method for solving third order Newton–Cotes quadrature system, Lobachevskii J. of Mathematics, 38, No 1 (2017), 73-81; doi: 10.1134/S1995080217010152.

[10] R. Rahman, N.F.M Ali, J. Sulaiman, F.A. Muhiddin, Application of the half-sweep egsoh iteration for two-point boundary value problems of fractional order, Advances in Science, Techn. and Engin. Systems J., 4, No 2 (2019), 237-243; doi: 10.25046/aj040231.

[11] M.S. Sahimi, A. Ahmad, A.A. Bakar, The Iterative Alternating Decomposition Explicit (IADE) method to solve the heat conduction equation, Intern. J. of Computer Mathematics, 47, No 3-4 (1993), 219-229; doi: 10.1080/00207169308804179.

[12] A. Saudi, J. Sulaiman, Red-Black strategy for mobile robot path planning, In: Proc. of the Intern. MultiConference of Engineers and Computer Scientists, Hong Kong (2010), Vol III, 2215-2220.

[13] J. Sulaiman, M.K. Hasan, M. Othman, The Half-Sweep Iterative Alternating Decomposition Explicit (HSIADE) method for diffusion equation, In: Proc. of the First Intern. Conf. on Computational and Information Science, Shanghai (2004), 57-63.

[14] J. Sulaiman, M. Othman, M.K. Hasan, Quarter-sweep Iterative Alternating Decomposition Explicit algorithm applied to diffusion equations, Intern. J. of Computer Mathematics, 81, No 12 (2004), 1559-1565; doi: 10.1080/00207160412331291125.

[15] J. Sulaiman, M. Othman, M.K. Hasan, Half-Sweep Geometric Mean (HSGM) method using fourth-order finite difference scheme for two-point boundary problems, In: Proc. of the First Intern. Conf. on Mathematics and Statistics, Bandung (2006), 25-33.

[16] A. Sunarto, J. Sulaiaman, A. Saudi, Caputo’s implicit solution of time-fractional diffusion equation using half-sweep AOR iteration, Global J. of Pure and Appl. Math., 12, No 4 (2016), 3469-3479.
[17] D.M. Young, Iterative methods for solving partial difference equations of elliptic type, *Trans. Amer. Math. Soc.*, 76, No 1 (1954), 92-111; doi: 10.1090/S0002-9947-1954-0059635-7.
