Numerical Solution of Helmholtz Equation Using a New Four Point EGMSOR Iterative Method

Mohd Kamalrulzaman Md Akhir
Institute of Mathematical Research
Universiti Putra Malaysia
43400, UPM Serdang Selangor, Malaysia
mkamalrulzaman@gmail.com

Mohamed Othman
Departments of Communication Technology and Network
Faculty of Computer Science and Info Technology
Universiti Putra Malaysia
43400, UPM Serdang Selangor D.E, Malaysia

Jumat Sulaiman
Schools of Science and Technology
Universiti Malaysia Sabah
Locked Bag 2073, 88999 Kota Kinabalu Sabah, Malaysia

Zanariah Abdul Majid and Mohamed Suleiman
Departments of Mathematics
Faculty of Science
Universiti Putra Malaysia
43400, UPM Serdang Selangor, Malaysia

Abstract
Recently, a family of block iterative method via Explicit Group (EG) iterative methods is shown to be one of the feasible and successful classes of iterative methods in solving any system of linear algebraic equations. The aim of this article is to examine the effectiveness of the new four Point-EGMSOR iterative methods in solving two-dimensional Helmholtz equations. The concept of a four Point-EGMSOR is inspired via combination between four Point-EG iterative method together with Modified Successive OverRelaxation (MSOR) approach,
namely four Point-EGMSOR. In addition, the formulation and implementation of
the proposed method are also presented. Some numerical experiments have been
carried out to show the effectiveness of the proposed method compared to the
standard method.

**Keywords**: Explicit Group, Finite Difference Method, Helmholtz Equation;
Modified Successive OverRelaxation (MSOR) method

1 Introduction

The theory and applications of partial differential equations (PDEs) have been one
of the main principal tools in various area of science and engineering such as
elastic waves in solids, time harmonic acoustic, electromagnetic fields and optical
waveguide [20,21]. Yet, it is still very difficult to gain the solution for these
problems either analytically or numerically. Many researchers have proposed
numerous methods such as numerical analytic, finite difference, finite element,
finite volume and boundary element methods to find the analytic or approximate
solutions for the proposed problems. Next, these approximation equations will be
used to generate the corresponding systems of linear algebraic equations.
However, due to the large scale of linear algebraic equations systems, finite
difference method needs a longer time to simulate these approximate equations.

Consequently, there have been various discussion and work to speed up the
convergence rate in solving any system of linear algebraic equations; see Young
[7,8,9] and Saad [22]. In addition, Evans [3] has also introduced the four point
block iterative method via EG iterative method for solving large system of linear
algebraic equations. Throughout this method, further studies of the block iterative
have been extensively conducted by Abdullah [1], Ibrahim and Abdullah [2], and
Evans and Yousif [4,5] for demonstrating the capability of the block iterative
method. In this article, we will examine the advantages of the four Point-EG
iterative method together with MSOR approach and later called as four
Point-EGMSOR iterative method to solve two-dimensional Helmholtz problem.
To show the effectiveness of four Point-EGMSOR method, let us consider the
following two-dimensional Helmholtz equation which is defined as

\[
\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - \alpha = ( , )\, ( , ) = \frac{f(x,y)}{\partial x} \quad \text{subject to the dirichlet boundary condition and satisfy the exact solution}
\]

where \(( , )\) is given function with suf-\(ficient smoothness and \(\alpha \) is the nonnegative constant. Before implementing a
formulation of the finite difference approximation equation, let us consider that
the solution domain, in Figure 1 needs to be partitioned uniformly in both and
directions with fixed mesh size \(h = \frac{1}{n}\), where \(n\) is arbitrary positive integer.
In order to obtain finite grid networks for approximate finite approximate equation over Eq. (1), let us refer to Figure 1. Figure 1 acts as a guide for the implementation of the proposed computational algorithms. For that reason, the implementation of the block iterative algorithms will be applied onto the node points of the same type until the iterative convergence fixed is achieved.

![Uniform nodal points distribution](image)

**Figure 1**: Distribution of uniform nodal points at m=8.

The outline of this article is organized as follows. In Section 2, the formulation of the finite difference approximation equations will be elaborated. The latter section of this article will discuss the formulations of the MSOR iterative method. Next in Section 4, we start derivation and implementation of the four-point-EGMSOR method to solve problem (1). Besides that, some numerical results are given in Section 5 and the concluding remarks are given in final section.

### 2 Second Order Finite Difference Approximation

Since four-point block iterative methods such as the four Point-EGGS, four Point-EGSOR, and four Point-EGMSOR methods are categorized as linear solvers, firstly Eq. (1) needs to be approximated by approximation equations. To make things simple, our next discussion will be restricted to construct second order finite difference approximation equations for the formulation of four Point-EGGS, four Point-EGSOR, and four Point-EGMSOR iterative methods. These method will be derived to be used in solving corresponding linear systems algebraic generated from second-order finite difference approximation equations. The finite grid networks as shown in Figure 1 are used as a guideline on how to implement these block iterative methods.

Eventually, some researchers have suggested high order scheme (Gupta et. al [17]) to derive high-order finite difference approximation equation. However both
lower and higher order schemes will result in their own system linear algebraic system with different properties of their coefficient matrix. In this paper, however, discretization schemes based on the second-order finite difference schemes will be considered to derive standard five-point approximation equations for Eq.(1). By using second-order central difference schemes, the standard five-point approximation for Eq.(1) can be expressed in the following formulae

\[ u_{i,j}^{n+1} = \left( 4 + \frac{\alpha}{h^2} \right) u_{i,j}^{n} - \left( \frac{\alpha}{h^2} \right) \left[ u_{i+1,j}^{n} + u_{i-1,j}^{n} \right], \quad \alpha = 4 \]

By considering interior node points \((i, j)\) of type as shown in Fig. 1, formulation and implementation of four Point-EGGS, four Point-EGSOR and four Point-EGMSOR iterative methods will be applied onto these interior node points until the criterion of convergence test is agreed.

3 The MSOR Iterative Method

Consider the following linear system into a general matrix form stated as

\[ Au = b \]

The methods for solving Eq. (1) can be classified into two categories which are direct and iterative method. Gauss elimination and LU factorization are some of the examples of the direct methods to solve system of linear algebraic equations. Meanwhile, in this article we are focusing on iterative linear system solvers. According to Young [8], the usage of the iterative methods has the advantage since the matrix is not distorted during the computation and the problem of the accumulation of rounding errors is less serious than direct methods. Let coefficient matrix in Eq. (1) decomposed in the form of

\[ A = D - L - U \]

where, and are the diagonals, negative lower triangulation and negative upper triangulation matrices, respectively. Thus, for real positive accelerated parameter, the general schemes for MSOR iterative method can be written as:

\[
\begin{align*}
(\theta+1)^{-1} &= (\theta) + \theta \\
(\theta+1)^{-1} &= (\theta) + \theta
\end{align*}
\]

Here, the relaxation parameter is for the “red” equations and the “black” equations. The idea of the MSOR method was introduced by De Vogelaere [11] is one of the most efficient method in solving system of linear algebraic equations. According to Kincaid and Young [10], MSOR method is best applied in solving large system
of linear equations when the two matrixes is two-cyclic are consistently ordered of equation (excluding red and black equation).

Thus, the performance of the MSOR methods often improved a proper choice of the accelerated parameters drastically. For example, this fact was discussed by Young [8] and Akhir, [13,14,15,16] who had extensively discussed the application of MSOR method for solving large system of linear algebraic equations. Again, this paper presents a new fast and reliable explicit group iterative method to solve two-dimensional Helmholtz equation. Several numerical experiments have been carried out and their results are presented to demonstrate the efficiency of the proposed method.

4 Formulation of Four-Point block Iterative Method

4.1 Four Point-EG Iterative Method

For reason of formulation four Point-EG iterative method, let consider a complete group of four points (4x4). By considering Eq. (1), this method can be generally expressed as

\[
\begin{bmatrix}
4 + 2\alpha & -1 & 0 & -1 \\
-1 & 4 + 2\alpha & -1 & 0 \\
0 & -1 & 4 + 2\alpha & -1 \\
-1 & 0 & -1 & 4 + 2\alpha \\
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\alpha +1 \\
\alpha +1 \\
\alpha +1 \\
\end{bmatrix}
= \begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
\end{bmatrix}
\]

(7)

where,

\[
1 = -1, + , -1 - 2 , \\
2 = +2, + , +1, -1 - 2 , +1, \\
3 = -1, +1 + , +2 - 2 , +1, \\
4 = +2, +1 + , +1, +2 - 2 , +1, +1, \\
\]

Now by determining the inverse matrix of Eq. (7), the four point-EG method can be generally shown as

\[
\begin{bmatrix}
\alpha \\
\alpha +1 \\
\alpha +1 \\
\alpha +1 \\
\end{bmatrix}
\begin{bmatrix}
\frac{1}{\psi} \\
\frac{1}{\psi} \\
\frac{1}{\psi} \\
\frac{1}{\psi} \\
\end{bmatrix}
= \begin{bmatrix}
1 \\
2 \\
3 \\
4 \\
\end{bmatrix}
\]

(8)

where,
\[
\psi = (2 + 2\alpha)(4 + 2\alpha)^2(6 + 2\alpha),
\]
\[
\gamma = 1 + \frac{1}{4}, \quad \omega = 2 + \frac{1}{3},
\]
\[
\delta = (1 + 2\alpha), \quad \epsilon = (2 + \frac{1}{2}).
\]

### 4.2 Four Point-EG SOR Iterative Method

Let assume that the solution at any group of four points, (4x4). From Eq. (1), this method can be expressed in the following system of linear algebraic equations

\[
\begin{bmatrix}
4 + \frac{2}{\alpha} & -1 & 0 & -1 \\
-1 & 4 + \frac{2}{\alpha} & -1 & 0 \\
0 & -1 & 4 + \frac{2}{\alpha} & -1 \\
-1 & 0 & -1 & 4 + \frac{2}{\alpha}
\end{bmatrix}
\begin{bmatrix}
\gamma \\
\omega \\
\delta \\
\epsilon
\end{bmatrix}
= \begin{bmatrix}
1 \\
\omega + \epsilon \\
\delta + \epsilon \\
\gamma + \epsilon
\end{bmatrix}.
\]

(9)

where,

\[
\gamma = -1, \quad \omega = 1 - 2, \quad \delta = 2, \quad \epsilon = 4.
\]

Similarly, the above Eq. (9) can be inverted to result the four point-EGSQR iterative method shown as

\[
\begin{bmatrix}
\gamma \\
\omega \\
\delta \\
\epsilon
\end{bmatrix}
= \frac{1}{\psi}
\begin{bmatrix}
1 + \gamma \\
2 + \omega \\
3 + \delta \\
4 + \epsilon
\end{bmatrix},
\]

(10)

Hence, by adding one accelerated parameter into Eq. (13), the four point-EGSQR iterative method can be rewritten as

\[
\begin{bmatrix}
\gamma \\
\omega \\
\delta \\
\epsilon
\end{bmatrix}
= (1-\theta)
\begin{bmatrix}
\gamma \\
\omega \\
\delta \\
\epsilon
\end{bmatrix}
+ \frac{\theta}{\psi}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4
\end{bmatrix}.
\]

(11)

where,
Numerical solution of Helmholtz equation

\[ \psi = \left(2 + \alpha^2\right) \left(4 + \alpha^2\right)^2 \left(6 + \alpha^2\right), \]

where,

\[ 1 = 1 + \frac{2}{4}, \quad 2 = 2 + \frac{3}{4}, \]

\[ = \left(1 + 2\right), \quad = \left(2 + 2\right). \]

4.3 Four Point-EG MSOR Iterative Method

Let a four point be considered to form (4x4) systems of linear algebraic equations as

\[
\begin{bmatrix}
4 + \alpha^2 & -1 & 0 & -1 \\
-1 & 4 + \alpha^2 & -1 & 0 \\
0 & -1 & 4 + \alpha^2 & -1 \\
-1 & 0 & -1 & 4 + \alpha^2
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix}
= \begin{bmatrix}
1 \\
2 \\
3 \\
4
\end{bmatrix},
\]

where,

\[ 1 = -i, + \cdot, -i - \frac{2}{2}, \quad 2 = +2, + \cdot, -i - \frac{2}{2}, \]

\[ 3 = -1, +1 + \cdot, +2 - \frac{2}{2}, \quad 4 = +2, +1 + \cdot, +1 - \frac{2}{2}, \]

Again this system in Eq. (12) can be rewritten by multiplying the inverse of the coefficients matrix, the four point-EGMSOR method can be simplified as

\[
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix}
= \frac{1}{\psi}
\begin{bmatrix}
1 + \\
2 + \\
3 + \\
4 +
\end{bmatrix},
\]

(13)

By adding two accelerated parameters, the four point-EGMSOR method shown as

\[
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
= \left(1 - \theta\right)
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
+ \frac{\theta \psi}{\psi}
\begin{bmatrix}
1 + \\
2 + \\
3 + \\
4 +
\end{bmatrix},
\]

(14)

\[
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
= \left(1 - \theta\right)
\begin{bmatrix}
\cdot \\
+1, \\
+1, +1
\end{bmatrix}
+ \frac{\theta \psi}{\psi}
\begin{bmatrix}
1 + \\
2 + \\
3 + \\
4 +
\end{bmatrix},
\]

(15)
where,
\[
\psi = \left(2 + 2 \alpha \right) \left(4 + 2 \alpha \right)^2 \left(6 + 2 \alpha \right),
\]
\[
1 = 1 + 4, \quad 2 = 2 + 3,
\]
\[
= \left(1 + 2 \right), \quad = \left(2 + 2 \right).
\]

These two accelerated parameters can be calculated practically by consecutively choosing a value with some precision until the optimal value is obtained. For the choice \( \theta = \theta = 1 \), MSOR method will reduce the Gauss-Seidel (GS). Take note that when and \( \theta = \theta \) will result the MSOR method coincides with the original Successive Over-Relaxation (SOR) method with red black ordering.

For a general system matrix, the MSOR method converges for both accelerated parameters such as \( 0 < \theta < 2 \) and \( 0 < \theta < 2 \), Young [8, 10]. In this paper, the approximate solution of the system of linear equations is restricted to four Point block schemes based on GS, SOR and MSOR technique. By determining the value of matrices, and as stated in (5), the general algorithm four point-EGMSOR method would be generally described in the following algorithm

Algorithm 1:
In this method, the \( \Omega \) is divided into one types of points (i.e. ●) as shown in Figure 2. The solutions on any group of points ● can only be implemented by only involving the same type of point ●. The four point-EGMSOR algorithm may be describes as follows

1. Divide the solution domain into two types as in Figure 2. Compute the values of \( ^2 \).
2. Iterate the intermediate solution \( u \) of point type ● using Eq. (3)
   \[\begin{align*}
   2.1 \quad & \quad -l, +l, +l, +l - \left(4 + \alpha^2 \right), = 2^2, \, , \\
   \end{align*}\]
3. Implement the relaxation parameter for \( \theta \) and \( \theta \):
   \[\begin{align*}
   3. \quad & \quad \left( +1 \right) = \left( - \theta \right)^{-1} \left[ \left( \theta \right) \left( - (1 - \theta) \right) \right] + \theta, \\
   & \quad \left( +1 \right) = \left( - \theta \right)^{-1} \left[ \left( \theta \right) \left( - (1 - \theta) \right) \right] + \theta, \\
   \end{align*}\]
4. Check the convergence. If converge evaluate the rest of points (i.e., ○) using,
   \[\begin{align*}
   & \quad \left( +1 \right) + l, + l, - l, - l, + l, - l, + l, - l, = 2^2, \\
   \end{align*}\]
   respectively. Otherwise repeat the iteration cycle (i.e., go to step 2)
4. Stop
Figure 2: Implementation of the four point-EGMSOR method at solution domain m=16.

5 Numerical Experiments

In order to confirm four point-EGMSOR method is superior than other three four point block methods, the following experiments are carried out on Intel(R) Core(TM) i7 CPU 860@2.80Ghz with memory is 4.00GB. In comparison, the Full-Sweep point Gauss-Seidel (FSGS) acts as the control of comparison of numerical result. Three criterion will be considered for methods such as the number of iterations, execution time and its maximum and absolute error in comparison. Throughout the simulations, the convergence test is considered the tolerance error $\varepsilon = 10^{-10}$ and carried out on several different value $n$.

Example 1 (Evans, [3])

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - 10 = 6 - 10(2^2 + 2^2), \ (x, y) \in [0,1] \times [0,1].$$

Exact solution

$$(x, y) = 2^2 + 2^2.$$
Example 2 (Evans et al., [6])

\[
\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - 2 = (12^2 + 3^4)\sin (x, y) \in [0,1] \times [0,1].
\]

Exact solution

\[(x, y) = 4\sin (x, y)\]

6 Conclusions

In the previous section, we have presented the formulation of four point-EGGS, four point-EGSOR and four point-EGMSOR methods based on the second order finite difference method that can easily generate a system of linear equations as shown in Eq.(3). In Tables 1 and 3, it is clearly shown that the application of four point-EGMSOR methods can reduce number of iterations and execution time compared to four point-EGSOR and four point EGGS methods. Table 2 and 4 shows percentages of number decrement of iterations for four point EGGS, four point-EGSOR and four point-EGMSOR methods. Through numerical results in Tables 1 and 3, we found that the application of four point-EGMSOR method approach has reduced the execution time of the iterative method. In fact, decrement percentages of the execution time for four point-EGMSOR, four point-EGSOR and four point-EGGS methods as compared to FSGS method are summarized in Tables 2 and 4. Meanwhile, the accuracy for approximate solutions for all methods are in good agreement as compared to the FSGS.

Overall, the numerical experiments prove that four point-EGMSOR iterative method is very significant among the four point-EGGS, four point-EGSOR and FSGS methods in a sense of iterations and execution time. This is due to both accelerated parameter to speed up the convergence rate in solving a large system of linear algebraic equations; refer Tables 1 and 3. In future, our study will extend in investigating the effectiveness of the half-sweep approach (Abdullah [1]; Ibrahim and Abdullah [2]; Othman and Abdullah [18,19]; Sulaiman [12]; Akhir [16]) to solve PDE mainly on multidimensional partial differential.
**Table 1**: Comparison of number of iterations, execution time (seconds) and maximum absolute errors for the iterative methods (Example 1)

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Methods</th>
<th>Numbers of Iterations</th>
<th>Execution Time (Seconds)</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>FSGS</td>
<td>1326</td>
<td>0.26</td>
<td>6.8176e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>640</td>
<td>0.20</td>
<td>3.3185e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>192</td>
<td>0.09</td>
<td>7.9741e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>92</td>
<td>0.04</td>
<td>1.9461e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(θ = θ = 1.59)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FSGS</td>
<td>4910</td>
<td>1.38</td>
<td>2.7407e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>2553</td>
<td>0.77</td>
<td>5.5044e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>301</td>
<td>0.16</td>
<td>7.6808e-10</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>186</td>
<td>0.08</td>
<td>7.3864E-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(θ = θ = 1.92)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FSGS</td>
<td>18085</td>
<td>15.43</td>
<td>1.1004e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>9426</td>
<td>7.44</td>
<td>1.3653e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>567</td>
<td>0.51</td>
<td>4.5575e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>369</td>
<td>0.37</td>
<td>4.8885E-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(θ = θ = 1.89)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FSGS</td>
<td>66177</td>
<td>210.11</td>
<td>4.4067e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>34618</td>
<td>102.81</td>
<td>2.2029e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>1985</td>
<td>5.82</td>
<td>5.7629e-11</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>1097</td>
<td>4.08</td>
<td>5.3462e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(θ = θ = 1.91)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(θ = θ = 1.96 &amp; θ = 1.97)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**: Reduction percentages of the number of iterations and execution time for four point-EGGS, four point-EGGS and four point-EGMSOR methods compared to the FSGS methods (Example 1)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Numbers of Iterations (%)</th>
<th>Execution Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-EGGS</td>
<td>47.69-51.73</td>
<td>23.08-51.78</td>
</tr>
<tr>
<td>4-EGSOR</td>
<td>85.52-93.87</td>
<td>65.38-96.76</td>
</tr>
<tr>
<td>4-EGMSOR</td>
<td>93.06-98.35</td>
<td>84.62-98.06</td>
</tr>
</tbody>
</table>
Table 3: Comparison of number of iterations, execution time (seconds) and maximum absolute errors for the iterative methods (Example 2)

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Methods</th>
<th>Numbers of Iterations</th>
<th>Execution Time (Seconds)</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>FSGS</td>
<td>1694</td>
<td>0.34</td>
<td>9.7880e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>886</td>
<td>0.28</td>
<td>9.7880e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>273</td>
<td>0.21</td>
<td>9.7880e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>102</td>
<td>0.07</td>
<td>9.7880e-3</td>
</tr>
</tbody>
</table>

\( \theta = 1.92 \)

\( \theta = 1.82 \) & \( \theta = 1.83 \)

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Methods</th>
<th>Numbers of Iterations</th>
<th>Execution Time (Seconds)</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>FSGS</td>
<td>6175</td>
<td>1.56</td>
<td>9.7889e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>3235</td>
<td>1.01</td>
<td>9.7889e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>374</td>
<td>0.24</td>
<td>9.7889e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>216</td>
<td>0.17</td>
<td>9.7889e-3</td>
</tr>
</tbody>
</table>

\( \theta = 1.94 \)

\( \theta = 1.90 \) & \( \theta = 1.91 \)

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Methods</th>
<th>Numbers of Iterations</th>
<th>Execution Time (Seconds)</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>FSGS</td>
<td>22340</td>
<td>16.8</td>
<td>9.7906e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>11748</td>
<td>9.61</td>
<td>9.7906e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>515</td>
<td>0.46</td>
<td>9.7909e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>416</td>
<td>0.43</td>
<td>9.7907e-3</td>
</tr>
</tbody>
</table>

\( \theta = 1.92 \)

\( \theta = 1.95 \) & \( \theta = 1.96 \)

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Methods</th>
<th>Numbers of Iterations</th>
<th>Execution Time (Seconds)</th>
<th>Maximum Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>FSGS</td>
<td>80028</td>
<td>225.82</td>
<td>9.7923e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGGS</td>
<td>42318</td>
<td>129.43</td>
<td>9.7922e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGSOR</td>
<td>1605</td>
<td>4.9</td>
<td>9.7926e-3</td>
</tr>
<tr>
<td></td>
<td>4-EGMSOR</td>
<td>1059</td>
<td>3.97</td>
<td>9.7926e-3</td>
</tr>
</tbody>
</table>

\( \theta = 1.94 \)

\( \theta = 1.98 \) & \( \theta = 1.99 \)

Table 4: Reduction percentages of the number of iterations and execution time for four point-EGGS, four point-EGGS and four point-EGMSOR methods compared to the FSGS methods (Example 2)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Numbers of Iterations (%)</th>
<th>Execution Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-EGGS</td>
<td>47.12-47.70</td>
<td>35.25-42.79</td>
</tr>
<tr>
<td>4-EGSOR</td>
<td>93.38-93.94</td>
<td>38.24-97.83</td>
</tr>
<tr>
<td>4-EGMSOR</td>
<td>93.98-98.68</td>
<td>79.41-98.24</td>
</tr>
</tbody>
</table>
References


Received: June, 2011