Application of Four-Point EGSOR Iteration with Nonlocal Arithmetic Mean Discretization Scheme for Solving Burger’s Equation

N F A Zainal\textsuperscript{1,} J Sulaiman\textsuperscript{2} and M U Alibubin\textsuperscript{3}

\textsuperscript{1,2}Mathematics with Economics Program, Faculty of Science and Natural Resources, Universiti Malaysia Sabah.
\textsuperscript{3}Mahsa College Sabah Campus, Kota Kinabalu, Sabah.

E-mail: farah.zainal19@gmail.com

Abstract. The main objective for this study is to examine the efficiency of block iterative method namely Four-Point Explicit Group Successive Over Relaxation (4EGSOR) iterative method. The nonlinear Burger’s equation is then solved through the application of nonlocal arithmetic mean discretization (AMD) scheme to form a linear system. Next, to scrutinize the efficiency of 4EGSOR with Gauss-Seidel (GS) and Successive Over Relaxation (SOR) iterative methods, the numerical experiments for four proposed problems are being considered. By referring to the numerical results obtained, we concluded that 4EGSOR is more superior than GS and SOR iterative methods in aspects of number of iterations and execution time.

1. Introduction

In recent years, the Burger’s equations have been known as one of the most important nonlinear equation. This equation can be found in applied mathematics, physics and engineering. There are many researchers proposed various methods in solving the problems of Burger’s equations such as Cole-Hopf Transformation [1], Standard Explicit Method [2], Weak Galerkin Finite Element Method [3] and Adomian Decomposition Method [4].

In this paper, we focus on the application of nonlocal arithmetic mean discretization scheme in getting the linear system of Burger’s equation. To solve the linear system since it is sparse and large scale, the family of iterative methods such as point and block iterative methods is used as a linear solver. To scrutinize the efficiency of 4EGSOR iterative method with nonlocal (AMD) scheme, there are four examples of Burger’s problems that can be solved by using the block iterative methods.

In order to get the approximation solution, let us consider the general form of nonlinear Burger’s equation:

\[ \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = \lambda \frac{\partial^2 v}{\partial x^2}, \quad \gamma_0 \leq x \leq \gamma_1, \quad t > 0 \] (1)

with initial and Dirichlet boundary conditions:

\[ v(x, 0) = f(x), \quad \gamma_0 \leq x \leq \gamma_1 \]
\[ v(\gamma_0, t) = f_1(t), \quad v(\gamma_1, t) = f_2(t), \quad t > 0. \]

where \( \lambda \) is a parameter \( (\lambda > 0) \) and \( v = \frac{\partial v}{\partial x} \) is the nonlinear term.

There are many sections in this paper. For the first section, we elaborate the introduction of Burger’s equation. In section 2, we state the formulation of nonlocal AMD scheme to form a linear system. Next, the formulation of the proposed iterative methods is shown to solve the linear system.
Then, in section 4, we introduced four examples of problem 1 and we also present the numerical results. Finally, in the last section we made the conclusion of this paper.

2. Formulation of Nonlocal AMD Scheme

To start this section, problem (1) can be rewritten as

$$\frac{\partial v}{\partial t} + F(x, t, v) \frac{\partial v}{\partial x} = \lambda \frac{\partial^3 v}{\partial x^3}$$

(2)

To discretize problem (2), we need to consider the following general formulations of nonlocal AMD scheme at any time level \((j+1)\) which given as follows [5]

$$V_{i,j+1}^2 = V_{i,j+1} V_{i+1,j+1}$$

(3)

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

(4)

By considering equations (3) and (4), we can eliminate any nonlinear term. Consequently, Problem (2) becomes

$$\frac{V_{i,j+1} - V_{i,j}}{\Delta t} + G_{i,j+1} \left( \frac{V_{i+1,j+1} - V_{i-1,j+1}}{2\Delta x} \right) = \frac{\lambda}{\Delta t} \left( V_{i-1,j+1} - 2V_{i,j+1} + V_{i+1,j+1} \right)$$

(5)

where

$$G_{i,j+1} = F(x_i, t_{j+1}, V_{i,j+1})$$

(6)

Actually, the expression \(G_{i,j+1} \left( \frac{V_{i+1,j+1} - V_{i-1,j+1}}{2\Delta x} \right)\) shows the nonlinear term of problem (2). Then, we need to use the nonlocal AMD scheme to form a system of linear equation. By substituting equation (4) into equation (6), we get the new expression as

$$G_{i,j+1} = F(x_i, t_{j+1}, \frac{V_{i+1,j+1} + V_{i-1,j+1}}{2})$$

(7)

Both equations (5) and (7) can be represented in general form as

$$-\alpha_i V_{i-1,j+1} + b_i V_{i,j+1} - V_{i+1,j+1} = F_{i,j}$$

(8)

where,

$$\alpha_i = \left( \frac{1}{4\Delta h} \right) G_{i,j+1} + \frac{\lambda}{\Delta t} \left( \frac{2\Delta x}{(\Delta t)^2} \right), \quad b_i = \frac{1 + \frac{2\lambda\Delta t}{(\Delta t)^2}}{\beta_i},$$

$$\beta_i = \frac{\lambda}{\Delta t} \left( \frac{1}{4\Delta h} \right) G_{i,j+1}, \quad F_{i,j} = \frac{V_{i,j}}{\Delta t}$$

By considering any group of two neighboring node points, \((x_{i-1}, t_{j+1})\) and \((x_{i+1}, t_{j+1})\), we can generate a system of linear equation in the form of

$$AV_{i,j} = F_{j}$$

(9)

where,

$$A = \begin{bmatrix}
    b_1 & -1 \\
    -\alpha_2 & b_2 & -1 \\
    -\alpha_3 & b_3 & -1 \\
    \vdots & \ddots & \ddots \\
    -\alpha_n & b_n & -1 \\
    -\alpha_n & b_n & -1
\end{bmatrix},$$

$$V_{i,j} = \begin{bmatrix} V_{i,j} \\
    V_{i+1,j} \\
    \vdots \\
    V_{i+n,j} \\
\end{bmatrix},$$

$$F_{j} = \begin{bmatrix} F_{j} \\
    \vdots \\
\end{bmatrix}$$
3. Formulation of EGSOR Method

In the previous section, by referring to the linear system [9], it is obviously show that the feature of its coefficient matrix is sparse and large scale. According to the previous studies, Evans [6] has proposed the Explicit Group iterative method to solve the sparse linear system. Hence, this paper attempts to scrutinize the effectiveness of the EGSOR method. Figure 1 shows the implementation of EGSOR iterative method and as can be seen in this figure, the last three node points are known as an ungroup case [7].

![Figure 1. Implementation of EGSOR iterative method over the solution domain](image)

The general formulation of the EG iterative method can be shown as

$$V_{i+1} = \begin{bmatrix} V_{1,j+1}, V_{2,j+1}, V_{3,j+1}, \ldots \end{bmatrix}^T$$

$$E_{j} = \begin{bmatrix} F_{1,j} + \alpha_1 V_{0,j+1}, F_{2,j}, F_{3,j}, \ldots F_{m-1,j+1} + V_{m+1,j} \end{bmatrix}^T$$

3.1 Formulation of EGSOR Method

By adding the weighted parameter, $\omega$ into equation (11), the general form of 4-Point EGSOR iterative method can be shown as

$$V_{i+1}^{(4,i)} = \begin{bmatrix} V_{i,j+1}^{(4,i)}, V_{i+1,j+1}^{(4,i)}, V_{i+2,j+1}^{(4,i)}, V_{i+3,j+1}^{(4,i)} \end{bmatrix} = \begin{bmatrix} b_1 & -1 & 0 & 0 \\ -\alpha_2 & b_2 & -1 & 0 \\ -\alpha_3 & b_3 & -1 & 0 \\ -\alpha_4 & b_4 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

By adding the weighted parameter, $\omega$ into equation (11), the general form of 4-Point EGSOR iterative method can be shown as

$$V_{i,j+1}^{(4,i)} = \begin{bmatrix} b_1 & -1 & 0 & 0 \\ -\alpha_2 & b_2 & -1 & 0 \\ -\alpha_3 & b_3 & -1 & 0 \\ -\alpha_4 & b_4 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + (1 - \omega) V_{i,j+1}$$
Thus, according to equation (12), let Algorithm 1 as shown below explained about the implementation of 4EGSOR method.

**Algorithm 1: 4EGSOR iterative method**

i. Let the starting value $V_{j+1}^{(0)} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$

ii. Assign the optimal value of $\omega$

iii. Calculate $V_{i,j+1}^{(k+1)}$ using:

$$
\begin{bmatrix}
V_{i+1,j+1}^{(k)} \\
V_{i+2,j+1}^{(k)} \\
V_{i+3,j+1}^{(k)}
\end{bmatrix} = \omega \begin{bmatrix}
b_1 & -1 & & & \\
-\alpha_2 & b_2 & -1 & & \\
& -\alpha_3 & b_3 & -1 & \\
& & -\alpha_4 & b_4 & \\
\end{bmatrix} \begin{bmatrix}
s_1 \\
s_2 \\
s_3 \\
s_4
\end{bmatrix} + (1-\omega) \begin{bmatrix}
V_{i+1,j+1}^{(k)} \\
V_{i+2,j+1}^{(k)} \\
V_{i+3,j+1}^{(k)}
\end{bmatrix}.
$$

For $i = m-4$, calculate $V_{i,j+1}^{(k+1)}$ using

$$
\begin{bmatrix}
V_{i,j+1}^{(k+1)} \\
V_{i+1,j+1}^{(k+1)} \\
V_{i+2,j+1}^{(k+1)}
\end{bmatrix} = \omega \begin{bmatrix}
b_1 & -1 & & & \\
-\alpha_2 & b_2 & -1 & & \\
& -\alpha_3 & b_3 & -1 & \\
& & -\alpha_4 & b_4 & \\
\end{bmatrix} \begin{bmatrix}
s_1 \\
s_2 \\
s_3 \\
s_4
\end{bmatrix} + (1-\omega) \begin{bmatrix}
V_{i+1,j+1}^{(k)} \\
V_{i+2,j+1}^{(k)} \\
V_{i+3,j+1}^{(k)}
\end{bmatrix}.
$$

iv. Convergence test $\left| V_{i,j+1}^{(k+1)} - V_{i,j+1}^{(k)} \right| \leq \varepsilon = 10^{-10}$. If yes, proceed to next step. Or else repeat step (iii).

v. Stop.

4. Numerical Examples

For the numerical comparison to testify the efficiency of 4EGSOR iterative method, there are four examples that are proposed. There are three aspects will be measured such as number of iterations, execution time (second) and maximum absolute error. The numerical results obtained are presented as shown in Table 1.

**Example 1 [8]**

Let the initial value equation be define as

$$
u(x, 1) = \frac{x}{1 + \exp \left(\frac{1}{4\lambda} \left(x^2 - \frac{1}{4}\right)\right)}, \quad t > 0.
$$

(13)

The exact solution of problem (13) is specified by

$$
u(x, t) = \frac{x}{1 + \left(\frac{t}{\tau_e}\right)^{\frac{1}{2}} \exp \left(\frac{x^2}{4\lambda t}\right)}, \quad \text{where} \quad \tau_e = \exp \left(\frac{1}{8\lambda}\right).
$$

(14)
Consider problem (1) with initial value equation are taken from the exact solution \[11\]
The exact solution of problem (17) is shown as

\[ v(x) = \frac{2x}{1+2t}, \]

Example 2 [9]
Let the initial value equation be define as

\[ v(x,0) = 2x, \quad t > 0. \] \hspace{1cm} (15)

The exact solution of problem (15) can be stated as shown

\[ v(x,t) = \frac{2x}{1+2t}. \] \hspace{1cm} (16)

Example 3 [10]
Let the initial value equation be define as

\[ v(x,0) = 2\lambda \frac{\pi \sin(\pi x)}{\sigma + \cos(\pi x)} \quad \text{for} \quad t > 0. \] \hspace{1cm} (17)

The exact solution of problem (17) is shown as

\[ v(x,t) = \frac{2\lambda \pi e^{-\delta t} \sin(\pi x)}{\sigma + e^{-\delta t} \cos(\pi x)}. \] \hspace{1cm} (18)

Example 4 [11]
Consider problem (1) with initial value equation are taken from the exact solution [11]

\[ v(x,0) = \frac{\lambda}{1+\lambda t} \left( x + \tan \left( \frac{x}{2+2\lambda t} \right) \right), \quad 0.5 \leq x \leq 1.5, \quad t \geq 0. \] \hspace{1cm} (19)

| Example | M | Number of iterations | Execution time (second) | Maximum absolute error |
|---------|---|----------------------|-------------------------|------------------------|
|        |   | GS       | SOR    | 4EGSOR | GS   | SOR    | 4EGSOR | GS   | SOR    | 4EGSOR |
| 1       | 256 | 113     | 26     | 15     | 0.14 | 0.08   | 0.06   | 1.603E-04 | 1.603E-04 | 1.603E-04 |
|  | 512 | 392     | 48     | 27     | 0.91 | 0.15   | 0.10   | 1.634E-04 | 1.632E-04 | 1.632E-04 |
|  | 1024 | 1402    | 89     | 50     | 6.27 | 0.51   | 0.28   | 1.645E-04 | 1.639E-04 | 1.639E-04 |
|  | 2048 | 5015    | 163    | 93     | 44.96| 1.90   | 1.02   | 1.661E-04 | 1.642E-04 | 1.642E-04 |
|  | 4096 | 17757   | 315    | 177    | 323.16| 7.20   | 3.79   | 1.722E-04 | 1.642E-04 | 1.642E-04 |
| 2       | 256 | 9391    | 317    | 158    | 10.12| 0.43   | 0.22   | 2.311E-04 | 2.317E-04 | 2.317E-04 |
|  | 512 | 34224   | 633    | 310    | 73.85| 1.55   | 0.78   | 2.292E-04 | 2.317E-04 | 2.317E-04 |
|  | 1024 | 123648  | 1328   | 614    | 535.63| 6.38   | 3.00   | 2.217E-04 | 2.317E-04 | 2.317E-04 |
|  | 2048 | 441778  | 2446   | 1220   | 3851.61| 24.28  | 11.79  | 1.918E-04 | 2.317E-04 | 2.317E-04 |
|  | 4096 | 1556249 | 4913   | 2419   | 30095.34| 94.12  | 46.44  | 7.776E-05 | 2.317E-04 | 2.317E-04 |
| 3       | 256 | 1092    | 143    | 55     | 1.20 | 0.20   | 0.11   | 7.396E-04 | 7.399E-04 | 7.399E-04 |
|  | 512 | 3986    | 274    | 105    | 8.72 | 0.68   | 0.30   | 7.372E-04 | 7.387E-04 | 7.387E-04 |
|  | 1024 | 14490   | 530    | 203    | 63.59| 2.54   | 1.05   | 7.322E-04 | 7.384E-04 | 7.384E-04 |
|  | 2048 | 52197   | 1020   | 389    | 459.38| 9.72   | 3.86   | 7.133E-04 | 7.383E-04 | 7.384E-04 |
|  | 4096 | 185762  | 2143   | 746    | 33310.30| 41.59  | 14.70  | 6.384E-04 | 7.383E-04 | 7.383E-04 |
| 4       | 256 | 30      | 22     | 9      | 0.11 | 0.08   | 0.06   | 1.044E-08 | 8.013E-10 | 1.688E-10 |
|  | 512 | 89      | 41     | 16     | 0.24 | 0.13   | 0.08   | 4.647E-08 | 4.839E-04 | 2.006E-10 |
|  | 1024 | 304     | 80     | 29     | 1.33 | 0.40   | 0.18   | 1.852E-07 | 2.363E-08 | 7.806E-09 |
|  | 2048 | 1076    | 154    | 57     | 9.24 | 1.46   | 0.58   | 7.601E-07 | 4.470E-08 | 1.296E-08 |
|  | 4096 | 3818    | 300    | 111    | 65.28| 5.63   | 2.17   | 3.048E-06 | 8.665E-08 | 2.163E-08 |
Table 2. Reduction percentages for SOR and 4EGSOR iterative methods compared with GS method.

| Methods | Number of iterations | Example 1 | Example 2 | Example 3 | Example 4 |
|---------|----------------------|-----------|-----------|-----------|-----------|
|         |                      |           |           |           |           |
| SOR     | 76.99% - 98.23%      | 96.62% - 99.68% | 86.90% - 98.85% | 26.67% - 92.14% |
| 4EGSOR  | 86.73% - 99.00%      | 98.32% - 99.84% | 94.96% - 99.60% | 70.00% - 97.09% |

| Methods | Execution time | Example 1 | Example 2 | Example 3 | Example 4 |
|---------|---------------|-----------|-----------|-----------|-----------|
|         |               |           |           |           |           |
| SOR     | 42.68% - 97.77% | 95.75% - 99.69% | 83.33% - 98.74% | 27.27% - 91.38% |
| 4EGSOR  | 57.14% - 98.83% | 97.83% - 99.85% | 90.83% - 99.56% | 45.45% - 96.68% |

By referring to Table 1, it shows the numerical results of proposed iterative method based on the four examples of this problem with different grid sizes, \( m = 256, 512, 1024, 2048 \) and \( 4096 \). Meanwhile, Table 2 show the reduction percentage of SOR and 4EGSOR iterative methods with GS iterative method. By referring to Table 1, the number of iterations of SOR iterative method has decreased by approximately 76.99% - 98.23%, 96.62% - 99.68%, 86.90% - 98.85% and 26.67% - 92.14% respectively compared to GS iterative method. Hence, it required less execution time than GS iterative method by 42.68% - 97.77%, 95.75% - 99.69%, 83.33% - 98.74% and 27.27% - 91.38%.

Meanwhile for the 4EGSOR iterative method, the number of iterations has decreased by approximately 86.73% - 99%, 98.32% - %, 94.96% - 99.60% and 70% - 97.09% respectively. Thus, the 4EGSOR iterative method also required less execution time than GS iterative method by 57.14% - 98.83%, 97.83% - %, 90.83% - 99.56% and 45.45% - 96.68%. By referring to the reduction percentage in Table 2, it can be concluded that the 4EGSOR iterative method is more superior as compared with GS and SOR iterative methods.

5. Conclusion
In this study, the nonlocal AMD scheme is successfully applied to solve the nonlinear Burger’s equation. The nonlinear equation are then transform to linear system. By referring to the numerical results obtained, it show that 4EGSOR iterative method is efficient than GS and SOR iterative methods in aspects of number of iterations and execution time. For the future research, other block iterative methods such as EDGAOR method can be considered in solving Burger’s equation.

6. References
[1] Gorguis A 2006 *Applied Mathematics and Computation* vol 173 pp 126-136.
[2] Kutluay S and Esen A 2004 *Appl. Maths. and Comp* vol 167 pp 21-33.
[3] Chen Y and Zhang T 2019 *Appl. Maths. and Comp* vol 348 pp 103-119.
[4] Bakodah H O, Al-Zaid N A, Mirzazadeh M and Zhou Q 2017 *Appl. Maths. and Comp* vol 130 pp 1339-1346.
[5] Alibubin M U, Sunarto A, Sulaiman J and Saudi A 2018 *Advanced Science Letter* vol 24 pp 1922-1926.

[6] Evans D J and Sahimi M S 1987 *Int. Journal of Comp. and Appl. Maths* vol 24 pp 311-341.

[7] Hong E J, Saudi A and Sulaiman J 2017 In Proceeding of the International Journal of Imaging and Robotics vol 17 pp 15-24.

[8] Tamsir M, Dhiman N and Srivastava V K 2016 *Beni-Suef University Journal of Basic and Applied Sciences* vol 5 pp 244-254.

[9] Biazar J and Aminikhah H 2009 *Mathematical and Computer Modelling* vol 49 pp 1394-1400.

[10] Liao W 2008 *Applied Mathematics and Computation* vol 206 pp 755-764

[11] Arora G and Joshi V 2017 *Alexandria Engineering Journal*.

[12] Zhang P and Wang J 2012 *Appl. Maths. and Comp.* vol 219 pp 892-898.

[13] Zainal N F A, Sulaiman J and Alibubin M U, 2018 *International Conference on Mathematics, Engineering and Industrial Applications*.

[14] Saudi A and Sulaiman J 2012 *Third International Conference on Intelligent Systems Modelling and Simulation*.

[15] Sulaiman J, Alibubin M U, Sunarto A and Akhir M K M 2016 *Global Journal of Pure and Applied Mathematics* vol 12 pp 3415-3424.