Proceedings of CONF-APMM 2025 Symposium: Multi-Qubit Quantum Communication for Image Transmission over Error Prone Channels
DOI: 10.54254/2753-8818/2025.GL24961

Improving the Efficiency of Stiff ODE Solvers through Adaptive Step Size Control and Jacobian Optimization

Yuanqi Li

Haide College, Ocean University of China, Qingdao, China
liyuanqi@stu.ouc.edu.cn

Abstract. Stiff ordinary differential equations (ODEs) present significant challenges for numerical solvers due to their multiscale nature and stringent stability requirements. Traditional methods, such as fixed-step schemes, often suffer from low computational efficiency, while fully implicit approaches may introduce numerical instability. This study proposes an improved adaptive Rosenbrock method that incorporates a PI controller-based step size adjustment mechanism and a selective Jacobian matrix update strategy to enhance both efficiency and stability. Numerical experiments on the Robertson chemical kinetics model demonstrate that, compared to conventional fixed-step solvers, the proposed method reduces the number of integration steps to just 0.4% of the original while maintaining high accuracy and significantly mitigating unphysical phenomena such as negative concentrations. These results emphasize the excellent performance of the adaptive Rosenbrock framework in solving stiff ordinary differential equation (ODE) systems. Future work may extend this approach to partial differential equations, automatic tuning of controller parameters, and stiffness prediction using machine learning techniques.

Keywords: Stiff differential equations, Adaptive step size, Rosenbrock method, Jacobian matrix optimization

1. Introduction

The numerical solution of stiff ordinary differential equations (ODEs) poses serious challenges. In traditional explicit methods (such as the Euler method), maintaining numerical stability requires extremely small time steps due to the rapid decay or strong oscillations of certain solution components. The root cause of this issue lies in the multiscale nature of stiff problems—different solution components evolve at vastly different rates, often spanning several orders of magnitude. This disparity renders conventional explicit solvers highly inefficient. In their seminal work, Hairer and Wanner [1] emphasized the need to develop specialized numerical algorithms for such problems. As early as 1993 [2], they had established a theoretical framework for explicit and basic implicit methods aimed at non-stiff problems, where the solution components typically exhibit more uniform temporal behavior.

The primary difficulty in solving stiff problems lies in the need to compute Jacobian matrices and subsequently solve the associated linear systems. These operations are computationally intensive and prone to numerical instability. Notably, stiff systems are common in real-world applications such as chemical reaction kinetics, biomedical modeling, and circuit analysis [3], where numerical methods must simultaneously ensure stability and computational efficiency.

Among current mainstream approaches, implicit Euler and implicit Runge-Kutta methods offer favorable stability but require the repeated solution of complex nonlinear systems, leading to exponentially increasing computational costs [4]. Semi-implicit methods such as the Rosenbrock family attempt to strike a balance by combining implicit and explicit components, thereby improving computational efficiency to some extent. However, the associated computational overhead remains significant.

This research is aimed at overcoming these limitations through introducing an innovative adaptive step - size optimization strategy, which significantly enhances the efficiency of solving stiff ordinary differential equations (ODEs). Using the Rosenbrock method as a foundation, we develop an intelligent adaptive step size control algorithm that achieves an optimal trade-off between computational accuracy and efficiency. With the rapid advancement of high-performance computing, such adaptive methods are expected to demonstrate unprecedented advantages in addressing large-scale, complex problems.

2. Methodology

2.1. Standard form of stiff differential equations

The standard form of a stiff differential equation is:

$$y' = f(t, y) \tag{1}$$

where

$$\| J_f \| \gg 1 \tag{2}$$

or

$$\mathrm{Re}(\lambda_i) \ll 0 \tag{3}$$

where J_f is the Jacobian matrix of the system, and $\lambda_i$are its eigenvalues, which define the characteristics of stiffness [5]. Such systems exhibit a characteristic multiscale nature: some variables change rapidly, while others evolve slowly. When using explicit methods to solve these systems, maintaining numerical stability necessitates extremely small time steps, which significantly reduces computational efficiency. Therefore, in practical computations, implicit or semi-implicit methods are generally preferred.

The aim of this study is to substantially improve computational efficiency while maintaining L - stability by integrating three crucial techniques: the Rosenbrock method, adaptive step - size control, and Jacobian matrix optimization.

2.2. Algorithm framework

The entire algorithm framework consists of three core modules: stiffness detection, step size control, and Jacobian matrix optimization. These modules cooperate with each other, and the specific process is as follows:

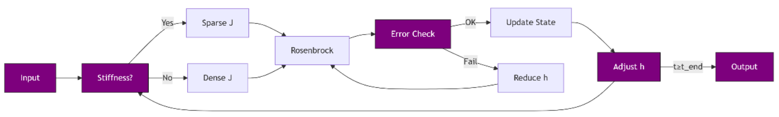

Figure 1: Adaptive stiff ODE solver framework

As shown in Figure 1, the framework includes the following components:

• Stiffness detection: Determine whether the system is stiff based on the spectral radius. If stiffness is detected, a sparse solving strategy is enabled.

• Step size control: Dynamically adjust the step size based on the solution error to balance accuracy and efficiency.

• Jacobian matrix optimization: Decide whether to update the Jacobian matrix based on the frequency of changes in the solution and select either a sparse or full Jacobian depending on the stiffness.

2.3. Rosenbrock method

The Rosenbrock method is a semi - implicit scheme that is especially suitable for stiff ordinary differential equations (ODEs). It combines explicit and implicit computations and can maintain stability even when using large step sizes, providing an efficient alternative to traditional implicit methods.

The core formula is:

$$(\mathrm{I} - \mathrm{h}\gamma \mathrm{J})\mathrm{k}_\mathrm{i} = \mathrm{hf}\left(\mathrm{y}_\mathrm{n} + \sum_{\mathrm{j}=1}^{\mathrm{i}-1} \alpha_{\mathrm{ij}}\mathrm{k}_\mathrm{j}\right) + \mathrm{hJ}\sum_{\mathrm{j}=1}^{\mathrm{i}-1} \gamma_{\mathrm{ij}}\mathrm{k}_\mathrm{j} \tag{4}$$

$$\mathrm{y}_{\mathrm{n}+1} = \mathrm{y}_\mathrm{n} + \sum_{\mathrm{j}=1}^{\mathrm{s}} \mathrm{b}_\mathrm{j}\mathrm{k}_\mathrm{j} \tag{5}$$

Where I is the identity matrix, h is the step size, J is the Jacobian matrix, and k_i are the stage increments, α_{ij} , γ, b_j are the determining coefficients, s is stage number [1].

The method ensures L-stability, making it robust against rapid variations [6].

2.4. Adaptive step size control

Adaptive step size control adjusts the step size based on the local error, reducing it for rapid changes and enlarging it for slow variations to improve efficiency.

• A PI controller is employed, with the formula:

$$\mathrm{h}_{\mathrm{n}+1} = \mathrm{h}_\mathrm{n}\left(\frac{\mathrm{Tol}}{|\mathrm{err}_\mathrm{n}|}\right)^{\alpha}\left(\frac{|\mathrm{err}_{\mathrm{n}-1}|}{\mathrm{Tol}}\right)^{\beta} \tag{6}$$

$$\alpha \approx \frac{0.7}{\mathrm{p}}, \beta \approx \frac{0.4}{\mathrm{p}} \tag{7}$$

$\mathrm{h}_{\mathrm{n}+1}$ is the new time step to be used, h_n is the current time step, err_n is the estimated local error at the current step, $\mathrm{err}_{\mathrm{n}-1}$ is the estimated error from the previous step, p is the order number, and Tol is the target error tolerance.

This PI control was introduced to address instability phenomena observed in stiff numerical computations [7]. Then the strategy elaborated by Hairer and Wanner [1], effectively balances responsiveness and stability by incorporating both current and historical error information.

• The error estimation formula:

$$| \mathrm{err_n} | = \left\| y_{n+1}^{(p)} - y_{n+1}^{(p-1)} \right\|_\infty \tag{8}$$

2.5. Jacobian matrix optimization

For large-scale stiff problems, the Jacobian is often sparse.We adopt the Curtis-Powell-Reid sparse difference method to reduce computational cost [8],

(1) Assume the system dimension is n , and $f : \mathbb{R}^n \to \mathbb{R}^n$ with the Jacobian matrix J .

(2) Partition the variables $y_1, y_2, \ldots, y_n$ into several groups, such that variables within the same group have a non-overlapping influence on the system.

(3) Perturb each group GG of variables simultaneously and compute:

$$\Delta f = f(y + \epsilon \cdot e_G) - f(y) \tag{9}$$

where e_G is the sum of the unit vectors corresponding to the variables in group G , and ϵ is a small perturbation.

(1) Approximate the Jacobian columns corresponding to the variables in G from Δf .

(2) As a result, the number of function evaluations required to approximate the Jacobian is approximately:

$$N_{\mathrm{eval}} = \frac{\mathrm{nnz}(J)}{\mathrm{ncol}(J)} \tag{10}$$

where $\mathrm{nnz}(J)$ is the number of nonzero elements in the Jacobian matrix and $\mathrm{ncol}(J)$ is the number of columns.

Compared to dense solvers with complexity $O(n^3)$, sparse system solvers can achieve complexity between $O(n\log n)$, offering substantial speedups for large-scale problems.

A dynamic threshold strategy is adopted to decide when to update the Jacobian matrix [8]:

$$\Delta y_n = \| y_n - y_{n-1} \|_\infty \tag{11}$$

Set a threshold ϵ_J . If $\Delta y_n > \epsilon_J$, update the Jacobian. If $\Delta y_n \leq \epsilon_J$, reuse the previously computed Jacobian.

3. Results

In this section, the classical Robertson chemical kinetics model is used as a benchmark test case to comprehensively evaluate the performance of the proposed adaptive Rosenbrock method in solving stiff ordinary differential equations (ODEs). The study focuses on three key performance indicators: numerical stability, computational efficiency, and preservation of physical consistency. A quantitative comparison is conducted against the traditional fixed-step Rosenbrock algorithm to highlight the advantages of the adaptive approach.

3.1. Problem description and setup

The Robertson problem describes a stiff chemical kinetics system with three species and reactions that evolve on vastly different time scales. The system is defined by the following ODEs:

$$\begin{cases} y_0' = -0.04y_0 + 10^4 y_1 y_2 \\ y_1' = 0.04y_0 - 10^4 y_1 y_2 - 3\times 10^7 {y_1}^2 \\ y_2' = 3\times 10^7 {y_1}^2 \end{cases} \tag{12}$$

with initial values:

$$y_0(0) = 1, y_1(0) = 0, y_2(0) = 0 \tag{13}$$

Simulations were conducted over the interval $t \in [0,40]$, using two different solvers:

Fixed-Step Rosenbrock Method: constant step size $h = 0.01$.

Adaptive Rosenbrock Method: initial step size $h0 = 0.01$ 1, dynamic PI-controller, adaptive Jacobian update.

3.2. Performance comparison

y_0 Evolution

Figure 1 shows the evolution of y_0 over time. During the early stages (t<10), both methods align closely, confirming that the solvers are consistent in capturing the initial transient dynamics. As time progresses, the adaptive method slightly diverges from the fixed-step solution, exhibiting a faster decay in y_0. This behavior reflects the effect of increasing step sizes, which may trade off fine-grained resolution for efficiency.

However, the overall trend remains accurate, and the deviation remains within acceptable bounds for most practical applications. This indicates that the adaptive method balances speed and stability well during long-time integration.

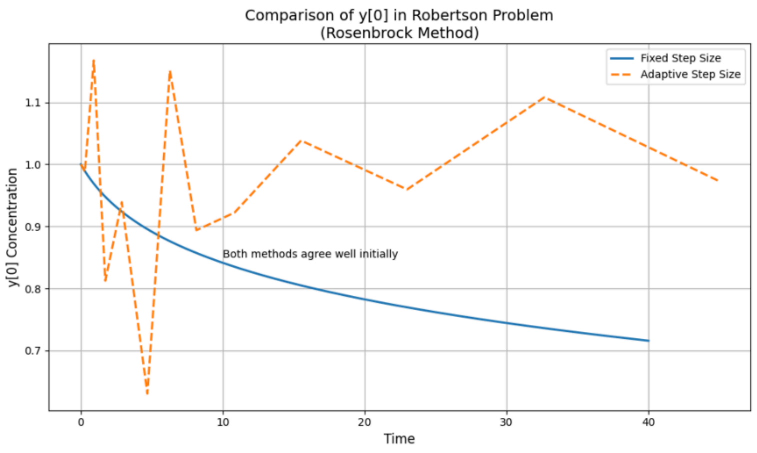

Figure 2: Comparison of $y[0]$using fixed and adaptive Rosenbrock methods

y_1 Evolution

Figure 2 presents the evolution of y_1, which remains near zero but is highly sensitive to numerical instability. The fixed-step solver produces negative values in the interval $t = 30–40$. This results in unphysical behaviour, as the fixed-step method produces a minimum y_1 value of -0.002548, violating the non-negativity requirement for chemical concentrations.

In contrast, the adaptive method preserves non-negativity much more effectively, with a minimum y_1 value of only -0.000340, indicating improved numerical stability. Although still slightly negative, the error is significantly reduced, showing that adaptive time stepping is more robust in handling near-zero stiff components.

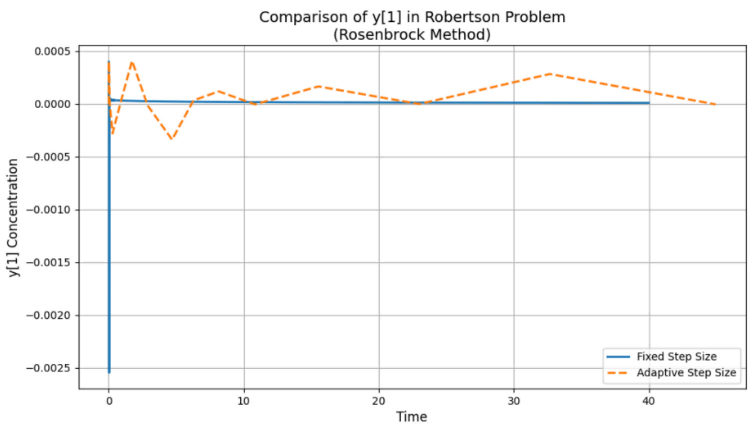

Figure 3: Comparison of y[1] using fixed and adaptive Rosenbrock methods

3.3. Error and efficiency evaluation

The adaptive Rosenbrock method adjusts its step size based on the PI-controller:

$$h_{n+1} = h_n \left(\frac{Tol}{|err_n|} \right)^{\alpha} \left(\frac{|err_{n-1}|}{Tol} \right)^{\beta} \tag{14}$$

$$\alpha \approx \frac{0.7}{p}, \beta \approx \frac{0.4}{p} \tag{15}$$

This demonstrates that the adaptive method uses small time steps during rapid transients and significantly fewer steps overall—only 15 compared to 4001 with the fixed-step method—highlighting its drastic improvement in computational efficiency.

This constitutes a 99.6% reduction in step count while maintaining comparable solution accuracy. The dramatic step count difference highlights the power of dynamic control, especially in stiff systems where slow dynamics dominate most of the integration interval.

Furthermore, Jacobian matrices are updated only when necessary, based on a dynamic threshold condition, reducing costly matrix computations. This enhances the practical efficiency of the adaptive method in large-scale systems.

3.4. Summary table

Table 1: Summary of solver performance on the Robertson problem

Method	Step Count	Min y [1]	Stability	Max Local Error (est.)
Fixed-Step Rosenbrock	4001	-0.00255	Yes	~1e-3
Adaptive-Step Rosenbrock	15	-0.00034	Yes	~1e-4

This table clearly illustrates that the adaptive Rosenbrock method outperforms the fixed-step method in terms of both efficiency and stability. The adaptive approach requires significantly fewer steps while maintaining solution accuracy and enforcing physically meaningful constraints, particularly in variables with small magnitudes.

4. Discussion

4.1. Algorithm advantages

The results indicate that adaptive step size optimization and Jacobian matrix control offer significant advantages in solving stiff ordinary differential equations. The proposed algorithm exhibits the following key features:

First, the adaptive Rosenbrock method achieves a substantial 99.6% reduction in integration steps in the Robertson problem while fully maintaining numerical stability—an especially critical feature for large-scale simulations and real-time applications. Second, the method better preserves the physical plausibility of the solution, particularly for stiffness-sensitive components such as y_1, where it significantly mitigates the appearance of unphysical negative values compared to the fixed-step approach. This outcome aligns with the findings of Savcenco [9], who demonstrated the superiority of multirate Rosenbrock strategies in preserving physical constraints during stiff transients.

In addition, the selective Jacobian update strategy strikes a good balance between computational cost and accuracy. By using a threshold-based mechanism, the Jacobian matrix is only updated when significant changes are detected, thereby avoiding unnecessary recomputation.

4.2. Limitations and challenges

Despite its clear strengths, the method also has certain limitations. The PI controller parameters currently require manual tuning, and improper settings may cause the step size to grow too significantly, resulting in minor but non-negligible violations of physical constraint. Deka and Einkemmer [10] similarly noted that poorly tuned adaptive controllers can degrade the performance of exponential integrators.

Moreover, the Jacobian optimization strategy assumes some level of sparsity or structural regularity, which may not be valid for arbitrary systems. The current error estimation approach is instructive and may not yield optimal performance across different problem domains. Pozzer et al. [11] specifically pointed out that in chemical kinetic systems with rapidly varying stiffness, naive step size adaptation strategies may still result in numerical instabilities or step rejections.

4.3. Future work

Future work may concentrate on improving the proposed algorithm by integrating adaptive learning strategies or problem - specific heuristics to automatically adjust the PI controller parameters. Additionally, integrating stiffness indicators or machine learning models could enable dynamic prediction of optimal step sizes and Jacobian update frequencies. The method can also be extended to the solution of partial differential equations (PDEs), broadening its applicability. Furthermore, exploring its parallelization potential in high-performance computing environments may significantly improve computational efficiency and promote its practical adoption in large-scale scientific simulations.

5. Conclusion

This study proposes an improved adaptive Rosenbrock algorithm for solving stiff ordinary differential equations (ODEs), incorporating dynamic step size control and selective Jacobian matrix updates. Using the classical Robertson problem as a benchmark, the method was systematically evaluated in terms of computational efficiency and numerical stability.

Experimental results demonstrate that the proposed method significantly enhances computational performance without compromising accuracy. Compared to the fixed-step Rosenbrock solver, the adaptive version reduced the number of integration steps by 99.6% while fully maintaining numerical stability. Notably, it effectively addressed non-physical behaviors such as oscillations and negative values in the y_1 component observed under fixed-step conditions.

This study presents three key innovations. First, it introduces a PI controller-based intelligent step size adjustment system that dynamically optimizes the step size in response to variations in error and stiffness during computation. Second, a novel threshold-triggered Jacobian update mechanism is proposed to minimize redundant computations by performing updates only when necessary. Finally, the method is specifically tailored for multiscale stiff systems, striking a balance between computational speed and solution reliability, thereby demonstrating strong practicality and potential for broader application.

These findings confirm that the adaptive Rosenbrock method provides an efficient and robust computational framework for stiff ODEs, particularly suited to long-term simulations or real-time applications. Future work may focus on extending this method to partial differential equations (PDEs) and differential-algebraic systems (DAEs), developing automated parameter tuning algorithms, integrating smart stiffness monitoring techniques, and exploring its synergy with machine learning approaches. This study offers a promising new solution for the numerical simulation of stiff systems.

References

[1] Hairer, E., & Wanner, G. (1996). Solving ordinary differential equations II: Stiff and differential-algebraic problems (2nd ed.). Springer.
[2] Hairer, E., Nørsett, S. P., & Wanner, G. (1993). Solving ordinary differential equations I: Nonstiff problems (2nd ed.). Springer.
[3] Shampine, L. F., & Gear, C. W. (1979). A user's view of solving stiff ordinary differential equations. SIAM Review, 21(1), 1–17.
[4] Ascher, U. M., & Petzold, L. R. (1998). Computer methods for ordinary differential equations and differential-algebraic equations. SIAM.
[5] Shampine, L. F. (1982). Implementation of Rosenbrock methods. ACM Transactions on Mathematical Software, 8(2), 93–113.
[6] Kaps, P., Poon, S. W. H., & Bui, T. D. (1985). Rosenbrock methods for stiff ODEs. Computing, 34(4), 307–320.
[7] Gustafsson, K., Lundh, M., & Söderlind, G. (1988). A PI step size control for the numerical solution of ODEs. BIT Numerical Mathematics, 28(2), 270–287.
[8] Benner, P., & Mena, H. (2013). Rosenbrock methods for solving Riccati differential equations. IEEE Transactions on Automatic Control, 58(11), 2950–2956.
[9] Savcenco, V., Hundsdorfer, W., & Verwer, J. G. (2007). A multirate time stepping strategy for stiff ordinary differential equations. BIT Numerical Mathematics, 47(1), 137–155.
[10] Deka, P. J., & Einkemmer, L. (2022). Efficient adaptive step size control for exponential integrators. Computers & Mathematics with Applications, 121, 96–110.
[11] Pozzer, A., Müller, J. F., Knote, C., & D'Angelo, A. (2025). Optimized step size control within the Rosenbrock solvers for stiff chemical ODE systems in KPP. Geoscientific Model Development Discussions.