

## ORIGINAL RESEARCH ARTICLE

## A Study on Some Numerical Methods for Simulating Mathematical Models of Ordinary Differential Equations

 Samuel Adamu<sup>1</sup>, Favour Anjikwi<sup>1</sup>, Hassan Bukar<sup>1\*</sup>, Ojo Olamiposi Aduroja<sup>2</sup> and Alhaji Tahir<sup>3</sup>
<sup>1</sup>Department of Mathematics, Nigerian Army University Biu, Borno State, Nigeria

<sup>2</sup>Department of Mathematics, University of Ilesa, Ilesa, Osun State, Nigeria

<sup>3</sup>Department of Mathematics, Modibbo Adama University Yola, Adamawa State, Nigeria

### ABSTRACT

This study presents a comparative analysis on some numerical methods for simulating mathematical models of ordinary differential equations, including Euler, Classical Runge-Kutta, Adomian Decomposition, Block, and Simulink. We examine each method's accuracy, stability, and consistency through a series of test cases. These methods are applied to simulate some selected mathematical models, and the results are shown in tables. The graph of each table is depicted in figures for discussion and comparative analysis. The results show that, while straightforward, the Euler method demonstrates significant limitations in accuracy compared to the Classical Runge-Kutta method, which provides reliable and precise results. The Adomian Decomposition Method solves the problems and yields results very close to the analytical solution, but the block method performs better due to its multistep approach. Simulink offers a more robust approach for modelling and simulation with visible and interpretable solutions for good understanding. This study revealed that numerical methods can easily be used to better simulate mathematical models that may not have analytical solutions and thus provide approximate solutions.

### ARTICLE HISTORY

Received January 05, 2025

Accepted April 25, 2025

Published May 04, 2025

### KEYWORDS

Decay, Growth, Model, Simulation, and Block-Methods



© The Author(s). This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 License [creativecommons.org](https://creativecommons.org/licenses/by-nc/4.0/)

### INTRODUCTION

Mathematical model is the application of methods to analyse complex real-world situations, which may consist of some elements abstracted from reality to make predictions of what might happen (Hairer *et al.*, 1993). Mathematical models are applicable across various fields of study, including physics, engineering, economics, biology, environmental science, medicine, social sciences, computer science, operations research, and finance (Aduroja & Adamu, 2025). These physical problems involve a number of separate elements linked together in some manner which lead to first order differential equations of the form,

$$u'(t) = f(t, u), u(t_0) = u_0, \quad (1)$$

where  $u'(t)$  and  $f(t, u)$  are of the form,

$$u'(t) = \begin{bmatrix} x' \\ \xi' \\ \vdots \\ z' \end{bmatrix} \text{ and } f(t, u) = \begin{bmatrix} f(t, x) \\ f(t, \xi) \\ \vdots \\ f(t, z) \end{bmatrix}, t_0 \in I, u_0 = (x_0, \xi_0, \dots, z_0) \in D$$

Numerical methods are essential for simulating mathematical models, particularly in complex systems with unattainable analytical solutions (Aduroja & Adamu, 2025).

Many real-world situations present a challenge in science, medicine, and engineering, which mathematical models handle to provide a solution via numerical methods to make an informed decision (Porgo *et al.*, 2018). This is necessary in order to accurately simulate real-world phenomena (Bretti, 2021).

On the other hand, first-order differential equations are solved using numerical methods, leading to two distinct types of solutions. The first type is a series solution of  $y$  in terms of  $x$ , which allows us to determine the value of  $y$  at a specific value of  $x$  through direct substitution in the series. Examples of this approach include the Taylor and Picard methods.

The second type of solution involves obtaining values of  $y$  at specified values of  $x$ . This category includes methods such as Euler, Runge-Kutta, Adam-Bashforth, and Milne, often referred to as step-by-step methods. These methods

**Correspondence:** Hassan Bukar. Department of Mathematics, Nigerian Army University Biu, Borno State, Nigeria. ✉ [bukar.hassan@naub.edu.ng](mailto:bukar.hassan@naub.edu.ng).

**How to cite:** Samuel, A., Favour, A., Hassan, B., Ojo, A. O., & Alhaji, T. (2025). A Study on Some Numerical Methods for Simulating Mathematical Models of Ordinary Differential Equations. *UMYU Scientifica*, 4(2), 7 – 15. <https://doi.org/10.56919/usci.2542.002>

work by calculating the values of  $y$  in short increments at equal intervals  $h$  of the independent variable  $x$ .

This study, therefore, focuses on these step-by-step methods for effectively simulating mathematical models.

Mathematical models are commonly simulated using ODE45, ODE15S or ODE23t. The ode45 works best for linear and non-stiff problems. ode15s and ode23t can handle nonlinear and stiff problems. Thus, we explore some numerical methods by examining the solutions' computational stability, accuracy and consistency. The performance of these methods are tested on different types of ordinary differential equations, highlighting the strengths and limitations of the methods.

Early in 1990, George Adomian introduced and developed the Adomian Decomposition Method (ADM). ADM, like the perturbation approach, it is a power series method and researchers in the field of initial and boundary value problems have been paying more attention to it lately (Rahman, 2007; Mariam, 2014). ADM is well known for its ability to solve nonlinear problems with closed form solution (Sunday, 2011).

Simulink graphically represents the model's simulation, which is perfect for broader application because it offers an easy-to-use interface for simulating dynamic systems (Aduroja et al., 2024).

Compared to traditional methods, block methods are self-starting, more accurate, more efficient (Omar & Adeyeye, 2016; Adamu, 2023; Adamu et al., 2019; Bukar et al., 2022). Modifying linear multi-step methods first addressed the Dahlquist barrier theorem by adding fractional step points in the derivation process to form a hybrid block method (Adamu et al., 2019; Adamu et al., 2020). This increased the result's accuracy and maintained good stability properties (Adamu et al., 2019; Adamu et al., 2020).

James et al. (2013) developed an order-6 half-step block approach of power series approximation. The method is used for SIR, growth, and mixing real-world situations. Using the Laguerre polynomial, Sunday et al. (2016) developed and used a quarter-step L-stable hybrid block approach of order 5 to handle a few real-world problems involving mixture, decay, growth, and SIR model difficulties. An implicit two-step obrechhoff type block approach of order 6 is constructed by Omar & Adeyeye (2016) and use the method to simulate SIR models and mixtures models.

Ashgi et al. (2021) simulate epidemiological models using Runge-Kutta and Euler methods and compare the behaviour of the results using the two methods. When comparing the two approaches' performance in solving the model, the results show that the Euler method produces a larger error, the classical Runge-Kutta method is more accurate but both of the methods are accurate.

Many scholars like (James et al., 2013; Omar & Adeyeye, 2016; Sunday, Yusuf & Andest, 2016; Adamu et al., 2019) have developed numerical methods for numerical approximations but there is still lack of adequate studies

that compare the performance of these methods in simulating mathematical models across different types of problems in ordinary differential equations. Thus, many researchers lack the guidance in selecting numerical methods to tackle problems in ordinary differential equations. Therefore, this study provides a clear picture of the strength and limitations of some numerical methods so that researchers will avoid trial and error when selecting methods for a specific problem or over relying on one method.

## METHODS

Five different methods are considered in this study for the numerical simulation of some mathematical models. The algorithm of these methods is as follows:

### Adomian Decomposition Method Algorithm

(i) Consider the following nonlinear equation,

$$\xi = f + L(\xi) + N(\xi) \tag{2}$$

where  $f$  is Lipschitz continuity,  $\xi$  is a function of  $x$ ,  $L$  and  $N$  are linear and nonlinear operators respectively.

(ii) Decompose the unknown function  $\xi(x)$  into an infinite series to give,

$$\xi(x) = \sum_{n=0}^{\infty} \xi_n(x), \tag{3}$$

or equivalently,

$$\xi = \xi_0 + \xi_1 + \xi_2 + \dots, \tag{4}$$

where the components  $\xi_n(x)$  will be determined recursively.

(iii) Decompose the nonlinear term  $N(\xi)$  into an infinite series of polynomials to give,

$$N(\xi) = \sum_{n=0}^{\infty} A_n(\xi_n), \tag{5}$$

where  $A_n$  are the Adomian polynomials defined as  $A_n(\xi_0, \xi_1, \dots, \xi_n)$ , given by,

$$A_n(\xi_0, \xi_1, \dots, \xi_n) = \frac{1}{n!} \frac{d^n}{d\lambda^n} [N(\sum_{k=0}^{\infty} \xi_k \lambda^k)]_{\lambda=0}, \tag{6}$$

$n = 0, 1, 2, 3, \dots$

(iv) Substitute (3) - (5) into (2) to give,

$$\sum_{n=0}^{\infty} \xi_n = f + \sum_{n=0}^{\infty} L(\xi_n) + \sum_{n=0}^{\infty} NA_n(\xi_0, \xi_1, \dots, \xi_n). \tag{7}$$

(v) Let  $L^{-1}$  be an inverse operator and apply to (2.6), to give,

$$L^{-1}(\sum_{n=0}^{\infty} \xi_n) = f + \sum_{n=0}^{\infty} \xi_n + L^{-1}(\sum_{n=0}^{\infty} NA_n(\xi_0, \xi_1, \dots, \xi_n)),$$

$$\sum_{n=0}^{\infty} \xi_{n+1} = f - L^{-1}(\sum_{n=0}^{\infty} \xi_n) + L^{-1}(\sum_{n=0}^{\infty} NA_n(\xi_0, \xi_1, \dots, \xi_n)), \tag{8}$$

or equivalently,

$$\xi_1 + \xi_2 + \dots = \xi_0 - L^{-1}(\xi_0 + \xi_1 + \xi_2 + \dots) - L^{-1}N(A_0 + A_1 + A_2 + \dots), \tag{9}$$

where  $\xi_0 = f$ .

(vi) Recursively, the relationship in (8) is given as,

$$\left. \begin{aligned} \xi_0 &= f \\ \xi_1 &= -L^{-1}(\xi_0) - L^{-1}(A_0(\xi_0)) \\ \xi_2 &= -L^{-1}(\xi_1) - L^{-1}(A_1(\xi_0, \xi_1)) \\ &\vdots \\ \xi_{n+1} &= -L^{-1}(\xi_n) - L^{-1}(A_n(\xi_0, \xi_1, \dots, \xi_n)) \end{aligned} \right\} \tag{10}$$

If  $\lim_{x \rightarrow \infty} (\xi_{n+1}(x))$  is convergent, then the solution follows.

Once the components  $\xi_n, n \geq 0$  are determined, the solution  $\xi$  in a series form follows immediately in a closed-form.

**Euler's Method**

The Euler's method approximates the solution by using a linear function based on the derivative at the current point, and it's given by,

$$\xi_{n+1} = \xi_n + hf(t_n, \xi_n), \tag{11}$$

where  $h$  is the step size and  $f(t_n, \xi_n)$  is the derivative.

**Classical Runge-Kutta Methods**

Classical Runge-Kutta Method (CRKM) is a one-step method with multi-stage evaluations of the derivative per step given as,

$$\xi_{n+1} = \xi_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{12}$$

$$k_1 = f(t_n, \xi_n),$$

$$k_2 = f\left(t_n + \frac{h}{2}, \xi_n + \frac{h}{2}\right),$$

$$k_3 = f\left(t_n + \frac{h}{2}, \xi_n + \frac{k_2}{2}\right),$$

$$k_4 = f(t_n + h, \xi_n + k_3),$$

**Block Method Algorithm**

(i) Consider the approximate solution,

$$\xi(x) = \sum_{n=0}^k \alpha_n x^n, \tag{13}$$

where  $x \in [a, b], a_n \in \mathbf{R}$  are the unknown parameters to be determined.

(ii) Evaluate the first and second derivative of equation (13) at  $x = x_{n+j}, j = 0, \dots, k$ , gives

$$XA = U, \tag{14}$$

where

$$A = [a_0, a_1, a_2, a_3, \dots, a_k], \quad k = r + s - 1.$$

$$U = [\xi_n, \xi_{n+1}, \dots, \xi_{n+r}, \xi'_n, \dots, \xi'_{n+s}]^T,$$

(iii) Impose the following conditions

$$\begin{aligned} \xi(x_{n+j}) &= \xi_{n+j}, \quad j = 0, 1, \dots, r \\ \xi'(x_{n+j}) &= f_{n+j}, \quad j = 0, 1, \dots, s' \end{aligned}$$

on equation (13), gives,

$$X = \begin{bmatrix} 1 & x_n & x_n^2 & \dots & x_n^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n+r} & x_{n+r}^2 & \dots & x_{n+r}^k \\ 0 & 1 & 2x_n & \dots & kx_n^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2x_{n+s} & \dots & kx_{n+s}^{k-1} \end{bmatrix}.$$

(iv) Solve (14) for the unknown parameters and substitute the results into (13) to give,

$$\xi_{n+t} = \sum_{j=0}^r \alpha_j(t) \xi_{n+j} + h \sum_{j=0}^s \beta_j(t) f_{n+j}, \tag{15}$$

where  $\alpha_j(t)$  and  $\beta_j(t)$  are polynomial of degree  $r + s - 1$  and  $t = \frac{x - x_n}{h}$ .

(v) Evaluate (15) at some selected grid and off-grid points to give,

$$A^{(1)}Y_{m+1} = A^{(0)}Y_m + hB^{(0)}F_m + hB^{(1)}F_{m+1}, \tag{16}$$

where

$$Y_{m+1} = [\xi_{n+1} \ \xi_{n+2} \ \dots \ \xi_{n+r}]^T, Y_m = [\xi_{n-1} \ \xi_{n-2} \ \dots \ \xi_n]^T, F_{m+1} = [f_{n+1} \ f_{n+2} \ \dots \ f_{n+s}]^T, F_m = [f_{n-1} \ f_{n-2} \ \dots \ f_n]^T,$$

$A^{(1)}, A^{(0)}, B^{(1)}$  and  $B^{(0)}$  are  $r \times r$  matrices. See (James et al., 2013; Omar & Adeyeye, 2016; Sunday, Yusuf

& Andest, 2016) for details of the block methods considered.

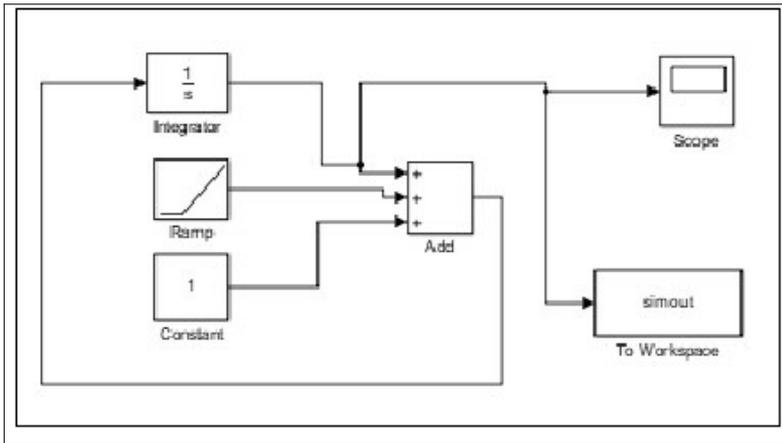
**MATLAB Simulink**

Simulink of ordinary differential equations are carryout by the following steps:

- (i) The highest state derivatives are identified.
- (ii) These states are integrated once or more to obtain all states.

- (iii) The highest states are computed with a simulation.
- (iv) The signals entering the summation are computed by means of gains multiplying known signals and inputs.

To demonstrate how the Simulink block simulate is constructed for first order differential equation  $\xi' = \xi + t + 1$ , the Simulink block is given as:



**Figure 1: MATLAB Simulink Block**

See (Aduroja, Adamu & Ajileye, 2024) for more details about Simulink.

**Areas of Comparison**

The effectiveness and accuracy of the numerical methods will be compared based on the following:

- (i) The type of equation to be solved.
- (ii) The numerical results i.e absolute error, consistence, convergence.
- (iii) The graphical representation of the numerical solutions.

**RESULTS**

In this subsection, well known Growth, Decay, Mixture and SIR models are simulated using the methods considered to illustrate the performance and accuracy of the methods. The numerical results are computed using MATLAB 2018a and compared. Let  $x_n(t)$  and  $x(t)$  be the approximate and numerical solutions, then the absolute error is given by  $|x_N(t) - x(t)|$ . The numerical results are presented in figure and tabular form.

Notation	Meaning
t	Point of evaluation for time
Exact	Exact Solution

SJ11 Sunday (2011)  
 OA16 Omar & Adeyeye (2016)  
 JE13 James et al., (2013)  
 SYA16 Sunday, Yusuf & Andest (2016)

**Example 1: Growth Model (Sunday, Yusuf & Andest, 2016)**

The growth rate of the bacterial culture growth model is proportionate to the current state. About 1000 strands of the bacteria are found in the culture after an hour, and 3000 strands are found four hours later. We considered  $\xi(t)$  the number of bacterial strands in the culture at time to compute the total number of strands of the bacteria accessible in the culture at time  $t : 0 \leq t \leq 1$ . The equation is modelled as:

$$\frac{d\xi}{dt} = 0.366\xi, \quad \xi(0) = 694, \quad h = 0.1, \quad t \in [0,1]$$

with analytical solution

$$\xi(t) = 694e^{0.366t}.$$

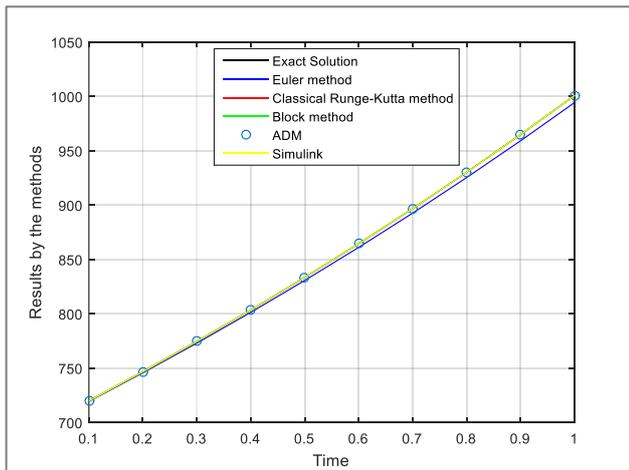
Using MATLAB 2024 the numerical results for example 1 are presented on Table 1 and 2, figure 2 and 3

**Table 1: Computed results for example 1**

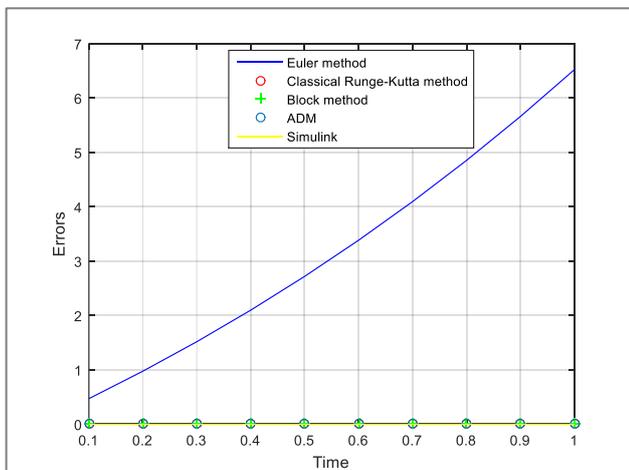
t	Exact	Block (SYA16)	Euler	CRKM	ADM	Simulink
0.1	719.87095	719.87095	719.40040	719.87095	719.87095	719.87089
0.2	746.70631	746.70631	745.73045	746.70631	746.70631	746.70621
0.3	774.54205	774.54205	773.02418	774.54205	774.54205	774.54188
0.4	803.41545	803.41545	801.31687	803.41545	803.41545	803.41522
0.5	833.36519	833.36519	830.64507	833.36519	833.36519	833.36489
0.6	864.43140	864.43140	861.04668	864.43140	864.43140	864.43103
0.7	896.65570	896.65570	892.56099	896.65570	896.65570	896.65525
0.8	930.08126	930.08126	925.22872	930.08125	930.08126	930.08072
0.9	964.75285	964.75285	959.09209	964.75285	964.75285	964.75222
1.0	1000.7169	1000.71693	994.19486	1000.71693	1000.7169	1000.71621

**Table 2: Error for example 1**

t	Block (SYA16)	Euler	CRKM	ADM	Simulink
0.1	0.00000e+00	4.7055e - 1	3. 8215e - 7	2.0000e - 14	5.2271e - 5
0.2	0.00000e+00	9.7586e - 1	7.9280e - 7	2.5000e - 13	1.0844e - 4
0.3	0.00000e+00	1.5179e + 0	1.2335e - 6	3.4000e - 13	1.6872e - 4
0.4	0.00000e+00	2.0986e + 0	1.7060e - 6	7.0000e - 14	2.3335e - 4
0.5	0.00000e+00	2.7201e + 0	2.2120e - 6	5.6000e - 13	3.0256e - 4
0.6	2.27374e-13	3.3847e + 0	2.7534e - 6	1.0800e - 12	3.7661e - 4
0.7	2.27374e-13	4.0947e + 0	3.3320e - 6	5.9100e - 12	4.5575e - 4
0.8	3.41061e-13	4.8525e + 0	3.9500e - 6	2.4840e - 11	5.4028e - 4
0.9	2.27374e-13	5.6608e + 0	4.6094e - 6	8.9120e - 11	6.3047e - 4
1.0	3.41061e-13	6.5221e + 0	5.3125e - 6	2.8320e - 10	7.2663e - 4



**Figure 2: Computed result for problem 1**



**Figure 3: Error for example 1**

**Example 2: Decay Model** (Sunday, 2011; Omar & Adeyeye, 2016)

<https://scientifica.umyu.edu.ng/>

The rate of decay of a particular radioactive material is related to its concentration, and there is a 100g block of this material visible. After 40 days, its bulk has reduced to 90g . Use the method to solve this problem for  $\forall t \in [0,1]$  and find an equation for the mass of the substance at any time. For the problem mentioned above, the differential equation is:

$$\frac{d\xi}{dt} = -\alpha\xi, \xi(0)=100, h = 0.1, t \in [0,1]$$

where the mass of the substance at any given time is represented by  $\xi$  . The variables specify the decay rate of this specific chemical  $t$  and  $\alpha$  . Thus, the theoretical outcome is

$$f(0) = 100 \text{ g}, t = 40 \text{ days}, f(40) = 90 \text{ g}$$

$$f(t) = f(0)e^{\alpha t},$$

$$90 = 100e^{40\alpha},$$

$$\alpha = \frac{\ln 9 - \ln 10}{40} = -0.0026$$

Hence the theoretical solution is

$$\xi(t) = 100e^{-0.0026t}.$$

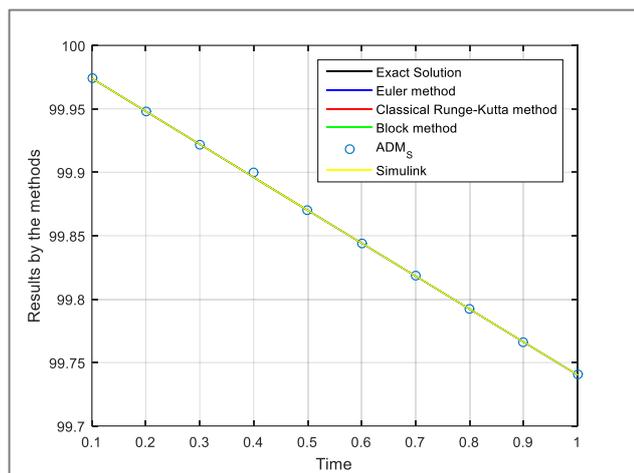
the expression for the mass of the substance at any time  $t$

**Table 3: Computed results for problem 2**

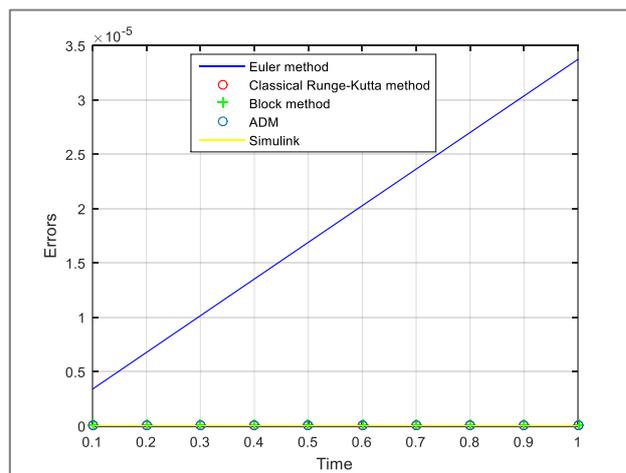
t	Exact	ADM (SJ11)	Block (JE13)	Euler	CRKM	Simulink
0.1	99.97400337	99.97400340	99.97400337	99.97400000	99.97400337	99.97400337
0.2	99.94801351	99.94801353	99.94801351	99.94800676	99.94801351	99.94801351
0.3	99.92203041	99.92203041	99.92203041	99.92202027	99.92203041	99.92203041
0.4	99.89605406	99.8960541	99.89605406	99.89604055	99.89605406	99.89605406
0.5	99.87008446	99.87008449	99.87008446	99.87006758	99.87008446	99.87008446
0.6	99.84412161	99.84412162	99.84412161	99.84410136	99.84412161	99.84412161
0.7	99.81816551	99.81816555	99.81816551	99.81814189	99.81816551	99.81816551
0.8	99.79221617	99.79221618	99.79221617	99.79218918	99.79221617	99.79221617
0.9	99.76627356	99.76627357	99.76627356	99.76624321	99.76627356	99.76627356
1.0	99.74033770	99.74033771	99.74033770	99.74030398	99.74033770	99.74033770

**Table 4: Error for problem 2**

t	ADM (SJ11)	Block (JE13)	Euler	CRKM	Simulink
0.1	2.0000e-8	0.000000e+00	3.3797e - 6	1.5e - 14	1.5000e - 14
0.2	1.0000e-8	1.421085e-14	6.7577e - 6	4.7000e - 14	4.7000e - 14
0.3	0.0000e+0	0.000000e+00	1.0134e - 5	4.0000e - 14	4.0000e - 14
0.4	0.0000e+0	0.000000e+00	1.3508e - 5	4.6e - 14	4.6000e - 14
0.5	3.0000e-8	1.421085e-14	1.6881e - 5	3.0000e - 14	1.3000e - 13
0.6	0.0000e+0	1.421085e-14	2.0252e - 5	3.3e - 14	6.7000e - 14
0.7	3.0000e-8	1.421085e-14	2.3621e - 5	6.1e - 14	1.6100e - 13
0.8	1.0000e-8	0.000000e+00	2.6988e - 5	3.0000e - 15	1.9700e - 13
0.9	0.0000e-0	0.000000e+00	3.0354e - 5	7.3e - 14	1.7300e - 13
1.0	0.0000e-0	0.000000e+00	3.3718e - 5	3.0e - 14	1.7000e - 13



**Figure 4: Computed result for problem 2**



**Figure 5: Error for example 2**

**Table 5: Computed results for problem 3**

t	Exact	ADM (SJ11)	Block (JE13)	Euler	CRKM	Simulink
0.1	020.201340	1020.201340	1020.201340	1020.000000	1020.201340	1020.201333
0.2	1040.810774	1040.801774	1040.810774	1040.400000	1040.810774	1040.810760
0.3	1061.836546	1061.836571	1061.836546	1061.208000	1061.836546	1061.836525
0.4	1083.287067	1083.287068	1083.287067	1082.432160	1083.287067	1083.287039
0.5	1105.170918	1105.170930	1105.170918	1104.080803	1105.170917	1105.170881
0.6	1105.170918	1127.496852	1127.496851	1126.162419	1127.496851	1127.496807
0.7	1150.273798	1150.273799	1150.273798	1148.685667	1150.273798	1150.273746
0.8	1173.510870	1173.510895	1173.510870	1171.659381	1173.510870	1173.510809
0.9	1197.217363	1197.217363	1197.217363	1195.092568	1197.217362	1197.217292
1.0	1221.402758	1221.402758	1221.402758	1218.994419	1221.402757	1221.402678

**Example 3:** Growth Model (Sunday, 2011; Omar & Adeyeye, 2016)

Let the differential equation be used to depict the growth of a particular microorganism.

$$\frac{d\xi}{dt} = \lambda \xi, \xi(0) = 1000, h = 0.1, t \in [0,1]$$

We make the assumption that the model expands unhindered and constantly. If a single organism produces

0.2 offspring on average per hour, how many of those microorganisms are there?

The above problem can be solved analytically to give:

$$\xi(t) = 1000e^{0.2t}.$$

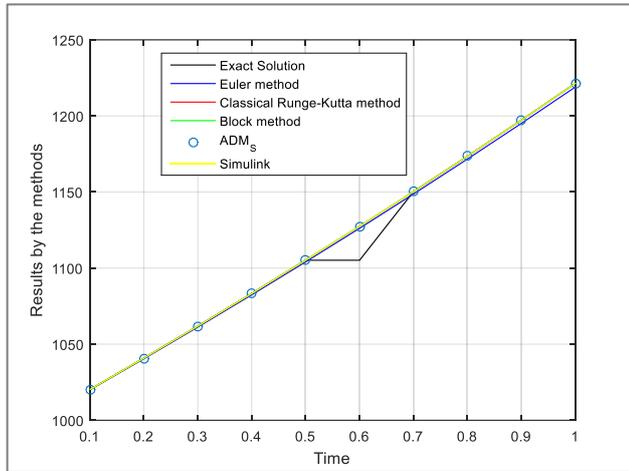


Figure 6: Computed result for problem 3

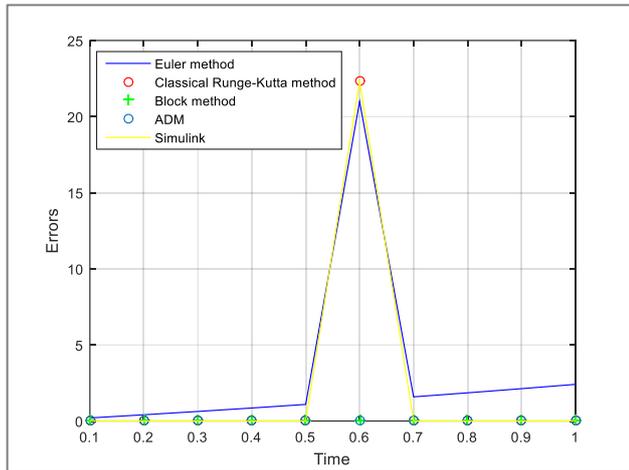


Figure 7: Error for example 3

**Example 4: SIR Model (Omar & Adeyeye, 2016)**

This model calculates the total number of individuals in a population who have a communicable infection over a given time period  $t$ . It deals with epidemic diseases. The development of susceptible-infected-recovered model is based on the fact that they include coupled equations regarding the number of individuals within the host  $S(t)$  whose susceptible disease level is appropriately high, the

number of individuals whose parasite  $I(t)$  level is high, and the total number of individuals who have recovered  $R(t)$ . The model is connected as follows:

$$\frac{dS}{dt} = \mu - \beta SI - \mu S$$

$$\frac{dI}{dt} = \beta SI - \gamma I - \mu I$$

$$\frac{dR}{dt} = \gamma I - \mu R$$

for  $\mu, \gamma$  and  $\beta$  are positive parameters.  $\xi$  is given as

$$\xi = S + I + R$$

Combining (3.1), (3.2) and (3.3)

$$\xi' = \frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = \mu(1 - \xi)$$

Therefore, the differential equation is

$$\xi' = \mu(1 - \xi), \xi(\alpha) = p$$

Taking  $\mu = \frac{1}{2}$  and attaching an initial condition

$$\xi(0) = -\frac{1}{2}, \text{ then}$$

$$\frac{d\xi}{dt} = \frac{1}{2}(1 - \xi), \xi(0) = \frac{1}{2}, h = 0.1, t \in [0, 1]$$

with analytical solution

$$\xi(t) = 1 - \frac{1}{2}e^{-t}.$$

**DISCUSSION**

Problem 1 is a growth model, and studying the results in Table 1 and errors in Table 2, show that all the methods considered give accurate results, but the Euler method has the largest error. The block method by Sunday, Yusuf & Andest (2016), and ADM perform very well with the least error, followed by CRKM and then Simulink. The block method, Euler method, CRKM, and Simulink are more consistent when compared with ADM, but the block method converges faster than all the other methods considered. All the problems in this study are solved with a consistent range.

The growth rate of the bacteria is shown in Figure 2, where the results obtained using various approaches closely resemble the analytical solution. Figure 3 displays the errors by the methods that are getting closer to zero while the Euler method advances slowly.

**Table 6: Error for problem 3**

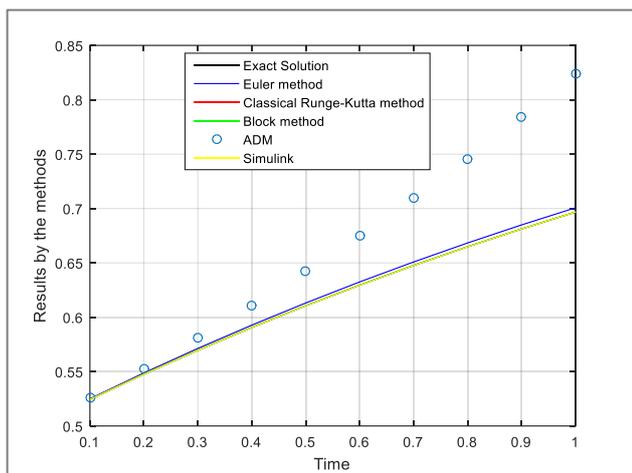
t	ADM (SJ11)	Block (JE13)	Euler	CRKM	Simulink
0.1	0.0000e+0	0.000000e+00	2.0134e - 1	2.675e - 8	6.6934e - 6
0.2	0.0000e+0	0.000000e+00	4.1077e - 1	5.4593e - 8	1.3657e - 5
0.3	2.4000e - 5	0.000000e+00	6.2855e - 1	8.3543e - 8	2.0900e - 5
0.4	0.0000e+0	2.273737e - 13	8.5491e - 1	1.1364e - 7	2.8429e - 5
0.5	1.2000e - 5	2.273737e - 13	1.0901e+0	1.4492e - 7	3.6254e - 5
0.6	0.0000e+0	2.273737e - 13	20.992e+0	22.326e+0	22.326e+0
0.7	0.0000e+0	2.273737e - 13	1.5881e+0	2.1117e - 7	5.2828e - 5
0.8	2.4000e - 5	0.000000e+00	1.8515e+0	2.4621e - 7	6.1594e - 5
0.9	0.0000e+0	0.000000e+00	2.1248e+0	2.8258e - 7	7.0693e - 5
1.0	0.0000e+0	0.000000e+00	2.4083e+0	3.2032e - 7	8.0135e - 5

**Table 7: Computed results for problem 4**

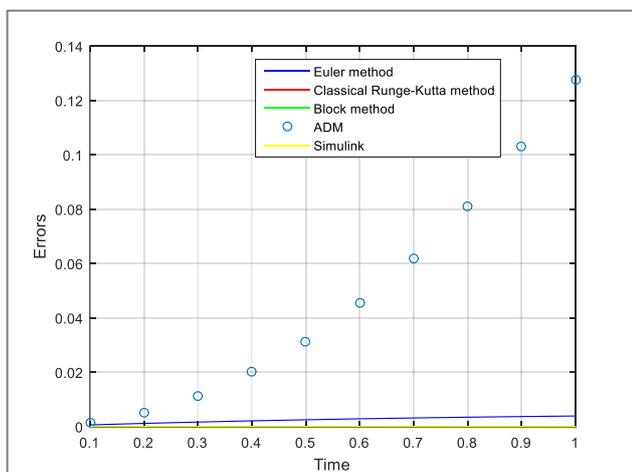
t	Exact	Block (OA16)	Euler	CRKM	ADM	Simulink
0.1	0.52438528	0.52438528	0.52500000	0.52438528	0.52563554	0.52438541
0.2	0.54758129	0.54758129	0.54875000	0.54758128	0.55258545	0.54758153
0.3	0.56964601	0.56964601	0.57131250	0.56964600	0.58091712	0.56964636
0.4	0.59063462	0.59063462	0.59274687	0.59063461	0.61070137	0.59063506
0.5	0.61059960	0.61059960	0.61310953	0.61059960	0.64201270	0.61060013
0.6	0.62959088	0.62959088	0.63245405	0.62959088	0.67492940	0.62959149
0.7	0.64765595	0.64765595	0.65083135	0.64765594	0.70953377	0.64765662
0.8	0.66483997	0.66483997	0.66828978	0.66483996	0.745912348	0.66484070
0.9	0.68118592	0.68118592	0.68487529	0.68118591	0.78415609	0.68118670
1.0	0.69673467	0.69673467	0.70063153	0.69673466	0.82436063	0.69673549

**Table 8: Error for problem 4**

t	Block (OA16)	Euler	CRKM	ADM	Simulink
0.1	3.826740e-14	6.1471e - 4	1.2913e - 9	1.2503e - 3	1.2892e - 7
0.2	7.484830e-14	1.1687e - 3	2.4567e - 9	5.0042e - 3	2.4526e - 7
0.3	1.058240e-13	1.6665e - 3	3.5053e - 9	1.1271e - 2	3.4995e - 7
0.4	1.354510e-13	2.1123e - 3	4.4458e - 9	2.0067e - 2	4.4384e - 7
0.5	1.601760e-13	2.5099e - 3	5.2862e - 9	3.1413e - 2	5.2774e - 7
0.6	1.838420e-13	2.8632e - 3	6.0340e - 9	4.5339e - 2	6.024e - 7
0.7	2.032250e-13	3.1754e - 3	6.6964e - 9	6.1878e - 2	6.6853e - 7
0.8	2.217960e-13	3.4498e - 3	7.2798e - 9	8.1072e - 2	7.2677e - 7
0.9	2.366300e-13	3.6894e - 3	7.7903e - 9	1.0297e - 1	7.7774e - 7
1.0	2.508620e-13	3.8969e - 3	8.2338e - 9	1.2763e - 1	8.2201e - 7



**Figure 8: Computed result for problem 4**



**Figure 9: Error for example 4**

Problem 2 is a Decay model, and as depicted in Table 3 and Table 4, the results clearly show that, the block

method, CRKM, and the Simulink perform better with the smallest error, while the Euler method has the largest error. ADM and block method by Sunday (2011) and James et al, (2013), respectively, give close-form solutions in some cases, but the results are not consistent. The Euler method, CRKM, and Simulink are more consistent.

Figure 4 presents the material's decay rate, where the results produced by the methods match the analytical solution. The errors in Figure 5 are getting closer to zero, but the Euler method is doing so slowly.

Problem 3 is a growth model. Observing the results in Table 5 and Table 6, it can be seen that ADM and block methods by Sunday (2011) and James et al. (2013) converge to the exact solution in some cases but are inconsistent. CRKM, Simulink, and Euler methods are more consistent, but CRKM and Simulink have better accuracy than the Euler method.

Figure 6 shows the microorganism's growth rate, and the results produced by the methods match the analytical solution. Figure 7 shows the errors which are approaching zero by most of the methods.

Problem 4 is a SIR model, and the results show that all the methods are consistent, but ADM and block method by Omar & Adeyeye (2016) are more accurate with the least error, while the Euler method has the largest error merging.

Figure 8 shows the SIR model's behavior, in which the results produced by the methods match the analytical solution. While Figure 9 shows the errors which is approach zero by most of the methods. But ADM exhibits an unusual behaviour with the SIR model, converging very slowly.

In all the problems considered, the Euler method shows consistency but generally gives the poorest result. ADAM and block methods are more consistent and accurate for all the problems. Simulink presents a reliable result, but the method cannot be improved on, block methods can be improved on.

## CONCLUSION

This study comparatively presents some numerical methods for simulating mathematical models in ordinary differential equations. The Euler method presented is one of the simplest methods for simulating mathematical models but has weak accuracy and is unstable for stiff problems. Classical Runge-Kutta method and Adomian decomposition methods, powerfully simulate the models of the problems under consideration with higher accuracy and stability. But ADM perform weakly when simulating the SIR problem. Both CRKM and ADM are powerful methods for simulating mathematical models but a good understanding of the methods is required.

Simulink present a graphical interface for simulating dynamic systems, which make it easier to simulate mathematical models. It demonstrates greater accuracy and consistency. Block method simulates the equations' systems simultaneously over time with high accuracy. It performs better than all the methods under consideration in terms of accuracy and convergence.

## REFERENCES

- Adamu, S. (2023). Numerical solution of optimal control problems using block method. *Electronic Journal of Mathematical Analysis and Applications*, 11(2), 1–12. <http://ejmaa.journals.ekb.eg>. [Crossref]
- Adamu, S., Bitrus, K., & Buhari, H. L. (2019). One step second derivative method using Chebyshev collocation point for the solution of ordinary differential equations. *Journal of the Nigerian Association of Mathematical Physics*, 51(May), 47–54. <http://nampjournals.org/publications-download/vol51/>
- Adamu, S., Danhausu, A. A., Stephen, L., & Williams, B. (2020). Two hybrid points block methods for the solution of initial value problems. *Journal of the Nigerian Association of Mathematical Physics*, 54(January issue), 7–12. <http://nampjournals.org/publications-download/vol54/>
- Aduroja, O. O., Adamu, S., & Ajileye, A. M. (2024). Radial basis neural network for the solution of optimal control problems via Simulink. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 15(2), 291–308. [Crossref]
- Ashgi, R., Pratama, M. A. A., & Purwani, S. (2021). Comparison of numerical simulation of epidemiological model between Euler method with 4th order Runge-Kutta method. *International Journal of Global Operations Research*, 2(1), 37–44. <http://www.iorajournal.org/index.php/ijgor/index>. [Crossref]
- Bretti, G. (2021). Differential models, numerical simulations and applications. *Axioms*, 10(4), 260. [Crossref]
- Bukar, H., Audu, B. A., & Garga, A. D. (2022). System of differential equation for two competing species. *Journal of Pure and Applied Science*, 22(4), 706–710. <https://www.bibliomed.org/?mno=110245>
- Hairer, E., Nørsett, S. P., & Wanner, G. (1993). *Solving ordinary differential equations I: Nonstiff problems*. Springer.
- James, A. A., Adesanya, A. O., Sunday, J., & Yakubu, D. G. (2013). Half-step continuous block method for the solution of modeled problems of ordinary differential equations. *American Journal of Computational Mathematics*, 3, 261–269. [Crossref]
- Mariam, B. H. A. (2014). *Iterative methods for the numerical solutions of boundary value problems* (Unpublished Ph.D. thesis). Faculty of the American University of Sharjah, College of Arts and Sciences, Sharjah, United Arab Emirates.
- Ojo, O. A., & Adamu, S. (2025). Numerical method for simulating the model of using insecticide for the optimal control of mosquitoes for the eradication of malaria. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 16(1), 15–24. [Crossref]
- Omar, Z., & Adeyeye, O. (2016). Numerical solution of first order initial value problems using a self-starting implicit two-step Obrechhoff-type block method. *Journal of Mathematics and Statistics*, 12(2), 127–134. [Crossref]
- Porgo, T. V., Norris, S. L., Salanti, G., Johnson, L. F., Simpson, J. A., Low, N., Egger, M., & Althaus, C. L. (2018). The use of mathematical modeling studies for evidence synthesis and guideline development: A glossary. *Research Synthesis Methods*, 10(1), 125–133. [Crossref]
- Rahman, M. (2007). *Integral equations and their applications*. WIT Press.
- Sunday, J. (2011). On the Adomian decomposition method for numerical solution of ODEs arising from a natural law of growth and decay. *The Pacific Journal of Science and Technology*, 12(1), 237–243.
- Sunday, J., Yusuf, D., & Andest, J. N. (2016). A quarter-step computational hybrid block method for first-order modeled differential equations using Laguerre polynomial. *Engineering Mathematics Letters*, 4, 1–17. <http://scik.org>. [Crossref]