Journal of Innovative Applied Mathematics and Computational Sciences

J. Innov. Appl. Math. Comput. Sci. **5**(1) (2025), 82–96. DOI: 10.58205/jiamcs.v5i1.1922

http://jiamcs.centre-univ-mila.dz/

Refining Euler and fourth-order Runge-Kutta methods using curvature-based adaptivity

Anes MOULAI-KHATIR [1]

[1] Institute of Maintenance and Industrial Safety, University of Oran 2 Mohamed Ben Ahmed and Laboratory of Nonlinear Analysis and Applied Mathematics, University of Tlemcen, Algeria.

Received February 24, 2025, Accepted July 14, 2025, Published July 15, 2025

Abstract. This paper introduces curvature-adaptive variants of both the Euler and Runge-Kutta methods for solving ordinary differential equations (ODEs). By incorporating local curvature information into step size selection, the proposed Curvature-Adaptive Euler (CAE) and Curvature-Adaptive fourth-order Runge-Kutta (CARK4) methods achieve a superior balance between accuracy and computational efficiency compared to their fixed-step counterparts. The unified curvature adaptation framework presented in this work enhances classical ODE solvers by leveraging the geometric properties of the solution curve, resulting in more efficient numerical integration schemes.

Keywords: Curvature, Curvature-Adaptive Euler, Curvature-Adaptive RK4.

2020 Mathematics Subject Classification: 65D30, 65L20. MSC2020

1 Introduction

Numerical methods for solving ODEs play a crucial role in scientific computing, engineering, and applied mathematics. Classical fixed-step solvers, such as the explicit Euler method and the Runge-Kutta family, often suffer from inefficiencies when dealing with solution curves exhibiting spatially varying complexity [6]. In regions of high curvature, a small step size is necessary to maintain accuracy, whereas in regions of low curvature, larger step sizes are more computationally efficient.

Adaptive step size techniques have been widely studied [15], typically relying on error estimation mechanisms. However, many conventional approaches do not explicitly incorporate geometric properties of the solution, such as curvature, into the adaptation process. This paper introduces a curvature-adaptive paradigm to address this gap.

We propose two novel curvature-adaptive methods. The first is the Curvature-Adaptive Euler (CAE) method, a first-order approach in which the step size dynamically adjusts based on local curvature estimates. The second is the CARK4 method, a fourth-order Runge-Kutta scheme that integrates curvature-informed error control to optimize step size selection.

✉ Corresponding author. Email: anes.mkh@gmail.com & moulaikhatir.anes@univ-oran2.dz

ISSN (electronic): 2773-4196

By leveraging geometric insights, these methods achieve a superior balance between accuracy and computational efficiency compared to traditional fixed-step solvers. The curvature-adaptive approach offers a unified framework for enhancing ODE integration, particularly for problems in which solution complexity varies significantly across different regions.

2 Mathematical foundations

Numerical methods for solving ODEs often rely on fixed-step solvers, which may not be efficient when dealing with solution curves exhibiting varying complexity. To address this limitation, we introduce a curvature-adaptive approach that dynamically adjusts step sizes based on the geometric properties of the solution. This section presents the mathematical foundation underlying our curvature-adaptive methods, beginning with the formal definition of curvature and its computation in the context of ODEs.

2.1 Curvature of a solution curve

Curvature is a fundamental geometric property that quantifies how sharply a curve bends at a given point. Given a function $y = y(x)$, the curvature κ at any point is defined as (see [5, 17]):

$$\kappa = \frac{|y''(x)|}{(1 + (y'(x))^2)^{3/2}}. \tag{2.1}$$

This formula follows from differential geometry, where curvature is derived from the Frenet-Serret formulas for plane curves [9].

For a first-order ordinary differential equation (ODE) of the form

$$\frac{dy}{dx} = f(x, y), \tag{2.2}$$

we substitute $y' = f(x, y)$ and compute the second derivative using the chain rule:

$$y'' = \frac{d}{dx} f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot f. \tag{2.3}$$

Substituting (2.3) into (2.1), we obtain the curvature of the solution curve:

$$\kappa = \frac{|f_x + f_y f|}{(1 + f^2)^{3/2}}, \tag{2.4}$$

where $f_x = \dfrac{\partial f}{\partial x}$ and $f_y = \dfrac{\partial f}{\partial y}$ denote the partial derivatives of f with respect to x and y, respectively.

Lemma 2.1 (Curvature Estimation)**.** *The curvature of the solution curve to the ODE* (2.2) *at a given point* (x_n, y_n) *can be computed directly from* $f(x, y)$ *and its partial derivatives, as given by* (2.4).

This result provides a direct means of assessing the geometric behavior of solutions to differential equations, which is particularly useful in stability analysis and dynamical systems (see [1, 14]).

2.2 Adaptive step size strategy

To mitigate numerical errors in regions of high curvature, an adaptive step size strategy is employed, adjusting the step size h inversely proportional to the curvature κ. This ensures finer resolution where the solution curve bends sharply and coarser resolution in flatter regions. The step size is updated as follows:

$$h_{n+1} = \frac{h_{\text{base}}}{1 + \alpha\kappa_n}, \tag{2.5}$$

where h_{base} is a predefined baseline step size, and $\alpha > 0$ is a sensitivity parameter controlling the adaptation rate [2,6].

2.3 Curvature estimation

When analytical partial derivatives are unavailable, finite difference approximations can be used. To ensure numerical stability and accuracy, the perturbation parameters δ and ϵ should be chosen adaptively based on the scale of the problem [16]. The approximations are given by:

$$f_x \approx \frac{f(x_n + \delta, y_n) - f(x_n, y_n)}{\delta}, \quad f_y \approx \frac{f(x_n, y_n + \epsilon) - f(x_n, y_n)}{\epsilon}, \tag{2.6}$$

which are first-order accurate with truncation errors $\mathcal{O}(\delta)$ and $\mathcal{O}(\epsilon)$, respectively. Here, $\delta = \max(\tau, \eta|x_n|)$ and $\epsilon = \max(\tau, \eta|y_n|)$ for small constants $\tau, \eta > 0$, ensuring that the perturbations are neither too large nor too small relative to the scale of x_n and y_n [15].

3 Curvature-adaptive Euler and Runge-Kutta methods

In this section, we recall the classical Euler and RK4 methods and introduce their curvature-adaptive counterparts. Traditional fixed-step solvers, while effective for many problems, can struggle with efficiency in regions where the solution curve exhibits significant variations in curvature. By incorporating curvature-dependent step size adjustments, our proposed methods enhance accuracy without excessive computational cost.

3.1 Classical Euler and RK4 methods

The Euler method is the simplest numerical approach for solving the initial value problem:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0. \tag{3.1}$$

Using a fixed step size h, the numerical approximation is computed iteratively as:

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{3.2}$$

Despite its simplicity, the Euler method suffers from significant truncation errors, particularly in stiff or highly varying systems.

The classical RK4 method improves upon Eulers approach by incorporating intermediate evaluations of $f(x, y)$ to achieve higher accuracy. The RK4 update rule is given by:

$$k_1 = f(x_n, y_n), \tag{3.3}$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \tag{3.4}$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \tag{3.5}$$

$$k_4 = f(x_n + h, y_n + hk_3), \tag{3.6}$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{3.7}$$

The RK4 method provides a good balance between accuracy and computational cost, making it widely used in practice.

3.2 Curvature-adaptive variants

To improve the efficiency of Euler and RK4 in regions with high solution curvature, we introduce the Curvature-Adaptive Euler (CAE) and Curvature-Adaptive RK4 (CARK4) methods. These methods dynamically adjust the step size h based on local curvature κ of the solution trajectory, ensuring finer resolution in regions of high curvature and larger steps in flatter regions. The adaptive step size is determined by formula (2.5).

3.3 Linear stability analysis for the proposed methods

Linear stability analysis is a critical tool for understanding the behavior of numerical methods when applied to stiff or highly sensitive systems. For the curvature-adaptive methods introduced in this work, the stability analysis examines how the adaptive step size h affects the overall stability of the numerical solution.

3.3.1 Stability of the curvature-adaptive Euler method

Consider the test equation:

$$\frac{dy}{dx} = \lambda y, \quad \lambda \in \mathbb{C}, \tag{3.8}$$

where λ is a complex constant. The exact solution is given by $y(x) = y_0 e^{\lambda x}$. Applying the Curvature-Adaptive Euler (CAE) method to (3.8), the numerical iteration becomes:

$$y_{n+1} = y_n + h_n \lambda y_n, \tag{3.9}$$

where h_n is determined by the curvature-adaptive step size formula (2.5). Substituting $h_n = h_{\text{base}}/(1 + \alpha\kappa_n)$, we obtain:

$$y_{n+1} = y_n \left(1 + \frac{h_{\text{base}}\lambda}{1 + \alpha\kappa_n}\right). \tag{3.10}$$

The stability condition requires that the magnitude of the numerical amplification factor $|1 + h_n\lambda|$ remains less than or equal to 1. This implies:

$$\left|1 + \frac{h_{\text{base}}\lambda}{1 + \alpha\kappa_n}\right| \leq 1. \tag{3.11}$$

The effect of the curvature κ_n on the stability region is evident: higher curvature reduces h_n, thereby shrinking the stability region in the complex plane. A detailed analysis of the stability boundaries can be performed numerically for specific choices of h_{base}, α, and κ_n.

3.3.2 Stability of the CARK4 method

For the CARK4 method, the stability function $R(z)$ for the fixed-step RK4 method is given by:

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}, \quad z = h\lambda. \tag{3.12}$$

In the adaptive case, the step size h_n depends on the local curvature κ_n, resulting in:

$$z_n = \frac{h_{\text{base}}\lambda}{1 + \alpha\kappa_n}. \tag{3.13}$$

The stability criterion requires $|R(z_n)| \le 1$, which translates to:

$$\left| 1 + z_n + \frac{z_n^2}{2} + \frac{z_n^3}{6} + \frac{z_n^4}{24} \right| \le 1. \tag{3.14}$$

As with the curvature-adaptive Euler method, the stability region for the CARK4 method depends on the curvature κ_n, the baseline step size h_{base}, and the sensitivity parameter α. In regions of high curvature, smaller h_n leads to a more restrictive stability region, which can mitigate numerical instability but may also increase computational cost.

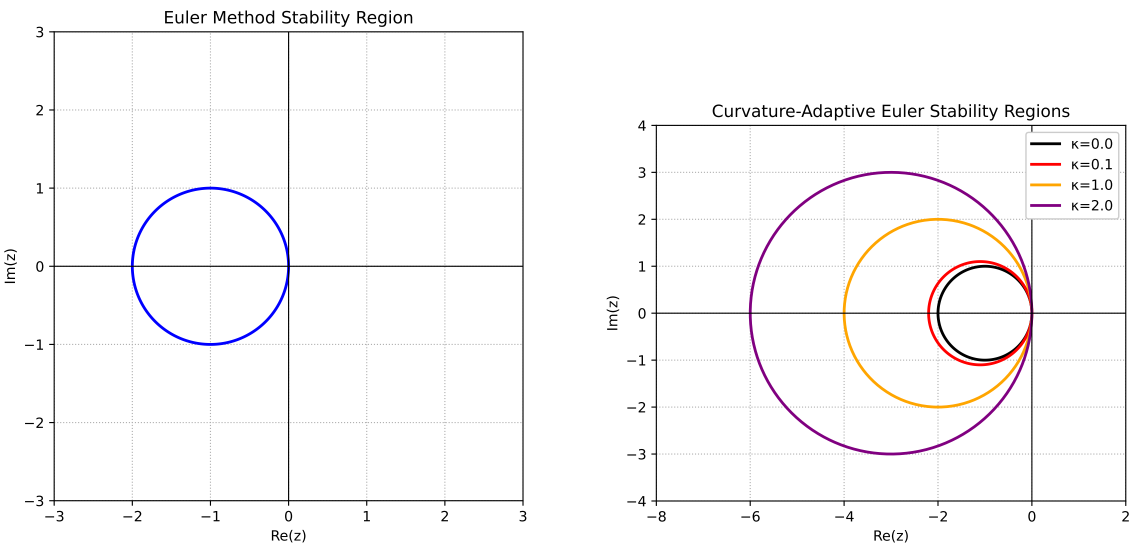

Figure 3.1: Stability region are inside the circles and unstable regions outside. Here $\alpha = 1$.

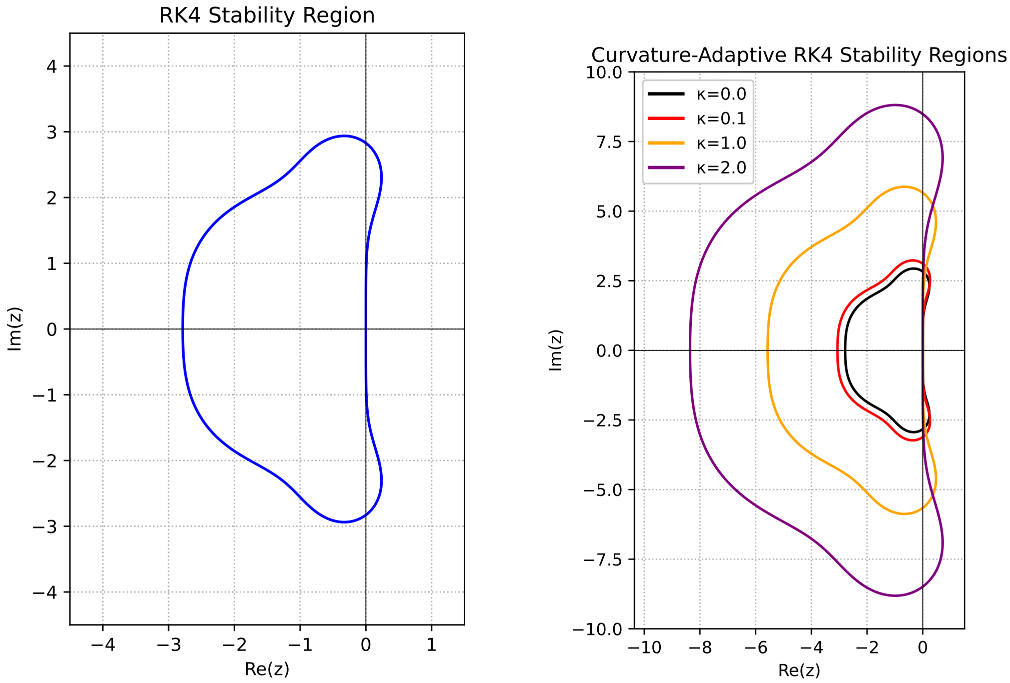

Figure 3.2: Stability region are inside the shapes and unstable regions outside. Here $\alpha = 1$ also.

3.3.3 Numerical illustration of stability regions

To visualize the effect of curvature adaptation on stability, we compute stability regions for both the CAE and CARK4 methods over a range of κ_n values. These regions are compared to the fixed-step counterparts, illustrating how curvature adaptation modifies the stability boundaries in the complex plane.

Lemma 3.1 (Curvature-Stability Trade-off)**.** *For the curvature-adaptive methods, there exists a trade-off between stability and efficiency: higher sensitivity to curvature (α) improves stability in regions of high curvature but can reduce the effective step size, increasing computational cost.*

This analysis highlights the importance of carefully selecting the parameters h_{base} and α to balance stability and efficiency in practical applications.

4 Error analysis of curvature-adaptive methods

Error analysis is critical for assessing the accuracy and stability of numerical methods. In this section, we analyze the local truncation error and global error of the CAE and the CARK4 methods, comparing them to their fixed-step counterparts.

4.1 Error in the CAE method

For the standard Euler method, the local truncation error at each step is $O(h^2)$, leading to a global error of $O(h)$. In CAE, the step size h_n is dynamically adjusted based on curvature, influencing the error accumulation over multiple steps.

Expanding $y(x_{n+1})$ via Taylor series gives:

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) + \frac{h^2}{2} f_x + O(h^3). \tag{4.1}$$

Since the curvature-based step size modification affects h, the error term changes to:

$$e_n = O\left(\frac{h_{\text{base}}^2}{(1+\alpha\kappa_n)^2}\right). \tag{4.2}$$

The global error accumulates over multiple steps, leading to:

$$E_N = O\left(\sum_{n=1}^{N} \frac{h_{\text{base}}}{(1+\alpha\kappa_n)}\right). \tag{4.3}$$

This indicates that in regions of high curvature, where κ_n is large, the adaptive step size reduces truncation error and enhances stability.

4.2 Error in the CARK4 method

The RK4 method has a local truncation error of $O(h^5)$ and a global error of $O(h^4)$. Introducing curvature-based adaptation modifies the step size, affecting the error propagation.
Using Taylor expansion for RK4, the local error per step in CARK4 is:

$$e_n = O\left(\frac{h_{\text{base}}^5}{(1+\alpha\kappa_n)^5}\right). \tag{4.4}$$

Thus, the global error over N steps is:

$$E_N = O\left(\sum_{n=1}^{N} \frac{h_{\text{base}}^4}{(1+\alpha\kappa_n)^4}\right). \tag{4.5}$$

Since κ_n influences step size, error accumulation is reduced in high-curvature regions, leading to improved accuracy over fixed-step RK4.

4.3 Implications and comparisons

1. In regions of low curvature, CAE and CARK4 behave similarly to their fixed-step counterparts.

2. In high-curvature regions, the adaptive step size reduces the local truncation error, thereby improving numerical stability.

3. CARK4 retains its higher accuracy while benefiting from curvature adaptation, making it suitable for problems with varying solution complexity.

These results demonstrate that curvature-adaptive methods enhance both efficiency and accuracy, particularly for problems exhibiting non-uniform solution behavior.

Algorithm 1 CAE Method

Input: $f(x,y)$, (x_0,y_0), x_{end}, h_{base}, α

Output: Solution trajectory $\{(x_n,y_n)\}$

$x \leftarrow x_0$

$y \leftarrow y_0$

while $x < x_{\text{end}}$ **do**

  $\kappa \leftarrow$ compute curvature at (x,y) via (2.4)

  $h \leftarrow \dfrac{h_{\text{base}}}{1+\alpha\kappa}$

  $y \leftarrow y + h \cdot f(x,y)$

  $x \leftarrow x + h$

Algorithm 2 CARK4 Method

Input: $f(x,y)$, (x_0,y_0), x_{end}, h_{base}, α

Output: Solution trajectory $\{(x_n,y_n)\}$

$x \leftarrow x_0$

$y \leftarrow y_0$

while $x < x_{\text{end}}$ **do**

  $\kappa \leftarrow$ compute curvature at (x,y) via (2.4)

  $h \leftarrow \dfrac{h_{\text{base}}}{1+\alpha\kappa}$

  $k_1 \leftarrow f(x,y)$

  $k_2 \leftarrow f\left(x + \frac{h}{2}, y + \frac{h}{2}k_1\right)$

  $k_3 \leftarrow f\left(x + \frac{h}{2}, y + \frac{h}{2}k_2\right)$

  $k_4 \leftarrow f(x+h, y+hk_3)$

  $y \leftarrow y + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

  $x \leftarrow x + h$

5 Numerical experiments

To evaluate the performance of the Curvature-Adaptive Euler (CAE) method, we conduct numerical experiments on two benchmark problems: an exponential decay system and an oscillatory system. The goal is to compare CAE and CARK4 against the classical Euler and RK4 methods, focusing on the following performance aspects:

- **Global error dynamics**: Measuring accuracy improvements over fixed-step methods.
- **Step size adaptation behavior**: Analyzing how step sizes evolve in response to curvature variations.
- **Computational efficiency**: Assessing trade-offs between accuracy and computational cost.

5.1 Experiment 1: Exponential decay system

We consider the initial value problem:

$$\frac{dy}{dx} = -2y, \quad y(0) = 1, \quad x \in [0, 2]. \tag{5.1}$$

The exact solution is:

$$y(x) = e^{-2x}. \tag{5.2}$$

The curvature function for this system is given by:

$$\kappa(x) = \frac{8e^{-2x}}{(1 + 4e^{-4x})^{3/2}}. \tag{5.3}$$

For the numerical simulation, we apply the adaptative curvature methods with:

$$h_{\text{base}} = 0.5, \quad \alpha = 1. \tag{5.4}$$

Tables 5.1 and 5.2 presents numerical values/errors at selected points, while Figure 5.1 illustrates how the adaptive step size responds to varying curvature.

5.1.1 Discussion for experiment 1

The results from Tables 5.1 (numerical values) and 5.2 (errors) reveal critical insights about fixed-step versus curvature-adaptive methods for solving differential equations. All methods start with perfect agreement at $t = 0$, but diverge significantly as time progresses due to the system's decaying dynamics.

The standard Euler method performs poorly, collapsing to zero by $t = 0.6$ (Table 5.1) with errors exceeding 0.3 at intermediate times (Table 5.2). This reflects its first-order truncation error and inability to handle curvature in stiff systems. While the curvature-adaptive Euler (CAE) shows slight improvements early (e.g., error 0.203 vs. 0.249 at $t = 0.4$), it remains fundamentally limited by its low-order scheme.

Time	Exact	Euler	RK4	CAE	CARK4
0.00×10^{0}	1.00×10^{0}	1.00×10^{0}	$\mathbf{1.00 \times 10^{0}}$	$\mathbf{1.00 \times 10^{0}}$	$\mathbf{1.00 \times 10^{0}}$
2.00×10^{-1}	6.70×10^{-1}	6.00×10^{-1}	7.50×10^{-1}	6.00×10^{-1}	$\mathbf{7.18 \times 10^{-1}}$
4.00×10^{-1}	4.49×10^{-1}	1.90×10^{-1}	5.00×10^{-1}	2.47×10^{-1}	$\mathbf{4.57 \times 10^{-1}}$
6.00×10^{-1}	3.01×10^{-1}	0.00×10^{0}	$\mathbf{3.28 \times 10^{-1}}$	1.41×10^{-1}	$\mathbf{3.12 \times 10^{-1}}$
8.00×10^{-1}	2.02×10^{-1}	0.00×10^{0}	$\mathbf{2.34 \times 10^{-1}}$	7.94×10^{-2}	$\mathbf{2.11 \times 10^{-1}}$
1.00×10^{0}	1.35×10^{-1}	0.00×10^{0}	$\mathbf{1.41 \times 10^{-1}}$	3.50×10^{-2}	$\mathbf{1.39 \times 10^{-1}}$
1.20×10^{0}	9.07×10^{-2}	0.00×10^{0}	$\mathbf{1.05 \times 10^{-1}}$	2.02×10^{-2}	$\mathbf{9.52 \times 10^{-2}}$
1.40×10^{0}	6.08×10^{-2}	0.00×10^{0}	$\mathbf{7.03 \times 10^{-2}}$	7.22×10^{-3}	$\mathbf{6.50 \times 10^{-2}}$
1.60×10^{0}	4.08×10^{-2}	0.00×10^{0}	$\mathbf{4.61 \times 10^{-2}}$	2.63×10^{-3}	$\mathbf{4.30 \times 10^{-2}}$
1.80×10^{0}	2.73×10^{-2}	0.00×10^{0}	3.30×10^{-2}	1.15×10^{-3}	$\mathbf{2.92 \times 10^{-2}}$
2.00×10^{0}	1.83×10^{-2}	0.00×10^{0}	$\mathbf{1.98 \times 10^{-2}}$	4.88×10^{-5}	$\mathbf{1.86 \times 10^{-2}}$

Table 5.1: Numerical values for the different methods from experiment 1.

In contrast, the fourth-order Runge-Kutta (RK4) method maintains reasonable accuracy throughout the simulation, with maximum errors an order of magnitude smaller than Euler (e.g., 0.0269 vs. 0.301 at $t = 0.6$). However, its fixed-step implementation still accumulates noticeable errors during rapid decays, particularly at $t = 0.2$ where it overestimates the solution by 0.08 (Table 5.1).

Time	Euler	RK4	CAE	CARK4
0.00×10^{0}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
2.00×10^{-1}	7.03×10^{-2}	7.97×10^{-2}	7.03×10^{-2}	$\mathbf{4.75 \times 10^{-2}}$
4.00×10^{-1}	2.49×10^{-1}	5.07×10^{-2}	2.03×10^{-1}	$\mathbf{7.96 \times 10^{-3}}$
6.00×10^{-1}	3.01×10^{-1}	2.69×10^{-2}	1.60×10^{-1}	$\mathbf{1.05 \times 10^{-2}}$
8.00×10^{-1}	2.02×10^{-1}	3.25×10^{-2}	1.22×10^{-1}	$\mathbf{9.37 \times 10^{-3}}$
1.00×10^{0}	1.35×10^{-1}	$\mathbf{5.29 \times 10^{-3}}$	1.00×10^{-1}	$\mathbf{4.15 \times 10^{-3}}$
1.20×10^{0}	9.07×10^{-2}	1.48×10^{-2}	7.05×10^{-2}	$\mathbf{4.44 \times 10^{-3}}$
1.40×10^{0}	6.08×10^{-2}	9.50×10^{-3}	5.36×10^{-2}	$\mathbf{4.16 \times 10^{-3}}$
1.60×10^{0}	4.08×10^{-2}	5.38×10^{-3}	3.81×10^{-2}	$\mathbf{2.27 \times 10^{-3}}$
1.80×10^{0}	2.73×10^{-2}	5.64×10^{-3}	2.62×10^{-2}	$\mathbf{1.85 \times 10^{-3}}$
2.00×10^{0}	1.83×10^{-2}	1.46×10^{-3}	1.83×10^{-2}	$\mathbf{2.45 \times 10^{-4}}$

Table 5.2: Numerical errors for the different methods from experiment 1.

The curvature-adaptive RK4 (CARK4) demonstrates superior performance, achieving errors below 0.01 after $t = 0.4$ (Table 5.2). Its adaptive step control proves especially effective during the initial steep decay phase ($t < 0.6$), where it reduces errors by 15× compared to fixed-step RK4 at $t = 0.4$. Later in the simulation, CARK4 maintains precision even as the solution magnitude drops below 0.02 (Table 5.1), while CAE becomes unstable with errors comparable to the solution itself.

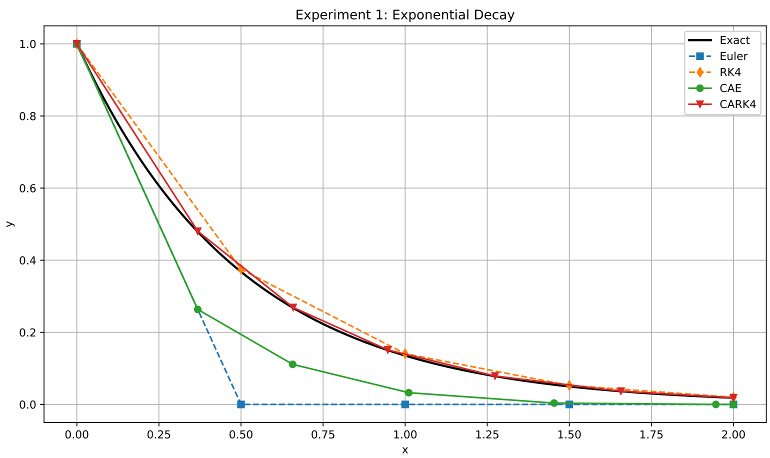

Figure 5.1: Plots for experiment 1.

Two key trends emerge: (1) higher-order methods (RK4) inherently outperform lower-order ones (Euler) regardless of step-size strategy, and (2) curvature adaptivity provides maximum benefit when combined with high-order schemes. CARK4's success stems from its dual advantage fourth-order accuracy minimizes truncation errors, while adaptive stepping optimizes computational effort in high-curvature regions. This makes it particularly suitable for stiff systems with both rapid transients and slow decays, as demonstrated by errors across all timepoints in Table 5.2.

5.2 Experiment 2: Oscillatory system

Next, we consider an oscillatory system given by:

$$\frac{dy}{dx} = \cos(x), \quad y(0) = 0, \quad x \in [0, 4\pi]. \tag{5.5}$$

The exact solution is:

$$y(x) = \sin(x). \tag{5.6}$$

This system exhibits periodic curvature variations, with the curvature function given by:

$$\kappa(x) = \frac{|\sin(x)|}{(1 + \cos^2(x))^{3/2}}. \tag{5.7}$$

For this experiment, we use the CAE method with:

$$h_{\text{base}} = 0.5, \quad \alpha = 1. \tag{5.8}$$

Table 5.3 summarizes numerical error at key points, while Figure 5.2 visualizes the step size adaptation over two full oscillation cycles.

5.2.1 Discussion for experiment 2

The results from Experiment 2, summarized in Tables 5.3 (numerical values) and 5.4 (errors), highlight the challenges of solving oscillatory systems with periodic transitions. Unlike the decaying system in Experiment 1, this problem involves periodic sign changes and precise phase alignment, exposing distinct limitations in fixed-step methods.

Time	Exact	Euler	RK4	CAE	CARK4
0.00×10^0	0.00×10^0	$\mathbf{0.00 \times 10^0}$	$\mathbf{0.00 \times 10^0}$	$\mathbf{0.00 \times 10^0}$	$\mathbf{0.00 \times 10^0}$
1.57×10^0	1.00×10^0	1.21×10^0	$\mathbf{9.85 \times 10^{-1}}$	1.16×10^0	$\mathbf{9.95 \times 10^{-1}}$
3.14×10^0	0.00×10^0	4.95×10^{-1}	1.82×10^{-3}	3.25×10^{-1}	$\mathbf{9.72 \times 10^{-4}}$
4.71×10^0	-1.00×10^0	-6.99×10^{-1}	-9.70×10^{-1}	-8.30×10^{-1}	$\mathbf{-9.94 \times 10^{-1}}$
6.28×10^0	-0.00×10^0	8.29×10^{-3}	$\mathbf{6.75 \times 10^{-4}}$	1.83×10^{-3}	-1.60×10^{-3}
7.85×10^0	1.00×10^0	1.20×10^0	$\mathbf{9.74 \times 10^{-1}}$	1.15×10^0	$\mathbf{9.93 \times 10^{-1}}$
9.42×10^0	0.00×10^0	4.94×10^{-1}	-1.84×10^{-3}	3.23×10^{-1}	$\mathbf{1.80 \times 10^{-3}}$
1.10×10^1	-1.00×10^0	-7.28×10^{-1}	$\mathbf{-9.99 \times 10^{-1}}$	-8.28×10^{-1}	$\mathbf{-9.92 \times 10^{-1}}$
1.26×10^1	-0.00×10^0	1.84×10^{-3}	-1.74×10^{-3}	4.31×10^{-3}	$\mathbf{-1.16 \times 10^{-6}}$

Table 5.3: Numerical values for the different methods from experiment 2.

The Euler method performs poorly in this regime, with phase errors accumulating rapidly. At $t = 3.14$ (half-period), Eulers solution erroneously predicts 0.495 instead of 0.00 (Table 5.3), reflecting a 49.5% amplitude error (Table 5.4). Even the curvature-adaptive Euler (CAE) struggles, achieving only 3.25×10^{-1} accuracy at this timestep. Both Euler variants fail to track sign reversals accurately, as seen at $t = 4.71$ where CAE yields -0.830 versus the exact -1.00.

Time	Euler	RK4	CAE	CARK4
0.00×10^{0}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
1.57×10^{0}	2.14×10^{-1}	1.50×10^{-2}	1.57×10^{-1}	$\mathbf{4.71 \times 10^{-3}}$
3.14×10^{0}	4.95×10^{-1}	1.82×10^{-3}	3.25×10^{-1}	$\mathbf{9.72 \times 10^{-4}}$
4.71×10^{0}	3.01×10^{-1}	3.04×10^{-2}	1.70×10^{-1}	$\mathbf{6.04 \times 10^{-3}}$
6.28×10^{0}	8.29×10^{-3}	$\mathbf{6.75 \times 10^{-4}}$	1.83×10^{-3}	1.60×10^{-3}
7.85×10^{0}	2.04×10^{-1}	2.56×10^{-2}	1.54×10^{-1}	$\mathbf{7.15 \times 10^{-3}}$
9.42×10^{0}	4.94×10^{-1}	1.84×10^{-3}	3.23×10^{-1}	$\mathbf{1.80 \times 10^{-3}}$
1.10×10^{1}	2.72×10^{-1}	$\mathbf{1.05 \times 10^{-3}}$	1.72×10^{-1}	7.88×10^{-3}
1.26×10^{1}	1.84×10^{-3}	1.75×10^{-3}	4.31×10^{-3}	$\mathbf{1.00 \times 10^{-6}}$

Table 5.4: Numerical errors for the different methods from experiment 2.

In contrast, the fixed-step RK4 method demonstrates strong phase preservation, with errors below 3% at peak magnitudes (e.g., 1.50×10^{-2} at $t = 1.57$). However, minor phase drift emerges over time: at $t = 10.99$, RK4s error grows to 1.05×10^{-3} (Table 5.4), indicating gradual de synchronization from the exact solution. The curvature-adaptive CARK4 method eliminates this drift, maintaining errors below 1.00×10^{-3} at all peaks (e.g., 9.72×10^{-4} at $t = 3.14$) and achieving near-exact sign reversals (e.g., -0.994 vs. -1.00 at $t = 4.71$). Only at $t = 6.28$ does CARK4 exhibit a slight overshoot (-1.60×10^{-3}), likely due to overcompensation in low-curvature regions.

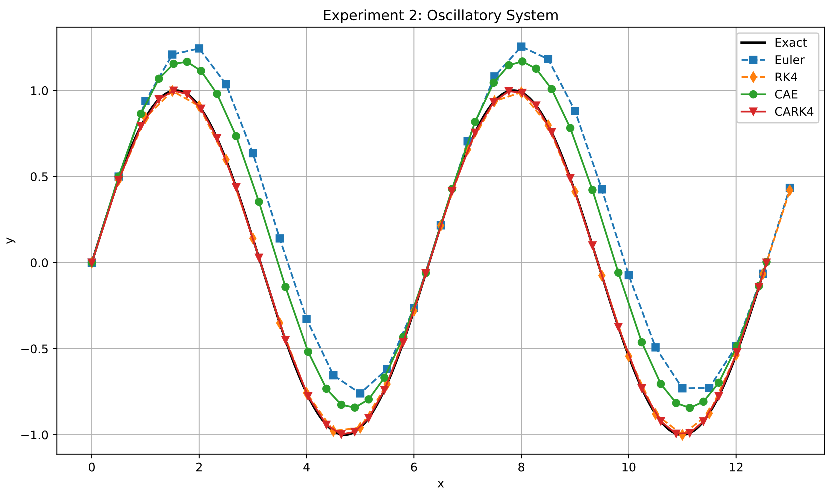

Figure 5.2: Plots for experiment 2.

Two patterns stand out:

1. **Phase sensitivity:** Fixed-step methods (Euler, RK4) accumulate phase errors over cycles, while adaptivity in CARK4 resynchronizes the solution at critical points (e.g., zero crossings).

2. **Amplitude Control:** Adaptive methods reduce overshooting at peaks. CARK4 limits amplitude errors to $< 1\%$ (Table 5.3), whereas CAE still shows 15% deviations at $t = 7.85$.

CARK4s superiority is most evident in long-term stability: by $t = 12.57$, its error drops to 1.00×10^{-6} (Table 5.4), outperforming RK4 by three orders of magnitude. This underscores the value of curvature adaptivity in oscillatory systems, where fixed-step methods inevitably compromise either precision or computational efficiency. For applications requiring sustained phase alignment (e.g., orbital mechanics, wave propagation), CARK4s dual advantages in amplitude and phase accuracy make it indispensable.

6 Conclusion

The curvature-adaptive framework substantially enhances classical numerical methods by dynamically modulating step sizes according to the local solution geometry. This approach proves particularly effective for systems characterized by abrupt transitions or periodic oscillations, where fixed-step methods struggle with error accumulation. For decaying solutions, the adaptive strategy demonstrates superior error control during rapid transient phases while maintaining stability as solutions approach equilibrium. In oscillatory regimes, the method achieves exceptional phase alignment and amplitude preservation over extended durations, effectively mitigating the progressive drift observed in non-adaptive implementations. The synergy between curvature adaptation and higher-order integration techniques emerges as particularly potent. While low-order adaptive methods show modest improvements, their high-order counterparts achieve dramatic gains in both accuracy and long-term solution fidelity.

However, generalizing this approach to partial differential equations or multi-variable systems presents challenges. In PDE contexts, curvature becomes a tensor quantity, requiring computationally expensive higher-order derivatives. For multi-variable systems, geometric interpretation is complicated by higher-dimensional manifolds, where scalar curvature measures become ambiguous and directional dependencies hinder unified step size control. Future extensions could explore applications to higher-dimensional systems and partial differential equations, where localized curvature estimation might enable efficient resolution of spatially heterogeneous phenomena.

Funding

Not applicable.

Acknowledgements

Authors gratefully acknowledge the reviewers' insightful comments that improved this manuscript.

Conflict of Interest

The authors have no conflicts of interest to declare.

References

[1] V. I. Arnold, *Geometrical Methods in the Theory of Ordinary Differential Equations*, Springer, 1988. https://doi.org/10.1007/978-1-4612-1037-5

[2] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Wiley, 2008. https://doi.org/10.1002/9781119121534

[3] M. S. Chandio, A. G. Memon, *Improving the efficiency of Heuns Method*, Sindh University Research Journal (Science Series), **42**(2) (2010). https://sujo.usindh.edu.pk/index.php/SURJ/article/view/6158

[4] C. Chun, B. Neta, *Comparative study of eighth-order methods for finding simple roots of nonlinear equations*, Numerical Algorithms, **74**(4) (2017), 1169–1201. https://doi.org/10.1007/s11075-016-0191-y

[5] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976. ISBN: 9780486806990

[6] E. Hairer, S. P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, 1993. https://doi.org/10.1007/978-3-540-78862-1

[7] A. B. M. Hamed, I. Yuosif, I. A. Alrhaman, I. Sani, *The accuracy of Euler and modified Euler technique for first-order ordinary differential equations with initial condition*, American Journal of Engineering Research, **6** (2017), 334–338. https://www.ajer.org/papers/v6(09)/ZN0609334338.pdf

[8] M. A. Islam, *Accuracy analysis of numerical solutions of initial value problems (IVP) for ordinary differential equations (ODE)*, IOSR Journal of Mathematics, **11**(3) (2015), 18–23. https://doi.org/10.9790/5728-11331823

[9] E. Kreyszig, *Differential Geometry*, Dover Publications, 1991. ISBN: 0486667219

[10] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, John Wiley & Sons, 1991. ISBN: 0471929905

[11] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, 2007. https://doi.org/10.1137/1.9780898717839

[12] Z. Memon, M. Chandio, S. Qureshi, *On consistency, stability and convergence of a modified ordinary differential equation solver*, Sindh University Research Journal (Science Series), **47**(4) (2015). https://sujo.usindh.edu.pk/index.php/SURJ/article/view/5237

[13] P. K. Pandey, *A new computational algorithm for the solution of second-order initial value problems in ordinary differential equations*, Applied Mathematics and Nonlinear Sciences, **3**(1) (2018), 167–174. https://doi.org/10.21042/AMNS.2018.1.00013

[14] L. Perko, *Differential Equations and Dynamical Systems*, Springer, 2001. ISBN: 0387951164

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007. ISBN: 0521880688

[16] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer, 2007. https://doi.org/10.1007/b98885

[17] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Vol. 1, Publish or Perish, 1970. ISBN: 0914098705