Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.034370
Article

Stable Computer Method for Solving Initial Value Problems with Engineering Applications

Mudassir Shams[1], Nasreen Kausar[2], Ebru Ozbilge[3,*] and Alper Bulut[3]

[1]Department of Mathematics and Statistics, Riphah International University, I-14, Islamabad, 44000, Pakistan
[2]Department of Mathematics, Yildiz Technical University, Faculty of Arts and Science, Esenler, 34210, Istanbul, Turkey
[3]Department of Mathematics & Statistics, American University of the Middle East, Egaila, 54200, Kuwait
*Corresponding Author: Ebru Ozbilge. Email: ebru.kahveci@aum.edu.kw
Received: 15 July 2022; Accepted: 22 September 2022

Abstract: Engineering and applied mathematics disciplines that involve differential equations in general, and initial value problems in particular, include classical mechanics, thermodynamics, electromagnetism, and the general theory of relativity. A reliable, stable, efficient, and consistent numerical scheme is frequently required for modelling and simulation of a wide range of real-world problems using differential equations. In this study, the tangent slope is assumed to be the contra-harmonic mean, in which the arithmetic mean is used as a correction instead of Euler's method to improve the efficiency of the improved Euler's technique for solving ordinary differential equations with initial conditions. The stability, consistency, and efficiency of the system were evaluated, and the conclusions were supported by the presentation of numerical test applications in engineering. According to the stability analysis, the proposed method has a wider stability region than other well-known methods that are currently used in the literature for solving initial-value problems. To validate the rate convergence of the numerical technique, a few initial value problems of both scalar and vector valued types were examined. The proposed method, modified Euler explicit method, and other methods known in the literature have all been used to calculate the absolute maximum error, absolute error at the last grid point of the integration interval under consideration, and computational time in seconds to test the performance. The Lorentz system was used as an example to illustrate the validity of the solution provided by the newly developed method. The method is determined to be more reliable than the commonly existing methods with the same order of convergence, as mentioned in the literature for numerical calculations and visualization of the results produced by all the methods discussed, Mat Lab-R2011b has been used.

Keywords: Local truncation error; consistency; computational time; stability; lorentz system

1 Introduction

Differential equations include ordinary and partial derivatives. Initial value problems (IVPs) are formed by utilizing ordinary differential equations and initial conditions to provide the value of an unknown function at a specific place in the domain. Differential equations are commonly used in physics, chemistry, biology, and economics, as well as in many physical problems in science and engineering.

Numerous analytical techniques are available to solve differential equations. Analytical techniques [1,2], however, are sometimes unable are very difficult to provide closed form solutions of linear or nonlinear differential equations. Therefore, numerical methods are frequently used to solve highly nonlinear and linear differential equations [3]. Numerical techniques are important tools for quickly solving difficult problems using computer programming. To solve IVPs, engineers and researchers from a variety of disciplines [4–7] needed numerical schemes that were more efficient, stable, and had a larger region of convergence than other techniques that already existed and had the same order of convergence.

Numerous numerical algorithms for solving ordinary differential equations (ODEs) with initial value problems have been developed by many researchers. Many authors have attempted to solve initial value problems with great accuracy and speed up using various methods such as Euler's method, Heun's method, and others (see e.g., [8–12]). Euler's method calculates the slope of the function over the interval by using the line, tangent to the function at the beginning of the interval. In comparison, Heun's method considers tangent lines to the solution curve at both ends of the interval. Some have attempted to improve these precision approaches, while others have enhanced them for greater accuracy, stability, and consistency [13,14]. Some changes have been made to numerical algorithms from time to time to improve the performance based on our needs.

The main aim of this study was to construct a novel numerical technique for solving both linear and nonlinear initial value problems. Using consistency and stability analyses, we illustrate that the region of stability of our recently proposed numerical techniques is wider than the methods currently available in the literature. Compared with other techniques already in use, our newly developed method is more reliable, consistent, and efficient.

2 Proposed Method's Methodology Approach

Consider the following first-order differential equation with an initial value problem:

$$\frac{dy}{dx} = f(x, y);\ y(x_0) = y_0. \tag{1}$$

The most well-known and simple approach for solving Eq. (1) is the explicit Euler equation' with step length h, which is given by

$$y^p_{n+1} = y_n + hf(x_n, y_n). \tag{2}$$

The explicit Euler method [15] approximates the derivative of y_0 at $x = x_n$ using forward difference. Heun's method has a slope that is the average of the slope of the tangents to the integral curve at the endpoint of the interval, given by

$$y^c_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y^p_{n+1})\right]. \tag{3}$$

In [16], using Euler's method as a predictor, a second stage of the second-order method (MRK2M) based on the harmonic mean is presented as:

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}\left(\frac{k_1^2+k_2^2}{k_1+k_2}\right), \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_{n+1}, y^p{}_{n+1}). \end{cases} \tag{4}$$

Here, we propose modified version (MMRK2M) of Eq. (4) as:

$$\begin{cases} y_{n+1} = y_n + h\left(\frac{k_1^2+k_3^2}{k_1+k_3}\right), \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_{n+1}, y_{n+1}^p), \\ k_3 = f\left(x_n + h, y_n + h\left(\frac{k_1+k_2}{2}\right)\right). \end{cases} \tag{5}$$

3 Analysis of the Proposed Method's Stability

Dahlquist's test problem [17] can be used to analyze the stability of the proposed scheme.

$$\frac{dy}{dx} = \lambda y_n;\ y(x_0) = y_0,\ \lambda \in C. \tag{6}$$

Eq. (5) can also be written as:

$$y_{n+1} = y_n + h\left(\frac{(f(x_n, y_n))^2 + \left(f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)\right)^2}{f(x_n, y_n) + f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)}\right). \tag{7}$$

To test the stability of the suggested approach, substitute Eq. (6) with Eq. (5), and we have

$$y_{n+1} = y_n + h\left(\frac{(f(x_n, y_n))^2 + \left(f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)\right)^2}{f(x_n, y_n) + f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)}\right), \tag{8}$$

$$y_{n+1} = y_n + \frac{h^2\lambda^2 y_n^2 + \left(h\lambda y_n + h^2\lambda^2 y_n + \frac{1}{2}h^3\lambda^3 y_n\right)^2}{h\lambda y_n + h\lambda y_n + \frac{1}{2}h^3\lambda^3 y_n}, \tag{9}$$

$$y_{n+1} = \left(1 + \frac{h^2\lambda^2 + \left(h\lambda + h^2\lambda^2 + \frac{1}{2}h^3\lambda^3\right)^2}{h\lambda + h\lambda + \frac{1}{2}h^3\lambda^3}\right) y_n. \tag{10}$$

Putting $z = h\lambda$ in Eq. (10), we have

$$Q(z) = \frac{y_{n+1}}{y_n} = 1 + \frac{z^2 + \left(z + z^2 + \frac{1}{2}z^3\right)^2}{z + z^2 + \frac{1}{2}z^3}, \tag{11}$$

The stability region of numerical systems is depicted in Figs. 1 and 2. The stability polynomial is given by

$$|Q(z)| = \left|\frac{y_{n+1}}{y_n}\right| = \left|1 + \frac{z^2 + \left(z + z^2 + \frac{1}{2}z^3\right)^2}{z + z^2 + \frac{1}{2}z^3}\right| \leq 1. \tag{12}$$

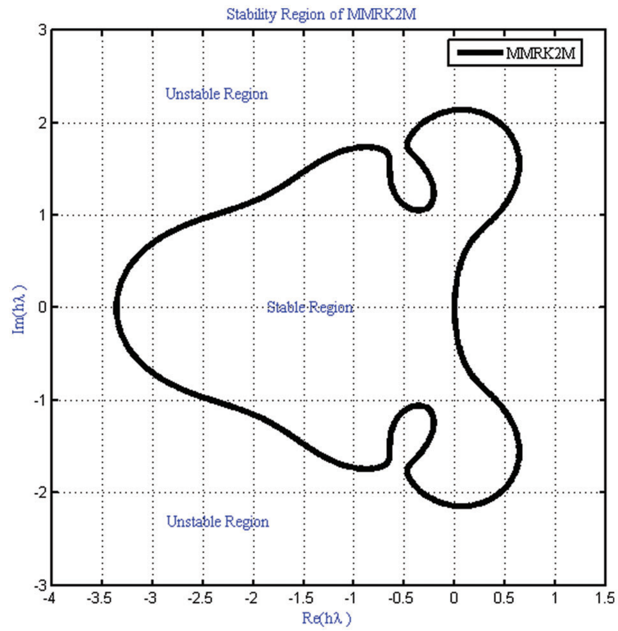

Figure 1: Stability region of MMRK2M is represented by white color where shading region shows instability region of the proposed method

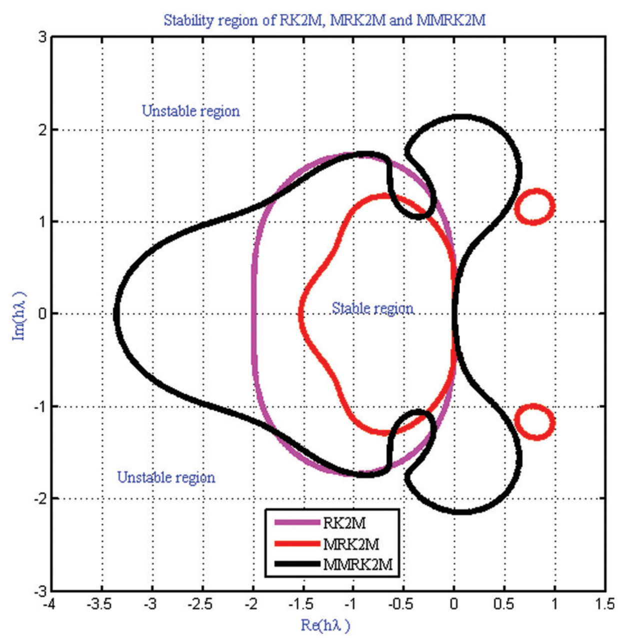

Figure 2: Comparison of the stability region of the proposed method MMRK2M (Black color) with MRK2 (Red color) and MRK2M (Cyan color) respectively

4 Consistency

To verify the consistency [17,18] of the proposed method, Eq. (5) can be written as:

$$y_{n+1} = y_n + h\vartheta(x_n, y_n, h), \tag{13}$$

and numerical scheme will be consistent if

$$\lim_{h\to 0} \vartheta(x_n, y_n, h) = f(x_n, y_n). \tag{14}$$

In this case, we used equation Eq. (13) to determine the consistency of the proposed technique as follows:

$$\begin{aligned}\lim_{h\to 0} \vartheta(x_n, y_n, h) &= \lim_{h\to 0}\left(\frac{(f(x_n, y_n))^2 + \left(f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)\right)^2}{f(x_n, y_n) + f\left(x_n + h, y_n + h\left(\frac{f(x_n,y_n)+f(x_{n+1},y_{n+1}^p)}{2}\right)\right)}\right),\\ &= f(x_n, y_n).\end{aligned} \tag{15}$$

Thus, proposed method is consistent.

5 Local Truncation Error

The local truncation error of a numerical technique is an estimate of the error introduced in a single iteration of the method, assuming that everything provided in the method is exact.

The Taylor series expansion of $y(x_{n+1})$ is given by:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + O(h^4), \tag{16}$$

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) + \frac{h^2}{2!}\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + \ldots \tag{17}$$

The Taylor series expansion of

$$k_1 = f(x_n, y_n), \tag{18}$$

$$\begin{aligned} k_2 = k_2 &= f(x_n + h, y_n + hf(x_n, y_n)),\\ &= f(x_n, y_n) + h\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) +\\ &\quad \frac{h^2}{2!}\left(\frac{\partial^2 f(x_n, y_n)}{\partial x^2} + 2f(x_n, y_n)\frac{\partial^2 f(x_n, y_n)}{\partial x \partial y} + (f(x_n, y_n))^2\frac{\partial^2 f(x_n, y_n)}{\partial^2 y}\right) + O(\,h^3),\end{aligned} \tag{19}$$

$$\frac{k_1 + k_2}{2} = f(x_n, y_n) + \frac{h}{2}\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + O(h^2), \tag{20}$$

$$k_3 = f(x_n, y_n) + h\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + O(h^2), \tag{21}$$

$$y_{n+1} = y_n + h\left(\frac{(f(x_n, y_n))^2 + \left(f(x_n, y_n) + h\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + O(h^2)\right)^2}{f(x_n, y_n) + f(x_n, y_n) + h\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + O(h^2)}\right), \tag{22}$$

$$y_{n+1} = y_n + hf(x_n, Y_n) + \frac{h^2}{2!}\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + O(h^3). \tag{23}$$

As a result of subtracting Eq. (23) or Eq. (22) from Eq. (17), the proposed numerical technique has a local truncation error of order $O(h^3)$; hence, the order of the proposed method is 2.

6 Numerical Outcomes

We execute the following iterative methods

1. Newly constructed method MMRK2M
2. Modified Euler method MRK2
3. Ram et al. method MRK2M

to solve IVPs and the system of IVPs [19], [20]. All numerical computations were performed using a Mat Lab 2011Rb laptop (Processor Intel® Core™ i3-3310m CPU@2.4GHz with 64-bit operating system) on Windows 8. In all Tables 1–7, the maximum absolute and local truncation errors are defined as $|y_n - y(x_n)|$.

Example 1:

Consider the IVP-I

$$\left\{\frac{dy(t)}{dt} = \frac{1}{4}\left(70 - 10\cos\left(\frac{\pi t}{12} - y(t)\right)\right); y(0) = 60; 0 \leq t \leq 12.\right.$$

Using MMRK2M, MRK2 and MRK2M for different values of step-size h, we find approximate solution of the non-linear IVP-I. The numerical results are presented in Tables 1 and 2 respectively. The exact and numerical approximate solutions are shown in Figs. 3 and 4.

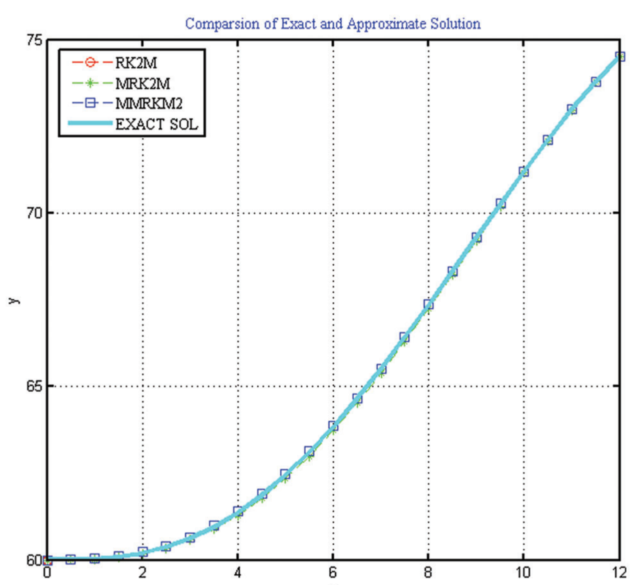

Figure 3: Comparison of exact and approximate solution of IVPs used in Example 1

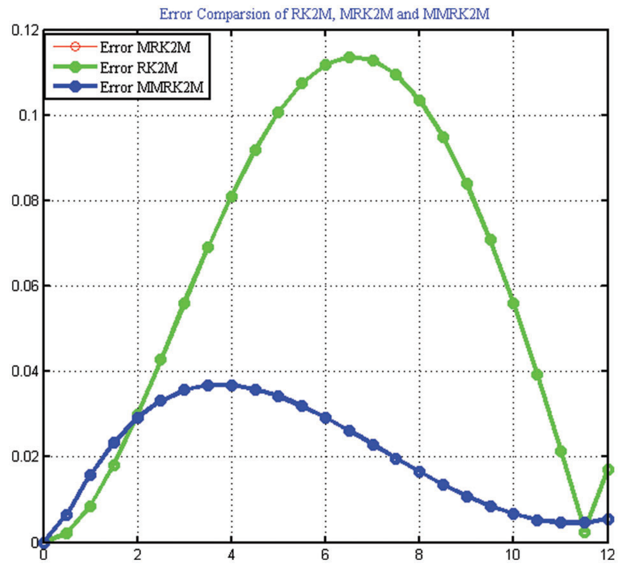

Figure 4: Error comparison of the numerical scheme MMRK2M, MRK2, MRK2M for solving IVPs used in Example 1

Table 1: Comparison of exact and approximate solution of system of IVPs used in Example 1

All numerical computation are done using 64-digit floating point arithmetic for finding exact and approximate solution using RK2M, MRK2M, MMRK2M for h = 0.5 and h = 0.1					
	RK2M		MMRK2M		Exact
t	y(t)	y(t)	y(t)	y(t)	y(t)
	h = 0.5	h = 0.1	h = 0.5	h = 0.1	
0	60	60	60	60	60
1	60.0182	60.02498	60.04256	60.02753	60.02676
2	60.16981	60.19366	60.22886	60.20092	60.19967
3	60.56893	60.61378	60.66059	60.62641	60.62495
4	61.28446	61.34943	61.4023	61.36701	61.36552
5	62.34187	62.42258	62.47667	62.44388	62.4425
6	63.72647	63.81614	63.8674	63.83942	63.83825
7	65.38819	65.47873	65.52397	65.50197	65.50105
8	67.248	67.33095	67.3679	67.35207	67.3514
9	69.20561	69.27293	69.30034	69.28998	69.28954
10	71.14802	71.19282	71.21048	71.20416	71.20389
11	72.95864	72.97571	72.98455	72.98015	72.97997
12	74.52627	74.51249	74.51472	74.50938	74.50917

Table 2: Comparison of the local error of the numerical methods for finding solution of the system of the IVPs used in Example 1

	RK2M		MMRK2M		MRK2M
t	y(t)	y(t)	y(t)	y(t)	y(t)
	h = 0.5	h = 0.1	h = 0.5	h = 0.1	h = 0.5
0	0	0	0	0	Div.
1	0.008567	0.001788	0.015798	0.000765	Div.
2	0.029856	0.006014	0.029188	0.001247	Div.
3	0.056015	0.011166	0.035645	0.001467	Div.
4	0.081063	0.016086	0.036784	0.001492	Div.
5	0.100629	0.019926	0.034166	0.001378	Div.
6	0.111778	0.02211	0.02915	0.001175	Div.
7	0.112862	0.022316	0.022919	0.000926	Div.
8	0.103399	0.020446	0.016504	0.00067	Div.
9	0.083929	0.016606	0.0108	0.000441	Div.
10	0.055867	0.011072	0.006589	0.000269	Div.
11	0.021322	0.004258	0.004586	0.000181	Div.
12	0.017099	0.003325	0.005552	0.000207	Div.

Table 3: Comparison of exact and approximate solution of system of IVPs used in Example 2

All numerical computation are done using 64-digit floating point arithmetic for finding exact and approximate solution using RK2M, MRK2M, MMRK2M								
	RK2M		MRK2M		MMRK2M		Exact	
t	$i_1(t)$	$i_2(t)$	$i_1(t)$	$i_2(t)$	$i_1(t)$	$i_2(t)$	Exact solution	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.1	4.7496	1.5902	4.8096	1.6055	4.7804	1.5903	4.7804	1.5903
0.2	5.98877	2.6643	6.0216	2.6932	6.0003	2.6730	6.0003	2.6730
0.3	6.418609	3.0857	6.4362	3.1030	6.4240	3.0909	6.4240	3.0909
0.4	6.575453	3.2421	6.5840	3.2507	6.5780	3.2447	6.5780	3.2447
0.5	6.633103	3.2997	6.6370	3.3037	6.6342	3.3009	6.6342	3.3009
0.6	6.654315	3.3209	6.6560	3.3227	6.6548	3.3214	6.6548	3.3214
0.7	6.662121	3.3287	6.6628	3.3295	6.6623	3.3290	6.6623	3.3290
0.8	6.664994	3.3316	6.6653	3.3319	6.6650	3.3317	6.6650	3.3317
0.9	6.666051	3.3327	6.6661	3.3328	6.6660	3.3327	6.6660	3.3327
1.0	6.66644	3.3331	6.6664	3.3331	6.6664	3.3331	6.6664	3.3331

Table 4: Comparison of the local error of the numerical methods for finding solution of the system of the IVPs used in Example 2

	MRK2	MRK2M	MMRK2M	MRK2	MRK2M	MMRK2M
	Er-F1	ER-F1	ER-F1	ER-F2	ER-F2	ER-F2
t	$i_1(t)$	$i_1(t)$	$i_1(t)$	$i_2(t)$	$i_2(t)$	$i_2(t)$
0.0	0.00	0.00	0.000	0.0000	0.000000	0.000000
0.1	0.059976	0.030743	0.029233	0.015247	0.000102	0.015145
0.2	0.032832	0.011586	0.021246	0.028923	0.008684	0.020239
0.3	0.017611	0.005463	0.012148	0.017348	0.005259	0.012091
0.4	0.008633	0.002576	0.006058	0.008618	0.002563	0.006055
0.5	0.003961	0.001164	0.002797	0.00396	0.001163	0.002797
0.6	0.001739	0.000508	0.001231	0.001739	0.000508	0.001231
0.7	0.000741	0.000216	0.000525	0.000741	0.000216	0.000525
0.8	0.000309	9.01E-05	0.000219	0.000309	9.01E-05	0.000219
0.9	0.000127	3.71E-05	8.96E-05	0.000127	3.71E-05	8.96E-05
1.0	5.13E-05	1.51E-05	3.62E-05	5.13E-05	1.51E-05	3.62E-05

Table 5: Comparison of exact and approximate solution of system of IVPs used in Example 3

All numerical computation are done using 64-digit floating point arithmetic for finding exact and approximate solution using RK2M, MRK2M, MMRK2M for h = 0.1

	RK2M			MRK2M			MMRK2M			Exact Solution		
t	x(t)	y(t)	z(t)	x(t)	y(t)	z(t)	x(t)	y(t)	z(t)	x(t)	y(t)	z(t)
0.0	44.57	37.37	35.02	44.58	37.36	35.02	45	35	35	45	35	35
0.1	44.30	39.50	35.10	44.31	39.49	35.11	44.58	37.36	35.02	44.58	37.36	35.02
0.2	44.15	41.42	35.23	44.16	41.41	35.23	44.31	39.49	35.11	44.31	39.49	35.11
0.3	44.10	43.16	35.39	44.11	43.14	35.39	44.16	41.41	35.23	44.16	41.41	35.23
0.4	44.14	44.74	35.57	44.15	44.72	35.58	44.11	43.15	35.39	44.11	43.15	35.39
0.5	44.24	46.17	35.78	44.25	46.15	35.79	44.15	44.72	35.58	44.15	44.72	35.58
0.6	44.40	47.48	36.01	44.41	47.45	36.02	44.25	46.15	35.79	44.25	46.15	35.79
0.7	44.60	48.67	36.25	44.61	48.65	36.26	44.41	47.46	36.01	44.41	47.46	36.01
0.8	44.84	49.77	36.51	44.85	49.75	36.51	44.61	48.66	36.26	44.61	48.66	36.26
0.9	45.11	50.79	36.76	45.12	50.76	36.77	44.85	49.75	36.51	44.85	49.75	36.51
1.0	44.57	37.37	35.02	44.58	37.36	35.02	45.11	50.77	36.77	45.11	50.77	36.77

Table 6: Comparison of the local error (ER) of the numerical methods for finding solution of the system of the IVPs used in Example 3

	RK2M	MRK2M	MMRK2M	RK2M	MRK2M	MMRK2M	RK2M	MRK2M	MMRK2M
	ER-F1	ER-F1	ER-F1	Er-F2	ER-F2	ER-F2	ER-F3	ER-F3	ER-F3
t	x(t)	y(t)	z(t)	x(t)	y(t)	z(t)	x(t)	y(t)	z(t)
0.0	0	0	0	0	0	0	0	0	0
0.1	0.003	0.003	0.0003	0.006	0.005	0.0003	0.001	0.001	0.0001
0.2	0.006	0.005	0.0009	0.01	0.01	0.001	0.002	0.002	5.5E-05
0.3	0.008	0.006	0.001	0.01	0.01	0.001	0.003	0.003	0.0002
0.4	0.009	0.007	0.002	0.01	0.01	0.002	0.003	0.003	0.0005
0.5	0.01	0.011	0.0004	0.02	0.017	0.003	0.004	0.003	0.0008
0.6	0.01	0.011	0.0004	0.02	0.017	0.005	0.004	0.003	0.001
0.7	0.01	0.010	0.001	0.02	0.01	0.006	0.004	0.002	0.001
0.8	0.01	0.009	0.002	0.02	0.01	0.006	0.004	0.002	0.001
0.9	0.01	0.008	0.002	0.02	0.01	0.007	0.003	0.002	0.001
1.0	0.01	0.007	0.003	0.02	0.01	0.008	0.003	0.001	0.001

Table 7: Shows the absolute maximum error (Max-Error) and central processing unit time (CPU-time) in seconds for different grid points

Absolut maximum error, computational CPU-time in seconds of RK2M, MRK2M and MMRK2M for solving IVP-I			
Numerical methods	MRK2	MRK2M	MMRK2M
Max-error (h = 0.5)	1.13E-02	2.11E-02	0.12E-03
Max-error (h = 0.1)	4.53E-03	5.73E-03	1.44E-05
CPU-time	0.03254	0.02189	0.001215
Absolut maximum error, computational CPU-time in seconds of RK2M, MRK2M and MMRK2M for solving system of IVPs used in example 2			
Max-error (h = 0.1)	1.13E-02	2.11E-02	0.12E-03
CPU-time	1.120356	2.015167	1.001341
Absolut maximum error, computational CPU-time in seconds of RK2M, MRK2M and MMRK2M for solving system of IVPs used in example 3			
Max-error (h = 0.1)	1.13E-02	2.11E-02	0.12E-03
CPU-time	2.01534	3.01523	1.012357
Absolut maximum error, computational CPU-time in seconds of RK2M, MRK2M and MMRK2M for solving system of IVPs used in example 4			
Max-error (h = 0.1)	1.13E-02	2.11E-02	0.12E-03
CPU-time	0.01251	0.06874	0.001423

6.1 Applications in Engineering

Here, we discuss some engineering applications that contain highly nonlinear differential equations with information provided at the initial points.

Example 2: Application in Electrical Engineering

The differential equation system for the current $i_1(t)$ and $i_2(t)$ in the electrical network depicted in Fig. 5 is

$$\begin{cases} \frac{di_1(t)}{dt} = \frac{-(R_1+R_2)}{L_2} i_1(t) + \frac{R_2}{L_2} i_2(t) + \frac{E}{L_2}, \\ \frac{di_2(t)}{dt} = \frac{R_2}{L_1} i_1(t) - \frac{R_2}{L_1} i_2(t), \\ i_1(t_0) = 0, \\ i_2(t_0) = 0. \end{cases} \tag{24}$$

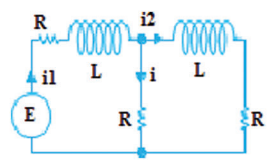

Figure 5: Shows the flow of current $i_1(t), i_2(t)$ through resistance R and inductance L along with source of voltage E

The proposed and existing numerical techniques are used to solve the system of differential equations to determine the behavior of the current $i_1(t)$ and $i_2(t)$ for $R_1 = R_2 = 10\Omega$, $E = 100V$,$L_1 = L_2 = L = 1h$ and $t_0 = 0$. By substituting the values of the resistance, voltage, and inductance, we obtain the following system of differential equations:

$$\begin{cases} \frac{di_1(t)}{dt} = -20i_1(t) + 10i_2(t) + 100, \\ \frac{di_2(t)}{dt} = 10i_1(t) - 20i_2(t), \\ i_1(0) = 0, \\ i_2(0) = 0. \end{cases} \tag{25}$$

The numerical results are presented in Tables 3 and 4 respectively. The exact and numerical approximate solutions are shown in Figs. 6 and 7.

Example 3: House Heating Problem (An Engineering Application)

Consider a conventional house that has an attic, a basement, and an insulated main floor. Insulation is present in the walls and ceilings of ordinary living spaces but not in the attic portion. The floor and walls of the basement were insulated by soil. Air gaps in the joists, a layer of flooring on the main level, and a layer of drywall all serve as insulation for the basement ceiling. Using the variables and Newton's cooling law [19], we examined how the temperatures in the three levels differ from one another. $z(t)$ = Humidity in the attic

$y(t)$ = Temperature in main living area

$x(t)$ = Basement Temperature

t = Time in hours.

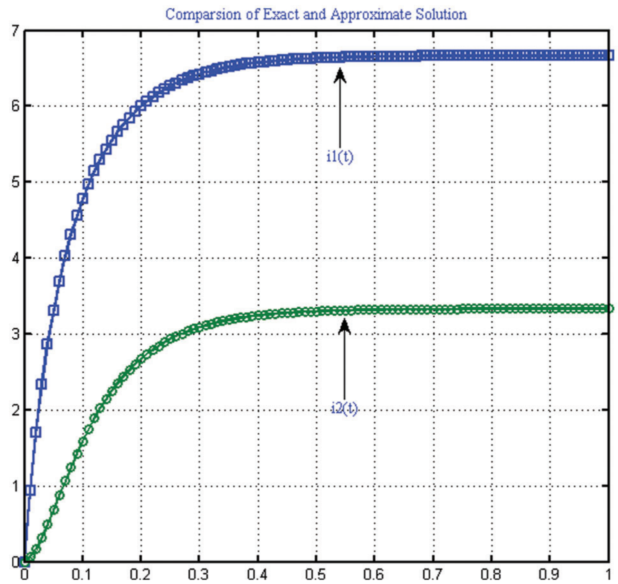

Figure 6: Comparison of exact and approximate solution of IVP used in Example 2

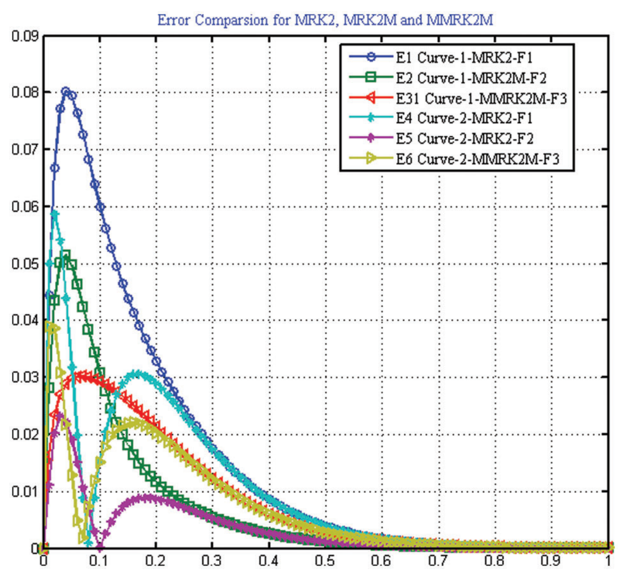

Figure 7: Error comparison of the numerical scheme MMRK2M, MRK2, MRK2M for solving IVP used in Example 2

Primary Data: Imagine its winter and the daily high is 35°F. Furthermore, a basement earth temperature of 45°F was predicted. The heating system was then gradually turned off. The initial values at midday ($t = 0$) are $x(t_0) = 45$, $y(t_0) = 35$, and $z(t_0) = 35$.

Portable heater: At noon, a small electric heater with a thermostat set to 100°F was turned on. When the heater is turned on, it raises the temperature by 20°F per hour, so it will take some time (may never be!) to reach 100°F.

The attic walls and ceiling, main floor walls, basement walls and floor, and basement ceiling were all subjected to temperature rate treatment. Newton's cooling law gives rise to the positive cooling constants h_0, h_1, h_2, h_3, and h_4, as well as the equations

$$\begin{cases} \frac{dx(t)}{dt} = h_0(45 - x(t)) + h_1(y(t) - x(t)), \\ \frac{dy(t)}{dt} = h_1(x(t) - y(t)) + h_2(35 - y(t)) + h_3(z(t) - y(t)) + 20, \\ \frac{dz(t)}{dt} = h_3(y(t) - z(t)) + h_4(35 - z(t)), \\ x(t_0) = \alpha_0, \\ y(t_0) = \alpha_1, \\ z(t_0) = \alpha_2. \end{cases} \tag{26}$$

The insulation constants were defined as $h_0 = \frac{1}{2}$, $h_1 = \frac{1}{2}$, $h_2 = \frac{1}{4}$, $h_3 = \frac{1}{4}$ and $h_4 = \frac{1}{2}$ to indicate insulation quality. Reciprocal $\frac{1}{h}$ represents the time required in hours to exchange 63 percent of the temperature differential. The main floor, for example, takes 4h. The model is:

$$\begin{cases} \frac{dx(t)}{dt} = \frac{1}{2}(45 - x(t)) + \frac{1}{2}(y(t) - x(t)), \\ \frac{dy(t)}{dt} = \frac{1}{2}(x(t) - y(t)) + \frac{1}{4}(35 - y(t)) + \frac{1}{4}(z(t) - y(t)) + 20, \\ \frac{dz(t)}{dt} = \frac{1}{4}(y(t) - z(t)) + \frac{1}{2}(35 - z(t)), \\ x(t_0) = \alpha_0, \\ y(t_0) = \alpha_1, \\ z(t_0) = \alpha_2. \end{cases} \tag{27}$$

The homogeneous solution in vector form is provided by the formula in terms of constants a = (7 + 21)/8, b = (7 + 21)/8, c = 1 + 21, and d = 1 + 21 and arbitrary constants c_1, c_2, c_3.

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = c_1 e^{\frac{-t}{2}} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} + c_2 e^{\left(\frac{7-\sqrt{21}}{8}\right)t} \begin{pmatrix} 4 \\ -1 + \sqrt{21} \\ 2 \end{pmatrix} + c_3 e^{\left(-1-\sqrt{21}\right)t} \begin{pmatrix} 4 \\ -1 - \sqrt{21} \\ 2 \end{pmatrix}. \tag{28}$$

The equilibrium solution is a specific solution. The exact solution to Eq. (27) is shown in Fig. 8.

$$\begin{pmatrix} x_p(t) \\ y_p(t) \\ z_p(t) \end{pmatrix} = \begin{pmatrix} \frac{455}{8} \\ \frac{275}{4} \\ \frac{185}{4} \end{pmatrix}. \tag{29}$$

The temperatures of the three spaces are approximately equal to $x = 57$, $y = 69$, and $z = 46$ degrees Fahrenheit because the homogeneous solution has a limit of zero at infinity. Solving for c_1, c_2, and c_3 with the initial values of $x(0) = \alpha_0 = 45$, $y(0) = \alpha_1 = 35$, and $z(0) = \alpha_2 = 35$ yields specific information. $c_1 = \frac{-85}{24}$, $c_2 = \frac{-25}{24} - \frac{115\sqrt{21}}{168}$ and $c_3 = \frac{-25}{24} + \frac{115\sqrt{21}}{168}$ are solutions.

Underpowered heater: Each hour, 20°F is added to the main floor, but the heat escapes quickly, so it is approximately 51°F after one hour. The temperature was approximately 65°F after five hours. In this case, the heater is so poor that the main living space remains cold after several hours. The forced-air furnace worked similarly. The numerical results of the system of IVPs are presented in Tables 5 and 6 respectively.

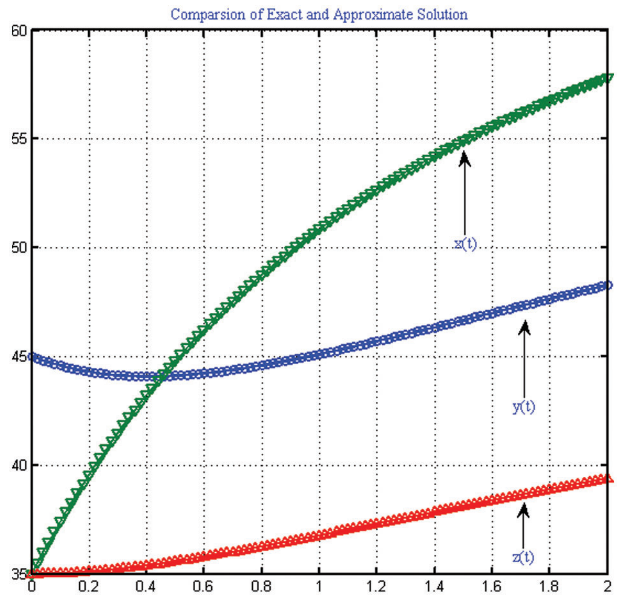

Figure 8: Comparison of exact and approximate solution of IVPs used in Example 3

Example 4: Solution of the Lorentz System

Consider three dimensional chaotic Lorentz system [20] as:

$$\begin{cases} \dfrac{dx(t)}{dt} = \sigma_1(y(t) - x(t)), \\ \dfrac{dy(t)}{dt} = \sigma_2 x(t) - y(t) - x(t); z(t); x(0) = 1; y(0) = 1; z(0) = 0, \\ \dfrac{dz(t)}{dt} = -\sigma_3\, z(t) + x(t)y(t). \end{cases} \tag{30}$$

MMRK2M, MRK2, and MRK2M methods are used to solve the Lorentz system for different values of parameter, that is, $\sigma_1 = -1.9, \sigma_2 = 120, \sigma_3 = 44$. The Computational CPU-time in seconds and the maximum error are listed given in Table 7. The exact solution of the Lorentz system is shown in Figs. 9–12.

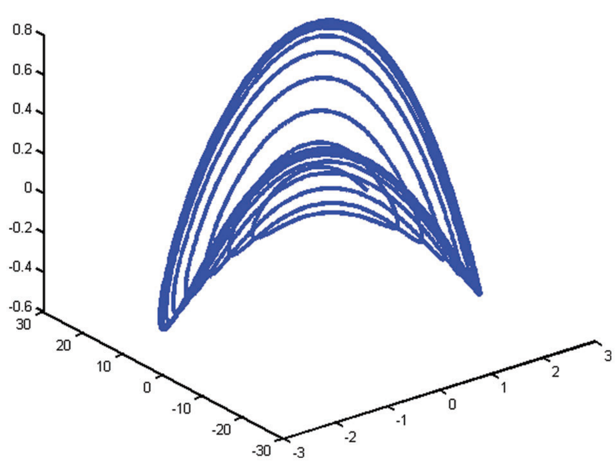

Figure 9: Solution of the Lorentz system in three dimensional space

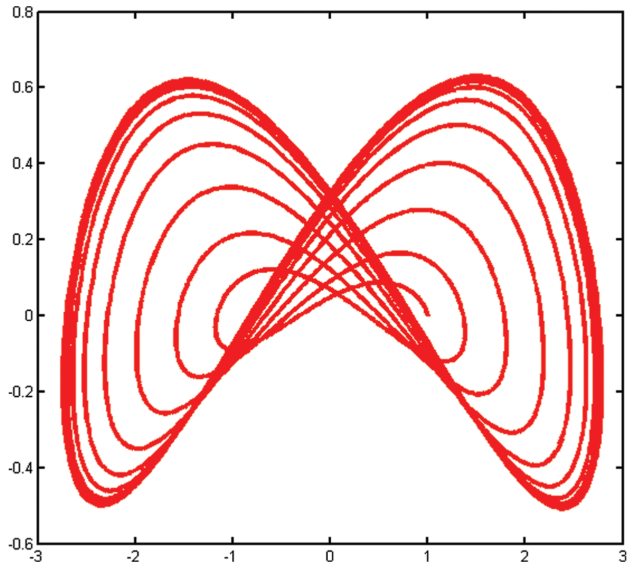

Figure 10: Phase diagram of the Lorentz system in xz-plane

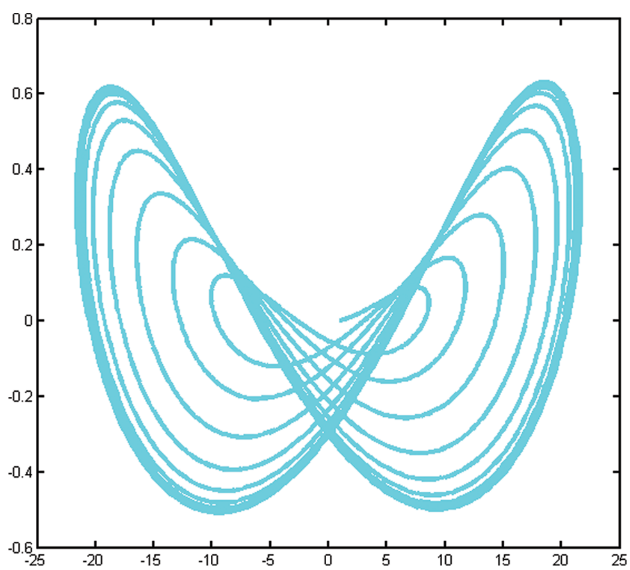

Figure 11: Phase diagram of the Lorentz system in yz-plane

6.2 Results and Discussion

Here, we discuss the superiority of the numerical method MMRK2M over existing methods in the literature in terms of stability, residual errors and computational time. Tables 1–7, show that the newly developed method, MMRK2M, is more stable and consistent than MRK2 and MRK2M. Figs. 1 and 2 shows the larger stability zone of MMRK2M compared to MRK2 and MRK2M. Figs. 3, 6, and 7 shows the exact and approximate solutions of MMRK2M, MRK2M, and MRK2, respectively, of the IVPs used in example 1–3. Figs. 4 and 7 show the maximum residual error computed by the numerical techniques MMRK2M, MRK2M, and MRK2 for solving IVPs. In terms of residual error, MMRK2M exhibited better convergence behavior than MRK2M and MRK2.

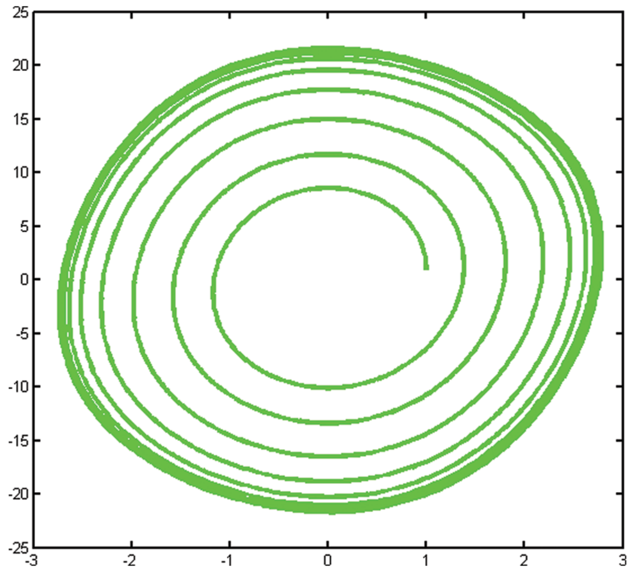

Figure 12: Projection of the Lorentz system in xy-plane

7 Conclusion

We developed a numerical approach for solving initial value problems with linear and nonlinear functions that are efficient, stable, reliable, and consistent. The numerical system exhibited consistency, second order local convergence, and absolute stability. The Examples from engineering show that the newly developed MMRK2M theoretical order of convergence matches computational outcomes. MMRK2M computes the maximum error significantly more accurately than MRK2 and MRK2M. In terms of the computation time measured in seconds, local truncation error and absolute maximum error, Tables 1–7 and Figs. 1–12 show that our proposed technique outperforms MRK2 and MRK2M.

Acknowledgement: The authors thank to the American University of the Middle East for their support.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. L. Burden and J. D. Faires, "Boundary-value problems for ordinary differential equations," in *Numerical Analysis*, 9th ed., vol. 1. Boston, MA 02210, USA: Belmont Thomson Brooks/Cole Press, pp. 1–863, 2005.

[2] G. Adomian and R. Rach, "Inversion of nonlinear stochastic operators," *Journal of Mathematical Analysis and Applications*, vol. 91, no. 1, pp. 39–46, 1983.

[3] S. D. Kim, X. Piao, D. H. Kim and P. Kim, "Convergence on error correction methods for solving initial value problems," *Journal of Computational and Applied Mathematics*, vol. 236, no. 17, pp. 4448–4461, 2012.

[4] P. A. Naik, M. Yavuz, S. Qureshi, J. Zu and S. Townley, "Modeling and analysis of COVID-19 epidemics with treatment in fractional derivatives using real data from Pakistan," *The European Physical Journal Plus*, vol. 135, no. 10, pp. 1–42, 2020.

[5] P. A. Naik, J. Zu and K. M. Owolabi, "Modeling the mechanics of viral kinetics under immune control during primary infection of HIV-1 with treatment in fractional order," *Physica A: Statistical Mechanics and its Applications*, vol. 545, no. 1, pp. 1–19, 2020.

[6] P. A. Naik, J. Zu and M. B. Ghori, “Modeling the effects of the contaminated environments on COVID-19 transmission in India,” *Results in Physics*, vol. 29, no. 19, pp. 1–11, 2021.

[7] P. A. Naik, K. M. Owolabi, J. Zu and M. U. D. Naik, “Modeling the transmission dynamics of COVID-19 pandemic in Caputo type fractional derivative,” *Journal of Multiscale Modelling*, vol. 12, no. 3, pp. 1–19, 2021.

[8] W. Gautschi, “Numerical integration of ODEs, based on trigonometric polynomials,” *Computer & Mathematics with Applications*, vol. 8, no. 3, pp. 231–239, 1982.

[9] A. Wambeeq, “Nonlinear methods in solving ordinary differential equations,” *Journal of Computational and Applied Mathematics*, vol. 2, no. 1, pp. 27–33, 1976.

[10] M. A. Aslam, “Accuracy analysis of numerical solutions of initial value problems (IVP) for ordinary differential equations (ODE),” *Applied Mathematics and Nonlinear Science*, vol. 3, no. 1, pp. 167–174, 2018.

[11] A. Ochoche, “Improving the improved modified Euler method for better performance on autonomous initial value problems,” *Leonardo Journal of Science*, vol. 12, pp. 57–66, 2008.

[12] H. Ramos, “A non-standard explicit integration scheme for initial value problems,” *Computer with Mathematics & Applications*, vol. 189, no. 1, pp. 710–718, 2007.

[13] M. A. Akanbi, “Third order Euler method for numerical solution of ordinary differential equations,” *ARPN Journal of Engineering and Applied Sciences*, vol. 5, pp. 42–49, 2010.

[14] N. A. Yahya, “Semi-implicit two-step hybrid method with FSAL property for solving second order differential equations,” *International Journal of Advanced and Applied Sciences*, vol. 4, no. 6, pp. 169–174, 2017.

[15] O. Abraham, “Improving the modified Euler method,” *Leonardo Journal of Science*, vol. 6, no. 10, pp. 1–8, 2007.

[16] T. Ram, M. A. Solangi and A. A. Sanghah, “A hybrid numerical method with greater efficiency for solving initial value problems,” *Mathematical Theory and Modeling*, vol. 10, no. 2, pp. 1–7, 2020.

[17] R. M. Corless, C. Y. Kaya and R. H. C. Moir, “Optimal residual and the Dahlquist test problem,” *Numerical Algorithms*, vol. 8, no. 4, pp. 1253–1274, 2019.

[18] Z. Jackiewicz, R. Renaut and A. Feldstein, “Two-step Runge-Kutta methods,” *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 1165–1182, 1991.

[19] S. Tracogna and B. Welfert, “Two-step Runge-Kutta: Theory and practice BIT,” *Numerical Mathematics*, vol. 40, no. 4, pp. 775–799, 2000.

[20] J. M. Li, Y. L. Wang and W. Zhang, “Numerical simulation of the Lorentz-Type chaotic system using bay centric Lagrange interpolation collocation method,” *Advance in Mathematical Physics*, vol. 2019, pp. 1–9, 2019.