Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Explicit Group Iterative Method with Semi-Approximate Implicit Approach for Solving One-Dimensional Burgers' Equation

Nur Farah Azira Zainal[1], Jumat Sulaiman[1,*], Azali Saudi[2], Nur Afza Mat Ali[1], Nor Syahida Mohamad[1], Andang Sunarto[3]

[1] Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
[2] Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
[3] Tadris Matematika, Universitas Islam Negeri (UIN) Fatmawati Sukarno, Bengkulu 38211, Indonesia

ABSTRACT

This paper is considering a semi-approximate approach in investigating the Burgers' problem. The process of the discretization of Burgers' problem has taken place which it starts with the second-order finite difference in the discretize process together with the linearization part by using the semi-approximate implicit scheme in a way to achieve the approximation equation, thus generating the corresponding linear system equations. Besides, the Gauss-Seidel (GS), Successive Over-Relaxation (SOR) and explicit group (EG) iterative method has combined together with the SOR iterative method, namely as 4-point EGSOR (4EGSOR) has been introduced in this study for resolving the linear system. To assess the proficiency of the suggested methods on the approximation equation, the numerical test has been conducted by considering three parameters, which are computational time, iteration number and maximum absolute error. The findings indicate that the 4EGSOR method outperforms both SOR and GS iterative methods.

Keywords:
Nonlinear Burgers' equation; Implicit finite difference scheme; Semi-approximate implicit approach; Explicit group iteration

1. Introduction

The early discovery of the Burgers' equation was in 1915 by Bateman. Burger itself introduced a new improvised and stable solution to the Burgers' problem in 1939 [1]. By referring to Bateman, Burgers' equation was categorized as a nonlinear (NL) equation, which is under the parabolic partial differential equations (PDEs). It has been applied in diverse areas of applied mathematics including instances like diffusion wave in fluid dynamics, nonlinear acoustics, gas dynamic, traffic flows and heat conduction [2,3].

In this study, we attempt to seek the approximate solutions of the following Burgers' equation, accompanied by a brief explanation of each characteristic, presented by Se Bonkile *et al.,* [4]:

* *Corresponding author.*
E-mail address: jumat@ums.edu.my

https://doi.org/10.37934/araset.61.1.127137

$$\frac{\delta k}{\delta t}+k\frac{\delta k}{\delta n}=\gamma\frac{\delta^2 k}{\delta n^2}, \tag{1}$$

subject to initial conditions:

$$k(n,0)=g(n),\ \ n\in[\alpha,\beta],$$

and the Dirichlet boundary conditions:

$$k(\alpha,t)=g_0(t),\ \ k(\beta,t)=g_1(t),\ \ t>0.$$

The characteristics:

i. γ is a viscosity with $\gamma > 0$
ii. $k\frac{\delta k}{\delta n}$ is known as NL convective term.

In many cases, obtaining the exact solution of Burgers' equation is not feasible. Thus, several numerical techniques have been outlined to develop the numerical solutions of Eq. (1) [5]. In previous studies, the approximate solution of Eq. (1) has been constructed using homotopy perturbation method (HPM), compact difference method (CDM), finite difference (FD) scheme, B-Spline method, variational iteration method (VIM), finite element (FE) scheme and differential quadrature method (DQM). For instance, Biazar and Ghazvini presented HPM [6] to acquire the approximate solution of Eq. (1). Next, Noorzad *et al.,* [7] applied the combination of VIM and HPM to handle Burgers' problem. Mohyud-Din *et al.,* [8] proposed a modified version of VIM by introducing He's polynomials. Furthermore, Pandey *et al.,* [9] applied the Douglas FD scheme in order to reduced Eq. (1) to the heat equation. Kaysar *et al.,* [10] proposed three new semi-implicit FD methods to find the solution of Eq. (1). Meanwhile, new fully implicit FD scheme has been proposed by Mohamed [11].

Besides that, Chen and Zhang [12] implemented a weak Galerkin FE method. The numerical test has been implied and the results show that the approach was applicable to the Burgers' equation. Based on the study of Tamsir *et al.,* [13], the study has imposed the cubic B-Spline DQM in extended version into Eq. (1). Mittal and Jain [14] solve 1D Burgers' equation by implementing cubic B-splines over FE with modified version. In 2017, Aswin *et al*., [15] also employed a polynomial based DQM and quasi-linearization is used to eliminate nonlinearity. Yang *et al.,* [16] presented high-order CDM to generate approximation equation of Eq. (1), then Thomas algorithm is used to acquire the approximate solution of the tridiagonal linear system (LS).

Recently, the implementation of semi-approximate (SA) approach based on the NL Burger's equations was found to be effective in converting a NL system into a LS [17]. The solution of LS can then be solve using direct methods or iterative methods. In this study, we consider a second-order implicit finite difference (SIFD) scheme to discretize the Eq. (1) to form the approximation equation, which leads into a system of NL. Then, we employ a semi-approximate implicit (SAI) approach to transform the NL system into a LS. By implementing such an approach, the highly computational complexity of formulating a linear system, such as by using Newton method, can be avoided. Following that, we consider 4EGSOR method to address the large and sparse LS generated from the process of discretization of one-dimensional Burgers' equation.

This paper consists of five parts. In the subsequent part, we elaborate on the formulation of SA approach employed to the Burgers' equation, leading to the generation of a system of linear equations. Following that, we discuss the formulation of 4EGSOR iterative method to achieve the approximate solution of the generated LS. Following that, three numerical examples were presented to demonstrate the computational time, iteration number and maximum absolute error of the 4EGSOR method in comparison to GS and SOR methods. Finally, we conclude the findings in the last section.

2. Formation of Semi-Approximate Implicit Approximation Equation

As discussed earlier, the SIFD scheme is applied to Eq. (1) to derive its corresponding NL approximation equation. Subsequently, the SAI approach is employed to formulate the NL approximation equation for the development of a LS. Prior to entering the discretization process, Eq. (1) can be simplified into new equation as follows

$$\frac{\delta k}{\delta t}+G\left(n,t,k\right)\frac{\delta k}{\delta n}=\gamma\frac{\delta^2 k}{\delta n^2}. \tag{2}$$

where $G\left(n,t,k\right)$ is denoted as a NL function. Next, we contemplate the segmentation of the solution domain, denoted as $y_i,\ i=0,1,2,\ldots,m-1$ and $t_j,\ j=0,1,2,\ldots,m-1$. Following this, implementing the SIFD discretization scheme on Eq. (2), yields the corresponding NL approximation equation as articulated by Zainal, Sulaiman and Alibubin [18].

$$\frac{k_{i.j+1}-k_{i.j}}{\Delta t}+g_{i.j}\left(k_{1.j+1},\ k_{2.j+1},\ \ldots,\ k_{m-1.j+1}\right)=\frac{\gamma}{\left(\Delta h\right)^2}\left(k_{i-1.j+1}-2k_{i.j+1}+k_{i+1.j+1}\right), \tag{3}$$

where

$$g_{i.j}\left(k_{1.j+1},\ k_{2.j+1},\ \ldots,\ k_{m-1.j+1}\right)=d\left(n_i,\ t_{j+1},\ k_{i.j+1},\ \frac{k_{i+1.j+1}-k_{i-1.j+1}}{2\Delta h}\right). \tag{4}$$

Due to the existence of NL term in Eq. (4), it is necessary to eliminate this term using the SA approach in order to formulate a LS for Eq. (1) [19]. To achieve this, the term $y_{i.j+1}$ in Eq. (4) is approximated as $y_{i.j}$, given the significantly low value of Δt. As a result, Eq. (4) can be transformed into:

$$g_{i.j}\left(k_{1.j+1},\ k_{1.j+1},\ \ldots,\ k_{m-1.j+1}\right)=d\left(n_i,\ t_{j+1},\ k_{i.j},\ \frac{k_{i+1.j+1}-k_{i-1.j+1}}{2\Delta h}\right). \tag{5}$$

To minimize computational complexity, the SAI approximation equation for Eq. (1), presented as:

$$-p_{i.j}k_{i-1.j+1}+q_i k_{i.j+1}-k_{i+1.j+1}=G_{i.j},\quad i=1,\ 2,\ 3,\ \ldots,\ m-1, \tag{6}$$

where

$$p_i = \frac{\left(\frac{1}{2(\Delta h)}\right) g_{i.j} + \frac{\gamma}{(\Delta h)^2}}{c_i}, \quad q_i = \frac{1 + \frac{2\gamma \Delta t}{(\Delta h)^2}}{c_i}, \quad c_i = \frac{\gamma}{(\Delta h)^2} - \left(\frac{1}{2(\Delta h)}\right) g_{i.j}, \quad G_{i.j} = \frac{k_{i.j}}{\Delta t . c_i}.$$

By referring to the approximation Eq. (6), it becomes apparent that a sequence of LS at every time step $(j+1)$ can be constructed as:

$$B\underline{k}_{j+1} = \underline{G}_j, \tag{7}$$

where

$$B = \begin{bmatrix} q_1 & -1 & & & & \\ -p_2 & q_2 & -1 & & & \\ & -p_3 & q_3 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -p_{m-2} & q_{m-2} & -1 \\ & & & & -p_{m-1} & q_{m-1} \end{bmatrix}_{(m-1)\times(m-1)},$$

$$\underline{k}_{j+1} = \left[k_{1.j+1},\ k_{2.j+1},\ k_{3.j+1},\ \ldots,\ k_{m-1.j+1}\right]^T,$$

$$\underline{G}_j = \left[G_{1.j} + p_1 k_{0.j+1},\ G_{2.j},\ G_{3.j},\ \ldots,\ G_{m-2.j},\ G_{m-1.j} + k_{m.j+1}\right]^T.$$

3. Formation of 4EGSOR Iterative Method

In the preceding section, the mention of the linear system in Eq. (7) makes it evident that its coefficient matrix exhibits sparsity and a large scale. This section will present the structure of the 4EGSOR iteration, which was introduced by Evans [20] to address sparse linear systems. This method aims to minimize the computational complexity by solving several small groups of points instead of handling a large system of linear equations per iteration. Hence, the coefficients matrix, B are divided into small groups of four points. The illustration of the 4EGSOR iteration is depicted in Figure 1, where it is apparent that the last block is incomplete block with three node points are identified as an ungrouped case [21].

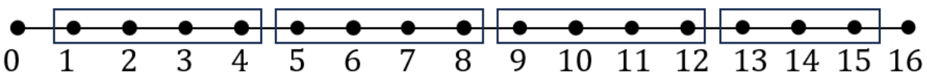

Fig. 1. Implementation of 4EGSOR method for solution domain $m = 16$

By considering SAI approximation equation of Burgers' problem in Eq. (8), let consider any group of four points represented in a (4x4) linear system at any given time step $(j+1)$ can be formulated as follows:

$$\begin{bmatrix} q_i & -1 & & \\ -p_{i+1} & q_{i+1} & -1 & \\ & -p_{i+2} & q_{i+2} & -1 \\ & & -p_{i+3} & q_{i+3} \end{bmatrix} \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}, \tag{8}$$

where

$$r_1 = p_i k_{i-1.j+1} + \frac{1}{\Delta t.q_i} k_{i.j},$$
$$r_2 = \frac{1}{\Delta t.q_{i+1}} k_{i+1.j},$$
$$r_3 = \frac{1}{\Delta t.q_{i+2}} k_{i+2.j},$$
$$r_4 = k_{i+4,j+1} + \frac{1}{\Delta t.q_{i+3}} k_{i+3.j}.$$

Now, upon obtaining the inverse matrix of Eq. (8) the 4-point EG (4EG) method are given as:

$$\begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix}^{(v+1)} = \begin{bmatrix} q_i & -1 & & \\ -p_{i+1} & q_{i+1} & -1 & \\ & -p_{i+2} & q_{i+2} & -1 \\ & & -p_{i+3} & q_{i+3} \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}, \tag{9}$$

By incorporating one weighted parameter, ω, into Eq. (9), the 4EGSOR iterative method can generally formulated as:

$$\begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix}^{(v+1)} = (1-\omega) \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix}^{(v)} + \omega \begin{bmatrix} q_i & -1 & & \\ -p_{i+1} & q_{i+1} & -1 & \\ & -p_{i+2} & q_{i+2} & -1 \\ & & -p_{i+3} & q_{i+3} \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}, \tag{10}$$

where the permissible range for the parameter ω is defined as $1 < \omega < 2$. For the last block comprising three node points at $i = m - 4$ as illustrated in Figure 1 can be stated as

$$\begin{bmatrix} q_i & -1 & \\ -p_{i+1} & q_{i+1} & -1 \\ & -p_{i+2} & q_{i+2} \end{bmatrix} \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \tag{11}$$

where

$r_1 = p_i k_{i-1.j+1} + \frac{1}{\Delta t.q_i} k_{i.j},$

$r_2 = \frac{1}{\Delta t.q_{i+1}} k_{i+1.j},$

$r_3 = k_{i+3,j+1} + \frac{1}{\Delta t.q_{i+2}} k_{i+2.j}.$

Next, through the computation of the inverse matrix of Eq. (11), the 3-point EG (3EG) method with parameter ω can be generally formulated as:

$$\begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \end{bmatrix}^{(v+1)} = (1-\omega) \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \end{bmatrix}^{(v)} + \omega \begin{bmatrix} q_i & -1 & \\ -p_{i+1} & q_{i+1} & -1 \\ & -p_{i+2} & q_{i+2} \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}. \quad (12)$$

Therefore, based on Eq. (10) and Eq. (12), Algorithm 1 provided below elucidates the implementation details of the 4EGSOR and 3EGSOR methods.

Algorithm 1: 4EGSOR scheme

i. Assign value $k_{j+1}^{(0)} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$
ii. Designate the optimum value for ω
iii. Calculate $k_{i.j+1}^{(v+1)}$:
For $i = 1, 5, 9, \dots, m-8$, calculate $k_{i.j+1}^{(v+1)}$ using

$$\begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix}^{(v+1)} = (1-\omega) \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \\ k_{i+3.j+1} \end{bmatrix}^{(v)} + \omega \begin{bmatrix} q_i & -1 & & \\ -p_{i+1} & q_{i+1} & -1 & \\ & -p_{i+2} & q_{i+2} & -1 \\ & & -p_{i+3} & q_{i+3} \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix},$$

For $i = m-4$, calculate $k_{i.j+1}^{(v+1)}$ using

$$\begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \end{bmatrix}^{(v+1)} = (1-\omega) \begin{bmatrix} k_{i.j+1} \\ k_{i+1.j+1} \\ k_{i+2.j+1} \end{bmatrix}^{(v)} + \omega \begin{bmatrix} q_i & -1 & \\ -p_{i+1} & q_{i+1} & -1 \\ & -p_{i+2} & q_{i+2} \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}.$$

iv. Conduct the test of convergence and $\left| k_{i.j+1}^{(v+1)} - k_{i.j+1}^{(v)} \right| \le \varepsilon = 10^{-10}$. If yes, proceed to the subsequent step. Otherwise, revert to preceding step.
v. Present approximate solution.

4. Numerical Experiments

To analyse the proficiency of the 4EGSOR, let's consider three examples of NL Burgers' problem. In order to conduct a comparative analysis, we evaluated the proposed iterations alongside the GS and SOR, designated as reference methods. The numerical results obtained from implementing all proposed iterations were scrutinized based on three criteria which are maximum absolute error, computational time (sec), iteration number [22-24].

4.1 Example [25]

Consider the given problem Eq. (1), where the initial value equation (IVE) is derived from the exact solution [26].

$$k\left(n,t\right)=\frac{\gamma}{1+\gamma t}\left(n+\tan\left(\frac{n}{2+2\gamma t}\right)\right),\quad t\geq 0. \tag{13}$$

4.2 Example [27]

Consider the given problem Eq. (1), where the IVE is derived from the exact solution [28].

$$k\left(n,t\right)=\frac{z}{\alpha}-(\frac{2\gamma}{\alpha})\tanh\left(n-zt\right),\quad t\geq 0. \tag{14}$$

4.3 Example [29]

Consider the given problem Eq. (1), where the IVE is derived from the exact solution [30].

$$k\left(n,t\right)=2\gamma\pi\left(\frac{\frac{1}{4}\sin\pi ne^{-\pi^2\gamma^2 t}+\sin 2\pi ne^{-4\pi^2\gamma^2 t}}{1+\frac{1}{4}\cos\pi ne^{-\pi^2\gamma^2 t}+\frac{1}{2}\cos 2\pi ne^{-4\pi^2\gamma^2 t}}\right),\quad t\geq 0. \tag{15}$$

As can be seen in Table 1, Table 2, Table 3 and Table 4, the numerical results that have been tabulated showing that there is a huge significant difference between those presented iteration especially for the 4EGSOR compared to SOR iterative method with the reduction percentage 90.46% - 99.07%, 87.43% - 98.86% and 89.53% - 99.16% respectively for the iteration number for example 1, 2 and 3. While the reading of the reduction percentage for computational time is quite huge too with 83.72% - 98.91%, 79.45% - 98.71% and 78.08% - 99.04% respectively. The 4EGSOR has demonstrated enhancements in both iteration number and computational time when compared to the SOR iterative method. The reduction percentage for both iteration number and computational time, showing that the 4EGSOR has higher percentage compared to the SOR iterative method.

Table 1
The comparison of results for three difference examples in term of iteration number

Example	m	Iteration number		
		GS	SOR	4EGSOR
1	1024	304	79	29
	2048	1076	154	57
	4096	3818	300	111
	8192	13395	584	217
	16384	46100	1138	427
2	1024	175	60	22
	2048	618	116	43
	4096	2218	225	84
	8192	7926	440	162
	16384	27969	857	319
3	1024	172	48	18
	2048	605	92	34
	4096	2165	175	64
	8192	7713	333	120
	16384	27111	632	227

Table 2
The comparison of results for three difference examples in term of computational time

Example	m	Computational Time		
		GS	SOR	4EGSOR
1	1024	1.29	0.40	0.21
	2048	8.51	1.38	0.59
	4096	61.32	5.47	2.14
	8192	417.43	20.57	8.04
	16384	2882.05	80.45	31.47
2	1024	0.73	0.31	0.15
	2048	4.84	1.06	0.45
	4096	34.74	4.18	1.60
	8192	249.34	15.50	6.03
	16384	1826.78	60.76	23.50
3	1024	0.73	0.25	0.16
	2048	4.86	0.88	0.37
	4096	35.07	3.28	1.24
	8192	248.15	11.72	4.55
	16384	1749.57	44.44	16.81

Table 3
The comparison of results for three difference examples in term of maximum absolute error

Example	m	Maximum Absolute Error		
		GS	SOR	4EGSOR
1	1024	1.85386E-07	6.34677E-08	7.42303E-08
	2048	7.60356E-07	4.53404E-08	6.95476E-08
	4096	3.04821E-06	1.96375E-08	5.68860E-08
	8192	1.21582E-05	1.21364E-07	5.03990E-07
	16384	4.83882E-05	2.89036E-07	3.31796E-07
2	1024	1.69446E-06	1.60216E-06	1.75745E-06
	2048	2.00203E-06	1.61364E-06	1.75730E-06
	4096	3.24059E-06	1.68263E-06	1.78119E-06
	8192	8.19095E-06	1.79016E-06	1.78824E-06
	16384	2.79898E-06	1.95586E-06	1.79461E-06
3	1024	3.06929E-04	3.06845E-04	3.06442E-04
	2048	3.07213E-04	3.06877E-04	3.06448E-04
	4096	3.08327E-04	3.06923E-04	3.06456E-04
	8192	3.12758E-04	3.07016E-04	3.06475E-04
	16384	3.30469E-04	3.07139E-04	3.06475E-04

Table 4
Percentage of decrement of iteration number and computational time of SOR and 4EGSOR iterative methods compared with GS iterative method

Example	Method	Iteration number	Computational time
1	SOR	74.01% - 97.53%	68.99% - 97.21%
	4EGSOR	90.46% - 99.07%	83.72% - 98.91%
2	SOR	65.71% - 96.94%	57.53% - 96.67%
	4EGSOR	87.43% - 98.86%	79.45% - 98.71%
3	SOR	72.09% - 97.67%	65.75% - 97.46%
	4EGSOR	89.53% - 99.16%	78.08% - 99.04%

5. Conclusions

In this study, we present the implementation of the second-order implicit finite difference for discretization, coupled with a semi-approximate approach employed in solving the Burgers' problem for the linearization process, resulting in the formulation of the corresponding approximation linear system. The numerical solutions, as outlined in Table 1, Table 2, Table 3 and Table 4, unequivocally manifest a substantial reduction in both the iteration number and computational time when utilizing the proposed iterative methods, particularly the 4EGSOR, in comparison to the conventional SOR iterative method. The comprehensive numerical findings ascertain the superior performance of the 4EGSOR iterative method over GS and SOR, evidenced by its notable efficiency in terms of iteration number and execution time. This discernible enhancement can be attributed to the diminished computational complexity inherent in the 4EGSOR approach, characterized by the involvement of fewer nodes points in the iteration process.

Acknowledgement
The financial support for this study was provided by Universiti Malaysia Sabah under the research grant scheme (GUG0490-1/2020).

References

[1] Rageh, Tamer M. and Hassan N. A. Ismail. "Numerical Solution of Modified Burger Equation." *East African Scholars Journal of Engineering and Computer Sciences*, no. 1 (2018): 21-28.
[2] Seydaoğlu, Muaz. "An accurate approximation algorithm for Burgers' equation in the presence of small viscosity." *Journal of Computational and Applied Mathematics* 344 (2018): 473-481. https://doi.org/10.1016/j.cam.2018.05.063
[3] Seydaoğlu, Muaz, Utku Erdoğan, and Turgut Öziş. "Numerical solution of Burgers' equation with high order splitting methods." *Journal of Computational and Applied Mathematics* 291 (2016): 410-421. https://doi.org/10.1016/j.cam.2015.04.021
[4] Bonkile, Mayur P., Ashish Awasthi, C. Lakshmi, Vijitha Mukundan, and V. S. Aswin. "A systematic literature review of Burgers' equation with recent advances." *Pramana* 90 (2018): 1-21. https://doi.org/10.1007/s12043-018-1559-4
[5] Nasir, Sharifah Noha Zahirah Syed Abdul, Nurul Ain Ab Wahab, and Mohd Agos Salim Nasir. "Graphical User Interface for Solving Non-Linear Equations for Undergraduate Students." *International Journal of Advanced Research in Future Ready Learning and Education* 30, no. 1 (2023): 25-34. https://doi.org/10.37934/frle.30.1.2534
[6] Biazar, Jafar, and H. Ghazvini. "Exact solutions for nonlinear Burgers' equation by homotopy perturbation method." *Numerical methods for partial differential equations* 25, no. 4 (2009): 833-842. https://doi.org/10.1002/num.20376
[7] Noorzad, Reza, A. Tahmasebi Poor, and Mehdi Omidvar. "Variational iteration method and homotopy-perturbation method for solving Burgers equation in fluid dynamics." *Journal of Applied Sciences* 8, no. 2 (2008): 369-373. https://doi.org/10.3923/jas.2008.369.373
[8] Mohyud-Din, Syed Tauseef, Muhammad Aslam Noor, and Khalida Inayat Noor. "Variational iteration method for Burgers' and coupled Burgers' equations using He's polynomials." *Zeitschrift für Naturforschung A* 65, no. 4 (2010): 263-267. https://doi.org/10.1515/zna-2010-0401
[9] Pandey, Kanti, Lajja Verma, and Amit K. Verma. "On a finite difference scheme for Burgers' equation." *Applied Mathematics and Computation* 215, no. 6 (2009): 2206-2214. https://doi.org/10.1016/j.amc.2009.08.018
[10] Rahman, Kaysar, Nurmamat Helil, and Rahmatjan Yimin. "Some new semi-implicit finite difference schemes for numerical solution of Burgers equation." In *2010 international conference on computer application and system modeling (ICCASM 2010)*, vol. 14, pp. V14-451. IEEE, 2010. https://doi.org/10.1109/ICCASM.2010.5622119
[11] Mohamed, Norhan A. "Solving one-and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes." *Arab Journal of Basic and Applied Sciences* 26, no. 1 (2019): 254-268. https://doi.org/10.1080/25765299.2019.1613746
[12] Chen, Yanli, and Tie Zhang. "A weak Galerkin finite element method for Burgers' equation." *Journal of Computational and Applied Mathematics* 348 (2019): 103-119. https://doi.org/10.1016/j.cam.2018.08.044
[13] Tamsir, Mohammad, Neeraj Dhiman, and Vineet K. Srivastava. "Extended modified cubic B-spline algorithm for nonlinear Burgers' equation." *Beni-Suef University journal of basic and applied sciences* 5, no. 3 (2016): 244-254. https://doi.org/10.1016/j.bjbas.2016.09.001
[14] Mittal, R. C., and RK2900117 Jain. "Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method." *Applied Mathematics and Computation* 218, no. 15 (2012): 7839-7855. https://doi.org/10.1016/j.amc.2012.01.059
[15] Aswin, V. S., Ashish Awasthi, and Mohammad Mehdi Rashidi. "A differential quadrature based numerical method for highly accurate solutions of Burgers' equation." *Numerical Methods for Partial Differential Equations* 33, no. 6 (2017): 2023-2042. https://doi.org/10.1002/num.22178
[16] Yang, Xiaojia, Yongbin Ge, and Lin Zhang. "A class of high-order compact difference schemes for solving the Burgers' equations." *Applied mathematics and computation* 358 (2019): 394-417. https://doi.org/10.1016/j.amc.2019.04.023
[17] Zainal, N. F. A., J. Sulaiman, A. Saudi, and N. A. M. Ali. "Semi-approximate Solution for Burgers' Equation Using SOR Iteration." In *Computational Science and Technology: 7th ICCST 2020, Pattaya, Thailand, 29–30 August, 2020*, pp. 475-486. Springer Singapore, 2021. https://doi.org/10.1007/978-981-33-4069-5_39
[18] Zainal, N. F. A., J. Sulaiman, and M. U. Alibubin. "Application of SOR iteration with nonlocal arithmetic discretization scheme for solving burger's equation." In *AIP Conference Proceedings*, vol. 2013, no. 1. AIP Publishing, 2018. https://doi.org/10.1063/1.5054234
[19] Öziş, Turgut, and Yaşar Aslan. "The semi-approximate approach for solving Burgers' equation with high Reynolds number." *Applied Mathematics and Computation* 163, no. 1 (2005): 131-145. https://doi.org/10.1016/j.amc.2004.01.032

[20] Evans, D. J. "Group explicit iterative methods for solving large linear systems." *International Journal of Computer Mathematics* 17, no. 1 (1985): 81-108. https://doi.org/10.1080/00207168508803452

[21] Zainal, N. F. A., J. Sulaiman, and M. U. Alibubin. "Application of four-point EGSOR iteration with nonlocal arithmetic mean discretization scheme for solving Burger's equation." In *Journal of Physics: Conference Series*, vol. 1358, no. 1, p. 012051. IOP Publishing, 2019. https://doi.org/10.1088/1742-6596/1358/1/012051

[22] Mohamad, N. S., and J. Sulaiman. "The piecewise polynomial collocation method for the solution of Fredholm equation of second kind by SOR iteration." In *AIP Conference Proceedings*, vol. 2013, no. 1. AIP Publishing, 2018. https://doi.org/10.1063/1.5054236

[23] Rahman, R., N. A. M. Ali, J. Sulaiman, and F. A. Muhiddin. "Caputo's Finite Difference Solution of Fractional Two-Point BVPs Using AGE Iteration." In *Journal of Physics: Conference Series*, vol. 1123, no. 1, p. 012044. IOP Publishing, 2018. https://doi.org/10.1088/1742-6596/1123/1/012044

[24] Alibubin, Mohd Usran, Jumat Sulaiman, Fatihah Anas Muhiddin, and Andang Sunarto. "Implementation of the KSOR Method for Solving One-Dimensional Time-Fractional Parabolic Partial Differential Equations with the Caputo Finite Difference Scheme Title of Manuscript." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 48, no. 1 (2024): 168-179. https://doi.org/10.37934/araset.48.1.168179

[25] Arora, Geeta, and Varun Joshi. "A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions." *Alexandria engineering journal* 57, no. 2 (2018): 1087-1098. https://doi.org/10.1016/j.aej.2017.02.017

[26] Raslan, Kamal R. "A collocation solution for Burgers equation using quadratic B-spline finite elements." *International journal of computer mathematics* 80, no. 7 (2003): 931-938. https://doi.org/10.1080/0020716031000079554

[27] Khater, A. H., R. S. Temsah, and MM2474624 Hassan. "A Chebyshev spectral collocation method for solving Burgers'-type equations." *Journal of computational and applied mathematics* 222, no. 2 (2008): 333-350. https://doi.org/10.1016/j.cam.2007.11.007

[28] Soliman, AA2206122. "The modified extended tanh-function method for solving Burgers-type equations." *Physica A: Statistical Mechanics and its Applications* 361, no. 2 (2006): 394-404. https://doi.org/10.1016/j.physa.2005.07.008

[29] Fu, Fangyan, Jiao Li, Jun Lin, Yanjin Guan, Fuzheng Gao, Cunsheng Zhang, and Liang Chen. "Moving least squares particle hydrodynamics method for Burgers' equation." *Applied Mathematics and Computation* 356 (2019): 362-378. https://doi.org/10.1016/j.amc.2019.03.040

[30] Cecchi, M. Morandi, R. Nociforo, and P. Patuzzo Grego. "Space-time finite elements numerical solutions of Burgers Problems." *Le Matematiche* 51, no. 1 (1996): 43-57.