Numerical Solutions of Nonlinear Ordinary Differential Equations by Using Adaptive Runge-Kutta Method

Sammie L. Clayton[1], Mulatu Lemma[2], Abhinandan Chowdhury*

Department of Mathematics, Savannah State University, Savannah, GA 31404, USA

[1]sclayto1@student.savannahstate.edu, [2]lemmam@savannahstate.edu

Abstract

We present a study on numerical solutions of nonlinear ordinary differential equations by applying Runge-Kutta-Fehlberg (RKF) method, a well-known adaptive Runge-kutta method. The adaptive Runge-kutta methods use embedded integration formulas which appear in pairs. Typically adaptive methods monitor the truncation error at each integration step and automatically adjust the stepsize to keep the error within prescribed limit. Numerical solutions to different nonlinear initial value problems (IVPs) attained by RKF method are compared with corresponding classical Runge-Kutta (RK4) approximations in order to investigate the computational superiority of the former. The resulting gain in the efficiency is compatible with theoretical prediction. Moreover, with the aid of a suitable time-stepping scheme, we show that the RKF method invariably requires less number of steps to arrive at the right endpoint of the finite interval where the IVP is being considered.

Keywords: Embedded Runge-Kutta methods, Duffing oscillator, Adaptive time-stepping schemes.

Introduction

The Runge-Kutta methods are an important family of predictor-corrector methods for approximation of solutions of ordinary differential equations (ODEs) which were developed by German mathematicians duo C. Runge (1856–1927) and M. W. Kutta (1867–1944). Let's consider an initial value problem (IVP)

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t, \boldsymbol{u}(t)) \text{ where } \boldsymbol{u}(t) = (u_1(t), u_2(t), \ldots, u_n(t))^{\mathrm{T}}, f \in [a,b] \times \mathbb{R}^n \to \mathbb{R}^n \tag{1}$$

with initial condition $\boldsymbol{u}(0) = \boldsymbol{u}_0$.

To numerically approximate the continuously differentiable solution $\boldsymbol{u}(t)$ of the IVP over the time interval $t \in [a,b]$, we subdivide the interval into N equal subintervals and select the mesh points $t_j = a + jh$, $j = 0, 1, \ldots, N$ and $h = (b-a)/N$, where h is called a step size.

The family of explicit Runge-Kutta (RK) methods of the m-th stage is given by

$$\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum_{i=1}^{m} c_i k_i,$$

$$\begin{aligned} \text{where} \qquad & k_1 = f(t_n, \boldsymbol{u}_n), \\ & k_2 = f\left(t_n + \alpha_2 h \boldsymbol{u}_n + h\beta_{21} k_1(t_n, \boldsymbol{u}_n)\right), \\ & k_3 = f\left(t_n + \alpha_3 h, \boldsymbol{u}_n + h\Big(\beta_{31} k_1(t_n, \boldsymbol{u}_n) + \beta_{32} k_2(t_n, \boldsymbol{u}_n)\Big)\right) \\ \text{and, in general,} \qquad & k_m = f\left(t_n + \alpha_m h + \boldsymbol{u}_n + h \sum_{i=1}^{m-1} \beta_{mj} k_j\right). \end{aligned}$$

To employ a specific method, the number of stages (m, in integer), the coefficients α_i (for $i = 2, 3, \ldots, m$), β_{ij} (for $1 \le j < i \le m$) and c_i (for $i = 1, 2, \ldots, m$) are required to be provided which are usually arranged in a tabular form, known as *Butcher tableau* (named after J. C. Butcher).

The RK4 method (with 4th-order convergence rate) represents one of the solutions corresponding to the stage $m = 4$. In fact, for $m \le 4$, the convergency order $p = m$. But, beyond that the amount of stages m grows faster than p. So, after some point (due to so-called *Butcher Barrier*), it becomes unreasonable to increase the convergency order p in order to increase the level of accuracy [2]. Which is why, the alternative stepsize adjustment algorithm based

*Corresponding author. Email: chowdhury@savannahstate.edu.

on the *embedded Runge-Kutta formulas* has turned out to be so effective in improving the accuracy. This method is developed by Fehlberg and is called the Runge-Kutta-Fehlberg methods (RKF) [5]. In this context, we must note that there exists another straightforward technique suitable for adaptive stepsize control in fourth-order Runge-Kutta which is known as the method of step-doubling (also referred as the Local Error method [10]). This method involves solving the problem twice using step sizes h and $h/2$ and comparing answers at the grid points corresponding to the larger step size. But it requires substantial amount of computation for smaller step size. Furthermore, computation must be repeated if agreement does not attain the desired level. The main difference between the RKF scheme and step-doubling scheme lies in use of the function in estimating the error. In step-doubling technique, the error estimation does not involve the evaluation of the function, while in RKF function must be advanced by using same evaluation points in order to estimate the error. This computational feature of step-doubling technique is considered to be advantageous, but in practice, algorithms based on embedded Runge-Kutta formulas are found to be more efficient performer. For more detail, we refer the reader to [9] for a comparative study between these two aforementioned techniques.

In the present study, we start reviewing the embedded Runge-Kutta formulas and run the numerical simulations by using algorithms based on RKF method on the test as well as real-world problems. Comparison will be made in the testing problems to demonstrate the enhanced computational efficiency of RKF method over the RK4 method. The main objective of this study is the practical demonstration of the adaptive stepsize control phenomenon of RKF method. In order to achieve this, we evaluate the grid points during the progress of the computations determined by prescribed adaptive step-size control strategy in a tabular form. In the same table, we also present the RK4 solutions with uniform distribution of the discretization grid points (uniform stepsize) over a finite interval. For a real-world problem, we have selected well-known Duffing equation, which is a nonlinear second-order differential equation discovered by German electrical engineer Georg Duffing. This equation which models Duffing oscillator with damping has been attracting remarkable research interest over the past few decades due to its applicability to numerous engineering fields, such as magneto-elastic mechanical system [6], large amplitude oscillation of centrifugal governor systems, nonlinear vibration of beam and plates and fluid flow induced vibrations [1]. We have also used another version of Duffing equation as one of the test examples which does not involve the damping term - which allows it to admit an exact solution. Typically, adaptive methods are not suitable for numerically solving eq. (1) if $\boldsymbol{f}(t,\boldsymbol{u}(t))$ contains discontinuity. The simulation may get stuck in an infinite loop while finding the appropriate value for the stepsize, which results in discrepancy occurring in estimated error. But, it should not be a concern for the problems in the current study, since all sought solutions are smooth and continuous in nature.

Description of the Method

The adaptive stepsize control algorithm proceeds in the following manner: at each step, two different approximations for the solution are found – one by using *a fourth-order method with five stages* and another with *a fifth-order method with six stages*.

The general form of a fifth-order Runge-Kutta formula is

$$\begin{aligned}
k_1 &= hf(t_n, u_n)\\
k_2 &= hf(t_n + a_2 h, u_n + b_{21}k_1)\\
&\cdots\\
k_6 &= hf(t_n + a_6 h, u_n + b_{61}k_1 + \cdots + b_{65}k_5)\\
y_{n+1} &= y_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + O(h^6)
\end{aligned}$$

The embedded fourth-order formula is

$$y^*_{n+1} = y_n + d_1k_1 + d_2k_2 + d_3k_3 + d_4k_4 + d_5k_5 + d_6k_6 + O(h^5)$$

and the error estimate is

$$\Delta = y_{n+1} - y^*_{n+1} = \sum_{i=6}^{6}(c_i - d_i)k_i$$

It must be noted that particular values of the various coefficients in the above equations are not unique. The table (1) gives the coefficients found by Cash and Karp [3]. Their values are preferred over the original values given by Fehlberg since it yields more efficient method with better error estimates.

Next, at each step, two attained approximations are compared. Absolute difference between these approximations gives the local interpolation or per-step error, ε. Error control is achieved by adjusting the increment h so that per-step error is approximately equal to a prescribed tolerance ε_{tol}. What follows is if the two answers are in close agreement, the approximation is accepted. If the two answers do not agree to a prescribed accuracy, the stepsize is reduced. If the two answers agree to more significant digits than required, the stepsize in increased. For example, if for current stepsize h_1, the produced error is ε, the corresponding optimal stepsize h_2 is estimated as $h_2 = \delta h_1(\varepsilon_{\text{tol}}/\varepsilon)^{0.2}$ where δ

i	a_i	b_{ij}					c_i	d_i
0	–	–	–	–	–	–	$\frac{37}{378}$	$\frac{2825}{27648}$
1	$\frac{1}{5}$	$\frac{1}{5}$	–	–	–	–	0	0
2	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	–	–	–	$\frac{250}{621}$	$\frac{18575}{48384}$
3	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$	–	–	$\frac{125}{594}$	$\frac{13525}{55296}$
4	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$	–	0	$\frac{277}{14336}$
5	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$

Table 1: Cash-Karp Parameters for Embedded Runge-Kutta Method

is a safety factor (≈ 1). If $\varepsilon > \varepsilon_{\text{tol}}$, the stepsize has to be decreased, and $\varepsilon < \varepsilon_{\text{tol}}$, the stepsize needs to be increased in the next step. For coding purpose we have used the following scheme [7]:

$$h_{\text{next}} = \begin{cases} \delta h\left(\dfrac{\varepsilon_{\text{tol}}}{\varepsilon}\right)^{0.2}, & \varepsilon \geq \varepsilon_{\text{tol}}, \\ \delta h\left(\dfrac{\varepsilon_{\text{tol}}}{\varepsilon}\right)^{0.25}, & \varepsilon < \varepsilon_{\text{tol}}, \end{cases}$$

where the value of δ is taken as 0.9.

Numerical Results and Discussion

In this section, we report a few numerical results concerning the numerical solutions of four test problems (example 1 – example 4) as well as one real-world problem (example 5) where nonlinear ODEs of different degrees are considered. Each higher order ODE is converted to a system of first-order ODEs. For each example, the numerical results are presented by depicting two numerical solution profiles (by RK4 and RKF) along with the profile of corresponding exact solution on the same window. The comparative error estimate for two numerical solutions are also presented. Additionally, we have presented the progression of both numerical solutions along the grid points to demonstrate the adaptive nature of the step-size control for RKF method for each problem. In almost every case, numerical result confirms that the number of steps required for RKF method to keep the error within the prescribed level is substantially less than that of RK4 method.

Example 1

First, we study the first order nonlinear ODE with constant coefficients which models the hybrid selection phenomenon as follows

$$\frac{\mathrm{d}u}{\mathrm{d}t} = ku(1-u)(2-u), \qquad u(0) = 0.5, \tag{2}$$

where k is a positive constant that depends on the genetic characteristic. In this hybrid model, $u(t)$ is the portion of population of a certain characteristic, and t is the time measured in generations. This equation admits the exact solution $u(t) = 1 - 1/\sqrt{1+3e^{2kt}}$ which satisfies the given initial condition $u(0) = 0.5$.

Example 2

Let's consider the following nonlinear initial value problem (IVP)

$$\frac{\mathrm{d}^2u}{\mathrm{d}t^2} = \left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)^2 - 1, \quad \text{with } u(0) = 0,\ u'(0) = \frac{e^2-1}{e^2+1}. \tag{3}$$

The exact solution of this initial value problem is $u(t) = t - \ln(e^{2t}+e^2) + \ln(1+e^2)$.

By letting

$$\boldsymbol{u} = \begin{bmatrix} u_0 & u_1 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u & u' \end{bmatrix}^{\mathrm{T}}$$

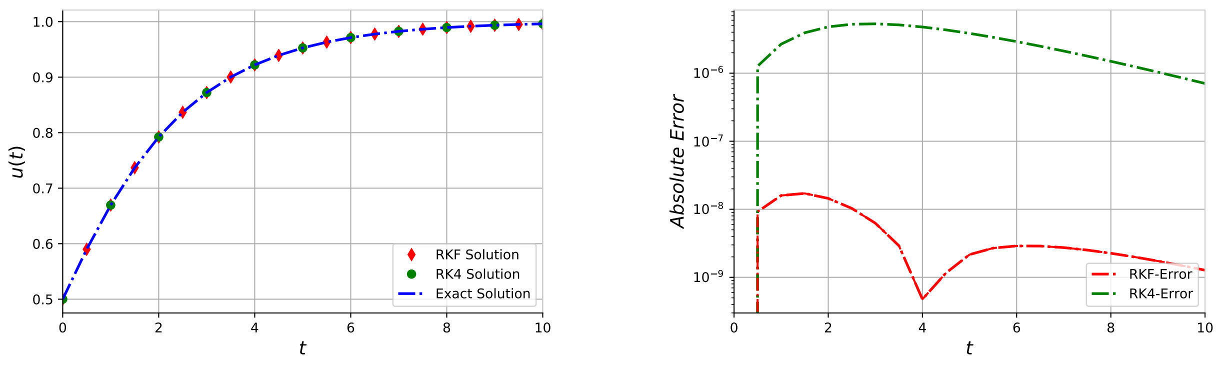

Figure 1: Left: The behavior of the exact and numerical solutions of eq. (2). Right: Absolute errors $\left(|u_a - u_{\text{RK4}}| \text{ and } |u_a - u_{\text{RKF}}|\right)$ between numerical solutions and the exact solution. Maximum absolute error 1.71298 E−08 (by RKF method) and 5.32023 E−06 (by RK4 method).

Stepsize used	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
$u(t) \times$ E01 RK4 Method	5.0000	6.6960	7.9223	8.7223	9.2210	9.5266	9.7126	9.8257	9.8942	9.9359	9.9611
Stepsize used	0.0	0.5	2.470	4.465	6.491	8.886	10.000				
$u(t) \times$ E01 RKF Method	5.0000	5.8991	8.3440	9.3816	9.7751	9.9320	9.9611				

Table 2: Comparison for numbers of steps between RK4 and RKF methods required to approximate $u(t)$ at the rightmost boundary value $t = 1$ of the chosen interval $[0, 10]$ for IVP (2). Per-step error tolerance of 10 E−04 is used.

the equivalent first-order equations become

$$\boldsymbol{u}' = \begin{bmatrix} u_0' & u_1' \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u_1 & (u_1^2 - 1) \end{bmatrix}^{\mathrm{T}}.$$

Also, the initial conditions are

$$\boldsymbol{u}(0) = \begin{bmatrix} u_0(0) & u_1(0) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & (e^2 - 1)/(e^2 + 1) \end{bmatrix}^{\mathrm{T}}$$

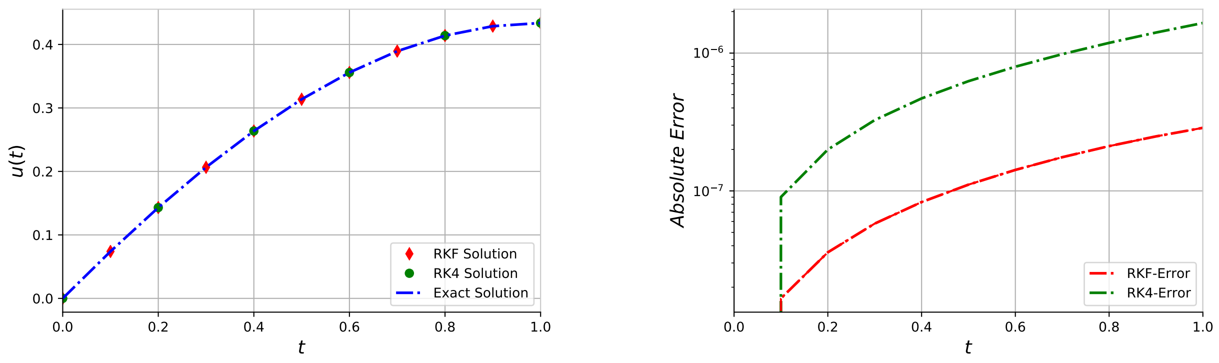

Figure 2: Left: The behavior of the exact and numerical solutions of eq. (3). Right: Absolute errors $\left(|u_a - u_{\text{RK4}}| \text{ and } |u_a - u_{\text{RKF}}|\right)$ between numerical solutions and the exact solution. Maximum absolute error 2.86821 E−07 (by RKF method) and 1.65371 E−06 (by RK4 method).

Stepsize used	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$u(t) \times$ E01 RK4 Method	0.00	0.7395	1.4302	2.0651	2.6365	3.1367	3.5583	3.8944	4.1391	4.2879	4.3378

Stepsize used	0.0	0.1	0.6960	1.00							
$u(t) \times$ E01 RKF Method	0.00	0.7395	3.8826	4.3379							

Table 3: Comparison for numbers of steps between RK4 and RKF methods required to approximate $u(t)$ at the rightmost boundary value $t = 1$ of the chosen interval $[0,1]$ for IVP (3). Per-step error tolerance of $10\text{E}{-}04$ is used.

Example 3

Consider Duffing equation which is a second order nonlinear ODE

$$\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} + 3u - 2u^3 = \cos(t)\sin(2t), \qquad \text{with } u(0) = 0,\ u'(0) = 1. \tag{4}$$

Similarly, by letting

$$\boldsymbol{u} = \begin{bmatrix} u_0 & u_1 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u & u' \end{bmatrix}^{\mathrm{T}}$$

the equivalent first-order equations become

$$\boldsymbol{u}' = \begin{bmatrix} u_0' & u_1' \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u_1 & \left(2u_0^3 - 3u_0 + \cos(t)\sin(2t)\right) \end{bmatrix}^{\mathrm{T}},$$

and, the initial conditions are

$$\boldsymbol{u}(0) = \begin{bmatrix} u_0(0) & u_1(0) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathrm{T}}$$

The exact solution of the IVP (4) is $u(x,t) = \sin(t)$.

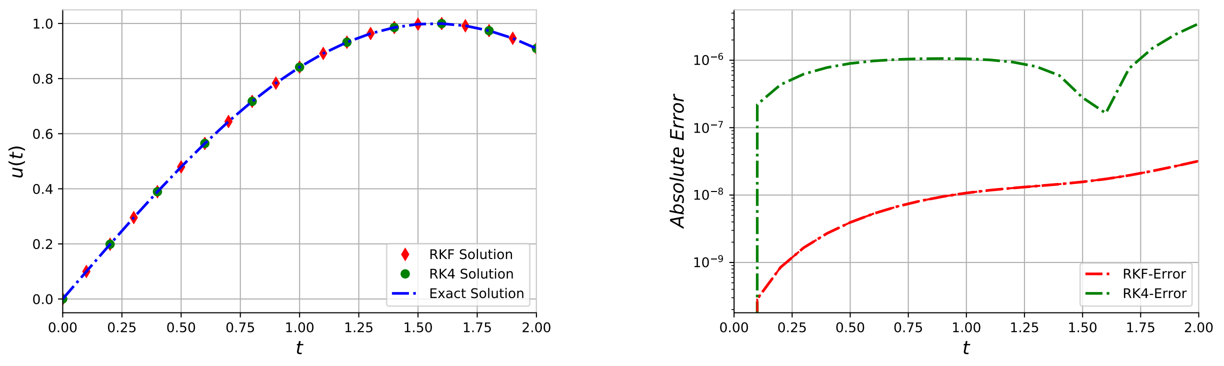

Figure 3: Left: The behavior of the exact and numerical solutions of eq. (4). Right: Absolute errors $\left(|u_a - u_{\mathrm{RK4}}| \text{ and } |u_a - u_{\mathrm{RKF}}|\right)$ between numerical solutions and the exact solution. Maximum absolute error $3.20104\text{E}{-}08$ (by RKF method) and $3.49696\text{E}{-}06$ (by RK4 method).

Example 4

Let's consider the following fourth-order nonlinear IVP

$$\frac{\mathrm{d}^4 u}{\mathrm{d}t^4} - 16u - 24u^5 = 40\tan(t), \qquad \text{with } u(0) = 0,\ u'(0) = 1,\ u''(0) = 1,\ u'''(0) = 2. \tag{5}$$

Again, by assuming similarly, by letting

$$\boldsymbol{u} = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u & u' & u'' & u''' \end{bmatrix}^{\mathrm{T}}$$

Stepsize used	0.00	0.20	0.40	0.60	0.80	1.00	1.20	1.40	1.60	1.80	2.00
$u(t)\times$E01 RK4 Method	0.00	1.9867	3.8942	5.6464	7.1736	8.4147	9.3204	9.8545	9.9957	9.7385	9.0930
Stepsize used	0.00	0.1000	0.6251	1.1551	1.7163	2.0000					
$u(t)\times$E01 RKF Method	0.00	0.9983	5.8517	9.1489	9.8952	9.0944					

Table 4: Comparison for numbers of steps between RK4 and RKF methods required to approximate $u(t)$ at the rightmost boundary value $t = 2$ of the chosen interval $[0,2]$ for IVP (4). Per-step error tolerance of 10E−04 is used.

the equivalent first-order equations become

$$\boldsymbol{u}' = \begin{bmatrix} u_0' & u_1' & u_2' & u_3' \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u_1 & u_2 & u_3 & (24u_0^5 + 16u_0 + 40\tan^3(t)) \end{bmatrix}^{\mathrm{T}},$$

and, the initial conditions are

$$\boldsymbol{u}(0) = \begin{bmatrix} u_0(0) & u_1(0) & u_2(0) & u_3(0) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 2 \end{bmatrix}^{\mathrm{T}}$$

The exact solution of the IVP (5) is $u(x,t) = \tan(t)$.

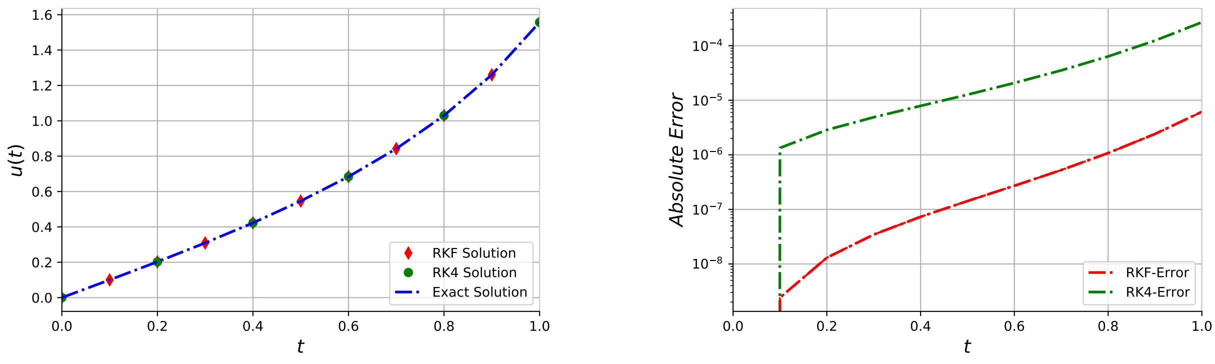

Figure 4: Left: The behavior of the exact and numerical solutions of eq. (5). Right: Absolute errors $\left(|u_a - u_{\mathrm{RK4}}|\right.$ and $\left.|u_a - u_{\mathrm{RKF}}|\right)$ between numerical solutions and the exact solution. Maximum absolute error 6.14703E−06 (by RKF method) and 2.69330E−04 (by RK4 method).

Stepsize used	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$u(t)$ RK4 Method	0.00	0.1003	0.2027	0.3093	0.428	0.5463	0.6841	0.8423	1.0296	1.2600	1.5571
Stepsize used	0.0	0.1	0.3547	0.5380	0.6818	0.7959	0.8880	0.9633	1.00		
$u(t)$ RKF Method	0.0000	0.1003	0.3704	0.5967	0.8116	1.0213	1.2296	1.4384	1.5574		

Table 5: Comparison for numbers of steps between RK4 and RKF methods required to approximate $u(t)$ at the rightmost boundary value $t = 1$ of the chosen interval $[0,1]$ for IVP (5). Per-step error tolerance of 10E−04 is used.

Example 5

The general form of Duffing oscillator with the damping effect is given in the following form:

$$\frac{\mathrm{d}^2u}{\mathrm{d}t^2} + \alpha\frac{\mathrm{d}u}{\mathrm{d}t} + \beta u + \gamma u^3 = f(t) \tag{6}$$

with initial conditions, $u(0) = a$, and $u'(0) = b$. Here, α, β, γ, a and b are real constants and $f(t)$ is a real valued function. Eq. (6) involves damping (the first derivative term) which means the amplitude of oscillations will reduce over time, which will subsequently give a non-conservative system. The coefficient α controls the magnitude of damping. Other two coefficients β and γ in the L.H.S. of eq. (6) are responsible for stiffness and the nonlinearity in the restoring force respectively.

Usually non-conservative oscillators can not be handled by semi-analytical methods – though two relatively new methods, one based on Laplace decomposition algorithm [11] and another based on modified differential transform method [8] have found some success while approximating the free responses of Duffing oscillator with different damping effects. In fact, in latter, the authors found solutions approximated by a semi-analytical method which is nothing but an intelligent combination of Differential transforms method, Laplace transform and Pade approximation. An efficient scheme based on modified Adomian decomposition method is proposed in [4] to obtain some accurate closed-form approximations of solutions to nonlinear duffing oscillator. Among numerical methods, RK4 method is considered as an effective method from the accuracy point of view, as quite often it is used as a benchmark solution. For the lack of exact solutions for eq. (6), we have used the analytical solution obtained in [8] for comparison purpose which is given below in (7).

$$u_a(t) = e^{-0.60107t}\big(0.000033846\cos(15.0816) - 0.00027134\sin(15.08t)\big) + 0.099966\,e^{-0.24894t}\cos(5.0125t), \qquad (7)$$

by choosing the equation parameters as $\alpha = 0.5$, $\beta = \gamma = 25$, $a = 0.1$, $b = 0$ and $f(t) = 0$ for eq. (6).

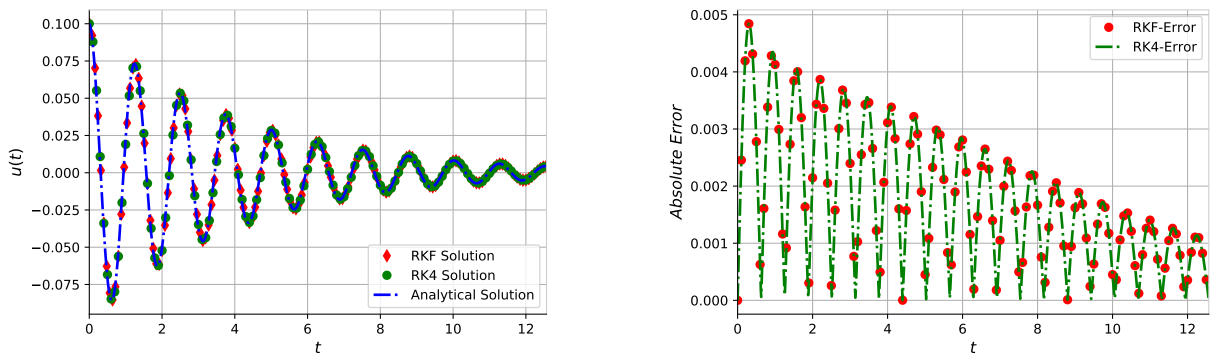

Figure 5: Left: The behavior of the analytical solution (7) and numerical solutions of eq. (6). Right: Absolute errors $\big(|u_a - u_{\text{RK4}}|$ and $|u_a - u_{\text{RKF}}|\big)$ between numerical solutions and the analytical solution. Maximum absolute error 4.84305 E−03 (by RKF method) and 4.84306 E−03 (by RK4 method).

Stepsize used	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$u(t) \times$ E01 RK4 Method	1.0000	0.8785	0.5519	0.1090	−0.3383	−0.6828	−0.8469	−0.7995	−0.5626	−0.2026	0.1899
Stepsize used	0.0	0.1	0.283	0.463	0.675	0.858	1.0				
$u(t) \times$ E01 RKF Method	1.0000	0.8785	0.1890	−0.5746	−0.8316	−0.3609	0.1907				

Table 6: Comparison for numbers of steps between RK4 and RKF methods required to approximate $u(t)$ at the rightmost boundary value $t = 1$ of the chosen interval $[0, 1]$ for eq. (6). Per-step error tolerance of 10 E−04 is used.

Conclusions

In this note we have revisited an embedded Runge-Kutta method proposed by Fehlberg which performs extra steps in order to estimate the error and adjusts the stepsize in an adaptive fashion. Adaptive time-stepping strategy makes it possible to achieve the desired accuracy at a relatively low cost. For all IVPs under considerations, the numerical results exhibit that RKF method provides more accurate approximations than the classical Runge-Kutta method. The amount of improvement on error estimation is in good agreement with the theoretical expectation. Moreover, in almost every example, the scheme based on RKF method is required to generate less number of approximations to arrive at the right endpoint of the finite interval - thus demonstrating the effectiveness of the adaptive stepsize control feature of the method. Though the cost per time step increases substantially, adaptive time-stepping enhances the robustness of the code, the overall efficiency and the credibility of the simulation results. Despite of their popularity in solving linear and nonlinear systems of ODEs which can be attributed to their computational efficiency as well as their self-starting nature, they suffer from one serious issue which limits their ability to derive high order approximations. It stems from the fact that they seek out a lot of information about the solution during one step, but all of that information is discarded before the start of the next step. In future, we would like to investigate the computational performances of various *multi-step methods* that use information from past steps to try to increase the accuracy of the solution without using additional function evaluations particularly while solving stiff systems of ODEs.

Conflicts of Interest

There are no conflicts of interest.

Acknowledgements

S.L.C. gratefully acknowledges the financial support in the form of research assistantship which was funded by the National Science Foundation (NSF) Catalyst Project Grant (Award Number – 1800798).

References

[1] M. T. Ahmadian, M. Mojahedi, and H. Moeenfard. Free vibration analysis of a nonlinear beam using homotopy and modified Lindstedt-Poincare methods. *J. Solid Mechanics*, 1:29–36, 2009.

[2] J. C. Butcher. On Fifth and Sixth Order Explicit Runge-Kutta Methods: Order Conditions and Order Barriers. *Can. Appl. Math. Q.*, 17(3):433–445, 2009.

[3] J. R. Cash and A. H. Karp. A Variable Order Runge-Kutta Method for Initial-value Problems with Rapidly Varying Right-hand Sides. *ACM Transactions on Mathematical Softwares*, 16:201–222, 1990.

[4] P. K. Das, D. Singh, and M. M. Panja. An Efficient Scheme for Accurate Closed-form Approximate Solution of Some Duffing- and Lienard-type Equations. *Journal of Advances in Mathematics*, 16:8213–8225, 2019.

[5] E. Fehlberg. Low-order classical Runge-Kutta Formulas with stepsize control. Technical report, 1969.

[6] J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems and bifurcations of vector fields*. Springer-Verlag, 1983.

[7] J. Kiusalaas. *Numerical methods in engineering with Python 3*. Cambridge University Press, 3rd edition, 2013.

[8] S. Nourazar and A. Mirzabeigy. Approximate solution for nonlinear Duffing oscillator with damping effect using the modified differential transform method. *Sci. Iran B*, 20(2):364–368, 2013.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77 - The Art of Scientific Computing*. Cambridge University Press, second edition.

[10] O. V. Sinkin, R. Holzlohner, J. Zweck, and C. R. Menyuk. Optimization of the Split-Step Fourier Method in Modeling Optical-fiber Communications Systems. *J. Lightwave Technol.*, 21(1):61–68, 2003.

[11] E. Yusufoglu. Numerical Solution of Duffing Equation by the Laplace Decomposition Algorithm. *Appl. Math. Comput.*, 177(2):572–580, 2006.