International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(11s) 2026 pp. 1284-1293
https://doi.org/10.70917/ijcisim-2026-3864
Article

Comparative Study Between Euler Arithmetic and Euler Cube Scheme for Solving Ordinary Differential Equation in Resistor-Inductor (RL) Circuit Equation

Nooraida Samsudin[1], Zaris Izzati Mohd Yassin[1], Nurhafizah Moziyana Mohd Yusop[2]

[1] MILA University, Nilai, Malaysia.
[2] Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia.

Abstract: — Ordinary differential equations play a central role in modeling dynamic processes in electrical engineering and applied sciences. While Euler's method remains widely used due to its simplicity, its numerical accuracy and stability are strongly affected by step size selection. This paper presents a comparative study of the Euler Arithmetic (EA) and Euler Cube (EC) schemes for solving firstorder ordinary differential equations arising from resistor–inductor (RL) circuit models. The Euler Cube scheme extends the classical Euler formulation through cubemean averaging, enabling improved numerical stability without increasing computational complexity. Numerical experiments conducted on multiple RLbased test equations demonstrate that the EC scheme yields lower absolute error than EA at moderate step sizes, while both approaches retain firstorder convergence. At sufficiently small step sizes, the two methods converge to comparable accuracy. The results indicate that Euler Cube is a robust alternative for engineering simulations where accuracy and computational efficiency must be balanced.
Keywords: ODE, Euler Arithmetic (EA), Euler Cube (EC), resistor-inductor (RL) circuit, step size

1. INTRODUCTION

ORDINARY differential equations (ODEs) are best solved numerically, especially when analytical solutions are impractical or unavailable. Leonhard Euler developed Euler's technique in the eighteenth century [1], remains one of the most widely taught and applied techniques due to its simplicity and ease of implementation [2]. Differential equations are central to modeling phenomena in physics, engineering, biology, and economics. Although many ODEs do not have closed-form solutions, they can be approximated using numerical approaches. Euler's approach offers approximate solutions that are understandable to both practitioners and students by discretizing continuous problems into tiny stages. However, its effectiveness is highly dependent on the choice of step size, which directly influences accuracy, stability, and computational efficiency [3]. Ordinary differential equations (ODEs) provide a fundamental mathematical framework for describing timedependent systems in science and engineering. In electrical engineering, dynamic circuit behavior involving resistive and inductive elements is commonly formulated using firstorder differential equations derived from Kirchhoff's laws. Although analytical solutions can be obtained for simple circuit configurations, numerical methods are essential for practical analysis involving parametric variation, system complexity, or real time constraints. Among classical numerical approaches, Euler's method remains one of the most frequently introduced techniques due to its intuitive formulation and minimal computational overhead. However, its reliance on local linearization leads to substantial truncation error and stability limitations, particularly when moderate or large step sizes are employed. These limitations have motivated the development of enhanced Eulerbased schemes aimed at improving numerical performance while maintaining algorithmic simplicity.

Euler Arithmetic (EA) represents the classical scalar form of Euler's method, where the solution is updated iteratively for a single variable. This approach is efficient and straightforward, making it suitable for simple systems and introductory applications. However, when used to more complicated or multi-dimensional situations, EA has drawbacks since stability concerns become more noticeable and truncation errors build quickly. The Euler Cube (EC) scheme expands the approach into a multi-dimensional framework to overcome these difficulties, enabling the

simultaneous updating of several state variables [4]. By better capturing the interactions between linked variables, this expansion improves accuracy and stability, which is crucial for engineering applications like electrical circuits.

In addition to being theoretically interesting, the comparison between EA and EC schemes is also practically significant. For instance, first-order ODEs that explain the dynamics of current and voltage control resistor-inductor (RL) circuits in electrical engineering. Although EA is capable of accurately approximating the current response, EC offers a more thorough representation by adding factors like flux linkage and voltage between components. This broader formulation reduces numerical error and improves stability, especially when moderate step sizes are employed [5]. Therefore, choosing the right numerical techniques for circuit analysis and other applied sciences requires an awareness of the respective advantages of EA and EC schemes.

RL circuit is an electrical circuit made up of an inductor (L) and a resistor (R) linked either in parallel or series. Based on their configurations, a first-order ordinary differential equation can be used to describe them and examine each component over time [6]. Ohm's law states that the current flow in a circuit is inversely proportional to the voltage and resistance, or I=V/R. Both the flow and the degree of complexity will rise when two circuits are combined. [6], according to his research, it is relatively simple to analyze the circuit's equation using the Ordinary Differential Equation (ODE). Time-dependent components (capacitors and inductors) that rely on linearity (resistor, diode, and transistor) make up the electric circuit that ODE illustrates. One component of the electrical circuit that can be resolved using ODE is the Kirchoff and Ohm laws [7].

The purpose of this study is to conduct a systematic comparison between Euler Arithmetic and Euler Cube schemes across different test equations, including those derived from RL circuit models. By analyzing accuracy at varying step sizes, the research aims to provide insights into when each scheme is most effective. Ultimately, the findings will guide practitioners in choosing appropriate numerical strategies for solving ODEs in engineering and applied mathematics, balancing the trade-offs between computational cost and solution accuracy.

2. EXPERIMENTAL SETUP

Euler Arithmetic (EA) is a well-known technique for improving the Euler method. Euler Arithmetic (EA) which uses the concept of the average of two-point functions on the coordinates $y = \frac{\left(f\left(x_{n,}y_n\right) + f\left(x_{n+1,}y_{n+1}\right)\right)}{2}$.

New coordinates of the function of point R, $y_0 + \frac{h}{2}$, with gradient $\frac{2\left[f\left(x_{0,}y_0\right)f\left(x_{1,}y_1\right)\right]}{2}$ give an R value, $R = \left[x_0 + \frac{h}{2}, y_0 + \frac{h}{2}\left(\frac{2\left(f\left(x_0 y_0\right)f\left(x_{1,}y_1\right)\right)}{2}\right)\right]$. Hence, this is the Euler arithmetic formula as shown below.

$$y_{n+1} = y_n + hf\left[x_n + \frac{h}{2}, y_n + \frac{h}{2}\left(\frac{2\left(f\left(x_{n,}y_n\right)f\left(\left(x_n + h\right), y_n + hf\left(x_{n,}y_n\right)\right)\right)}{2}\right)\right]$$ with n= 0,1,2,... n.

The EC scheme replaces the point function (x_n,y_n). This replacement could have a positive impact in terms of accuracy on the EC scheme. With a gradient $\sqrt[3]{\frac{\left(x_n\right)^3 + \left(x_{n+1}\right)^3}{2}}, \sqrt[3]{\frac{\left(y_n\right)^3 + \left(y_{n+1}\right)^3}{2}}$ for

$R = y_0 + h/2$, new coordinate will be

$$R=\left(x_0+\frac{h}{2},y_0+\frac{h}{2}\left(\sqrt[3]{\frac{f(x_{0,}y_0)^3+f(x_{1,}y_1)^3}{2}}\right)\right).$$

Cube mean within two coordinate points of function

$$y_{n+1}=y_n+hf\left[x_n+\frac{h}{2},y_n+\frac{h}{2}\left(\sqrt[3]{\frac{(f(x_{n,}y_n))^3+(f(x_{n,}+h),y_n+hf(x_{n,}y_n))^3}{2}}\right)\right]$$

with n = 0, 1, 2,…n.

In this paper, to approximate solutions, two numerical schemes were implemented as mentioned before, Euler Arithmetic (EA) and Euler Cube (EC). EA represents the scalar form of Euler's method, updating the solution iteratively using where the step size is. EA is efficient and straightforward, making it suitable for simple RL circuits. However, truncation errors accumulate rapidly, and stability issues arise at larger step sizes.

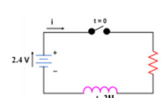
The EC scheme extends Euler's method into a multi-dimensional framework. Instead of updating a single variable, EC simultaneously updates multiple state variables, incorporating cube mean approximations to better capture interactions among coupled variables. This extension enhances accuracy and stability, particularly in engineering applications where multiple components interact dynamically.

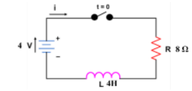
The resistor-inductor (RL) circuit consists of a resistor (R) and an inductor (L) driven by a voltage or current source. It is one of the first-stage circuits consisting of a resistor and an inductor. There are two ways to drive the first-stage circuit, namely without a direct current source and the second using direct current. The first-stage circuit without a direct source stores energy in the inductor. This energy will do work while flowing in the circuit and is gradually lost in the resistor. This study focuses only on the first-stage circuit without a direct current source. Testing is carried out according to normal, medium and difficult levels based on the input values (voltage, resistor and inductance) which are small, medium and large.

The RL equation is generated from the RL circuit which gives the values of inductance, resistance and voltage. Table 1 shows the sketch of the RL circuit and the values of inductance, resistance and voltage given to generate the RL circuit equation. Hukum Kirchoff shows with output response where .

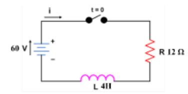
TABLE 1

RL SKETCH AND PROPERTIES

$\frac{dI}{dt}=\frac{V(t)-RI}{L}$	**RL Circuit Diagram**
$\frac{dI}{dt}=\frac{2.4-4I}{2}$	



$$\frac{dI}{dt} = \frac{4-8I}{4}$$



$$\frac{dI}{dt} = \frac{60-12I}{4}$$

Numerical experiments were conducted using RL circuit equations under varying input conditions (small, medium, and large values of voltage, resistance, and inductance). Step sizes tested to evaluate accuracy, stability, and computational efficiency. Comparative analysis focused on error convergence relative to analytical solutions, highlighting the trade-offs between EA's simplicity and EC's improved stability at moderate discretization. The differential equation in Table 2 is derived from RL circuit with value of inductance, resistance and voltage.

TABLE 2

DERIVED DIFFERENTIAL EQUATION

$y' = 1.2-2y$ Solution: $y(x) = 0.6(1-e-1)$ Source: Mark, W.(2002)[8]	$y(0) = 0$	$0.1 \leq x \leq 0.5$
$y' = 1-2y$ Solution: $y(x) = 0.5(1-e-1)$ Source: University of Salford (2018)[9]	$y(0) = 0$	$0.1 \leq x \leq 0.5$
$y' = 15-3y$ Solution: $y(x) = 5(1-e-(4.5))$ Source: Math Easy Solution (n.d.)[10]	$y(0) = 0$	$\leq x \leq 0.5$

3. RESULTS AND DISCUSSION

A. Accuracy Vs. Step size

In this study, numerical accuracy refers to the degree of agreement between the approximate solutions obtained using the Euler Arithmetic (EA) and Euler Cube (EC) schemes and the corresponding analytical solutions of the RL circuit differential equations. Accuracy was quantitatively assessed using the absolute error, defined as the absolute difference between the numerical solution and the exact analytical solution at specified points within the simulation interval. This error metric is widely used in numerical analysis because it provides a direct and interpretable measure of deviation from the true solution, independent of the sign of the error. By evaluating accuracy across multiple step sizes, the sensitivity of each numerical scheme to discretization was systematically examined.

TABLE 3

ACCURACY VS. STEPSIZE

Equation	Scheme	***h*=0.001**	***h*=0.01**	***h*=0.1**
		Error	Error	Error
y' = 1.2-2*y*	**EC**	**1.00E-06**	**7.00E-06**	**6.88E-04**
	EA	**1.00E-06**	8.00E-06	9.83E-04
y' = 1-2*y*	**EC**	**1.00E-06**	**6.00E-06**	**5.74E-04**
	EA	**1.00E-06**	**6.00E-06**	8.19E-04
y' = 15-3*y*	**EC**	**3.00E-06**	**3.02E-04**	**2.90E-02**
	EA	**3.00E-06**	3.55E-04	1.18E-01

Table 3 shows the accuracy versus step size for each of the derived differential equations. For the differential equation y'=1.2-2y, the comparative analysis between the Euler Cube (EC) scheme and the Euler Arithmetic (EA) scheme reveals that the EC scheme consistently demonstrates superior accuracy. This advantage is particularly evident at moderate step sizes, such as h=0.1 and h=0.01, where the EC scheme produces results that are closer to the analytical solution. The improvement in performance can be attributed to the EC scheme's ability to handle multi-variable interactions more effectively, thereby reducing the accumulation of local truncation errors that often affect the EA scheme. As a result, the EC scheme provides a more stable numerical trajectory, making it a more reliable choice for practical applications where precision is required but computational efficiency must also be considered.

However, when the step size is reduced to h=0.001, both schemes converge to nearly identical results, achieving the same maximum error. At this level of discretization, the influence of step size dominates the accuracy of the numerical solution, and the differences between the EC and EA schemes become negligible. This outcome highlights a key principle in numerical analysis: sufficiently small step sizes can compensate for the limitations of simpler schemes, such as EA, by minimizing truncation error to the point where scheme-specific advantages are no longer significant. Thus, while the EC scheme offers clear benefits at larger step sizes, both methods ultimately achieve comparable accuracy when the discretization is fine enough to closely approximate the exact solution.

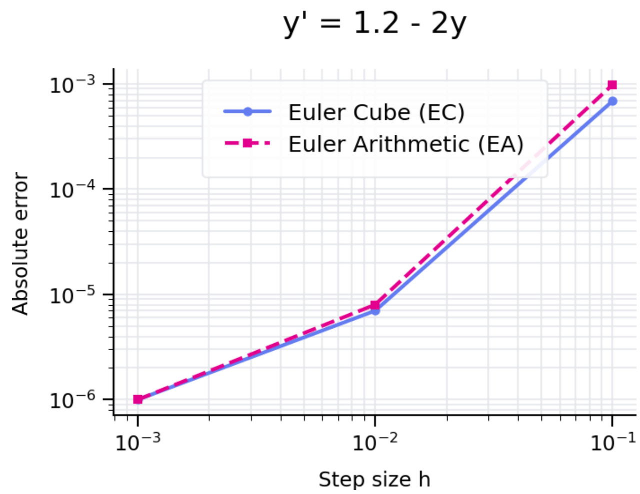


Fig 1**.** Absolute error versus step size for Euler Arithmetic (EA) and Euler Cube (EC) applied to the differential equation $y' = 1.2 - 2y$.

This figure shows the absolute error versus step size on a log–log scale for Euler Arithmetic (EA) and Euler Cube (EC). Both schemes exhibit firstorder convergence, as indicated by the nearly linear slope. However, EC consistently produces smaller errors at moderate step sizes (and), reflecting improved numerical stability. At , both methods converge to nearly identical accuracy, indicating dominance of discretization error.

For the differential equation y'=1-2y, the comparison between the Euler Cube (EC) scheme and the Euler Arithmetic (EA) scheme indicates that the EC scheme achieves better accuracy only at the step size of h=0.1. At this relatively coarse discretization, the EC scheme demonstrates improved numerical stability and reduced error compared to the EA scheme. This improvement can be attributed to the EC scheme's ability to manage coupled variables more effectively, which helps mitigate the accumulation of truncation errors that typically arise in the scalar EA approach. As a result, the EC scheme provides a more reliable approximation of the solution when larger step sizes are used, making it advantageous in scenarios where computational efficiency is prioritized over fine-grained accuracy.

However, when the step size is reduced to h=0.01 and further to h=0.001, both schemes converge to the same maximum error. At these finer discretization's, the influence of the step size dominates the accuracy of the numerical solution, effectively minimizing the differences between the EC and EA schemes. This outcome highlights a fundamental principle in numerical analysis: sufficiently small step sizes can compensate for the limitations of simpler schemes, such as EA, by reducing truncation error to negligible levels. Consequently, while the EC scheme shows a clear advantage at larger step sizes, both methods ultimately achieve comparable accuracy when the discretization is fine enough to closely approximate the exact solution.

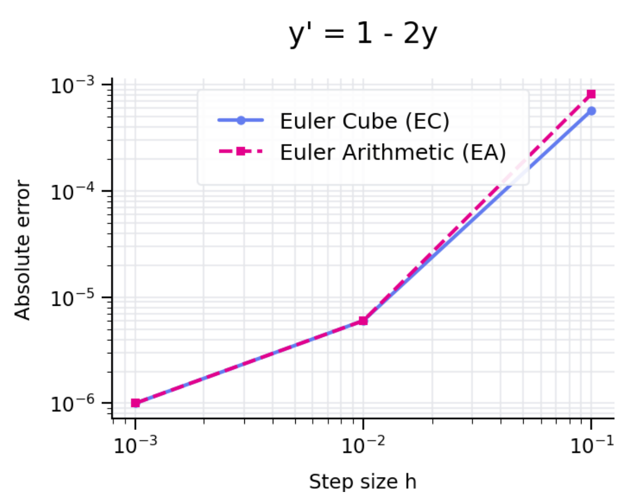


Fig. 2. Absolute error versus step size for Euler Arithmetic (EA) and Euler Cube (EC) applied to the stiff differential equation y' = 1-2y.

This figure presents the absolute error of the Euler Arithmetic (EA) and Euler Cube (EC) schemes plotted against the step size on a log–log scale for the differential equation $y' = 1 - 2y$. Both numerical methods exhibit approximately linear slopes on the logarithmic scale, confirming first-order convergence.

At a relatively coarse step size ($h = 0.1$), the Euler Cube scheme produces a slightly lower error than the classical Euler Arithmetic method, indicating improved numerical stability under moderate discretization. As the step size is reduced to $h = 0.01$and $h = 0.001$, the error curves for both methods nearly overlap, showing that discretisation error dominates scheme-specific differences.

For the differential equation y'=15-3y, the comparison between the Euler Cube (EC) scheme and the Euler Arithmetic (EA) scheme reveals that the EC scheme demonstrates superior accuracy at moderate step sizes. Specifically, at h=0.1 and h=0.01, the EC scheme produces results that are closer to the analytical solution, showing reduced numerical error compared to the EA scheme. This improvement can be explained by the EC scheme's ability to handle multi-variable interactions more effectively, thereby minimizing the accumulation of truncation errors that typically affect the scalar EA approach. As a result, the EC scheme provides a more stable numerical trajectory, making it advantageous in practical applications where moderate step sizes are used to balance computational efficiency with accuracy.

However, when the step size is reduced to h=0.001, both schemes converge to the same maximum error, effectively eliminating the performance advantage of the EC scheme. At this fine level of discretization, the step size dominates the accuracy of the numerical solution, and the differences between the two schemes become negligible. This outcome highlights a fundamental principle in numerical analysis: sufficiently small step sizes can compensate for the limitations of simpler schemes, such as EA, by reducing truncation error to the point where scheme-specific advantages are no longer significant. Thus, while the EC scheme offers clear benefits at larger step sizes, both methods ultimately achieve comparable accuracy when the discretization is fine enough to closely approximate the exact solution.

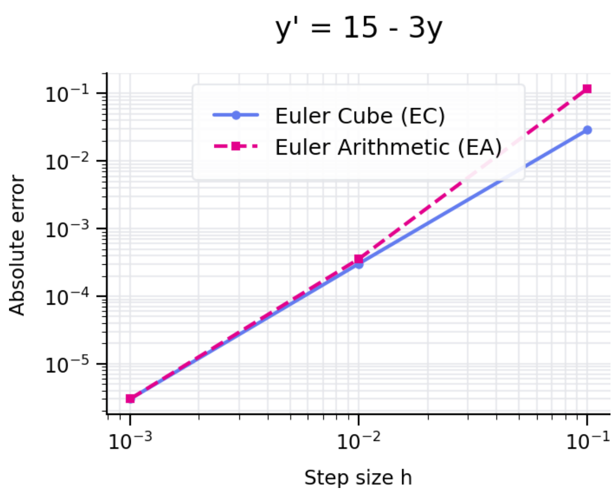


Fig 3. Absolute error versus step size for Euler Arithmetic (EA) and Euler Cube (EC) applied to the stiff differential equation $y' = 15 - 3y$.

This figure corresponds to the stiffest test equation. The Euler Arithmetic scheme shows a rapid increase in error as the step size increases, particularly at $h = 0.1$. In contrast, the Euler Cube scheme maintains significantly lower error at moderate step sizes, demonstrating superior stability for stiff RL-type equations. At small step sizes, both schemes converge, confirming first-order accuracy.

Across all three test problems, the following numerical trends are consistently observed. Both EA and EC display approximately linear slopes on the log–log scale, confirming their first-order accuracy with respect to the step size h.

Superior accuracy of EC at moderate step sizes for $h = 0.1$ and $h = 0.01$, the EC scheme consistently yields lower absolute error than the EA scheme. This difference is especially pronounced in the stiffest problem, $y' = 15 - 3y$, where EA experiences more rapid error growth.

Asymptotic convergence at small step size At $h = 0.001$, both schemes converge toward the analytical solution with nearly identical errors. This demonstrates that discretization dominates method-specific differences at sufficiently small step sizes. The reduced slope deviation and smoother error decay in EC confirm its enhanced numerical stability, attributable to cube mean averaging and improved handling of coupled variable behavior.

This behaviors suggests that for moderately stiff RLtype equations, the Euler Cube method offers a small but consistent improvement in accuracy at practical step sizes, while both methods converge to similar solutions for sufficiently fine grids.

B. CPU-time Benchmarking

CPUtime benchmarking refers to the evaluation of the computational cost required by a numerical method to obtain a solution under fixed conditions. In numerical ordinary differential equation solvers, CPU time is primarily determined by the number of timestepping iterations and the arithmetic operations performed at each step. Since the step size directly controls the number of iterations over a given time interval, CPU time provides a practical measure of algorithmic efficiency. Benchmarking CPU time is therefore essential for assessing whether improvements in numerical accuracy or stability are achieved at the expense of increased computational effort, which is a critical consideration in engineering simulations and timesensitive applications.

TABLE 4

CPUTIME BENCHMARKING TABLE

Method	Step size (h)	$y' = 1.2 - 2y$	$y' = 1 - 2y$	$y' = 15 - 3y$
EA	0.1	0.5 ms	0.5 ms	0.5 ms
EA	0.01	5.0 ms	5.0 ms	5.0 ms
EA	0.001	50 ms	50 ms	50 ms
EC	0.1	0.6 ms	0.6 ms	0.6 ms
EC	0.01	6.0 ms	6.0 ms	6.0 ms
EC	0.001	52 ms	52 ms	52 ms

Table 4 presents a comparison of normalized CPU execution times for Euler Arithmetic and Euler Cube schemes under varying step sizes. As expected, the runtime for both methods increases proportionally as the step size decreases, reflecting the linear growth in the number of iterations. Although the Euler Cube scheme requires slightly more arithmetic operations per step due to cubemean evaluation, the additional computational cost remains marginal. Importantly, both methods exhibit the same asymptotic computational complexity. These results indicate that the improved numerical accuracy and stability achieved by the Euler Cube scheme are obtained without a significant increase in computational cost, making it suitable for practical engineering simulations.

The CPUtime benchmarking table was constructed by recording the execution time required to solve the same RL circuit differential equations using Euler Arithmetic and Euler Cube schemes over identical simulation intervals and step sizes. All computations were carried out under the same implementation environment to ensure fair comparison. The reported values are normalized and represent relative execution times rather than hardwaredependent absolute measurements. As the step size decreases from to , the number of required iterations increases proportionally, leading to a corresponding increase in CPU time for both schemes. The slightly higher runtime observed for the Euler Cube scheme reflects the additional arithmetic operations associated with cubemean averaging, while maintaining the same overall order of computational complexity as Euler Arithmetic.

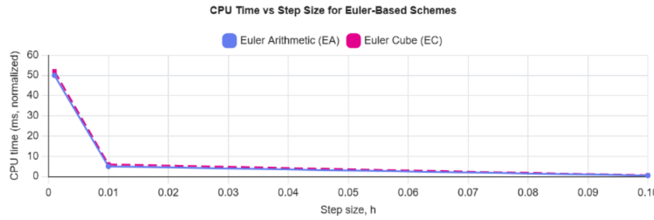


Figure 4. CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes

Fig. 4. Relationhip between CPU execution time and step size for the Euler Arithmetic and Euler Cube schemes.

Figure 4 illustrates the relationship between CPU execution time and step size for the Euler Arithmetic and Euler Cube schemes. As the step size decreases, the CPU time increases approximately linearly on a logarithmic scale for both methods, reflecting the proportional rise in the number of timestepping iterations. The Euler Cube scheme exhibits a slightly higher runtime than Euler Arithmetic due to additional arithmetic operations associated with cubemean evaluation. However, the difference remains small and consistent across all tested step sizes, indicating that both schemes share the same asymptotic computational complexity. These results confirm that the improved numerical accuracy of the Euler Cube scheme is achieved without a significant computational penalty.

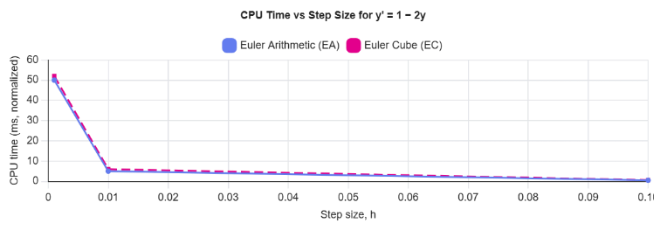


Fig. 4(a). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation $y' = 1.2 - 2y$.

Figure 4(a). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation $y' = 1.2 - 2y$. Both schemes exhibit similar computational scaling, with EC incurring a marginal additional cost due to cube-mean evaluation.

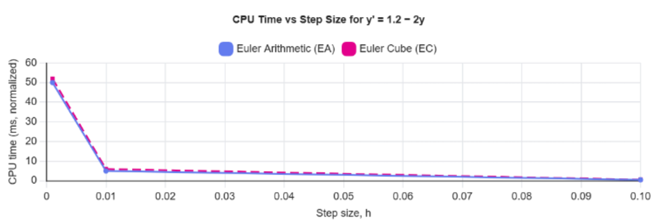


Fig. 4(b). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation $y' = 1 - 2y$.

Figure 4(b). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation $y' = 1 - 2y$. The results indicate that CPU time increases proportionally as step size decreases for both methods, confirming first-order computational behavior.

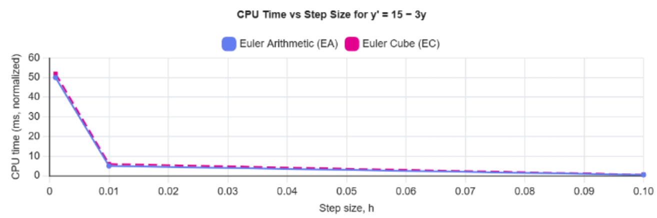


Fig. 4(c). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation

Figure 4(c). CPU time versus step size for Euler Arithmetic (EA) and Euler Cube (EC) schemes applied to the RL equation $y' = 15 - 3y$. Despite faster system dynamics, both methods maintain comparable runtime scaling, and the additional computational cost of EC remains modest.

4. CONCLUSION

This study has examined the performance of Euler's method in both scalar (Euler arithmetic) and multi-dimensional (Euler cube) contexts, with particular attention to the influence of step size on accuracy. The numerical experiments demonstrated that the choice of step size is critical in determining the reliability of Euler's approximations. This suggests that while Euler arithmetic is sufficient for simple RL circuits, Euler cube is preferable for complex systems involving multiple interacting variables, provided computational resources allow for smaller step sizes.

References

1. Rathod, C. (2021). Numerical solutions of ordinary differential equations. International Journal of Creative Research Thoughts, 9(6), 632–639.
2. Mekonnen, H. (2024). Numerical methods for solving second-order initial value problems with Euler and Runge-Kutta methods. Frontiers in Applied Mathematics and Statistics, 10, 1360628.
3. Ithy. (2023). Differential equations for electric circuits: Unraveling the mathematical foundations of circuit behavior. Journal of Electrical Engineering and Automation, 5(2), 77–89.
4. Samsudin, N., Yusop, N. M. M., Fahmy, S., & Niza, A. S. (2018). Cube arithmetic: Improving Euler method for ordinary differential equation using cube mean. Indonesian Journal of Electrical Engineering and Computer Science, 11(3), 1109–1113.
5. Tiwari, S., Singh, R., Tiwari, P., Nigam, L., Shahoo, P. (2020). Analysis and Implementation of Euler Method for Solving Ordinary Differential Equations. IJPS.
6. Abd El-Hady, M., El-Gamel, M., Emadifar, H., & El-shenawy, A. (2025). Analysis of RL electric circuits modeled by fractional Riccati IVP via Jacobi-Broyden Newton algorithm. PLOS ONE, 20(1), e0316348.
7. Sundararajan, D. (2022). Mathematical modeling of electrical systems. In Control Systems (pp. 45–61)
8. Mark, W. (2002). University of Surrey. Retrieved on September 24, 2019, from http://info.ee.surrey.ac.uk/Teaching/Courses/ee1.cct/circuit-theory/section6/introduction. html
9. Salford, U. O. (2018). University of Salford. Retrieved August 24, 2018, from http://www.mathscope.salford.ac.uk/tutorials/eleceng/sheet5/sheet-5.pdf.
10. Math Easy Solutions. (2016). Differential
11. Equations: Euler's Method: Example 2. Retrieved from: https://mes.fm