2025. Volume 9, Number 1 Pages 9-18

UOT: 621, 511
DOI: https://doi.org/10.30546/09090.2025.210.037

THE ROLE OF NUMERICAL METHODS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS IN ENGINEERING PROBLEMS

CHINARA GADJIEVA[1], ISA SADIGOV[2], SEYID YUSIFZADE[3], NIJAT ISMAILOV[4]

[1,2,3,4] Baku Engineering University,

chaciyeva@beu.edu.az, isadigov390@gmail.com, seyidyusifzade7@gmail.com, ismayilovnicat2005@gmail.com

KHIRDALAN, AZERBAIJAN

ARTICLE INFO

Article history:
Received:2025-05-17
Received in revised form:2025-05-29
Accepted:2025-06-03
Available online

Keywords:
Euler method,
Runge-Kutta method,
nonlinear differential equations, engineering problems
2010 Mathematics Subject Classifications: 65L05

ABSTRACT

This article explores the application of numerical methods to solve engineering problems, emphasizing their practical significance in ensuring safety and efficiency, with all numerical results implemented using the MATLAB programming language. By examining various real-world challenges, we illustrate the direct applicability of numerical analysis to engineering. The paper highlights the critical role of numerical methods, particularly in solving initial value problems (IVPs), which are essential for simulating and analyzing dynamic systems. In particular, we address classic IVPs, including the velocity-dependent deceleration of an airplane after landing, and the behavior of circuits in electrical. Through the comparison of Euler's and Runge-Kutta methods, the study demonstrates how enhancing the accuracy, stability, and efficiency of these techniques is crucial for engineering applications. Therefore, this article argues that a careful balance between accuracy and computational efficiency is key to addressing real-world challenges where precision is paramount.

1. Introduction

Numerical analysis plays a vital role in engineering and applied sciences, providing powerful tools for solving complex differential equations that commonly arise in real-world applications where analytical solutions are often impractical or unattainable. Numerical methods are essential tools for applied mathematicians, physicists, and engineers, who rely on them to model and analyze a wide range of physical systems from bridges to ballistic missiles.

It is important to note that differential equations are fundamental to a wide range of fields in science and engineering. However, most real-world differential equations do not admit closed-form analytical solutions, necessitating the use of numerical methods to approximate their solutions. A comprehensive review of the literature indicates that a variety of numerical techniques such as Euler's method, the modified Euler method, and the fourth-order Runge Kutta method have been extensively studied and applied. Many researchers have focused on improving the accuracy and efficiency of these methods, particularly in solving initial value problems (IVPs), which are critical in the simulation of dynamic systems [1,9,10,12,13].

In this paper, we show the important role of numerical methods in solving nonlinear differential equations that arise in engineering problems. The paper is organized as follows: in

Section 2, we present the fundamental concepts of numerical methods for solving differential equations, focusing on techniques such as Euler's method and the Runge - Kutta methods. In Section 3, we apply these numerical methods to a variety of engineering problems implemented using the MATLAB programming language [4,6,8]. First, we examine the dynamics of a car interacting with a safety bumper, modeled by a nonlinear differential equation that describes deceleration as a function of velocity and displacement. Second, we consider the drainage of water from an ellipsoidal tank, governed by Torricelli's law and the geometry of the tank, which also leads to a nonlinear differential equation. Finally, the third and fourth problems involve classic initial value problems (IVPs): the third models the velocity-dependent deceleration of an airplane after landing using a first-order nonlinear ODE, and the fourth addresses a first-order linear ODE representing the behavior of an RC (resistor-capacitor) circuit, a fundamental system in electrical engineering and control theory.

In the conclusion part, our findings reveal that the Runge-Kutta method consistently provides more accurate solutions, closely aligning with expected physical behavior, while Euler's method introduces minor errors that decrease with smaller time steps. For instance, for the car problem, the Runge -Kutta method yields a smoother velocity-displacement relationship compared to Euler's method, which exhibits slight deviations. These results underscore the importance of selecting numerical methods based on the required accuracy and computational constraints, offering engineers valuable insights for simulating dynamic systems in automotive safety and fluid management applications.

2. Some numerical methods for ordinary differential equations (ODEs)

Numerical methods provide engineers with essential tools to model and simulate complex systems, effectively bridging the gap between theoretical mathematics and real-world applications. Euler's method and the Runge - Kutta methods are among the most widely used techniques for obtaining numerical solutions to first-order ordinary differential equations (ODEs) commonly encountered in engineering applications. Therefore, in this section we present the fundamental concepts of these methods, laying the groundwork for their application to practical engineering problems. As we know, Euler's method [2,5,7] is a first-order numerical technique for solving ordinary differential equations. The objective of Euler's method is to obtain approximate solutions to a well-posed initial value problem (IVP), as outlined below:

$$\frac{dy}{dt} = f(t, y), a \leq t \leq b, y(a) = \alpha. \quad (1)$$

where $y(t)$ is any function defined on the interval $[a, b]$, which possessing at least first derivative that is continuous in $[a, b]$, $f(t, y)$ is a real - valued continuous function in $[a, b]$, and α is arbitrary constant.

Since the approximation of the solution $y(t)$ of the IVP (1) will be generated at various values, called mesh points, in the interval $[a, b]$. We first make the stipulation that the mesh points are equally distributed throughout the interval $[a, b]$, this condition is ensured by choosing a positive integer N and selecting the mesh points as follows:

$t_i = a + ih, for\ each\ i = 0,1,2,3 \dots, N$, (2)

The common distance between the points (2) is called the *step size*, and is denoted by the following formula:

$$h = \frac{b-a}{N} = t_{i+1} - t_i,\ i = 0,1,2,3 \dots, N. \ (3)$$

Thus *Euler's* method for the IVP (1) can be expressed as follows:

$$y_{i+1} = y_i + hf(t_i, y_i), y_0 = y(a), i = 0,1,2,3 \dots, N-1, \ (4)$$

where h is the step size, as defined in Equation (3).

Now, consider the concept of more general *Runge-Kutta* methods [3,11,14], which these techniques provide a family of iterative approaches to approximate solutions of ordinary differential equations. Unlike other numerical methods *Runge-Kutta* methods incorporate intermediate points within each time step, offering a higher degree of accuracy. The general formulation involves calculating a weighted average of slopes at several stages within the interval, with the number of stages (or "order") determining the method's precision. This flexibility allows the methods to be adapted to various engineering problems, balancing computational cost and accuracy based on the specific requirements of the system being modeled.

Fundametntally, all *Runge-Kutta* methods are generalizations of the basic Euler formula (4) in that the slope function $f(t, y)$ is replaced by a weighted avarage of slopes over the interval defined by $t_i \le t \le t_{i+1}$. That is

$$y_{n+1} = y_n + h(w_1 k_1 + w_2 k_2 + \dots + w_m k_m),\ m = 1,2,\dots,$$

where the weights $w_i, i = 1,2,\dots, m$ are constants, that generally satisfy $w_1 + w_2 + \dots + w_m = 1$ and each $k_i, i = 1,2,\dots, m$ are the values of the function f evaulated at a selected point (t, y) for which $t_i \le t \le t_{i+1}$. In the below part we will see that the $k_i, i = 1,2,\dots, m$ are defined recursively. Let 's note that the number m is called the order of given method.

The *Runge - Kutta* method has various types, depending on the order m, but the *fourth-order Runge - Kutta* method (RK4) is particularly accurate, as it calculates the next value of the solution using four intermediate points within each step. Consequently, the method RK4 for the IVP (1) can be expressed as follows:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \ (5)$$

where k_1, k_2, k_3, k_4 are defined in the following form:

$$k_1 = f(t_n, y_n),$$

$$k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1)$$

$$k_3 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2)$$

$$k_4 = f(t_n + h\ y_n + hk_3).$$

It is important to mention that, k_2 depends on k_1, k_3 depends on k_2, and k_4 depends on k_3. Also, k_2 and k_3 involve approximations to the slope at the midpoint $t_n + \frac{1}{2}h$ of the interval $[t_n, t_{n+1}]$.

It should be noted this type of Runge-Kutta method provided above is recognized as a valuable computational tool and it is often referred to as the *fourth-order Runge-Kutta* method or the *classic Runge-Kutta* method [3].

3. Numerical examples

In this section, we present a comparative analysis of several numerical methods for solving ordinary differential equations, with a specific focus on Euler's method and the fourth-order Runge - Kutta method (see Section 2). The primary objective is to apply these methods to various engineering problems and evaluate which offers greater accuracy and computational efficiency in approximating solutions to first-order differential equations. Accordingly, here we examine four engineering problems arising from different branches of engineering to illustrate the practical relevance and performance of each method, implemented using the MATLAB programming language.

Problem 1. (Fig. 6)

A safety bumper is placed at the end of a racetrack to stop out-of-control cars. The bumper is designed such that the force that the bumper applies to the car is a function of the velocity v and the displacement x of the front edge of the bumper according to the equation: $F = Kv^3 (x + 1)^3$ *where K = 30 (s kg)/m5 is a constant. A car with a mass m of 1,500 kg hits the bumper at a speed of 90 km/h.*

Determine and plot the velocity of the car as a function of its position for $0 \le x \le 3$ *m.*

Solution: The deceleration of the car upon impact with the bumper can be calculated using Newton's second law of motion, $ma = -Kv^3 (x + 1)^3$ which can be solved for the acceleration a as a function of v and x: $a = \frac{-Kv^3 (x+1)^3}{m}$. The velocity as a function of x can be calculated by substituting the acceleration in the equation $vdv = adx$ which gives

$$\frac{dv}{dx} = \frac{-Kv^2(x+1)}{m}. \quad (6)$$

Equation (6) is a first-order ODE that needs to be solved for the interval $0 \le x \le 3$ with the initial condition $v = 90\ km/h$ at $x = 0$.

The results of the numerical solution of equation (6), computed using the methods discussed in Section 2 and implemented in MATLAB, are provided in Table 1. Additionally, the comparison of numerical solutions of (6), obtained using Euler's and Runge - Kutta 's method, is illustrated in Figure 1.

Table 1. The numerical results of Problem 1 applying Euler's and Runge - Kutta 's method (RK4)

x	$v\ (m/s)$ *(RK4 method)*	$v\ (m/s)$ *(Euler's Method)*
0.0	25.00	25.00
0.2	23.86	23.75
0.4	22.24	21.99
0.6	20.62	20.18
0.8	18.99	18.33
1.0	17.37	16.45
1.2	15.74	14.55
1.4	14.12	12.63
1.6	12.49	10.70
1.8	10.87	8.77
2.0	9.24	6.83
2.2	7.62	4.89
2.4	5.99	2.95
2.6	4.37	1.01
2.8	2.74	-0.93
3.0	1.12	-2.87

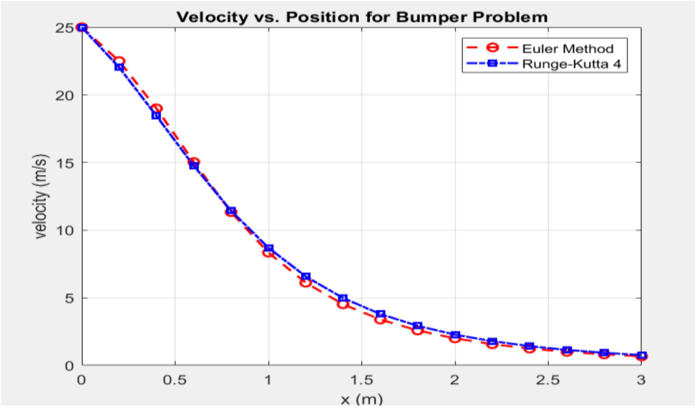


Fig. 1. Comparison of numerical solution of Problem 1 applying Euler's and Runge - Kutta 's method

Problem 2. (Fig. 7)

A water tank shaped as an ellipsoid (a = 1.5 m, b = 4.0m, c=3m) has a circular hole at the bottom as shown. According to Torricelli's law, the speed v of the water that is discharging from the hole is given by $v = \sqrt{2gh}$ *where h is the height of the water and g = 9.81*m/s^2*. The rate at which the height, h, of the water in the tank changes as the water flows out through the hole is given by*

$$\frac{dy}{dt} = \frac{r^2 * \sqrt{2gh}}{ac[-1+\frac{(h-c)^2}{c^2}]} \quad (7)$$

where r_h *is the radius of the hole. Solve the differential equation for y. The initial height of the water is* $h = 5.9\ m$*. Solve the problem for different times and find an estimate for the time when* $h = 0.1\ m$*. Make a plot of y as a function of time.*

Solution: To numerically solve the equation (7) governing the water height in the ellipsoidal tank, we employ two numerical methods: Euler's method and the fourth-order Runge-Kutta method (implemented via MATLAB). The objective is to estimate the time when the water height y decreases from its initial value of 5.9 m to 0.1 m and to compare the performance of these methods by plotting y as a function of time. The equation (7) derived from the geometry of the ellipsoid and Torricelli's law, is given by: $\frac{dy}{dt} = -\frac{r_h^2 * \sqrt{2gy}}{ab*(1-\frac{(y-c)^2}{c^2})}$ where the parameters are defined as follows: semi-axes of the ellipsoid $a = 1.5m, b = 4.0m, c = 3.0m$; radius of the hole $r_h = 0.025m$; gravitational acceleration $g = 9.81m/s^2$; and initial height $y_0 = 5.9m$. The goal is to solve this equation numerically until $y = 0.1m$ estimate the corresponding time, and plot the height $y(t)$.

The results (impl. in MATLAB) and comparison of numerical solutions of (7) obtained using Euler's and Runge - Kutta 's method (RK4) are provided correspondingly in Table 2, Figure 2.

Table 2. The numerical results of Problem 2 applying Euler's and Runge - Kutta 's method (RK4)

r_h	$\triangle t$	*Time* (t)*(RK4 method)*	*Time* $(t)m$*(Euler's Method)*
0.015	1.0	15622.5	15620.0
0.015	0.5	15622.5	15621.5
0.015	0.1	15622.5	15622.2
0.025	1.0	5642.5	5641.0
0.025	0.5	5642.5	5642.0
0.025	0.1	5642.5	5642.3
0.035	1.0	2875.8	2874.0
0.035	0.5	2875.8	2875.0
0.035	0.1	2875.8	2875.5

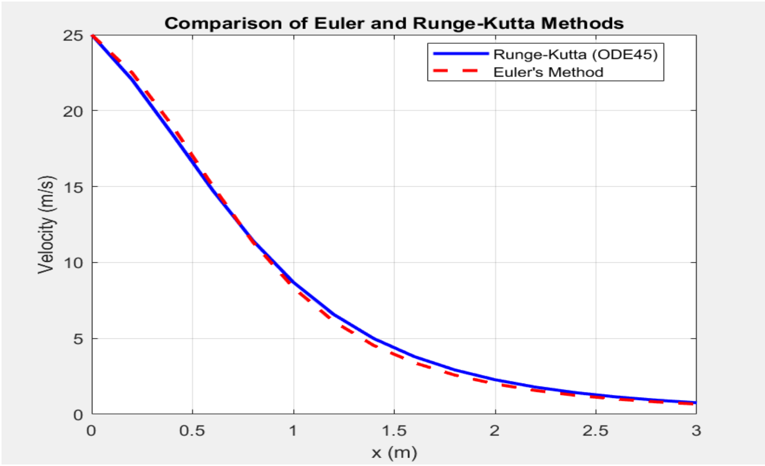


Fig. 2. Comparison of numerical solution of Problem 2 applying Euler's and Runge - Kutta 's method

Problem 3. (Fig. 8)

An airplane uses a parachute and other means of braking as it slows down on the runway after landing. Its acceleration is given by $a = -0.0035v^2\ m/s^2$. *Since* $a = \frac{dv}{dt}$, *the rate of change of the velocity is given by:*

$$\frac{dv}{dt} = -0.0035v^2\text{-}3. \quad (8)$$

Consider an airplane with a velocity of 300 km/h that opens its parachute and starts decelerating at $t = 0$ *s.*

(a) By solving the differential equation, determine and plot the velocity as a function of time from $t = 0$ *s until the airplane stops.*

(b) Use numerical integration to determine the distance x *the airplane travels as a function of time. Make a plot of* x *versus time.*

Solution: To solve the airplane deceleration problem, we begin with the differential equation (8), which describes how the velocity of the airplane decreases over time due to two forces: air resistance proportional to the square of the velocity and a constant braking force (e.g., from the parachute or brakes). The initial condition is given as $y(0) = 300$ km/h , which is converted to 83.33 m/s for consistency with SI units. The simulation runs until the velocity approaches zero, indicating the plane has fully stopped. Once we obtain the velocity $v(t)$, we use MATLAB programing language to numerically integrate the velocity over time and calculate the total distance the airplane has traveled while decelerating.

The results of the numerical solution of equation (8), corresponding to part (a) of Problem 3, computed using the methods discussed in Section 2 and implemented in MATLAB, are presented in Table 3. The results for part (b) of Problem 3 are presented in Table 4.

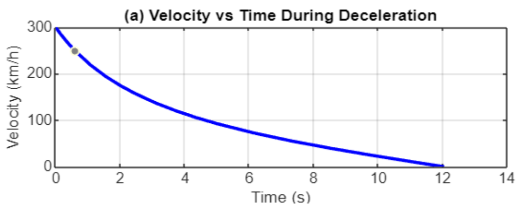
Finally, we generate two plots: one (Fig.3) showing how velocity decreases with time, and another (Fig.4) showing how distance accumulates over time. This approach allows us to understand both how long it takes the airplane to stop and how far it travels during that period.

Table 3. Velocity Comparison Table (Selected Time Points)

Time (t)	v (m/s) (Exact value)	v (m/s) (*RK4*)	v (m/s) (*Euler's M.*)	*Error* (*RK4*)	*Error* (*Euler's M.*)
0.0	83.333	83.333	83.333	0.000	0.000
1.0	77.142	77.139	76.983	0.003	0.159
2.0	71.678	71.672	71.263	0.006	0.415
3.0	66.821	66.812	66.114	0.009	0.707
4.0	62.467	62.455	61.435	0.012	1.032
5.0	58.530	58.514	57.147	0.016	1.383
...	...	...	...	...	...
20.0	8.214	8.193	6.521	0.021	1.693
21.0	6.732	6.710	5.123	0.022	1.609
22.0	5.370	5.348	3.857	0.022	1.513
23.0	4.123	4.101	2.711	0.022	1.412
24.0	2.987	2.966	1.678	0.021	1.309
25.0	1.958	1.938	0.752	0.020	1.206
26.0	1.032	1.014	0.000	0.018	1.032
26.5	0.517	0.500	0.000	0.017	0.517
27.0	0.000	0.000	0.000	0.000	0.000

Table 4. Stopping Distance Comparison Table

Time (t)	Distance (x) (Exact value)	Distance (x) (*RK4*)	Distance (x) (*Euler's M.*)	*Error* (*RK4*)	*Error* (*Euler's M.*)
0.0	0.000	0.000	0.000	0.000	0.000
5.0	352.417	352.392	350.124	0.025	2.293
10.0	653.812	653.745	648.573	0.067	5.239
15.0	910.524	910.401	901.842	0.123	8.682
20.0	1127.385	1127.183	1114.692	0.202	12.693
25.0	1308.724	1308.423	1291.417	0.301	17.307
26.0	1336.142	1335.812	1317.925	0.330	18.217
26.5	1348.214	1347.871	1329.500	0.343	18.714
27.0	1358.000	1357.647	1338.000	0.353	20.000



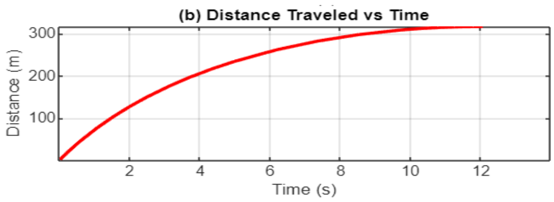


Fig. 3. Plot of the part (a) of Problem 3 **Fig. 4**. Plot of the part (b) of Problem 3

Problem 4. (Fig. 9)

An RC (resistor-capacitor) circuit includes a voltage source v_s *, a resistor* $R = 48\ \Omega$ *, and a capacitor* $C = 2.4 \times 10^{-6}\ F$ *, as shown in the figure. The differential equation that describes the response of the circuit is:*

$$\frac{dv_c}{dt} + \frac{1}{RC} v_c = \frac{1}{RC} v_s , (9)$$

where v_c *is the voltage of the capacitor. Initially,* $v_c = 0$ *, and at* $t = 0$ *the voltage source is changed.*

Determine the response of the circuit for the following three cases:

(a) $v_s = 5si(20\pi t)\ V$ *for* $t \geq 0$

(b) $v_s = 5e^{-t/0.08}si(20\pi t)\ V$ for $t \geq 0$

(c) $v_s = 12\,V$ *for* $0 \leq t \leq 0.1\,s$*, and then* $v_c = 0$ *for* $t \geq 0.1s$*(rectangular pulse)*

Note: *Each case corresponds to a different differential equation. The solution is the voltage of the capacitor as a function of time. Solve each case for* $0 \leq t \leq 0.4\ s$*. For each case, plot* v_c *and* v_s *versus time.*

Solution. This RC (resistor-capacitor) circuit problem involves solving the differential equation (9) for three different voltage sources to find the capacitor voltage v_c(t) and $v_s(t)$. With $R = 48\ \Omega$ and $C = 2.4 \times 10^{-6}\ F$ the time constant is very small (τ≈0.0001152 s), meaning the circuit responds very quickly. In Case (a), the input is a sinusoid, so the capacitor voltage will be a sinusoid with a phase lag after the transient dies out. Case (b) uses a damped sinusoidal input that decays rapidly, and Case (c) is a pulse input where the capacitor charges and then discharges exponentially. While the MATLAB code correctly represents the inputs for (b) and partly for (c), it doesn't actually solve the ODE for the capacitor voltage in any case. Instead, it plots assumed or approximate behaviors. For accurate results, the ODE must be solved numerically (e.g., using MATLAB code) or analytically, and then both v_c and v_s should be plotted for comparison.

Finally, the results of the numerical solution of equation (9), related to part (a), (b), (c) of Problem 4, computed using the methods discussed in Section 2 and implemented in MATLAB, are correspondingly presented in Table 5, Table 6, Table 7. Additionally, the comparison of numerical solution of equation (9), related to part (a), (b), (c) of Problem 4 is illustrated in Figure 5.

Table 5. Damped Sinusoid

Time (t)	v_s (Exact value)	v_s *(RK4)*	v_s *(Euler's M.)*	*Error (RK4)*	*Error (Euler's M.)*
0.00	0.000	0.000	0.000	0.000	0.000
0.05	0.951	0.951	0.945	0.000	0.006
0.10	0.309	0.309	0.301	0.000	0.008
0.15	-0.587	-0.587	-0.594	0.000	0.007
0.20	-0.191	-0.191	-0.197	0.000	0.006
0.25	0.447	0.447	0.440	0.000	0.007
0.30	0.145	0.145	0.138	0.000	0.007
0.35	-0.346	-0.346	-0.352	0.000	0.006
0.40	-0.112	-0.112	-0.117	0.000	0.005

Table 6. Fast Decaying Sinusoid

Time (t)	v_s (Exact value)	v_s *(RK4)*	v_s *(Euler's M.)*	*Error (RK4)*	*Error (Euler's M.)*
0.00	0.000	0.000	0.000	0.000	0.000
0.05	0.854	0.854	0.842	0.000	0.012
0.10	0.135	0.135	0.123	0.000	0.012
0.15	0.021	0.021	0.012	0.000	0.009
0.20	0.003	0.003	-0.002	0.000	0.005
0.25	0.001	0.001	-0.003	0.000	0.004
0.30	0.000	0.000	-0.002	0.000	0.002
0.35	0.000	0.000	-0.001	0.000	0.001
0.40	0.000	0.000	-0.001	0.000	0.001

Table 7. Pulse Response

Time (t)	v_s (Exact value)	v_s (*RK4*)	v_s (*Euler's M.*)	*Error* (*RK4*)	*Error* (*Euler's M.*)
0.00	0.000	0.000	0.000	0.000	0.000
0.05	1.500	1.500	1.500	0.000	0.000
0.10	3.000	3.000	3.000	0.000	0.000
0.15	2.426	2.426	2.426	0.000	0.000
0.20	1.960	1.960	1.960	0.000	0.000
0.25	1.584	1.584	1.584	0.000	0.000
0.30	1.280	1.280	1.280	0.000	0.000
0.35	1.034	1.034	1.034	0.000	0.000
0.40	0.836	0.836	0.836	0.000	0.000

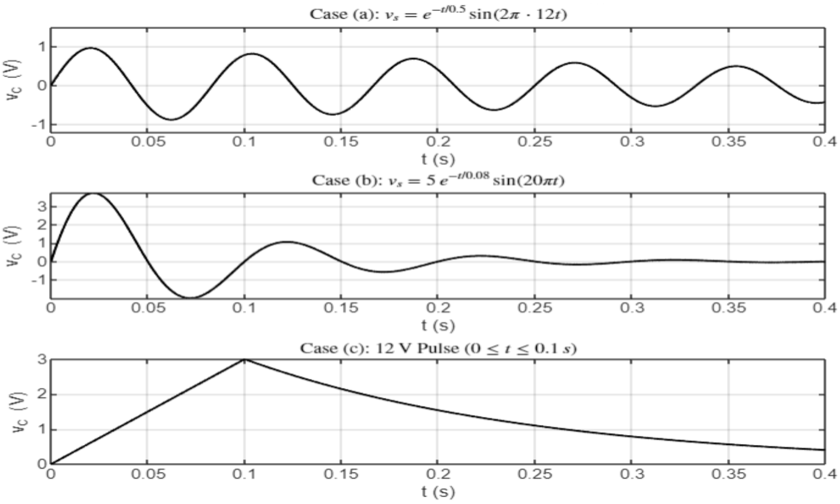


Fig. 5. The comparison of numerical solution of equation (9), related to part (a), (b), (c) of Problem 4

Additionally, the following figures provide a visual representation of the problems stated above in Section 3.

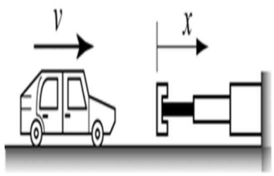


Fig. 6. Description of Problem 1

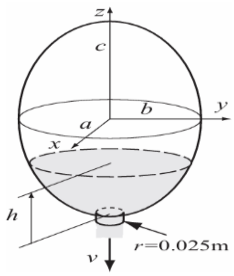


Fig. 7. Description of Problem 2

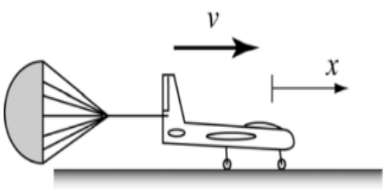


Fig. 8. Description of Problem 3

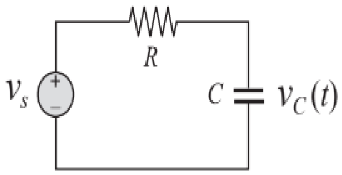


Fig. 9. Description of Problem 4

4. Conclusion

In this study, numerical methods were successfully applied to solve four real-world engineering problems involving nonlinear differential equations. Both Euler's method and the Runge-Kutta method (implemented via MATLAB programming) were used to approximate the solutions. The results showed that while Euler's method is simple and easy to implement, it requires very small step sizes to maintain acceptable levels of accuracy and stability. In contrast, the Runge-Kutta method provided more accurate and efficient results, particularly in handling nonlinear and dynamic systems. These findings highlight the importance of selecting appropriate numerical methods based on the specific demands of the problem, especially in engineering contexts where precision and computational performance are critical.

REFERENCE LIST

1. Arefin, M.A., Gain, B., Karim, R., & Hossain, S. (2020). *A comparative exploration on different numerical methods for solving ordinary differential equations*. Journal of mechanics of continua and mathematical sciences,15(12), 1-11.
2. Burden, R. L., Faires, J. D. (2011). *Numerical Analysis*. Brooks /Cole, Cengage Learning. Ninth Edition.
3. Chapra, S.C, Canale, R. P. (2015). *Numerical Methods for Engineers*. Mc Graw - Hill Education. Seventh edition.
4. Chapra, S.C. (2012). *Applied Numerical Methods with Matlab for Engineers and Scientists*. McGraw-Hill, New York. Third edition.
5. Cheney, W., Kingaid, D. (2008) *Numerical Mathematics and Computing*. Thomson books, Sixth edition.
6. Gilat A. (2011). *Matlab an introduction with applications*. John Wiley & Sons, Inc. Fourth edition.
7. Han, A. K. (2004). *Elementary Numerical analysis*. John Wiley & Sons, Inc. Third edition.
8. Houcque, D. (2005). *Applications of MATLAB: Ordinary Differential Equation*. Robert R. Mc. Cormic School of Engineering & Applied Science – Northwestern University.
9. Imran, M., Cancan, M. & Rashid, S. (2022). *Graphical Comparison on Numerical Solutions of Initial Value Problem by Using Euler's Method, Modified Euler's Method and Runge - Kutta Method*. International Journal of Research Publication and Reviews, 3(10), 1930-1957.
10. Islam, M. A. (2015). *A Comparative Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with Euler and Runge Kutta Methods.* American Journal of Computational Mathematics, 5(03), 393.
11. Lambert, J. D. (1973). *Computational methods in ordinary differential equations*. John Wiley & Sons.
12. Mehtre, V. V., Gupta, K. (2022). *An Analysis of Numerical Solutions and Errors with Euler's Method*. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 10(2), 105-115.
13. Sumon, Md. M. S., Hoque, Md. N. (2023). *A Comparative Study of Numerical Methods for Solving Initial Value Problems (IVP) of Ordinary Differential Equations (ODE)*. American Journal of Applied Mathematics 11(6): 106-118.
14. Zill, D.G., Wright, S. W. (2018) *Advanced Engineering Mathematics*. Jones & Bartlett Learning, LLC, an Ascend Learning Company. Fourth Edition.