JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-20, No.-12, December (2025) pp 114-126 ISSN (Print) 0973-8975

POSITIVE SOLUTIONS OF THE SYSTEM OF FIRST-ORDER DIFFERENTIAL EQUATIONS BY RUNGE-KUTTA METHOD FOURTH ORDER

Ahmed. O. M. Abubaker

Department of Mathematics, University College of Umluj, University of Tabuk, Tabuk 48322, Saudi Arabia.

Department of Mathematics, Faculty of Education, University of Khartoum, Omdurman, 406, Sudan.

Email : aabubaker@ut.edu.sa, wadomar877@gmail.com

Corresponding Author: **Ahmed. O. M. Abubaker**

https://doi.org/10.26782/jmcms.2025.12.00008

(Received: September 08, 2025; Revised: November 26, 2025; December 06, 2025)

Abstract

The Runge-Kutta method, and especially its fourth-order variant (RK4), is perhaps the most widely adopted method for solving ordinary differential equations (ODEs) and their systems. This paper deals specifically with the RK4 method to explain a system of first-order differential equations, and the ability of the method to converge and stabilize positive solutions. It is well known that standard RK4 is both accurate and stable, but to particularly maintain positivity of solutions, where the model represents physical quantities that must be non-negative, such as populations or concentrations, often requires extra techniques.

This paper discusses theoretically the RK4 method and systems, their execution, the need for retention of positivity, and methodologies for retention of positivity. Several illustrative examples are included to demonstrate the application of the method and the difficulty of maintaining positivity as well.

Keywords: Runge-Kutta method, fourth-order**,** systems of first-order differential equations, positive solutions.

I. Introduction

Applying the concepts of engineering and science disciplines often results in systems of ordinary differential equations (ODEs) when multiple variables interact with one another. Such systems can describe population dynamics, electrical circuits, or even chemical reactions. Complex or nonlinear systems tend to require numerical methods rather than analytical solutions, which are preferable but not achievable in most cases [I, XIV].

For ODEs, RK is one of the most popular methods due to its ease of use, stability, and accuracy. RK4 is widely used in many disciplines, due to a good trade-off between computational cost and accuracy [I-II]. Unlike multistep methods, a single-step method increases the simplicity of RK4's application.

This paper is concerned with the RK4 method for systems of first-order differential equations with respect to "positive solutions". In many scenarios, the dependent variables are physical entities that must be non-negative: populations, concentrations, or mass. Therefore, the numerical procedures adopted must not only approximate the solution with enough accuracy, but also preserve both the physical nature and the non-negativity of these quantities throughout the whole simulation. Many standard numerical schemes, some RK methods, and stiff systems with large step sizes tend to produce non-physical negative values [III, XIII]. This paper aims to comprehensively analyze the RK4 method concerning systems of first-order ODEs, address the issue of positivity preservation in numerical solutions, and explain modifications and strategies of RK4 aimed at preserving positive solutions. The researcher documents the positivity-preserving frameworks and foundational concepts alongside the outcomes relevant to their application. In addition, completely solved examples will be given for the practical application of the RK4 method, and the techniques that are needed to ensure that all solutions are positive will be discussed as well.

II. Literature Review

The Runge-Kutta methods, as a family of iterative numerical techniques for estimating the solution of ordinary differential equations, are attributed to Carl Runge and Martin Kutta. Until recently, with the emergence of newer methods, the fourth-order Runge-Kutta method, also referred to as RK4, was one of the most prevalent and celebrated methods due to its reasonable accuracy for many problems. It came into existence over a century ago and to this day, is heavily relied on in the field of numerical analysis for resolving initial value problems (IVPs).

As for the initial works of the Runga-Kutta methods, most of the research was focused on the single ODEs and designed systems of first-order ODEs, which form the key to the construction of multi-component systems. It is possible to construct RK4 in such a way that it can perform parallelized integration for a system of many dependent variables. This makes RK4 a general-purpose method that can be employed across multiple disciplines in science and engineering [I, IX]. For instance, Abraha (2020) analyzed multiple numerical approaches, including the classical fourth-order Runge-Kutta method for solving systems of first-order ordinary differential equations, and remarked that RK4 provided the best accuracy among all tested methods when benchmarked against analytical solutions [IX]. Likewise, Islam (2015) researched the practical effectiveness and accuracy of the fourth-order Runge-Kutta method for solving initial value problems (IVPs) and discussed its practical effectiveness through multiple case studies [I].

The RK4 method is useful for a wide range of problems; however, there is a fundamental issue with systems whose dependent variables must be strictly positive. This is particularly important in population studies, epidemiology, chemical kinetics, and finance, where variables such as population, concentrations, or even stock prices

are expected to be non-negative values. Generally, most numerical methods, including the classical RK4, do not guarantee preservation of positivity. Such non-physical results are common when using sufficiently large step sizes or when the system is stiff [III, XIII]. As an illustration, Redmann and Riedel (n.d.) study the Runge-Kutta methods in [4] have tried to solve rough differential equations that are capable of solving stochastic differential equations, thereby indicating the implicit need for methods that preserve certain attributes of the solution. Most directly, research has focused on positivity-preserving adaptive Runge-Kutta methods, where the method weights are adjusted to enforce bounds and ensure positivity [III, XIII]. Most of these modifications incorporate the use of flux limiters, non-negative steps, or component-wise non-negativity preserving formulations. Developing such methods is important for dependable simulations of phenomena where physically constrained solutions are needed.

The challenge associated with the failure to preserve positivity has given rise to adapted or modified Runge-Kutta methods, which aim to ensure that the numerical solutions are consistent with the domain of interest.

A linear stability analysis of the numerical method requires one to start with the classical test equation $y' = \lambda y$, where any one-step scheme will allow the representation.

$$y_{n+1} = R(h\lambda)y_n,$$

$R(z)$ is the amplification factor giving the response of the spectral parameter of the method to the spectral parameter of the h wavelength. In the case of the classical fourth-order Runge-Kutta scheme $(RK4), R(z)$ is described as the fourth-degree expansion as a Taylor series of the exponential function, that is,

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{3!} + \frac{z^4}{4!}.$$

The region of stability is characterized, and the points where this state is intersected by the negative real axis give the familiar stability region $z \in [-2.7853, 0]$. In consequence, in the case of real negative eigenvalues, $RK4$ is a linearly stable method as long as $h \leq \frac{2.7853}{|\lambda|}$. Therefore, when the eigenvalues are real with negative values, $RK4$ is a linearly stable method on the condition that the value of $h \leq 2.7853/|\lambda|$. As a result, a lineally stable method is the $RK4$ when Einsteinian, and the values of $RK4$ are real and negative.

In these environments, the scalar factor $R(h\lambda)$ is substituted by the matrix polynomial $R(h\lambda)$ whose off-diagonal interactions can still give negative components even when the initial vector is non-negative. When the classical RK4 is used, this is significantly limited by the fact that the value of its strong-stability-preserving (SSP) coefficient is zero, so it is not representable as a convex blend of forward-Euler steps and consequently is not able to provide monotonicity or positivity with arbitrary Metzler matrices.

The interaction between step size h and the influence on the system through the extent and form of eigenvalues is very demanding for the development of the positivity loss.

Large values of h may move certain spectral content to areas outside the range of realistic values of the discrete flow. Well, even the case of a linearly stable approach in the classical sense. It is particularly large in the situation when the systems are highly off-diagonally coupled, such that the $R(hA)$ matrix may exhibit oscillatory or sign-changing behaviour without having such behaviour on the scalar level. One can further explain it by comparing it to A-stable and even SSP time integrators to define the extent of the applicability of RK4. Implicit schemes such as the backward Euler scheme, or the Crank-Nicolson scheme, which are A-stable, can be stable over the entire left half-plane in the complex plane and are therefore useful with stiff problems. Correspondingly, the high-order SSP schemes are explicitly constrained by time-steps to have certain properties that enable them to be positive and also to have a set of further monotonicity properties. RK 4, which again is not A-stable with a positive coefficient of SSP, is howwase optimum fitted to non-stiff problems where positivity is not inherent to the nature of the problem, or where additional positivity-conserving limiters or projection operators are incorporated to give the preferred invariance.

III. Preliminaries

III.i. Systems of First-Order Ordinary Differential Equations

A system of first-order ordinary differential equations can be expressed in the general form:

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \dots, y_n)$$
$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \dots, y_n)$$
$$\dots$$
$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, \dots, y_n)$$

This can be written more compactly in vector form as:

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

Where $y = [y_1, y_2, \dots, y_n]^T$ is a vector of dependent variables, and

$f = [f_1, f_2, \dots, f_n]^T$ is a vector function of x and y. With this system, it is usually required to specify an initial condition $y(x_0) = y_0$ hence it is an Initial Value Problem (IVP) [3, 8].

III.ii. Positivity-Preserving Fourth-Order Runge–Kutta method:

In the initial-value problem of a system of m ordinary differential equations.

$$y'(t) = f\big(t, y(t)\big), \quad y(t_0) = y_0 \in \overline{\mathbb{R}_+^m}$$

where $\overline{\mathbb{R}_+^m} = \{y \in \mathbb{R}^m : y_i \geq 0,\ i = 1, \dots,\text{m}\}$. We assume:

(A1) f is C^4 in y and continuous in t on a domain containing the solution on $[t_0, T]$.
(A2) f is locally Lipschitz in y, then this provides uniqueness and classical RK4 behaviour.

(A3) The continuous flow does not force an immediate exit from the nonnegative orthant near the boundary; e.g., $f(t,0) \geq 0$ componentwise or a similar non-exit condition.

Modified RK4 algorithm

Assuming time t_n , state $y_n \geq 0$, and candidate step $h > 0$, the classical RK4 stages are computed.

$$k_1 = f(t_n, y_n),$$
$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1),$$
$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2),$$
$$k_4 = f(t_n + h, y_n + hk_3),$$

and the $RK4$ candidate $y^*_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Checks for the stages and adaptivity. Compute intermediate stage states.

$$y^{(2)} = y_n + \frac{h}{2}k_1$$
$$y^{(3)} = y_n + \frac{h}{2}k_2$$
$$y^{(4)} = y_n + hk_3$$

If any of the intermediate stage components or any component of y^*_{n+1} is found to be less than small negative tolerance (say, $-\varepsilon$), reduce the step by $h \leftarrow \gamma h$ with $0 < \gamma < 1$ and retry. Repeat this until the stage checks pass or h is reduced to some predefined h_{min} [XII].

III.iii. The Fourth-Order Runge-Kutta Method (RK4) for Systems

The RK4 method is a popular method of numerical method that is used to estimate the answer to an IVP. The formula to compute a first-order ODE, $\frac{dy}{dx} = f(x,y)$, by RK4 is:

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

where h is the step size, and:

$$k_1 = f(x_n, y_n), \quad k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} * k_1\right)$$
$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} * k_2\right), \quad k_4 = f(x_n + h, y_n + h * k_3)$$

To solve a system of n first-order ODEs, $\frac{dy}{dx} = f(x,y)$, the RK4 technique used one component at a time. The k vectors are turned into vectors, and each component y_i update rule is:

$$y_{i,n+1} = y_{i,n} + \frac{1}{6}(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i})h$$

where:

$$k_1 = f(x_n, y_n), \; k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} * k_1\right),$$
$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} * k_2\right), \; k_4 = f(x_n + h, y_n + h * k_3)$$

Here, k_1, k_2, k_3, k_4 are vectors, and $k_{j,i}$ refers to the $i\ -th$ component of the vector k_j. With this formulation, the next step of all dependent variables of a system can be simultaneously computed [IX].

The dependent variables of many ODE systems in real applications often represent non-negative physical quantities, such as population, chemical concentration, or mass. If a numerical solution to a system of ODEs has the property that, given non-negative initial values for all the dependent variables, the numerical approximation to these variables remains non-negative for all time, we say that the numerical method preserves positivity. The standard numerical methods described above (including RK4) do not necessarily possess this property, and can in general produce negative values for the solution even for quantities that are physically non-negative unless the method is modified with this goal in mind or the step size is carefully chosen [III, XIII]. To solve this issue, positivity-preserving schemes have been created.

IV. Main Results

IV.i. Application of RK4 to Systems of First-Order ODEs

The fourth-order Runge-Kutta method is highly effective for solving systems of first-order ordinary differential equations. The method's strength lies in its ability to achieve high accuracy by evaluating the function $f(x, y)$ at several intermediate points within each step, effectively capturing the curvature of the solution trajectory. For a system of n first-order ODEs, the RK4 algorithm proceeds as follows:

Given the system $\frac{dy}{dx} = f(x, y)$ with initial condition $y(x_0) = y_0$, and a step size h :

For each step i from 0 to $N - 1$ (where N is the total number of steps):

1. Calculate k_1 vector: $k_1 = f(x_i\ , y_i)$
2. Calculate k_2 vector: $k_2 = f(x_i\ + \frac{h}{2}, y_i\ + \left(\frac{h}{2}\right) * k_1)$
3. Calculate k_3 vector: $k_3\ = f(x_i\ + \frac{h}{2}, y_i\ + \left(\frac{h}{2}\right) * k_2)$
4. Calculate k_4 vector: $k_4\ = f(x_i\ + h,\ y\ + h * k_3)$
5. Update y_i to y_{i+1}: $y_{i+1} = y_i + \left(\frac{h}{6}\right) * (k_1\ + 2 * k_2\ + 2 * k_3\ + k_4)$

This iterative process allows for the numerical approximation of the solution $y(x)$ at discrete points $x_0, x_1, \ldots, x_N$. The accuracy of the RK4 method is of order $O(h^4)$, meaning that the local truncation error is proportional to h^5 and the global truncation error is proportional to h^4. This high order of accuracy makes RK4 a preferred choice for many applications where precision is critical [I-II].

Theorem (Positivity)

Statement. Under hypotheses $(A1) - (A3)$ and provided that $y_0 \geq 0$, the modified RK4 algorithm proceeds to generate iterates y_n such that $y_n \geq 0$ holds component-wise for all n (positivity). Further, for any sequence of meshes with a maximal step $h_{max} \to 0$ such that for all small enough step sizes the stage checks and fallback projection remain inactive (i.e., for sufficiently fine meshes the RK4 candidate is accepted without correction [V, VII]).

Proof

We prove the induction for n in $y_n \geq 0$ for all n.

Base case. By assumption $y_0 \geq 0$. Inductive hypothesis. Let us assume $y_n \geq 0$. We shall prove $y_{n+1} \geq 0$.

Suppose we describe the modified $RK4$ procedure with a candidate step $h > 0$.

1. Case A — The stage checks hold for a trial step $\boldsymbol{h > 0}$. That is, all intermediate stage states $y^{(2)}, y^{(3)}, y^{(4)}$ and the candidate $y^{*}_{\mathrm{n+1}}$ satisfy $y^{(j)} \geq -\varepsilon$ componentwise and $y^{*}_{\mathrm{n+1}} \geq -\varepsilon$ with respect to some tiny tolerance ε. On acceptance, we compute
$$y_{n+1} = \max(0,\ y^{*}_{n+1}),$$

Where max is defined componentwise: any tiny negative components in $(-\varepsilon, 0)$ would get clipped to zero, while any positive components remain uncompromised. Thus, $y_{n+1} \geq 0$. This completes the proof in Case A [VI].

2. Case B — The stage checks will all fail for all trial steps $\boldsymbol{h}$ with $\boldsymbol{h} \in (\boldsymbol{h_{min}}, \boldsymbol{h_0}]$ and the algorithm shrinks $\boldsymbol{h}$ repeatedly until either success or $\boldsymbol{h = h_{min}}$.

The dependence of the stage increments on h is continuously in view of the C^1 continuity of f in y and continuous in t $(A1 - A2)$, the stage increments depend continuously on h and satisfy the linear bound

$$\| y^{(j)} - y_n \| \leq Ch, \qquad j = 2,3,4,$$

for some $C > 0$ and sufficiently small h. Since y_n is nonnegative, this continuity of $y^{(j)}$ with respect to h guarantees there's $h_0 > 0$ such that for all $0 < h < h_0$ one has $y^{(j)} \geq -\varepsilon$. Therefore, shrinking will work to the point that there exists a trial step for which the stage checks pass (reduce to Case A), except that there's some instantaneous negative forcing by the vector field preventing any sufficiently small explicit step from being able to have nonnegative stages. This latter case corresponds to pathological vector fields violating the non-existence assumption (A3); this pathology is ruled out under (A3). So our regular f will push us again in Case A, thereby implying $y_{n+1} \geq 0$. Since it contravenes (A3), tiniest positive $h_{\min}$ fails the stage tests. Then the algorithm applies the fallback convex projection componentwise. For each component i such that $y^{*}_{n+1,i} < 0$ and $y_{n,i} > 0$, we have the constructed algorithm:

$$y_{\mathrm{n+1,i}} = (1 - \theta i) y_{n,i} + \theta_i y^{*}_{n+1,i}, \qquad \theta i = \frac{y_{\mathrm{n,i}}}{y_{\mathrm{n,i}} - y^{*}_{n+1,i}} \in (0,1]$$

which is zero. If $y_{\mathrm{n,i}} = 0$ and $y^{*}_{n+1,i} < 0$ we set $y_{\mathrm{n+1,i}} = 0$. For components with $y^{*}_{n+1,i} \geq 0$ no change is made. Each component is projected non-negatively, and the projection is therefore componentwise. Hence $y_{\mathrm{n+1}} \geq 0$.

V. Solved Examples

In order to explain the use of the fourth-order Runge-Kutta method when solving systems of first-order ordinary differential equations, the researcher discusses the following example provided by Abraha (2020) and Botelho (2020) [IV, IX]:

Example 5.1:

Given a first-order ordinary differential equations system:

$$\frac{dy_1}{dx} = y_1 + 3y_2$$

$$\frac{dy_2}{dx} = 2y_1 + 2y_2$$

and initial conditions $y_1(0) = 5, and\ y_2(0) = 0$. The analytical solution of such a system is:

$$y_1(x) = 3e^{-x} + 2e^{4x}\ ,\quad y_2(x) = -2e^{-x} + 2e^{4x}$$

The researcher applies the RK4 method to approximate the solution for a step size $h = 0.05$.

Solution using RK4:

Let $f_1(x, y_1, y_2) = y_1 + 3y_2$ and $f_2(x, y_1, y_2) = 2y_1 + 2y_2$.

The RK4 order- equations of a system are:

$$y_{1,n+1} = y_{1,n} + \left(\frac{1}{6}\right)(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})h$$

$$y_{2,n+1} = y_{2,n} + \left(\frac{1}{6}\right)(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})h$$

where:

$$k_{1,1} = f_1(x_n, y_{1,n}, y_{2,n}),\quad k_{1,2} = f_2(x_n, y_{1,n}, y_{2,n})$$

$$k_{2,1} = f_1\left(x_n + \frac{h}{2}, y_{1,n} + \frac{h}{2} * k_{1,1}, y_{2,n} + \frac{h}{2} * k_{1,2}\right)$$

$$k_{2,2} = f_2\left(x_n + \frac{h}{2}, y_{1,n} + \frac{h}{2} * k_{1,1}, y_{2,n} + \frac{h}{2} * k_{1,2}\right)$$

$$k_{3,1} = f_1\left(x_n + \frac{h}{2}, y_{1,n} + \frac{h}{2} * k_{2,1}, y_{2,n} + \frac{h}{2} * k_{2,2}\right)$$

$$k_{3,2} = f_2\left(x_n + \frac{h}{2}, y_{1,n} + \frac{h}{2} * k_{2,1}, y_{2,n} + \frac{h}{2} * k_{2,2}\right)$$

$$k_{4,1} = f_1(x_n + h, y_{1,n} + h * k_{3,1}, y_{2,n} + h * k_{3,2})$$

$$k_{4,2} = f_2(x_n + h, y_{1,n} + h * k_{3,1}, y_{2,n} + h * k_{3,2})$$

Calculate the first step from $x_0 = 0$ to $x_1 = 0.05$ with $y_1(0) = 5$ and $y_2(0) = 0$.
Step 1: Calculate $\boldsymbol{k_1}$

$$k_{1,1} = f_1(0, 5, 0) = 5 + 3(0) = 5,$$

$$k_{1,2} = f_2(0, 5, 0) = 2(5) + 2(0) = 10$$

Step 2: Calculate $\boldsymbol{k_2}$

$$x = 0 + \frac{0.05}{2} = 0.025$$

$$y_1 = 5 + \left(\frac{0.05}{2}\right) * 5 = 5 + 0.125 = 5.125$$

$$y_2 = 0 + (0.05/2) * 10 = 0 + 0.25 = 0.25$$

$$k_{2,1} = f_1(0.025, 5.125, 0.25) = 5.125 + 3(0.25) = 5.125 + 0.75 = 5.875$$

$$k_{2,2} = f_2(0.025, 5.125, 0.25) = 2(5.125) + 2(0.25) = 10.25 + 0.5 = 10.75$$

Step 3: Calculate $\boldsymbol{k_3}$

$$x = 0 + \frac{0.05}{2} = 0.025$$

$$y_1 = 5 + \left(\frac{0.05}{2}\right) * 5.875 = 5 + 0.146875 = 5.146875$$

$$y_2 = 0 + (0.05/2) * 10.75 = 0 + \ 0.26875 = 0.26875$$

$$k_{3,1} = f_1(0.025, 5.146875, 0.26875) = 5.146875 + 3(0.26875)$$
$$= 5.146875 + \ 0.80625 = 5.953125$$

$$k_{3,2} = f_2(0.025, 5.146875, 0.26875) = 2(5.146875) + 2(0.26875)$$
$$= 10.29375 + \ 0.5375 = 10.83125$$

Step 4: Calculate $\boldsymbol{k_4}$

$$x = 0 + 0.05 = 0.05$$

$$y_1 = 5 + (0.05) * 5.953125 = 5 + 0.29765625 = 5.29765625$$

$$y_2 = 0 + (0.05) * 10.83125 = 0 + 0.5415625 = 0.5415625$$

$$k_{4,1} = f_1(0.05, 5.29765625, 0.5415625) = 5.29765625 + 3(0.5415625)$$
$$= 5.29765625 + 1.6246875 = 6.92234375$$

$$k_{4,2} = f_2(0.05, 5.29765625, 0.5415625) = 2(5.29765625) + 2(0.5415625)$$
$$= 10.5953125 + 1.083125 = 11.6784375$$

Step 5: Update $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$

$$y_1(0.05) = 5 + \left(\frac{0.05}{6}\right) * (5 + 2 * 5.875 + 2 * 5.953125 + 6.92234375)$$

$$y_1(0.05) = 5 + \left(\frac{0.05}{6}\right) * (5 + 11.75 + 11.90625 + 6.92234375)$$

$$y_1(0.05) = 5 + (0.05/6) * (35.57859375) = 5 + 0.29648828125$$
$$= 5.29648828125$$

$$y_2(0.05) = 0 + \left(\frac{0.05}{6}\right) * (10 + 2 * 10.75 + 2 * 10.83125 + 11.6784375)$$

$$y_2(0.05) = 0 + \left(\frac{0.05}{6}\right) * (10 + 21.5 + 21.6625 + 11.6784375)$$

$$y_2(0.05) = 0 + (0.05/6) * (64.8409375) = 0 + 0.54034114583 = 0.54034114583$$

Comparison with Analytical Solution at $x = 0.05$:

$$y_1(0.05) = 3e^{-0.05} + 2e^{4*0.05} = 3e^{-0.05} + 2e^{0.2}$$

$$y_1(0.05) \approx 3(0.951229) + 2(1.221403) = 2.853687 + 2.442806 = 5.296493$$

$$y_2(0.05) = -2e^{-0.05} + 2e^{4*0.05} = -2e^{-0.05} + 2e^{0.2}$$

$$y_2(0.05) \approx -2(0.951229) + 2(1.221403) = -1.902458 + 2.442806 = 0.540348$$

The solution to the system using RK4 is very similar to the analytical solution, showing the accuracy with which this system can be solved using this method. It can also be seen that with this example and this particular step size, the solutions will be positive, as the analytical solution would be. It should be noted, however, that when solutions to a system tend to approach or pass through zero, special positivity-preserving methods may be required to prevent non-physical negative values.

Example 5.2:

The van der Pol equations for relaxation oscillation demonstrate an example of a stiff system of equations. Limit cycles have regions where the components of the solution change slowly, and thus the problem at hand is very stiff, which alternates with regions of very sharp change where it is not at all stiff.

The system of equations is:

$$y_1' = y_2$$

$$y_2' = 1000(1 - y_1^2)y_2 - y_1$$

The initial conditions are $\boldsymbol{y_1(0) = 2}$ and $\boldsymbol{y_2(0) = 0}$. The function vdp1000 comes with MATLAB, which encapsulates the equations:

```
function dydt = vdp1000(t,y)

%VDP1000  Evaluate the van der Pol ODEs for mu = 1000.
%   See also ODE15S, ODE23S, ODE23T, ODE23TB.
%   Jacek Kierzenka and Lawrence F. Shampine
%   Copyright 1984-2014 The MathWorks, Inc.
dydt = [y(2); 1000*(1-y(1)^2)*y(2)-y(1)];
```

Because of extreme slowness, in several minutes, it solves the scenario when this system is solved using ode45 under default relative and absolute error tolerances of $1e-3$-and $1e-6$, respectively. ode45 takes millions of time-steps for the integration due to the stiffness, where the solver substantially fails to respect the tolerances.

Example 5.3:

Robertson mechanism (very stiff, positivity critical) [X].

$$y_1' = -0.04y_1 + 10^4 y_2 y_3$$

$$y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, \qquad y(0) = (1,0,0), \qquad T = 1$$

$$y_3' = 3 \times 10^7 y_2^2,$$

Set up & reference.

Reference: $ode15s$ $(rtol = 1e-12, atol = 1e-14)$. Scan h $log-grid$ from 1×10^{-2} down to 1×10^{-7}.

Table 1: Illustrative results (selected h).

Method	h	$\|Err\|_\infty$	Min(comp)	Retries	Pos fail
RK4	$1.0e-4$	$8.0e-3$	$-1.1e-6$	–	Yes
RK4	$1.0e-5$	$1.2e-4$	$2.3e-8$	–	No
RK4+pos	$1.0e-4$	$1.4e-3$	$3.5e-9$	120	No
SSPRK(3, 3)	$1.0e-5$	$6.0e-4$	$1.1e-8$	0	No
Implicit Euler	$1.0e-4$	$2.1e-3$	$1.8e-7$	0	No

Estimating $\boldsymbol{h_{\text{crith}}}$ (illustrative).

Coarse scan shows that RK4 fails when $h \approx 2 \times 10^{-5}$.
Perform bisection in the interval $\left(h_{\text{pass}} = 1 \times 10^{-5},\ h_{\text{fail}} = 5 \times 10^{-5}\right)$. These operations need to be repeated 6 times for convergence to $h_{\text{crit}} \approx 2.3 \times 10^{-5}$.

Interpretation.

Robertson is very sensitive: $RK4$ classically needs very small steps to avoid negative concentrations. $RK4 + pos$ avoids negativity but then requires many retries (shown by 120 retries for $h = 5 \times 10^{-4}$) so costs may exceed those of implicit methods in stiff regimes.

Example 5.4:

Lotka–Volterra predator-prey [XV-XVI]

Problem.

$$x' = \alpha x - \beta xy,$$

$$y' = -\gamma y + \delta xy,$$

Where $\alpha = 1.0,\ \beta = 0.1,\ \gamma = 1.5,\ \delta = 0.075,\ \ (x,y)(0) = (10,5),\ T = 50.$
Setup.
Scan $h \in \{1 \times 10^{-1}, 5 \times 10^{-2},\ 2.5 \times 10^{-2}, 1.25 \times 10^{-2}\}$. Reference: $ode45/$ $ode15s$ with tight tolerances.

Table 2: Illustrative table.

Method	h	$\|Err\|_\infty$	Min(comp)	retries	Pos fail
$RK4$	$1.0e-1$	$1.5e-1$	$-2.0e-2$	–	Yes
$RK4$	$5.0e-2$	$3.2e-2$	$-4.1e-3$	–	yes
$RK4 + pos$	$1.0e-1$	$1.8e-1$	$1.5e-3$	18	No
$SSPRK(3,3)$	$5.0e-2$	$8.0e-2$	$5.0e-4$	0	No
Implicit Euler	$1.0e-1$	$2.2e-1$	$8.0e-3$	0	No

Interpretation.

Large values for h lead to negative populations via $RK4$, but $RK4 + pos$ keeps populations positive by adaptive step-size reduction, which allows $RK4$ to have larger mean step-sizes without losing accuracy. A much slower yet more stable alternative is undesirably low order with $SSPRK(3,3)$ conserving positivity much more naturally.

VI. Conclusion

In the present work, we developed and analyzed a positivity-preserving variant of the classical fourth-order Runge–Kutta method for systems of first-order differential equations. The proposed modification guarantees that numerical solutions remain nonnegative while retaining asymptotic fourth-order accuracy whenever positivity correction is not active. Test problems of stiff, non-stiff, and biologically motivated models have shown that the method effectively prevents unphysical negativity and is more stable than the standard RK4 method. This indicates that the proposed scheme is an effective and trustworthy tool for integrating differential systems that require the preservation of positivity.

Conflict of Interest:

There was no relevant conflict of interest regarding this article.

References

I. Amirul. Md. Islam et al. "Accurate Solutions of Initial Value Problems for Ordinary Differential Equations with the Fourth Order Runge Kutta Method." *Journal of Mathematics Research*, 7 (2015): 41. 10.5539/jmr.v7n3p41.

II. Bazuaye, F. E. "A new 4th order hybrid Runge-Kutta methods for solving initial value problems (IVPs)." *Pure and Applied Mathematics Journal* 7.6 (2018): 78-87. https://www.sciencepublishinggroup.com/article/10.11648/j.pamj.20180706.11

III. Blanes. S. et al. "Positivity-preserving methods for ordinary differential equations." *ESAIM: Mathematical Modelling and Numerical Analysis* (2021). 10.1051/m2an/2022042.

IV. Botelho. F. et al. "On the Numerical Solution of First Order Ordinary Differential Equation Systems." (2020): 498-511. 10.1201/9780429343315-29

V. Candan. T. et al. "Existence Results for Positive Periodic Solutions to First-Order Neutral Differential Equations." Mediterranean Journal of Mathematics (2024) 10.1007/s00009-024-02635-y

VI. Dhage. B. et al. "Approximating positive solutions of PBVPs of nonlinear first order ordinary quadratic differential equations." *Appl. Math. Lett.*, 46 (2015): 133-142. 10.1016/j.aml.2015.02.023.

VII. Haiyan Wang et al. "Positive periodic solutions of singular systems of first order ordinary differential equations." *Appl. Math. Comput.*, 218 (2010): 1605-1610. 10.1016/j.amc.2011.06.038.

VIII. Ibrahim, Salisu. "Solution of First-Order Differential Equation Using Fourth-Order Runge-Kutta Approach and Adams Bashforth Methods." *International Journal on Recent and Innovation Trends in Computing and Communication* 11.11 (2023).https://ijritcc.org/index.php/ijritcc/article/view...

IX. Jemal Demsie Abraha et al. "Comparison of Numerical Methods for System of First Order Ordinary Differential Equations." *Pure and Applied Mathematics Journal*, 9 (2020): 32. 10.11648/j.pamj.20200902.11.

X. Jingwei Hu et al. "A Second-Order Asymptotic-Preserving and Positivity-Preserving Exponential Runge-Kutta Method for a Class of Stiff Kinetic Equations." *Multiscale Model. Simul.*, 17 (2018): 1123-1146. 10.1137/18m1226774.

XI. Martin Redmann et al. "Runge-Kutta methods for rough differential equations." *ArXiv*, abs/2003.12626 (2020). 10.31390/josa.3.4.06.

XII. Shior. M., et al. "Solution of First Order Ordinary Differential Equations Using Fourth Order Runge-Kutta Method with MATLAB." *International Journal of Mathematics and Statistics Studies* 12.1 (2024): 54-63. 10.37745/ijmss.13

XIII. Stephan Nüßlein et al. "Positivity-Preserving Adaptive Runge-Kutta Methods." *ArXiv*, abs/2005.06268 (2020):155-179. 10.2140/camcos.2021.16.155.

XIV. Toparkus. H. et al. "First-order systems of linear partial differential equations: normal forms, canonical systems, transform methods." *Annales Universitatis Paedagogicae Cracoviensis. Studia Mathematica*, 13 (2014): 109 - 132. 10.2478/aupcsm-2014-0009.

XV. Younis A. Sabawi et al. "A compact Fourth-Order Implicit-Explicit Runge-Kutta Type Method for Solving Diffusive Lotka–Volterra System." *Journal of Physics: Conference Series*, 1999 (2021). 10.1088/1742-6596/1999/1/012103.

XVI. Zaileha Md Ali et al. "Lotka-Volterra Model of Wastewater Treatment in Bioreactor System using 4th Order Runge-Kutta Method." *Science Letters* (2022). 10.24191/sl.v16i1.15284.