APPLICATION OF HALF-SWEEP SOR ITERATION WITH NONLOCAL ARITHMETIC DISCRETIZATION SCHEME FOR SOLVING BURGER’S EQUATION

N. F. A. Zainal¹, J. Sulaiman¹ and M. U. Alibubin²
¹Faculty of Science and Natural Resources, Universiti Malaysia Sabah Kota Kinabalu, Sabah, Malaysia
²Mahsa College Sabah Campus Kota Kinabalu, Sabah, Malaysia
E-Mail: farah.zainal19@gmail.com

ABSTRACT
In this paper, the Burger’s equations have been approximated by using the second-order finite difference scheme and the half-sweep nonlocal arithmetic discretization scheme to construct the half-sweep generated linear system. Then, we investigate the applicable formulation of Half-sweep Successive Over Relaxation (HSSOR) iterative method for solving this linear system. In order to verify the effectiveness of the HSSOR iterative method, this paper also included the Full-sweep Successive Over Relaxation (FSSOR) and Full-Sweep Gauss-Seidel (FSGS) iterative methods. The performance analysis of these three proposed iterative methods is illustrated by solving four proposed Burger’s problems. The numerical results illustrate the great performance of the HSSOR iterative method together with half-sweep nonlocal arithmetic discretization scheme to solve the Burger’s equations in senses of execution time and number of iterations.

Keywords: burger’s equations, nonlocal arithmetic mean discretization, HSSOR iteration.

1. INTRODUCTION
Burger’s equation is categorized as nonlinear parabolic partial differential equations. In the last past decades, nonlinear Burger’s equation has attracted attention of many researchers. They have studied the solution of the Burger’s problem using various numerical methods in order to prove its effectiveness in solving the problems. Hence, Tamsir et al. [1] proposed modified B-spline cubic differential quadrature method. Moreover, Liao [2] used the compact finite difference in fourth-order scheme to solve Burger’s problem. Arora et al. [3] considered hybrid trigonometric differential quadrature method to get a solution of nonlinear Burger’s equation. Then, Raslan [4] proposed the quadratic B-spline finite elements together with a collocation method to solve Burger’s equation. Meanwhile, Biazar and Aminikhah [5] used He’s variational iteration method in which this can be used for solving fractional partial equations, Helmholtz equation, Burger and coupled Burger’s equations. Consequently, this study will examine the feasibility of the nonlocal arithmetic mean discretization scheme as an alternative method to eliminate the nonlinear term for solving Burger’s equation.

To examine the performance of HSSOR iteration together with nonlocal arithmetic mean discretization scheme, this paper needs to transform the corresponding nonlinear implicit approximation equations into the corresponding linear system by applying nonlocal arithmetic mean discretization scheme to get the approximate solution of Burger’s problem.

To get the numerical solution, we consider the Burger’s equation as follows:

\[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = v \frac{\partial^2 u}{\partial x^2}, \; a \leq x \leq b, \; t > 0 \]  

Subject to initial condition:

\[ u(x,0) = f(x), \; a \leq x \leq b \]

and boundary conditions:

\[ u(a,t) = f_1(t), \; u(b,t) = f_2(t), \; t > 0. \]

where \( u \) is an unknown function, \( v \) is a parameter in which \( v > 0 \) and \( \frac{\partial u}{\partial x} \) indicates represents the nonlinear term.

For simplifying in discretization process, problem (1) needs to be rewritten by the following equation

\[ \frac{\partial u}{\partial t} + F(x,t,u) \frac{\partial u}{\partial x} = v \frac{\partial^2 u}{\partial x^2} \]
Figure-1. a) and b) illustrate the distribution of mesh points over the solution domain of the problem for full- and half-sweep cases, respectively.

To do the discretization process, we need to construct the finite grid network as denoted in Figure-1. According to Figure-1, these three proposed iterative methods used in this paper has be applied against node points until achieved its convergence criterion. We let solution domain of problem (1) be divided uniformly with grid \( \Delta x \) and \( \Delta t \) in \( x \) and \( t \) directions respectively in which each grid of \( \Delta h \) and \( \Delta t \) can be defined as:

\[
\Delta h = \frac{b - a}{m},
\]

\[
\Delta t = \frac{1}{n}.
\]

According to the finite grid network in Figure-1, the unknown value of \( U(x_i, t_j) = u_{i,j} \) at any node point in Figure-1 can be calculated which is solved iteratively via three proposed iterative methods which is FSGS, FSSOR and HSSOR.

2. FORMULATION OF NONLOCAL ARITHMETIC DISCRETIZATION SCHEMES

In this part, formulation of full- and half-sweep nonlocal arithmetic discretization schemes at any time level \( (j + 1) \) has been used to remove the nonlinear. Practically, for solving one-dimensional nonlinear steady-state problems, the several formulations of the nonlocal arithmetic discretization schemes for full- and half-sweep cases can be stated respectively as follows:

a) Full-sweep case (Moaddy et al. [6])

\[
\begin{align*}
U_i^2 & = U_i U_{i+1} \\
U_i^2 & = \left( \frac{U_{i-1} + U_{i+1}}{2} \right) U_{i+1}
\end{align*}
\]

b) Half-sweep case (Alibubin et al. [7,8])

\[
\begin{align*}
U_i^2 & = U_i U_{i+2} \\
U_i^2 & = \left( \frac{U_{i-2} + U_{i+2}}{2} \right) U_{i+2}
\end{align*}
\]

By using the same concept for derivation of equations (4) and (5), we need to derive the formulation of the full- and half-sweep nonlocal arithmetic discretization schemes at any time level \( (j + 1) \) for solving Burger’s equation via two level approach. This means that equations (4) and (5) at any time level \( (j + 1) \) can be rewritten as follows:

a) Full-sweep case

\[
\begin{align*}
U_{i,j+1}^2 & = U_{i,j+1} U_{i+1,j+1} \\
U_{i,j+1}^2 & = \left( \frac{U_{i-1,j+1} + U_{i+1,j+1}}{2} \right) U_{i+1,j+1}
\end{align*}
\]

b) Half-sweep case

\[
\begin{align*}
U_{i,j+1}^2 & = U_{i,j+1} U_{i+2,j+1} \\
U_{i,j+1}^2 & = \left( \frac{U_{i-2,j+1} + U_{i+2,j+1}}{2} \right) U_{i+2,j+1}
\end{align*}
\]

Actually, equations (4) to (7) are categorized as a family of non-standard finite difference (NSFD) methods. By referring to equations (6b) and (7b), clearly the formulation of these nonlocal arithmetic discretization schemes for the internal node point \((x_{i}, t_{j+1})\) involves the two neighboring node points. Based on both equations, the
nonlinear term of the generated nonlinear approximation equation of problem (1) can be eliminated.

3. DERIVATION OF HALF-SWEEP NONLINEAR IMPLICIT FINITE DIFFERENCE APPROXIMATION EQUATION

In getting the finite difference solution, actually, problem (1) can be approximated at a reference node point \((x_i, t_{j+1})\) by using several different finite differences schemes. According to Figure 1(b), we restrict our discussion on derivation of the second-order half-sweep nonlinear implicit approximation equation for approximating problem (1). It means that the second-order implicit finite difference discretization scheme has been required to discretize over problem (1), therefore the second-order half-sweep nonlinear implicit finite difference approximation equation for approximating problem (1) can be easily given as

\[
\begin{align*}
U_{i, j+1} + G_{i, j+1} \left( \frac{U_{i+2, j+1} - U_{i-2, j+1}}{2(2\Delta x)} \right) \\
= \frac{v}{(2\Delta x)^2} \left( U_{i-2, j+1} - 2U_{i, j+1} + U_{i+2, j+1} \right)
\end{align*}
\]  

where

\[
G_{i, j+1} = F \left( x_i, t_{j+1}, U_{i, j+1} \right).
\]  

Since, the expression \(G_{i, j+1} \left( \frac{U_{i+2, j+1} - U_{i-2, j+1}}{2(2\Delta x)} \right)\) is denoted as the nonlinear term of problem (1), the half-sweep nonlocal arithmetic mean discretization scheme in equation (7b) can be imposed into equation (8) to construct the corresponding linear system. By considering a group of two neighboring node points, \((x_{i-2}, t_{j+1})\) and \((x_{i+2}, t_{j+1})\) and applying equation (7b), equation (9) can be easily expressed as

\[
G_{i, j+1} = F \left( x_i, t_{j+1}, \frac{U_{i-2, j+1} + U_{i+2, j+1}}{2} \right).
\]  

Referring to equation (8), the half-sweep three point implicit approximation equation at any time level \((j + 1)\) for the entire interior node points \((x_i, t_{j+1})\) of type • as depicted in Figure-1(b) can be rewritten as,

\[
-a_i U_{i-2, j+1} + b_i U_{i, j+1} - B_i U_{i+2, j+1} = F_i, \quad i = 2, 4, 6, ..., m - 2
\]  

\[
a_i = \left( \frac{1}{4\Delta x} \right) G_{i, j+1} + \frac{v}{(2\Delta x)^2},
\]

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

\[
B_i = \frac{v}{(2\Delta x)^2} - \left( \frac{1}{4\Delta x} \right) G_{i, j+1}, \quad F_{i, j} = \frac{U_{i, j}}{\Delta t},
\]  

\[
U_{j+1} = \begin{bmatrix} U_{2, j+1}, U_{4, j+1}, U_{6, j+1}, \ldots, U_{m-2, j+1} \end{bmatrix}^T, \quad F_j = \begin{bmatrix} F_{2, j} + a_0 U_{0, j+1}, F_{4, j}, F_{6, j}, \ldots, F_{m-4, j} \end{bmatrix}^T
\]  

\[
A = \begin{bmatrix}
 b & -B_2 & \cdots & \cdots & -B_{m-2} \\
-a_4 & b & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \cdots & \ddots & \ddots & \cdots \\
-a_{m-4} & \cdots & \cdots & b & \cdots \\
\end{bmatrix}
\]  

4. FORMULATION OF SUCCESSIVE OVER RELAXATION ITERATION FAMILY

As can be seen in equation (11), the computational molecule for equation (11) is depicted in Figure-2. Considering the approximation equation (11), the corresponding system of three point linear implicit approximation equations at any time level \((j + 1)\) for each interior node point of the solution domain in Figure-1 can be expressed by using a matrix representation as

\[
AU_{j+1} = F_j
\]  

where,

\[
U_{j+1} = \begin{bmatrix} U_{2, j+1}, U_{4, j+1}, U_{6, j+1}, \ldots, U_{m-2, j+1} \end{bmatrix}^T, \quad F_j = \begin{bmatrix} F_{2, j} + a_0 U_{0, j+1}, F_{4, j}, F_{6, j}, \ldots, F_{m-4, j} \end{bmatrix}^T
\]  

Based on the tridiagonal linear system in equation. (12), it can be revealed that the coefficient matrix, \(A\) in equation (12) are sparse and large scale. In the effort of getting the numerical solution of the sparse and large linear system, many studies on various iterative methods had been proposed to get the approximate solution of this linear system. Based on the literature review, Connolly et al. [6] proposed Gauss-Seidel iterative method and Young [9,10,11] has proposed the SOR iterative method.
Based on the half-sweep three-point implicit approximation equation in equation (11), the general formulation of the HSSOR iterative method at any time level \((j + 1)\), can be stated as [9,10,11]

\[
U_{i,j+1}^{(k)} = (1 - \omega)U_{i,j+1}^{(k-1)} + \frac{\omega}{b} \left( F_{i,j} + aU_{i-1,j+1}^{(k-1)} + B_{i+1,j+1}^{(k-1)} \right)
\]

(13)

for \(i = 2,4,\ldots,m - 2\) where \(\omega\) represents as a relaxation factor in which this parameter can be calculated practically by selecting values periodically until the optimum value of \(\omega\) is obtained among the range \(1 \leq \omega < 2\). The approximate value of \(\omega\) is considered optimal where its iteration number is the smallest. Hence, based on equation (13), the general implementation of HSSOR iteration for solving the linear system (12) can generally be explained in Algorithm 1.

**Algorithm 1 : HSSOR iterative method**

i. Initialize \(U_{i,j}^{(0)} \leftarrow 0\) and \(\varepsilon \leftarrow 10^{-10}\)

ii. Assign the optimal value of \(\omega\)

iii. For \(i = 1,2,\ldots, n\), perform

\[
U_{i,j+1}^{(k+1)} = (1 - \omega)U_{i,j+1}^{(k)} + \frac{\omega}{b} \left( F_{i,j} + aU_{i-1,j+1}^{(k)} + B_{i+1,j+1}^{(k)} \right)
\]

iv. Perform the convergence test,

\[
\left| U_{i,j+1}^{(k+1)} - U_{i,j+1}^{(k)} \right| \leq \varepsilon = 10^{-10}
\]

If yes, proceed to step (v). Otherwise go back to step (iii).

v. State finite difference solutions approximate solutions.

5. NUMERICAL EXPERIMENTS

In this section, we attempt to examine the effectiveness of the HSSOR iterative method compared to FSSOR and HGS iterative methods, we proposed four examples of problem (1). There are three criteria that will be used for numerical comparison. The following abbreviation are used to indicate three parameter, number of iterations (Iter.), time (T) and maximum absolute error (MaxEr) to be recorded in Table-1.

**Table-1.** Numerical results of FSGS, FSSOR and HSSOR iterative methods.

<table>
<thead>
<tr>
<th>Example</th>
<th>M</th>
<th>Number of iterations</th>
<th>Execution time</th>
<th>Maximum absolute error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FSFG</td>
<td>FSSOR</td>
<td>HSSOR</td>
</tr>
<tr>
<td>1</td>
<td>256</td>
<td>113</td>
<td>26</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>392</td>
<td>48</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>1402</td>
<td>89</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>2048</td>
<td>5015</td>
<td>163</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td>17757</td>
<td>315</td>
<td>163</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>9391</td>
<td>317</td>
<td>161</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>34224</td>
<td>633</td>
<td>324</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>123648</td>
<td>1328</td>
<td>633</td>
</tr>
<tr>
<td></td>
<td>2048</td>
<td>441778</td>
<td>2446</td>
<td>1249</td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td>1556249</td>
<td>4913</td>
<td>2446</td>
</tr>
<tr>
<td>3</td>
<td>256</td>
<td>1092</td>
<td>143</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>3986</td>
<td>274</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>14490</td>
<td>530</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>2048</td>
<td>52197</td>
<td>1020</td>
<td>530</td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td>185762</td>
<td>2143</td>
<td>1020</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>30</td>
<td>22</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>89</td>
<td>41</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>304</td>
<td>80</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>2048</td>
<td>1076</td>
<td>154</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td>3818</td>
<td>300</td>
<td>154</td>
</tr>
</tbody>
</table>

**Example 1 [1]**

Let the initial value equation as follows:

\[
u(x,1) = \frac{x}{1 + \exp \left( \frac{1}{4v} \left( x^3 - 1 \right) \right)}
\]

(14)

for \(t > 0\).
with exact solution of problem (14) is given by

$$u(x,t) = \frac{x}{1 + \left(\frac{t}{t_0}\right)^\frac{1}{2} \exp\left(\frac{x^2}{4vt}\right)},$$

where $t_0 = \exp\left(\frac{1}{8\nu}\right)$.

**Example 2 [5]**

Let the initial value equation as follows:

$$u(x,0) = 2x, \quad \text{for } t > 0.$$  \hspace{1cm} (16)

with exact solution of problem (16) is given by

$$u(x,t) = \frac{2x}{1 + 2t}.$$  \hspace{1cm} (17)

**Example 3 [12]**

Let the initial value equation as follows:

$$u(x,0) = 2\nu \frac{\pi \sin\left(\frac{\pi x}{\sigma + \cos\left(\frac{\pi x}{\sigma}\right)}\right)}{\sigma + e^{-\pi^2 \sigma \nu} \cos\left(\frac{\pi x}{\sigma}\right)} , \quad \text{for } t > 0.$$  \hspace{1cm} (18)

with exact solution of problem (18) is given by

$$u(x,t) = \frac{2\nu \pi e^{-\pi^2 \sigma \nu} \sin\left(\frac{\pi x}{\sigma}\right)}{\sigma + e^{-\pi^2 \sigma \nu} \cos\left(\frac{\pi x}{\sigma}\right)}.$$  \hspace{1cm} (19)

**Example 4 [3]**

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

$$u(x,0) = \nu \left[\frac{x}{1 + \nu \left(\frac{x}{2 + 2\nu t}\right)}\right],$$  \hspace{1cm} (20)

where $0.5 \leq x \leq 1.5, \quad t \geq 0$.

### Table-2

<table>
<thead>
<tr>
<th>Methods</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
<th>Example 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSSOR</td>
<td>77.0% – 98.2%</td>
<td>96.6% – 99.7%</td>
<td>86.9% – 98.9%</td>
<td>26.7% – 92.1%</td>
</tr>
<tr>
<td>HSSOR</td>
<td>87.6% – 99.1%</td>
<td>98.3% – 99.8%</td>
<td>93.2% – 99.5%</td>
<td>60.0% – 96.0%</td>
</tr>
</tbody>
</table>

### 4. CONCLUSIONS

This paper has been successfully generated a system of linear equation as shown in equation (11) based on the formulation of approximation equations using the nonlocal arithmetic discretization scheme. By referring to the numerical result obtained in Table-1, it clearly showed that HSSOR iterative method is better than FSGS and FSSOR iterative methods in terms of number of iterations and the execution time. HSSOR iterative method requires less number of iterations approximately by 87.6%–99.1%, 98.3%–99.8%, 93.2%–99.5% and 60.0%–96.0% respectively compared to FSGS iterative method. However, in terms of execution time, HSSOR are faster about 64.3%–98.8%, 98.0%–99.9%, 90.0%–99.4% and 45.5%–95.4% as compared with FSGS iterative methods. In terms of the accuracy of numerical solution obtained, all result of the proposed iterative methods gives in a good agreement. Finally, we concluded that HSSOR iterative method is better than other SOR iterative family.

### REFERENCES


