

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

M. A. Sandoval-Hernandez^{1*}, R. E. Morales-Cortez², C. Hoyos-Reyes², J. Sanchez-Orea², F. Ivan Molina-Herrera³, H. Jimenez-Islas⁴, N. Carrillo-Ramon², R. Ruiz-Gomez², R. A. Callejas-Molina², U. A. Filobello-Nino², H. Vazquez-Leal²

¹Centro de Bachillerato Tecnológico industrial y de servicios No. 190, Av. 15 Col. Venustiano Carranza 2da Sección, Boca del Río, 94297, Veracruz, México.

²Facultad de Instrumentación Electrónica, Universidad Veracruzana, Circuito Gonzalo Aguirre Beltrán S/N, Xalapa, 91090, Veracruz, México

³Departamento de Ingeniería, Universidad de Guanajuato, Av. Ing. Javier Barros Sierra 201 esq. Av. Baja California, Ejido Santa María del Refugio, Celaya 38110, Guanajuato, Mexico

⁴Departamento de Ingeniería Bioquímica y Ambiental, Tecnológico Nacional de México en Celaya, Antonio García Cubas Pte. #600 esq. Av. Tecnológico, Celaya 38010, Guanajuato, Mexico

ABSTRACT: This article presents a didactic application of the fourth-order Runge-Kutta method to nonlinear initial-value problems. The work continues and extends a previously reported teaching strategy based on reduction to first-order systems and a double-index notation for the RK4 stage constants. Four nonlinear cases are considered: the Lorenz system, the Duffing oscillator, the van der Pol oscillator, and a coupled strongly nonlinear system proposed by Cveticanin. For each case, the first three RK4 iterations are developed explicitly before the complete numerical simulation is carried out in Maple 2021. This makes the relation between the mathematical model, the stage constants, and the recursive update visible before software is used for the full trajectory. Numerical results are presented through time histories, phase portraits, and three-dimensional trajectories. In addition, surrealist visual compositions are integrated with the scientific graphs while the numerical curves, axes, scales, and data are preserved. From an educational perspective, this visual layer is explored as a way of broadening the conventional presentation of numerical results and of partially moving away from standardized forms associated with school mathematical discourse. To the best of the authors' knowledge, this is the first educational numerical-methods study in which surrealist visual compositions are systematically integrated with RK4 simulations of nonlinear initial-value problems.

KEYWORDS: Fourth-order Runge-Kutta method; Nonlinear initial-value problems; Mathematics education; School mathematical discourse; Surrealist scientific visualization.

I. INTRODUCTION

Initial-value problems (IVPs) are central in the mathematical description of dynamical processes in engineering, physics, biology, and other applied fields. In many models, nonlinear terms prevent the direct construction of a closed-form solution, so numerical integration becomes the practical route for studying the evolution of the dependent variables. Standard numerical-analysis texts present the Runge-Kutta family as one of the main tools for this purpose because it combines a clear algorithmic structure with good accuracy for a wide variety of ordinary differential equations (ODE) [1,2].

The fourth-order Runge-Kutta method (RK4) remains especially useful in teaching because each integration step can be followed through four intermediate evaluations before the state is updated. Its use is not limited to classroom examples. Recent applications include the simulation of a pneumatically operated valve with nonlinear damping [3], nonlinear Duffing-type dynamics [4], GPU acceleration of RK4 algorithms for reactor-kinetics calculations [5], and nonlinear spring-mass-damper systems [6]. These examples show that the same mathematical structure used in introductory numerical methods remains relevant in contemporary computational work.

The present article continues and extends the didactic strategy reported in [7]. That work proposed a systematic way to reduce higher-order ODE to first-order systems and to organize RK4 calculations by means of double-index stage constants. The first index

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

identifies the equation or dependent variable, whereas the second index identifies the RK4 stage. In the present work, this organization is applied to four nonlinear IVPs of increasing structural complexity. A further didactic decision is to calculate the first three iterations explicitly for every case before the complete simulation is performed. The purpose is not to replace software, but to make the transition from the differential model to the computational result transparent.

A second component of the study concerns the way numerical results are visually communicated. From the socioepistemological perspective, school mathematical discourse (dME) has been described as a system of reason that can privilege established objects, procedures, meanings, and forms of validation, sometimes producing exclusion or symbolic violence when alternative forms of mathematical construction are displaced [8, 9]. One way to identify the persistence of this discourse is to examine what remains unchanged even when instructional innovations are introduced: the method used in class may change while the forms of representation and the type of mathematical object being privileged remain essentially the same [10, 11].

In this article, surrealist visual compositions are explored as a controlled modification of the presentation layer. The scientific graphs are not altered: the numerical trajectory, axes, scales, and variable relationships are preserved. What changes is the surrounding visual environment through displacement, altered spatial relations, unexpected objects, and dreamlike scenes. This choice does not imply that a surreal image by itself removes school mathematical discourse or improves learning. Rather, it provides an exploratory way to partially move away from standardized graphical presentation while keeping the mathematical content intact.

The article is organized as follows. Section II summarizes the RK4 method. Section III presents the didactic strategy and the double-index notation used throughout the paper. Section IV develops the four nonlinear case studies, including the first three numerical iterations and the complete Maple simulations. Section V discusses the mathematical progression of the cases, the surrealist visual treatment, and its relation to school mathematical discourse. Section VI gives the conclusions.

II. FOURTH-ORDER RUNGE-KUTTA METHOD

Consider the first-order IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

For a step size h , RK4 advances the numerical solution from (t_m, y_m) to (t_{m+1}, y_{m+1}) through four slope evaluations:

$$\begin{aligned} k_1 &= f(t_m, y_m), \\ k_2 &= f\left(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_1\right), \\ k_3 &= f\left(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_2\right), \\ k_4 &= f(t_m + h, y_m + hk_3), \end{aligned}$$

followed by

$$y_{m+1} = y_m + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad t_{m+1} = t_m + h.$$

In this article, the k quantities are treated as slopes, not as increments already multiplied by h . With this convention, the factor h appears only in the intermediate states and in the final weighted update. RK4 has local truncation error of order $O(h^5)$ and global error of order $O(h^4)$ under the usual smoothness assumptions [1], [2].

For a vector problem

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0,$$

the same four-stage structure is applied component by component. Higher-order scalar equations are first rewritten as first-order systems. This step is particularly important from a didactic point of view because it makes the number of state variables, equations, and RK4 constants explicit before any program is executed.

III. DIDACTIC STRATEGY FOR THE IMPLEMENTATION OF RK4

A. Reduction and Variable Assignment

For an n th-order equation written in explicit form,

$$y^{(n)} = F(t, y, y', \dots, y^{(n-1)}),$$

the variables are assigned as

$$x_1 = y, \quad x_2 = y', \quad \dots, \quad x_n = y^{(n-1)}.$$

The equivalent first-order system is then

$$x_1' = f_1 = x_2, \quad x_2' = f_2 = x_3, \quad \dots, \quad x_n' = f_n = F(t, x_1, \dots, x_n).$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

The iteration index m is attached to every state variable, so the numerical state at t_m is $(x_{1,m}, x_{2,m}, \dots, x_{n,m})$. The initial conditions are assigned at $m = 0$.

B. Double-Index RK4 Constants

Following the strategy introduced in [7], the RK4 constants are written as $k_{i,j}$, where $i = 1, \dots, n$ identifies the equation and $j = 1, 2, 3, 4$ identifies the RK4 stage. Thus,

$$\begin{aligned} k_{i,1} &= f_i(t_m, x_{1,m}, \dots, x_{n,m}), \\ k_{i,2} &= f_i\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, \dots, x_{n,m} + \frac{h}{2}k_{n,1}\right), \\ k_{i,3} &= f_i\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, \dots, x_{n,m} + \frac{h}{2}k_{n,2}\right), \\ k_{i,4} &= f_i(t_m + h, x_{1,m} + hk_{1,3}, \dots, x_{n,m} + hk_{n,3}). \end{aligned}$$

The recursive update is

$$x_{i,m+1} = x_{i,m} + \frac{h}{6}(k_{i,1} + 2k_{i,2} + 2k_{i,3} + k_{i,4}), \quad i = 1, \dots, n.$$

This notation avoids the ambiguity that appears when a single symbol k_1, k_2, k_3, k_4 is used for a system containing several differential equations. The first index answers *which equation?* and the second answers *which RK4 stage?*

C. Common Sequence Used in the Four Case Studies

The same sequence is used throughout the paper: (1) state the nonlinear model and initial conditions; (2) reduce higher-order equations to first-order form when necessary; (3) define the functions f_i ; (4) calculate the double-index RK4 constants; (5) develop the first three numerical iterations explicitly; and (6) execute the complete simulation. Figure 1 summarizes this sequence. Its technical structure is adapted from the didactic strategy reported in [7], while its visual background is presented in a surrealist form.

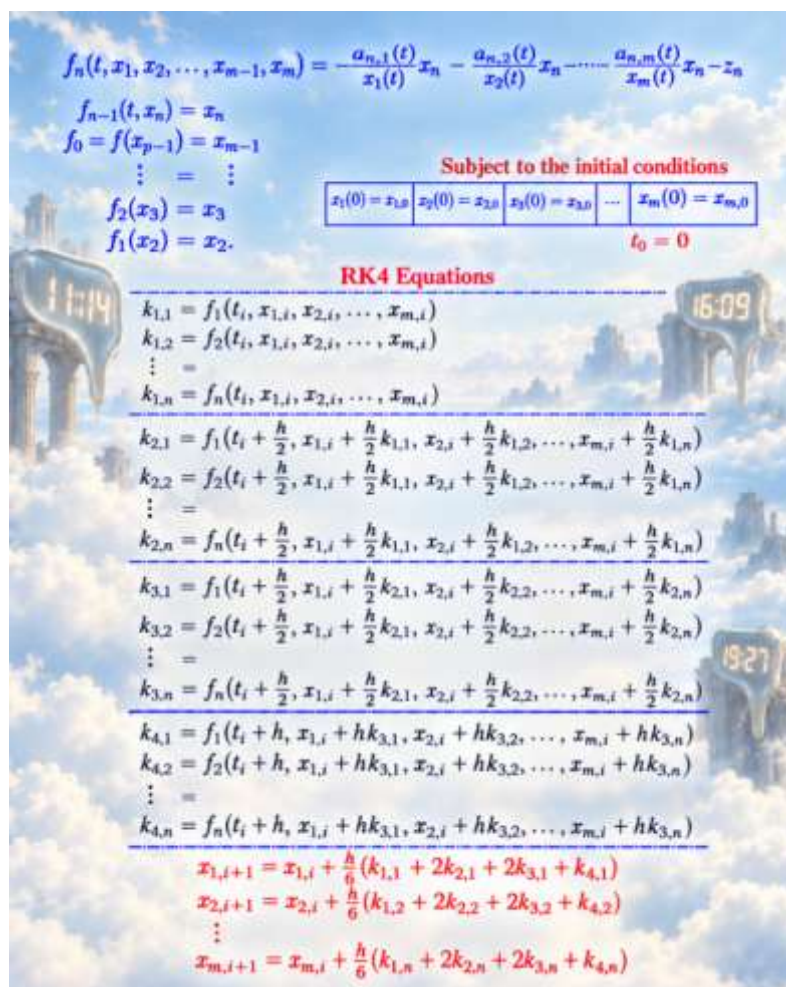


Figure 1. Didactic strategy for the implementation of the RK4 method in higher-order ODEs and systems. Adapted from [7].

IV. CASE STUDIES

A. Case 1: Lorenz System

The Lorenz system is a classical nonlinear model associated with deterministic aperiodic behavior [12]. It is written as

$$\begin{aligned}\frac{dx_1}{dt} &= 10(x_2 - x_1). \\ \frac{dx_2}{dt} &= x_1(28 - x_3) - x_2. \\ \frac{dx_3}{dt} &= x_1x_2 - \frac{8}{3}x_3.\end{aligned}$$

The initial conditions are

$$\begin{aligned}x_1(0) &= 0.1. \\ x_2(0) &= 0. \\ x_3(0) &= 0.\end{aligned}$$

Since the Lorenz model is already expressed as a system of three first-order differential equations, no additional variable transformation is required. Following the methodology described in Figure 1 and [7], the system can be rewritten in function notation as

$$\begin{aligned}f_1(t, x_1, x_2, x_3) &= 10(x_2 - x_1). \\ f_2(t, x_1, x_2, x_3) &= x_1(28 - x_3) - x_2. \\ f_3(t, x_1, x_2, x_3) &= x_1x_2 - \frac{8}{3}x_3.\end{aligned}$$

For a system of three first-order equations, the symbolic RK4 formulation becomes

$$\begin{aligned}k_{1,1} &= f_1(t_m, x_{1,m}, x_{2,m}, x_{3,m}). \\ k_{2,1} &= f_2(t_m, x_{1,m}, x_{2,m}, x_{3,m}). \\ k_{3,1} &= f_3(t_m, x_{1,m}, x_{2,m}, x_{3,m}). \\ k_{1,2} &= f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}\right). \\ k_{2,2} &= f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}\right). \\ k_{3,2} &= f_3\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}\right). \\ k_{1,3} &= f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}\right). \\ k_{2,3} &= f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}\right). \\ k_{3,3} &= f_3\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}\right). \\ k_{1,4} &= f_1(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}). \\ k_{2,4} &= f_2(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}). \\ k_{3,4} &= f_3(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}). \\ x_{1,m+1} &= x_{1,m} + \frac{h}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}). \\ x_{2,m+1} &= x_{2,m} + \frac{h}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}). \\ x_{3,m+1} &= x_{3,m} + \frac{h}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}).\end{aligned}$$

For this case, $h = 0.01$. The first numerical iteration starts from

$$\begin{aligned}t_0 &= 0.000000000. \\ x_{1,0} &= 0.100000000. \\ x_{2,0} &= 0.000000000. \\ x_{3,0} &= 0.000000000.\end{aligned}$$

Then,

$$\begin{aligned}k_{1,1} &= f_1(t_0, x_{1,0}, x_{2,0}, x_{3,0}) = -1.000000000. \\ k_{2,1} &= f_2(t_0, x_{1,0}, x_{2,0}, x_{3,0}) = 2.800000000.\end{aligned}$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$k_{3,1} = f_3(t_0, x_{1,0}, x_{2,0}, x_{3,0}) = 0.000000000.$$

Showing the numerical procedure

$$\begin{aligned} k_{1,1} &= 10(0.000000000 - 0.100000000) = -1.000000000, \\ k_{2,1} &= 0.100000000(28 - 0.000000000) - 0.000000000 = 2.800000000, \\ k_{3,1} &= 0.100000000(0.000000000) - \frac{8}{3}(0.000000000) = 0.000000000. \end{aligned}$$

At stage 2

$$\begin{aligned} k_{1,2} &= f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}) = -0.810000000. \\ k_{2,2} &= f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}) = 2.646000000. \\ k_{3,2} &= f_3(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}) = 0.001330000. \end{aligned}$$

Showing the numerical procedure.

$$\begin{aligned} k_{1,2} &= 10(0.014000000 - 0.095000000) = -0.810000000, \\ k_{2,2} &= 0.095000000(28 - 0.000000000) - 0.014000000 = 2.646000000, \\ k_{3,2} &= 0.095000000(0.014000000) - \frac{8}{3}(0.000000000) = 0.001330000. \end{aligned}$$

At stage 3

$$\begin{aligned} k_{1,3} &= f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}) = -0.827200000. \\ k_{2,3} &= f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}) = 2.673369362. \\ k_{3,3} &= f_3(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}) = 0.001251685. \end{aligned}$$

Showing the numerical procedure.

$$\begin{aligned} k_{1,3} &= 10(0.013230000 - 0.095950000) = -0.827200000, \\ k_{2,3} &= 0.095950000(28 - 0.000006650) - 0.013230000 = 2.673369362, \\ k_{3,3} &= 0.095950000(0.013230000) - \frac{8}{3}(0.000006650) = 0.001251685. \end{aligned}$$

At stage 4

$$\begin{aligned} k_{1,4} &= f_1(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}) = -0.649943064. \\ k_{2,4} &= f_2(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}) = 2.541649158. \\ k_{3,4} &= f_3(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}) = 0.002418850. \end{aligned}$$

Showing the numerical procedure.

$$\begin{aligned} k_{1,4} &= 10(0.026733694 - 0.091728000) = -0.649943064, \\ k_{2,4} &= 0.091728000(28 - 0.000012517) - 0.026733694 = 2.541649158, \\ k_{3,4} &= 0.091728000(0.026733694) - \frac{8}{3}(0.000012517) = 0.002418850. \end{aligned}$$

The recursive equations give

$$\begin{aligned} x_{1,1} &= x_{1,0} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.091792762. \\ x_{2,1} &= x_{2,0} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.026633980. \\ x_{3,1} &= x_{3,0} + \frac{0.01}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 0.000012637. \end{aligned}$$

Showing the numerical procedure. Recursive update gives

$$\begin{aligned} x_{1,1} &= 0.100000000 + \frac{0.01}{6}(-1.000000000 + 2(-0.810000000) + 2(-0.827200000) + -0.649943064) = 0.091792762. \\ x_{2,1} &= 0.000000000 + \frac{0.01}{6}(2.800000000 + 2(2.646000000) + 2(2.673369362) + 2.541649158) = 0.026633980. \\ x_{3,1} &= 0.000000000 + \frac{0.01}{6}(0.000000000 + 2(0.001330000) + 2(0.001251685) + 0.002418850) = 0.000012637. \end{aligned}$$

For the second numerical iteration, for $m = 1$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$\begin{aligned}t_1 &= 0.010000000. \\x_{1,1} &= 0.091792762. \\x_{2,1} &= 0.026633980. \\x_{3,1} &= 0.000012637.\end{aligned}$$

Then,

$$\begin{aligned}k_{1,1} &= f_1(t_1, x_{1,1}, x_{2,1}, x_{3,1}) = -0.651587818. \\k_{2,1} &= f_2(t_1, x_{1,1}, x_{2,1}, x_{3,1}) = 2.543562184. \\k_{3,1} &= f_3(t_1, x_{1,1}, x_{2,1}, x_{3,1}) = 0.002411108.\end{aligned}$$

Showing the numerical procedure

$$\begin{aligned}k_{1,1} &= 10(0.026633980 - 0.091792762) = -0.651587818, \\k_{2,1} &= 0.091792762(28 - 0.000012637) - 0.026633980 = 2.543562184, \\k_{3,1} &= 0.091792762(0.026633980) - \frac{8}{3}(0.000012637) = 0.002411108.\end{aligned}$$

At stage 2

$$\begin{aligned}k_{1,2} &= f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}) = -0.491830317. \\k_{2,2} &= f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}) = 2.439621052. \\k_{3,2} &= f_3(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}) = 0.003418157.\end{aligned}$$

Showing the numerical procedure.

$$\begin{aligned}k_{1,2} &= 10(0.039351791 - 0.088534822) = -0.491830317, \\k_{2,2} &= 0.088534822(28 - 0.000024693) - 0.039351791 = 2.439621052, \\k_{3,2} &= 0.088534822(0.039351791) - \frac{8}{3}(0.000024693) = 0.003418157.\end{aligned}$$

At stage 3

$$\begin{aligned}k_{1,3} &= f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}) = -0.505015249. \\k_{2,3} &= f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}) = 2.462506338. \\k_{3,3} &= f_3(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}) = 0.003389736.\end{aligned}$$

Showing the numerical procedure

$$\begin{aligned}k_{1,3} &= 10(0.038832085 - 0.089333610) = -0.505015249, \\k_{2,3} &= 0.089333610(28 - 0.000029728) - 0.038832085 = 2.462506338, \\k_{3,3} &= 0.089333610(0.038832085) - \frac{8}{3}(0.000029728) = 0.003389736.\end{aligned}$$

At stage 4

$$\begin{aligned}k_{1,4} &= f_1(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}) = -0.354835659. \\k_{2,4} &= f_2(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}) = 2.377529974. \\k_{3,4} &= f_3(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}) = 0.004322251.\end{aligned}$$

Showing the numerical procedure

$$\begin{aligned}k_{1,4} &= 10(0.051259043 - 0.086742609) = -0.354835659, \\k_{2,4} &= 0.086742609(28 - 0.000046534) - 0.051259043 = 2.377529974, \\k_{3,4} &= 0.086742609(0.051259043) - \frac{8}{3}(0.000046534) = 0.004322251.\end{aligned}$$

The recursive equations give

$$\begin{aligned}x_{1,2} &= x_{1,1} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.086792571. \\x_{2,2} &= x_{2,1} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.051176225. \\x_{3,2} &= x_{3,1} + \frac{0.01}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 0.000046552.\end{aligned}$$

Showing the numerical procedure. Recursive update gives

$$x_{1,2} = 0.091792762 + \frac{0.01}{6}(-0.651587818 + 2(-0.491830317) + 2(-0.505015249) + -0.354835659) = 0.086792571.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$x_{2,2} = 0.026633980 + \frac{0.01}{6} (2.543562184 + 2(2.439621052) + 2(2.462506338) + 2.377529974) = 0.051176225.$$

$$x_{3,2} = 0.000012637 + \frac{0.01}{6} (0.002411108 + 2(0.003418157) + 2(0.003389736) + 0.004322251) = 0.000046552.$$

For the third numerical iteration,

$$t_2 = 0.020000000.$$

$$x_{1,2} = 0.086792571.$$

$$x_{2,2} = 0.051176225.$$

$$x_{3,2} = 0.000046552.$$

Then,

$$k_{1,1} = f_1(t_2, x_{1,2}, x_{2,2}, x_{3,2}) = -0.356163458.$$

$$k_{2,1} = f_2(t_2, x_{1,2}, x_{2,2}, x_{3,2}) = 2.379011710.$$

$$k_{3,1} = f_3(t_2, x_{1,2}, x_{2,2}, x_{3,2}) = 0.004317577.$$

$$k_{1,2} = f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}) = -0.219404700.$$

$$k_{2,2} = f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}) = 2.317252015.$$

$$k_{3,2} = f_3(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}) = 0.005180093.$$

$$k_{1,3} = f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}) = -0.229330623.$$

$$k_{2,3} = f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}) = 2.336706624.$$

$$k_{3,3} = f_3(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}) = 0.005185258.$$

$$k_{1,4} = f_1(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}) = -0.099559734.$$

$$k_{2,4} = f_2(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}) = 2.291427795.$$

$$k_{3,4} = f_3(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}) = 0.006036440.$$

Finally,

$$x_{1,3} = x_{1,2} + \frac{0.01}{6} (k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.084537247.$$

$$x_{2,3} = x_{2,2} + \frac{0.01}{6} (k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.074473486.$$

$$x_{3,3} = x_{3,2} + \frac{0.01}{6} (k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 0.000098360.$$

After the third iteration,

$$t_3 = 0.030000000.$$

$$x_{1,3} = 0.084537247.$$

$$x_{2,3} = 0.074473486.$$

$$x_{3,3} = 0.000098360.$$

The complete numerical calculation was then continued to $t = 35.2$ in Maple 2021. The resulting three-dimensional trajectory is shown in Figure 2.



Figure 2. Three-dimensional trajectory of the Lorenz system obtained with the RK4 method for $h = 0.01$ and $0 \leq t \leq 35.2$.

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

B. Case 2: Duffing Oscillator

The forced Duffing equation considered here is

$$x'' + x' + x + x^3 = \cos(t).$$

The initial conditions are

$$x(0) = 0.$$

$$x'(0) = 0.$$

Duffing-type oscillators remain important models for nonlinear vibration analysis, and recent work continues to use them in analytical and numerical studies [13]. Introducing the variables

$$x_1 = x.$$

$$x_2 = x'.$$

the second-order differential equation is rewritten as

$$\frac{dx_1}{dt} = x_2.$$

$$\frac{dx_2}{dt} = \cos(t) - x_2 - x_1 - x_1^3.$$

The corresponding initial conditions are

$$x_1(0) = 0.$$

$$x_2(0) = 0.$$

Following the methodology described in Figure 1 and [7], the system is written in function notation as

$$f_1(t, x_1, x_2) = x_2.$$

$$f_2(t, x_1, x_2) = \cos(t) - x_2 - x_1 - x_1^3.$$

For a system of two first-order differential equations, the symbolic RK4 formulation is

$$k_{1,1} = f_1(t_m, x_{1,m}, x_{2,m}).$$

$$k_{2,1} = f_2(t_m, x_{1,m}, x_{2,m}).$$

$$k_{1,2} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}\right).$$

$$k_{2,2} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}\right).$$

$$k_{1,3} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}\right).$$

$$k_{2,3} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}\right).$$

$$k_{1,4} = f_1(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}).$$

$$k_{2,4} = f_2(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}).$$

$$x_{1,m+1} = x_{1,m} + \frac{h}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}).$$

$$x_{2,m+1} = x_{2,m} + \frac{h}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}).$$

For this case, $h = 0.02$. The first numerical iteration starts from

$$t_0 = 0.000000000.$$

$$x_{1,0} = 0.000000000.$$

$$x_{2,0} = 0.000000000.$$

Then,

$$k_{1,1} = f_1(t_0, x_{1,0}, x_{2,0}) = 0.000000000.$$

$$k_{2,1} = f_2(t_0, x_{1,0}, x_{2,0}) = 1.000000000.$$

$$k_{1,2} = f_1(t_0 + 0.01, x_{1,0} + 0.01k_{1,1}, x_{2,0} + 0.01k_{2,1}) = 0.010000000.$$

$$k_{2,2} = f_2(t_0 + 0.01, x_{1,0} + 0.01k_{1,1}, x_{2,0} + 0.01k_{2,1}) = 0.989950000.$$

$$k_{1,3} = f_1(t_0 + 0.01, x_{1,0} + 0.01k_{1,2}, x_{2,0} + 0.01k_{2,2}) = 0.009899500.$$

$$k_{2,3} = f_2(t_0 + 0.01, x_{1,0} + 0.01k_{1,2}, x_{2,0} + 0.01k_{2,2}) = 0.989950500.$$

$$k_{1,4} = f_1(t_0 + 0.02, x_{1,0} + 0.02k_{1,3}, x_{2,0} + 0.02k_{2,3}) = 0.019799010.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$k_{2,4} = f_2(t_0 + 0.02, x_{1,0} + 0.02k_{1,3}, x_{2,0} + 0.02k_{2,3}) = 0.979803007.$$

The recursive equations give

$$x_{1,1} = x_{1,0} + \frac{0.02}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.000198660.$$

$$x_{2,1} = x_{2,0} + \frac{0.02}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.019798680.$$

For the second numerical iteration,

$$t_1 = 0.020000000.$$

$$x_{1,1} = 0.000198660.$$

$$x_{2,1} = 0.019798680.$$

Then,

$$k_{1,1} = f_1(t_1, x_{1,1}, x_{2,1}) = 0.019798680.$$

$$k_{2,1} = f_2(t_1, x_{1,1}, x_{2,1}) = 0.979802667.$$

$$k_{1,2} = f_1(t_1 + 0.01, x_{1,1} + 0.01k_{1,1}, x_{2,1} + 0.01k_{2,1}) = 0.029596707.$$

$$k_{2,2} = f_2(t_1 + 0.01, x_{1,1} + 0.01k_{1,1}, x_{2,1} + 0.01k_{2,1}) = 0.969556680.$$

$$k_{1,3} = f_1(t_1 + 0.01, x_{1,1} + 0.01k_{1,2}, x_{2,1} + 0.01k_{2,2}) = 0.029494247.$$

$$k_{2,3} = f_2(t_1 + 0.01, x_{1,1} + 0.01k_{1,2}, x_{2,1} + 0.01k_{2,2}) = 0.969561160.$$

$$k_{1,4} = f_1(t_1 + 0.02, x_{1,1} + 0.02k_{1,3}, x_{2,1} + 0.02k_{2,3}) = 0.039189903.$$

$$k_{2,4} = f_2(t_1 + 0.02, x_{1,1} + 0.02k_{1,3}, x_{2,1} + 0.02k_{2,3}) = 0.959221658.$$

The recursive equations give

$$x_{1,2} = x_{1,1} + \frac{0.02}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.000789228.$$

$$x_{2,2} = x_{2,1} + \frac{0.02}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.039189547.$$

For the third numerical iteration,

$$t_2 = 0.040000000.$$

$$x_{1,2} = 0.000789228.$$

$$x_{2,2} = 0.039189547.$$

Then,

$$k_{1,1} = f_1(t_2, x_{1,2}, x_{2,2}) = 0.039189547.$$

$$k_{2,1} = f_2(t_2, x_{1,2}, x_{2,2}) = 0.959221331.$$

$$k_{1,2} = f_1(t_2 + 0.01, x_{1,2} + 0.01k_{1,1}, x_{2,2} + 0.01k_{2,1}) = 0.048781760.$$

$$k_{2,2} = f_2(t_2 + 0.01, x_{1,2} + 0.01k_{1,1}, x_{2,2} + 0.01k_{2,1}) = 0.948787375.$$

$$k_{1,3} = f_1(t_2 + 0.01, x_{1,2} + 0.01k_{1,2}, x_{2,2} + 0.01k_{2,2}) = 0.048677420.$$

$$k_{2,3} = f_2(t_2 + 0.01, x_{1,2} + 0.01k_{1,2}, x_{2,2} + 0.01k_{2,2}) = 0.948795792.$$

$$k_{1,4} = f_1(t_2 + 0.02, x_{1,2} + 0.02k_{1,3}, x_{2,2} + 0.02k_{2,3}) = 0.058165463.$$

$$k_{2,4} = f_2(t_2 + 0.02, x_{1,2} + 0.02k_{1,3}, x_{2,2} + 0.02k_{2,3}) = 0.938272295.$$

Finally,

$$x_{1,3} = x_{1,2} + \frac{0.02}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.001763473.$$

$$x_{2,3} = x_{2,2} + \frac{0.02}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.058165080.$$

After the third iteration,

$$t_3 = 0.060000000.$$

$$x_{1,3} = 0.001763473.$$

$$x_{2,3} = 0.058165080.$$

The numerical integration was then extended to

$$t = 300.$$

to obtain a long-time view of the response. This final time is a choice of the present study and is not attributed to the source problem in [13]. Figure 3 shows the time response and Figure 4 the associated phase portrait.



Figure 3. Time response $x_1(t)$ of the Duffing oscillator obtained with the RK4 method for $h = 0.02$ and $0 \leq t \leq 300$.

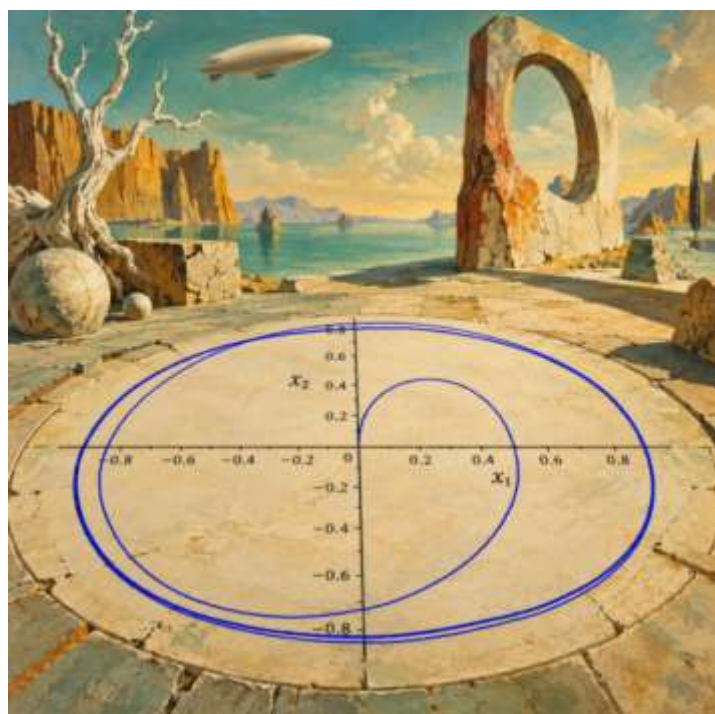


Figure 4. Phase portrait (x_1, x_2) of the Duffing oscillator obtained with the RK4 method.

C. Case 3: Van der Pol Oscillator

The van der Pol equation is a standard nonlinear model for self-sustained oscillation and provides a useful example for combining qualitative and numerical analysis [14]. The form used here is

$$x'' + \varepsilon(x^2 - 1)x' + x = 0.$$

For this case,

$$\varepsilon = 1.$$

The initial conditions are

$$x(0) = 0.01.$$

$$x'(0) = 0.01.$$

Introducing the variables

$$x_1 = x.$$

$$x_2 = x'.$$

the second-order differential equation is rewritten as

$$\frac{dx_1}{dt} = x_2.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$\frac{dx_2}{dt} = \varepsilon(1 - x_1^2)x_2 - x_1.$$

The corresponding initial conditions are

$$x_1(0) = 0.01.$$

$$x_2(0) = 0.01.$$

Following the methodology described in Figure 1 and [7], the system is written in function notation as

$$f_1(t, x_1, x_2) = x_2.$$

$$f_2(t, x_1, x_2) = \varepsilon(1 - x_1^2)x_2 - x_1.$$

For a system of two first-order differential equations, the symbolic RK4 formulation is

$$k_{1,1} = f_1(t_m, x_{1,m}, x_{2,m}).$$

$$k_{2,1} = f_2(t_m, x_{1,m}, x_{2,m}).$$

$$k_{1,2} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}\right).$$

$$k_{2,2} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}\right).$$

$$k_{1,3} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}\right).$$

$$k_{2,3} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}\right).$$

$$k_{1,4} = f_1(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}).$$

$$k_{2,4} = f_2(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}).$$

$$x_{1,m+1} = x_{1,m} + \frac{h}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}).$$

$$x_{2,m+1} = x_{2,m} + \frac{h}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}).$$

For this case, $h = 0.01$. The first numerical iteration starts from

$$t_0 = 0.000000000.$$

$$x_{1,0} = 0.010000000.$$

$$x_{2,0} = 0.010000000.$$

Then,

$$k_{1,1} = f_1(t_0, x_{1,0}, x_{2,0}) = 0.010000000.$$

$$k_{2,1} = f_2(t_0, x_{1,0}, x_{2,0}) = -0.000001000.$$

$$k_{1,2} = f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}) = 0.009999995.$$

$$k_{2,2} = f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}) = -0.000051015.$$

$$k_{1,3} = f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}) = 0.009999745.$$

$$k_{2,3} = f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}) = -0.000051265.$$

$$k_{1,4} = f_1(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}) = 0.009999487.$$

$$k_{2,4} = f_2(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}) = -0.000101530.$$

The recursive equations give

$$x_{1,1} = x_{1,0} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.010099998.$$

$$x_{2,1} = x_{2,0} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.009999488.$$

For the second numerical iteration,

$$t_1 = 0.010000000.$$

$$x_{1,1} = 0.010099998.$$

$$x_{2,1} = 0.009999488.$$

Then,

$$k_{1,1} = f_1(t_1, x_{1,1}, x_{2,1}) = 0.009999488.$$

$$k_{2,1} = f_2(t_1, x_{1,1}, x_{2,1}) = -0.000101530.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$\begin{aligned}k_{1,2} &= f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}) = 0.009998981. \\k_{2,2} &= f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}) = -0.000152045. \\k_{1,3} &= f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}) = 0.009998728. \\k_{2,3} &= f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}) = -0.000152295. \\k_{1,4} &= f_1(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}) = 0.009997965. \\k_{2,4} &= f_2(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}) = -0.000203061.\end{aligned}$$

The recursive equations give

$$\begin{aligned}x_{1,2} &= x_{1,1} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.010199986. \\x_{2,2} &= x_{2,1} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.009997966.\end{aligned}$$

For the third numerical iteration,

$$\begin{aligned}t_2 &= 0.020000000. \\x_{1,2} &= 0.010199986. \\x_{2,2} &= 0.009997966.\end{aligned}$$

Then,

$$\begin{aligned}k_{1,1} &= f_1(t_2, x_{1,2}, x_{2,2}) = 0.009997966. \\k_{2,1} &= f_2(t_2, x_{1,2}, x_{2,2}) = -0.000203061. \\k_{1,2} &= f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}) = 0.009996951. \\k_{2,2} &= f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}) = -0.000254076. \\k_{1,3} &= f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}) = 0.009996696. \\k_{2,3} &= f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}) = -0.000254326. \\k_{1,4} &= f_1(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}) = 0.009995423. \\k_{2,4} &= f_2(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}) = -0.000305591.\end{aligned}$$

Finally,

$$\begin{aligned}x_{1,3} &= x_{1,2} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 0.010299954. \\x_{2,3} &= x_{2,2} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = 0.009995424.\end{aligned}$$

After the third iteration,

$$\begin{aligned}t_3 &= 0.030000000. \\x_{1,3} &= 0.010299954. \\x_{2,3} &= 0.009995424.\end{aligned}$$

The numerical integration was then continued to

$$t = 30.$$

The temporal response shows the transition from the small prescribed initial values toward a sustained nonlinear oscillation, whereas the phase portrait shows the development toward the characteristic closed orbit. Figure 5 presents the time response and Figure 6 the corresponding phase portrait.



Figure 5. Time response $x_1(t)$ of the van der Pol oscillator for unit nonlinear parameter, $h = 0.01$, and $0 \leq t \leq 30$.

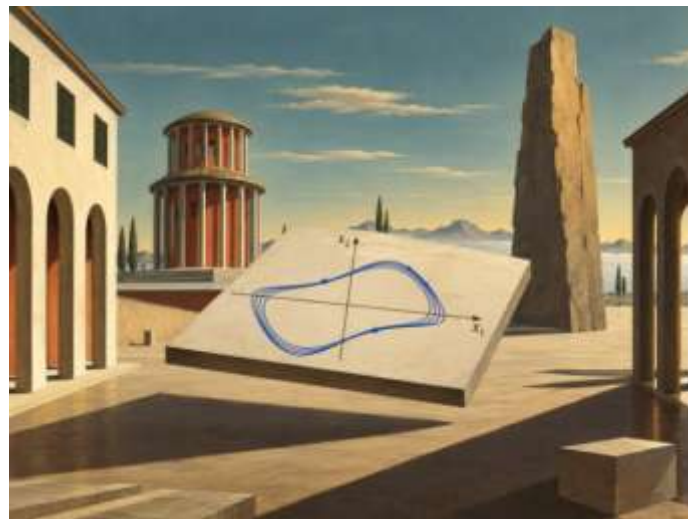


Figure 6. Phase portrait (x_1, x_2) of the van der Pol oscillator obtained with the RK4 method.

D. Case 4: Coupled Strongly Nonlinear System of Cveticanin

The fourth case is based on the coupled strongly nonlinear system studied by Cveticanin [15]:

$$\begin{aligned}\ddot{x} + x^3 - y &= -\varepsilon x^4 y, \\ \ddot{y} + 0.54x^5 - x &= 0.\end{aligned}$$

The initial conditions are

$$\begin{aligned}x(0) &= 1.2, \\ \dot{x}(0) &= 0, \\ y(0) &= 1.6412, \\ \dot{y}(0) &= 0.\end{aligned}$$

For this case,

$$\varepsilon = 0.1.$$

Introducing the variables

$$\begin{aligned}x_1 &= x, \\ x_2 &= \dot{x}, \\ x_3 &= y, \\ x_4 &= \dot{y}.\end{aligned}$$

the coupled second-order system is rewritten as

$$\begin{aligned}\frac{dx_1}{dt} &= x_2, \\ \frac{dx_2}{dt} &= -x_1^3 + x_3 - \varepsilon x_1^4 x_3.\end{aligned}$$

$$\frac{dx_3}{dt} = x_4.$$

$$\frac{dx_4}{dt} = x_1 - 0.54x_1^5.$$

The corresponding initial conditions are

$$x_1(0) = 1.2.$$

$$x_2(0) = 0.$$

$$x_3(0) = 1.6412.$$

$$x_4(0) = 0.$$

Following the methodology described in Figure 1 and [7], the system is written in function notation as

$$f_1(t, x_1, x_2, x_3, x_4) = x_2.$$

$$f_2(t, x_1, x_2, x_3, x_4) = -x_1^3 + x_3 - \varepsilon x_1^4 x_3.$$

$$f_3(t, x_1, x_2, x_3, x_4) = x_4.$$

$$f_4(t, x_1, x_2, x_3, x_4) = x_1 - 0.54x_1^5.$$

For a system of four first-order differential equations, the symbolic RK4 formulation is

$$k_{1,1} = f_1(t_m, x_{1,m}, x_{2,m}, x_{3,m}, x_{4,m}).$$

$$k_{2,1} = f_2(t_m, x_{1,m}, x_{2,m}, x_{3,m}, x_{4,m}).$$

$$k_{3,1} = f_3(t_m, x_{1,m}, x_{2,m}, x_{3,m}, x_{4,m}).$$

$$k_{4,1} = f_4(t_m, x_{1,m}, x_{2,m}, x_{3,m}, x_{4,m}).$$

$$k_{1,2} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}, x_{4,m} + \frac{h}{2}k_{4,1}\right).$$

$$k_{2,2} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}, x_{4,m} + \frac{h}{2}k_{4,1}\right).$$

$$k_{3,2} = f_3\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}, x_{4,m} + \frac{h}{2}k_{4,1}\right).$$

$$k_{4,2} = f_4\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,1}, x_{2,m} + \frac{h}{2}k_{2,1}, x_{3,m} + \frac{h}{2}k_{3,1}, x_{4,m} + \frac{h}{2}k_{4,1}\right).$$

$$k_{1,3} = f_1\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}, x_{4,m} + \frac{h}{2}k_{4,2}\right).$$

$$k_{2,3} = f_2\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}, x_{4,m} + \frac{h}{2}k_{4,2}\right).$$

$$k_{3,3} = f_3\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}, x_{4,m} + \frac{h}{2}k_{4,2}\right).$$

$$k_{4,3} = f_4\left(t_m + \frac{h}{2}, x_{1,m} + \frac{h}{2}k_{1,2}, x_{2,m} + \frac{h}{2}k_{2,2}, x_{3,m} + \frac{h}{2}k_{3,2}, x_{4,m} + \frac{h}{2}k_{4,2}\right).$$

$$k_{1,4} = f_1(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}, x_{4,m} + hk_{4,3}).$$

$$k_{2,4} = f_2(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}, x_{4,m} + hk_{4,3}).$$

$$k_{3,4} = f_3(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}, x_{4,m} + hk_{4,3}).$$

$$k_{4,4} = f_4(t_m + h, x_{1,m} + hk_{1,3}, x_{2,m} + hk_{2,3}, x_{3,m} + hk_{3,3}, x_{4,m} + hk_{4,3}).$$

The recursive equations are

$$x_{1,m+1} = x_{1,m} + \frac{h}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}).$$

$$x_{2,m+1} = x_{2,m} + \frac{h}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}).$$

$$x_{3,m+1} = x_{3,m} + \frac{h}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}).$$

$$x_{4,m+1} = x_{4,m} + \frac{h}{6}(k_{4,1} + 2k_{4,2} + 2k_{4,3} + k_{4,4}).$$

For this case, $h = 0.01$. The first numerical iteration starts from

$$t_0 = 0.000000000.$$

$$x_{1,0} = 1.200000000.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$x_{2,0} = 0.000000000.$$

$$x_{3,0} = 1.641200000.$$

$$x_{4,0} = 0.000000000.$$

Then,

$$k_{1,1} = f_1(t_0, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}) = 0.000000000.$$

$$k_{2,1} = f_2(t_0, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}) = -0.427119232.$$

$$k_{3,1} = f_3(t_0, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}) = 0.000000000.$$

$$k_{4,1} = f_4(t_0, x_{1,0}, x_{2,0}, x_{3,0}, x_{4,0}) = -0.143692800.$$

Next,

$$k_{1,2} = f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}, x_{4,0} + 0.005k_{4,1}) = -0.002135596.$$

$$k_{2,2} = f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}, x_{4,0} + 0.005k_{4,1}) = -0.427119232.$$

$$k_{3,2} = f_3(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}, x_{4,0} + 0.005k_{4,1}) = -0.000718464.$$

$$k_{4,2} = f_4(t_0 + 0.005, x_{1,0} + 0.005k_{1,1}, x_{2,0} + 0.005k_{2,1}, x_{3,0} + 0.005k_{3,1}, x_{4,0} + 0.005k_{4,1}) = -0.143692800.$$

Then,

$$k_{1,3} = f_1(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}, x_{4,0} + 0.005k_{4,2}) = -0.002135596.$$

$$k_{2,3} = f_2(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}, x_{4,0} + 0.005k_{4,2}) = -0.427063838.$$

$$k_{3,3} = f_3(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}, x_{4,0} + 0.005k_{4,2}) = -0.000718464.$$

$$k_{4,3} = f_4(t_0 + 0.005, x_{1,0} + 0.005k_{1,2}, x_{2,0} + 0.005k_{2,2}, x_{3,0} + 0.005k_{3,2}, x_{4,0} + 0.005k_{4,2}) = -0.143643696.$$

Finally,

$$k_{1,4} = f_1(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}, x_{4,0} + 0.01k_{4,3}) = -0.004270638.$$

$$k_{2,4} = f_2(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}, x_{4,0} + 0.01k_{4,3}) = -0.427008445.$$

$$k_{3,4} = f_3(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}, x_{4,0} + 0.01k_{4,3}) = -0.001436437.$$

$$k_{4,4} = f_4(t_0 + 0.01, x_{1,0} + 0.01k_{1,3}, x_{2,0} + 0.01k_{2,3}, x_{3,0} + 0.01k_{3,3}, x_{4,0} + 0.01k_{4,3}) = -0.143594594.$$

The recursive equations give

$$x_{1,1} = x_{1,0} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 1.199978645.$$

$$x_{2,1} = x_{2,0} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = -0.004270823.$$

$$x_{3,1} = x_{3,0} + \frac{0.01}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 1.641192816.$$

$$x_{4,1} = x_{4,0} + \frac{0.01}{6}(k_{4,1} + 2k_{4,2} + 2k_{4,3} + k_{4,4}) = -0.001436601.$$

For the second numerical iteration,

$$t_1 = 0.010000000.$$

$$x_{1,1} = 1.199978645.$$

$$x_{2,1} = -0.004270823.$$

$$x_{3,1} = 1.641192816.$$

$$x_{4,1} = -0.001436601.$$

Then,

$$k_{1,1} = f_1(t_1, x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}) = -0.004270823.$$

$$k_{2,1} = f_2(t_1, x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}) = -0.427008450.$$

$$k_{3,1} = f_3(t_1, x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}) = -0.001436601.$$

$$k_{4,1} = f_4(t_1, x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}) = -0.143594598.$$

Next,

$$k_{1,2} = f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}, x_{4,1} + 0.005k_{4,1}) = -0.006405865.$$

$$k_{2,2} = f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}, x_{4,1} + 0.005k_{4,1}) = -0.426897677.$$

$$k_{3,2} = f_3(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}, x_{4,1} + 0.005k_{4,1}) = -0.002154574.$$

$$k_{4,2} = f_4(t_1 + 0.005, x_{1,1} + 0.005k_{1,1}, x_{2,1} + 0.005k_{2,1}, x_{3,1} + 0.005k_{3,1}, x_{4,1} + 0.005k_{4,1}) = -0.143496410.$$

Then,

$$k_{1,3} = f_1(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}, x_{4,1} + 0.005k_{4,2}) = -0.006405311.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

$$k_{2,3} = f_2(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}, x_{4,1} + 0.005k_{4,2}) = -0.426842301.$$

$$k_{3,3} = f_3(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}, x_{4,1} + 0.005k_{4,2}) = -0.002154083.$$

$$k_{4,3} = f_4(t_1 + 0.005, x_{1,1} + 0.005k_{1,2}, x_{2,1} + 0.005k_{2,2}, x_{3,1} + 0.005k_{3,2}, x_{4,1} + 0.005k_{4,2}) = -0.143447327.$$

Finally,

$$k_{1,4} = f_1(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}, x_{4,1} + 0.01k_{4,3}) = -0.008539246.$$

$$k_{2,4} = f_2(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}, x_{4,1} + 0.01k_{4,3}) = -0.426676189.$$

$$k_{3,4} = f_3(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}, x_{4,1} + 0.01k_{4,3}) = -0.002871074.$$

$$k_{4,4} = f_4(t_1 + 0.01, x_{1,1} + 0.01k_{1,3}, x_{2,1} + 0.01k_{2,3}, x_{3,1} + 0.01k_{3,3}, x_{4,1} + 0.01k_{4,3}) = -0.143300100.$$

The recursive equations give

$$x_{1,2} = x_{1,1} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 1.199914591.$$

$$x_{2,2} = x_{2,1} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = -0.008539431.$$

$$x_{3,2} = x_{3,1} + \frac{0.01}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 1.641171275.$$

$$x_{4,2} = x_{4,1} + \frac{0.01}{6}(k_{4,1} + 2k_{4,2} + 2k_{4,3} + k_{4,4}) = -0.002871238.$$

For the third numerical iteration,

$$t_2 = 0.020000000.$$

$$x_{1,2} = 1.199914591.$$

$$x_{2,2} = -0.008539431.$$

$$x_{3,2} = 1.641171275.$$

$$x_{4,2} = -0.002871238.$$

Then,

$$k_{1,1} = f_1(t_2, x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = -0.008539431.$$

$$k_{2,1} = f_2(t_2, x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = -0.426676184.$$

$$k_{3,1} = f_3(t_2, x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = -0.002871238.$$

$$k_{4,1} = f_4(t_2, x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = -0.143300096.$$

Next,

$$k_{1,2} = f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}, x_{4,2} + 0.005k_{4,1}) = -0.010672812.$$

$$k_{2,2} = f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}, x_{4,2} + 0.005k_{4,1}) = -0.426454724.$$

$$k_{3,2} = f_3(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}, x_{4,2} + 0.005k_{4,1}) = -0.003587738.$$

$$k_{4,2} = f_4(t_2 + 0.005, x_{1,2} + 0.005k_{1,1}, x_{2,2} + 0.005k_{2,1}, x_{3,2} + 0.005k_{3,1}, x_{4,2} + 0.005k_{4,1}) = -0.143103828.$$

Then,

$$k_{1,3} = f_1(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}, x_{4,2} + 0.005k_{4,2}) = -0.010671704.$$

$$k_{2,3} = f_2(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}, x_{4,2} + 0.005k_{4,2}) = -0.426399397.$$

$$k_{3,3} = f_3(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}, x_{4,2} + 0.005k_{4,2}) = -0.003586757.$$

$$k_{4,3} = f_4(t_2 + 0.005, x_{1,2} + 0.005k_{1,2}, x_{2,2} + 0.005k_{2,2}, x_{3,2} + 0.005k_{3,2}, x_{4,2} + 0.005k_{4,2}) = -0.143054801.$$

Finally,

$$k_{1,4} = f_1(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}, x_{4,2} + 0.01k_{4,3}) = -0.012803425.$$

$$k_{2,4} = f_2(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}, x_{4,2} + 0.01k_{4,3}) = -0.426122693.$$

$$k_{3,4} = f_3(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}, x_{4,2} + 0.01k_{4,3}) = -0.004301786.$$

$$k_{4,4} = f_4(t_2 + 0.01, x_{1,2} + 0.01k_{1,3}, x_{2,2} + 0.01k_{2,3}, x_{3,2} + 0.01k_{3,3}, x_{4,2} + 0.01k_{4,3}) = -0.142809610.$$

The recursive equations give

$$x_{1,3} = x_{1,2} + \frac{0.01}{6}(k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}) = 1.199807871.$$

$$x_{2,3} = x_{2,2} + \frac{0.01}{6}(k_{2,1} + 2k_{2,2} + 2k_{2,3} + k_{2,4}) = -0.012803609.$$

$$x_{3,3} = x_{3,2} + \frac{0.01}{6}(k_{3,1} + 2k_{3,2} + 2k_{3,3} + k_{3,4}) = 1.641135405.$$

$$x_{4,3} = x_{4,2} + \frac{0.01}{6}(k_{4,1} + 2k_{4,2} + 2k_{4,3} + k_{4,4}) = -0.004301949.$$

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

After the third iteration,

$$\begin{aligned}t_3 &= 0.030000000. \\x_{1,3} &= 1.199807871. \\x_{2,3} &= -0.012803609. \\x_{3,3} &= 1.641135405. \\x_{4,3} &= -0.004301949.\end{aligned}$$

The numerical integration was then continued to

$$t = 30.$$

Figure 7 presents the simultaneous temporal behavior of the two displacement variables, while Figure 8 shows the trajectory in the (x, y) plane.



Figure 7. Time histories $x(t)$ and $y(t)$ for the coupled strongly nonlinear Cveticanin system with $\varepsilon = 0.1$ and $h = 0.01$.



Figure 8. Trajectory in the (x, y) plane for the coupled strongly nonlinear Cveticanin system.

V. DISCUSSION

Surrealism emerged in the early twentieth century as a cultural and artistic movement concerned with the relation between conscious experience, dreams, automatic processes, unexpected association, and the unconscious. Guillaume Apollinaire used the term *surrealism* in 1917, while André Breton gave the movement a programmatic formulation in the *Manifesto of Surrealism* in 1924 [16]. Surrealist practice developed across national boundaries and through a wide variety of visual languages [17]. Recent historical and critical studies published in 2024 also show that surrealism remains an active subject of interpretation rather than a closed historical episode [18, 19]. Among its major visual exponents are Salvador Dalí, René Magritte, Max Ernst, and Joan Miró, together with artists such as Leonora Carrington, Remedios Varo, Dorothea Tanning, and Man Ray.

The characteristics relevant to this article are not a particular color palette or a single pictorial technique. Surrealist imagery is more closely associated with dreamlike relations, displacement of familiar objects, altered scale and perspective, ambiguous

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

environments, suspended or impossible objects, and combinations that contradict ordinary physical expectations. The matte stone, plaster, ochre, sand, and desaturated blue textures used in several figures of this study are therefore a visual choice for the present compositions, not a defining historical property of surrealism.

All numerical simulations were independently performed in Maple 2021 on a computer running Ubuntu 20.04.6 LTS (64-bit), with an Intel Core i7-7700 processor at 3.60 GHz, 62.7 GiB of RAM, and 2.0 TB of storage. The graphical device was reported by the operating system as NV137, with GNOME 3.36.8 under X11. The same computational environment was used for the four cases so that the mathematical procedure and the implementation remained consistent.

The four problems follow the didactic perspective introduced in [7]. Figure 1 summarizes the common guide: model and initial conditions, reduction to first-order form when required, definition of the functions f_i , construction of the double-index RK4 constants, explicit development of the first numerical iterations, and complete computational simulation. In the original didactic strategy, this organization was intended to make higher-order ODE calculations systematic. Here it is extended to nonlinear systems of increasing dimension and combined with a distinct visual presentation layer.

The surreal background of Figure 1 uses soft, deformed digital clocks displaying 11:14, 16:09, and 19:27. The conceptual reference is the association between melting or soft clocks and Dalí's *The Persistence of Memory* (1931) [20]. The composition does not reproduce Dalí's painting. Instead, it updates the soft-clock idea through seven-segment digital displays, a device that did not belong to the visual technology of Dalí's period. The mathematical diagram remains the methodological guide, whereas the clock environment establishes the surrealist visual context.

The Lorenz system is the simplest case in terms of preprocessing because it is already written as three first-order equations. Nevertheless, each RK4 step requires twelve constants, so the double-index notation immediately becomes useful for distinguishing the equation number from the stage number. The trajectory in Figure 2 shows the familiar two-lobed geometric organization associated with Lorenz dynamics. The surrounding spatial treatment places the graph on an architectural plane that does not behave as an ordinary scientific plotting surface. Floating architecture, unusual depth, and ambiguous perspective create the surrealist component while the attractor itself is retained. This conceptual use of impossible spatial placement can be related to the broader logic of Magritte's suspended and displaced objects, such as the castle-bearing rock in *The Castle of the Pyrenees* [21], without reproducing its composition.

For the Duffing oscillator, the second-order equation must first be converted into two first-order equations. The problem therefore requires eight stage constants per step. The time response in Figure 3 and the phase portrait in Figure 4 are complementary views of exactly the same numerical solution: one preserves time explicitly, while the other emphasizes the geometry of the state variables. Duffing-type models continue to be relevant in nonlinear vibration control and energy-harvesting research [22]. The long perspectives, arches, isolated geometric objects, extended shadows, and large architectural planes used in the visual setting are conceptually connected with the metaphysical spatial language associated with Giorgio de Chirico, an important precursor of surrealism. *Gare Montparnasse (The Melancholy of Departure)* provides an institutional example of this type of disquieting perspective [23]. The connection is conceptual rather than compositional.

The van der Pol oscillator introduces self-excited dynamics. Starting from the small initial values $x_1(0) = x_2(0) = 0.01$, the numerical solution evolves toward sustained oscillation. The temporal response and phase portrait therefore allow the transient and the emergence of the stable oscillatory regime to be observed from two different representations. Modern coupled electromechanical studies continue to use van der Pol-type self-excited oscillators, including energy-harvesting applications [24]. In Figure 5, the landscape contains floating landforms, a seated candle-faced figure, and a disc-shaped flying object suspended above an impossible environment. The invented terrain can be conceptually related to the otherworldly landscapes associated with Yves Tanguy, represented for example by *The Satin Tuning Fork* (1940) [25]. Figure 6 uses a more architectural and metaphysical setting: the phase plane is presented as a tilted slab in a space governed by ambiguous scale and perspective.

The Cvetcanin system is the most demanding of the four examples. Two coupled second-order equations become four first-order equations, and each RK4 step therefore contains sixteen constants. This case makes the pedagogical value of the double-index notation particularly visible: the first index tracks four equations and the second tracks four stages. The time histories in Figure 7 show bounded oscillatory behavior of the two principal variables, whereas Figure 8 removes the independent variable and shows the geometric coupling between x and y . The two views are complementary descriptions of the same numerical data.

The visual treatment of Figure 7 uses a graph embedded in a stone-like floor beneath a cosmic landscape containing a suspended shark and a large ringed planet. The surreal effect comes from incompatible environments, contradictory scale, and the apparent suspension of objects against ordinary gravity. Figure 8 places the (x, y) trajectory on an architectural wall opening toward the sea, with a floating heavy cube and a suspended sailboat. Again, the matte palette is only a pictorial treatment; the surrealist quality lies in displacement, altered spatial relation, and impossible physical coexistence.

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

From the mathematical point of view, the four cases form a clear progression. The Lorenz system uses three first-order equations and twelve constants per step. The Duffing and van der Pol oscillators each begin as one second-order equation, become two first-order equations, and require eight constants. The Cveticanin model begins as two coupled second-order equations, becomes four first-order equations, and requires sixteen constants. Despite this increasing complexity, the same sequence in Figure 1 is preserved. This invariance is pedagogically useful because students can recognize one algorithmic structure across different nonlinear models instead of treating every problem as an unrelated procedure.

From the perspective of school mathematical discourse, the proposal should be interpreted cautiously. The mathematical object, the RK4 algorithm, the numerical data, and the formal reasoning are not replaced by the surrealist compositions. What changes is one layer of communication: the conventional isolated scientific graph is placed in a different visual environment. In this sense, the work explores a partial movement away from standardized modes of presentation without claiming that the visual intervention alone redesigns or eliminates dME. No experiment with students was performed in this study, so no claim is made that the surrealist treatment improves learning, motivation, or retention. Those questions require direct educational evaluation.

VI. CONCLUSIONS

This article presented a didactic application of the fourth-order Runge-Kutta method to four nonlinear initial-value problems: the Lorenz system, Duffing oscillator, van der Pol oscillator, and a coupled strongly nonlinear system proposed by Cveticanin. The work extends the strategy reported in [7] by applying the same double-index organization to systems of different dimensions and by developing the first three RK4 iterations explicitly before the complete numerical simulation. This sequence makes the method visible as a numerical construction rather than as a result produced only by software.

The four cases also show the scalability of the notation. The number of RK4 constants increases from eight in the reduced second-order oscillators to twelve in the Lorenz system and sixteen in the Cveticanin problem, but the structure of the calculation does not change. The Maple 2021 routines included in the Appendix reproduce the same organization and were used to obtain the complete numerical trajectories.

A second contribution is the systematic integration of surrealist visual compositions with scientific numerical graphs. The curves, axes, scales, and numerical relationships were preserved, whereas the surrounding environments were changed through displacement, unusual perspective, altered scale, suspended objects, and dreamlike spatial relations. In relation to school mathematical discourse, this visual intervention is presented as an exploratory way of broadening how a numerical result can be communicated, not as evidence that a different visual style by itself changes mathematical understanding.

The present work is methodological and visual rather than an educational experiment. Future work should evaluate how students interpret these compositions, whether the visual setting affects attention or engagement, whether it helps or interferes with reading the numerical information, and how it compares with conventional graphs. Similar studies could also examine other multimodal or artistic languages while keeping the mathematical content controlled.

APPENDIX. MAPLE 2021 ROUTINES

The following routines were executed and verified in Maple 2021. Each program prints the first three iterations and then performs the complete simulation for the corresponding case. The didactic developments presented in the main text follow Figure 1, where the complete double-index RK4 formulation is retained in order to make each stage of the method explicit for instructional purposes. However, the Maple routines included in this Appendix were constructed according to the optimized RK4 strategy presented in Figure 11 of the previous work [7]. In that formulation, auxiliary intermediate variables are introduced to avoid repeated calculations during the evaluation of the Runge-Kutta stages and, consequently, to reduce computational time. Thus, Figure 1 is used here as the pedagogical formulation of the method, whereas the Appendix routines follow the optimized computational implementation. Further details on both formulations and their comparison can be found in [7].

A. Lorenz System

```
restart;
with(plots):
Digits := 9;
```

```
# =====
# CASE STUDY 1: LORENZ SYSTEM
# Classical fourth-order Runge-Kutta method
# =====
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
# Final time and step size
t0 := 0.0:
tf := 35.2:
h := 0.01:

# -----
# First-order system
# -----
f1 := proc(T, X1, X2, X3)
  local CA:
  CA := 10*(X2-X1):
  return CA:
end proc:

f2 := proc(T, X1, X2, X3)
  local CB:
  CB := X1*(28-X3)-X2:
  return CB:
end proc:

f3 := proc(T, X1, X2, X3)
  local CC:
  CC := X1*X2-(8/3)*X3:
  return CC:
end proc:

# -----
# Initial conditions
# -----
x1nt := 0.1:
x2nt := 0.0:
x3nt := 0.0:
indice := 1:

# -----
# Lists used to store the numerical solution
# -----
t_pts := [t0]:
x1_pts := [x1nt]:
x2_pts := [x2nt]:
x3_pts := [x3nt]:

# CPU time
st := time():

# -----
# RK4 loop
# -----
for t from h by h to tf do
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
tn := t-h:

# Stage 1
K11 := f1(tn, x1nt, x2nt, x3nt):
K21 := f2(tn, x1nt, x2nt, x3nt):
K31 := f3(tn, x1nt, x2nt, x3nt):

# Stage 2
K12 := f1(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2):
K22 := f2(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2):
K32 := f3(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2):

# Stage 3
K13 := f1(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2):
K23 := f2(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2):

K33 := f3(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2):

# Stage 4
K14 := f1(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33):
K24 := f2(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33):
K34 := f3(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33):

# Update
x1n := x1nt+(h/6)*(K11+2*K12+2*K13+K14):
x2n := x2nt+(h/6)*(K21+2*K22+2*K23+K24):
x3n := x3nt+(h/6)*(K31+2*K32+2*K33+K34):

# Print the first three iterations
if indice <= 3 then
  print("Indice", indice):
  print("t", t):
  print("K11",K11): print("K21",K21): print("K31",K31):
  print("K12",K12): print("K22",K22): print("K32",K32):
  print("K13",K13): print("K23",K23): print("K33",K33):
  print("K14",K14): print("K24",K24): print("K34",K34):
  print("x1",x1n): print("x2",x2n): print("x3",x3n):
  print("-----"):
end if:

# Update current values
x1nt := x1n:
x2nt := x2n:
x3nt := x3n:

# Store values
t_pts := [op(t_pts),t]:
x1_pts := [op(x1_pts),x1nt]:
x2_pts := [op(x2_pts),x2nt]:
x3_pts := [op(x3_pts),x3nt]:
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
indice := indice+1:
end do:

# -----
# Calculation time
# -----
time()-st;

# -----
# Three-dimensional trajectory
# -----
pointplot3d( x1_pts, x2_pts, x3_pts,
color=blue, style=line,
labels=["x1", "x2", "x3"]);
```

B. Duffing Oscillator

```
restart;
with(plots):
Digits := 9:

# =====
# CASE STUDY 2: DUFFING OSCILLATOR
#  $x'' + x' + x + x^3 = \cos(t)$ 
# =====

t0 := 0.0:
tf := 300.0:
h := 0.02:

# -----
# First-order system
# -----
f1 := proc(T, X1, X2)
local CA:
CA := X2:
return CA:
end proc:

f2 := proc(T, X1, X2)
local CB:
CB := cos(T)-X2-X1-X1^3:
return CB:
end proc:

# -----
# Initial conditions
# -----
x1nt := 0.0:
x2nt := 0.0:
indice := 1:

# -----
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
# Lists used to store the numerical solution
# -----
t_pts := [t0]:
x1_pts := [x1nt]:
x2_pts := [x2nt]:

st := time():

# -----
# RK4 loop
# -----
for t from h by h to tf do

  tn := t-h:

  # Stage 1
  K11 := f1(tn, x1nt, x2nt):
  K21 := f2(tn, x1nt, x2nt):

  # Stage 2
  K12 := f1(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2):
  K22 := f2(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2):

  # Stage 3
  K13 := f1(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2):
  K23 := f2(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2):

  # Stage 4
  K14 := f1(tn+h, x1nt+h*K13, x2nt+h*K23):
  K24 := f2(tn+h, x1nt+h*K13, x2nt+h*K23):

  # Update
  x1n := x1nt+(h/6)*(K11+2*K12+2*K13+K14):
  x2n := x2nt+(h/6)*(K21+2*K22+2*K23+K24):

  # Print the first three iterations
  if indice <= 3 then
    print("Indice",indice):
    print("t",t):
    print("K11",K11): print("K21",K21):
    print("K12",K12): print("K22",K22):
    print("K13",K13): print("K23",K23):
    print("K14",K14): print("K24",K24):
    print("x1",x1n): print("x2",x2n):
    print("-----"):
  end if:

  # Update current values
  x1nt := x1n:
  x2nt := x2n:
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
# Store values
t_pts := [op(t_pts),t]:
x1_pts := [op(x1_pts),x1nt]:
x2_pts := [op(x2_pts),x2nt]:

indice := indice+1:
end do:

time()-st;

# -----
# Time response x1(t)
# -----
P1 := plot( [seq([t_pts[i],x1_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, labels=["t","x1"]):
display(P1);

# -----
# Phase portrait x1-x2
# -----
P2 := plot( [seq([x1_pts[i],x2_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, labels=["x1","x2"]):
display(P2);
```

C. Van der Pol Oscillator

```
restart;
with(plots):
Digits := 9:

# =====
# CASE STUDY 3: VAN DER POL OSCILLATOR
#  $x'' + \epsilon(x^2 - 1)x' + x = 0$ 
# =====

t0 := 0.0:
tf := 30.0:
h := 0.01:
epsilon := 1.0:

# -----
# First-order system
# -----
f1 := proc(T, X1, X2)
local CA:
CA := X2:
return CA:
end proc:

f2 := proc(T, X1, X2)
local CB:
global epsilon:
CB := epsilon*(1-X1^2)*X2-X1:
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
return CB:
end proc:

# -----
# Initial conditions
# -----
x1nt := 0.01:
x2nt := 0.01:
indice := 1:

# -----
# Lists used to store the numerical solution
# -----
t_pts := [t0]:
x1_pts := [x1nt]:
x2_pts := [x2nt]:

st := time():

# -----
# RK4 loop
# -----
for t from h by h to tf do

  tn := t-h:

  # Stage 1
  K11 := f1(tn, x1nt, x2nt):
  K21 := f2(tn, x1nt, x2nt):

  # Stage 2
  K12 := f1(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2):
  K22 := f2(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2):

  # Stage 3
  K13 := f1(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2):
  K23 := f2(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2):

  # Stage 4
  K14 := f1(tn+h, x1nt+h*K13, x2nt+h*K23):
  K24 := f2(tn+h, x1nt+h*K13, x2nt+h*K23):

  # Update
  x1n := x1nt+(h/6)*(K11+2*K12+2*K13+K14):
  x2n := x2nt+(h/6)*(K21+2*K22+2*K23+K24):

  # Print the first three iterations
  if indice <= 3 then
    print("Indice", indice):
    print("t", t):
    print("K11", K11): print("K21", K21):
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
print("K12",K12): print("K22",K22):
print("K13",K13): print("K23",K23):
print("K14",K14): print("K24",K24):
print("x1",x1n): print("x2",x2n):
print("-----"):
end if:

# Update current values
x1nt := x1n:
x2nt := x2n:

# Store values
t_pts := [op(t_pts),t]:
x1_pts := [op(x1_pts),x1nt]:
x2_pts := [op(x2_pts),x2nt]:

indice := indice+1:
end do:

time()-st;

# -----
# Time response x1(t)
# -----
P1 := plot( [seq([t_pts[i],x1_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, labels=["t","x1"]):
display(P1);

# -----
# Phase portrait x1-x2
# -----
P2 := plot( [seq([x1_pts[i],x2_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, labels=["x1","x2"]):
display(P2);
```

D. Coupled Cveticanin System

```
restart;
with(plots):
Digits := 9:

# =====
# CASE STUDY 4: COUPLED STRONGLY NONLINEAR SYSTEM
#
#  $x'' + x^3 - y = -\epsilon x^4 y$ 
#  $y'' + 0.54 x^5 - x = 0$ 
# =====

t0 := 0.0:
tf := 30.0:
h := 0.01:
epsilon := 0.1:
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
# -----
# First-order system
# x1=x, x2=x', x3=y, x4=y'
# -----
f1 := proc(T, X1, X2, X3, X4)
  local CA:
  CA := X2:
  return CA:
end proc:

f2 := proc(T, X1, X2, X3, X4)
  local CB:
  global epsilon:
  CB := -X1^3+X3-epsilon*X1^4*X3:
  return CB:
end proc:

f3 := proc(T, X1, X2, X3, X4)
  local CC:
  CC := X4:
  return CC:
end proc:

f4 := proc(T, X1, X2, X3, X4)
  local CD:
  CD := X1-0.54*X1^5:
  return CD:
end proc:

# -----
# Initial conditions
# -----
x1nt := 1.2:
x2nt := 0.0:
x3nt := 1.6412:
x4nt := 0.0:
indice := 1:

# -----
# Lists used to store the numerical solution
# -----
t_pts := [t0]:
x1_pts := [x1nt]:
x2_pts := [x2nt]:
x3_pts := [x3nt]:
x4_pts := [x4nt]:
st := time():

# -----
# RK4 loop
# -----
```

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

for t from h by h to tf do

tn := t-h:

Stage 1

K11 := f1(tn,x1nt,x2nt,x3nt,x4nt):

K21 := f2(tn,x1nt,x2nt,x3nt,x4nt):

K31 := f3(tn,x1nt,x2nt,x3nt,x4nt):

K41 := f4(tn,x1nt,x2nt,x3nt,x4nt):

Stage 2

K12 := f1(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2, x4nt+h*K41/2):

K22 := f2(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2, x4nt+h*K41/2):

K32 := f3(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2, x4nt+h*K41/2):

K42 := f4(tn+h/2, x1nt+h*K11/2, x2nt+h*K21/2, x3nt+h*K31/2, x4nt+h*K41/2):

Stage 3

K13 := f1(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2, x4nt+h*K42/2):

K23 := f2(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2, x4nt+h*K42/2):

K33 := f3(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2, x4nt+h*K42/2):

K43 := f4(tn+h/2, x1nt+h*K12/2, x2nt+h*K22/2, x3nt+h*K32/2, x4nt+h*K42/2):

Stage 4

K14 := f1(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33, x4nt+h*K43):

K24 := f2(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33, x4nt+h*K43):

K34 := f3(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33, x4nt+h*K43):

K44 := f4(tn+h, x1nt+h*K13, x2nt+h*K23, x3nt+h*K33, x4nt+h*K43):

Update

x1n := x1nt+(h/6)*(K11+2*K12+2*K13+K14):

x2n := x2nt+(h/6)*(K21+2*K22+2*K23+K24):

x3n := x3nt+(h/6)*(K31+2*K32+2*K33+K34):

x4n := x4nt+(h/6)*(K41+2*K42+2*K43+K44):

Print the first three iterations

if indice <= 3 then

print("Indice",indice):

print("t",t):

print("K11",K11): print("K21",K21):

print("K31",K31): print("K41",K41):

print("K12",K12): print("K22",K22):

print("K32",K32): print("K42",K42):

print("K13",K13): print("K23",K23):

print("K33",K33): print("K43",K43):

print("K14",K14): print("K24",K24):

print("K34",K34): print("K44",K44):

print("x1",x1n): print("x2",x2n):

print("x3",x3n): print("x4",x4n):

print("-----"):

end if:

Solving Differential Equations Using Fourth-Order Runge-Kutta in The Context of Surrealism

```
# Update current values
x1nt := x1n:
x2nt := x2n:
x3nt := x3n:
x4nt := x4n:

# Store values
t_pts := [op(t_pts),t]:
x1_pts := [op(x1_pts),x1nt]:
x2_pts := [op(x2_pts),x2nt]:
x3_pts := [op(x3_pts),x3nt]:
x4_pts := [op(x4_pts),x4nt]:

indice := indice+1:
end do:

time()-st;

# -----
# Time histories x(t) and y(t)
# -----
P1 := plot( [seq([t_pts[i],x1_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, legend="x(t)":

P2 := plot( [seq([t_pts[i],x3_pts[i]],i=1..nops(x3_pts))],
style=line, color=black, legend="y(t)" ):
display([P1,P2],labels=["t", "x,y"]);

# -----
# x-y trajectory
# -----
P3 := plot( [seq([x1_pts[i],x3_pts[i]],i=1..nops(x1_pts))],
style=line, color=blue, labels=["x", "y"]):
display(P3);
```

DISCLAIMER ON AI-ASSISTED TOOLS

ChatGPT (OpenAI) was used as an auxiliary generative-AI tool to produce surrealist visual compositions according to concepts, prompts, and iterative instructions provided by the authors; to assist with language, grammar, and style revision; and to support bibliographic searches. The numerical solutions were obtained independently with Maple 2021. All mathematical formulations, simulations, numerical verification, reference selection, scientific interpretation, and final decisions were reviewed by and remain the responsibility of the authors.

ACKNOWLEDGMENT

The authors acknowledge their respective institutions for the academic and computational environment that supported this work.

REFERENCES

- 1) R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 10th ed. Boston, MA, USA: Cengage Learning, 2016.
- 2) S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2021.
- 3) D. Das, S. Waghmare, P. Padmanabhan, V. Kumaresan, and D. P. Sudhakar, "Control of opening duration in a

- pneumatically operated valve with two-fluid combination and quadratic damping,” *Journal of Engineering and Applied Science*, vol. 70, art. 41, 2023, doi: 10.1186/s44147-023-00207-7.
- 4) G. M. Moatimid, T. S. Amer, and W. S. Amer, “Dynamical analysis of a damped harmonic forced Duffing oscillator with time delay,” *Scientific Reports*, vol. 13, art. 6507, 2023, doi: 10.1038/s41598-023-33461-z.
 - 5) D. F. Fernandes, M. C. Santos, A. C. Silva, and A. M. M. Lima, “Comparative study of CUDA-based parallel programming in C and Python for GPU acceleration of the 4th order Runge-Kutta method,” *Nuclear Engineering and Design*, vol. 421, art. 113050, 2024, doi: 10.1016/j.nucengdes.2024.113050.
 - 6) K.-B. Lin and T.-L. Tien, “Analysis of One-Degree-of-Freedom Spring-Mass-Damper System with Nonlinear Spring Using Runge-Kutta Method,” *Engineering Proceedings*, vol. 92, no. 1, art. 1, 2025, doi: 10.3390/engproc2025092001.
 - 7) M. A. Sandoval-Hernandez, H. Jimenez-Islas, S. E. Torreblanca-Bouchan, F. I. Molina-Herrera, J. J. Villegas-Saucillo, and H. Vazquez-Leal, “Estrategia didáctica para la solución numérica de ecuaciones diferenciales ordinarias de orden superior,” *Revista digital Matemática, Educación e Internet*, vol. 26, no. 2, 2026, doi: 10.18845/rdmei.v26i2.8420.
 - 8) D. Soto and R. Cantoral, “Discurso Matemático Escolar y Exclusión. Una Visión Socioepistemológica,” *Bolema*, vol. 28, no. 50, pp. 1525-1544, 2014, doi: 10.1590/1980-4415v28n50a25.
 - 9) R. Cantoral, G. Montiel, and D. Reyes-Gasperini, “Análisis del discurso Matemático Escolar en los libros de texto, una mirada desde la Teoría Socioepistemológica,” *Avances de Investigación en Educación Matemática*, no. 8, pp. 9-28, 2015, doi: 10.35763/aiem.v1i8.123.
 - 10) M. A. Sandoval-Hernandez, S. Hernandez-Mendez, S. E. Torreblanca-Bouchan, and G. U. Diaz-Arango, “Actualización de contenidos en el campo disciplinar de matemáticas del componente propedéutico del bachillerato tecnológico: el caso de las funciones especiales,” *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol. 12, no. 23, e278, 2021, doi: 10.23913/ride.v12i23.1044.
 - 11) M. G. Simón Ramos, “Elementos para el diseño de Situaciones de Aprendizaje para la enseñanza de las matemáticas con perspectiva de género,” *Educación Matemática*, vol. 37, no. 3, pp. 191-213, 2025, doi: 10.24844/EM3703.07.
 - 12) E. N. Lorenz, “Deterministic Nonperiodic Flow,” *Journal of the Atmospheric Sciences*, vol. 20, pp. 130-141, 1963, doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.
 - 13) M. Szymura and M. Pleszczyński, “Beyond Numerical Discretization: The Differential Transform Method as an Efficient Framework for Solving the Duffing Equation,” *Symmetry*, vol. 18, no. 6, art. 1030, 2026, doi: 10.3390/sym18061030.
 - 14) V. J. Pinto and L. Salgado, “The van der Pol equation: qualitative and numerical study,” *Revista de Matemática: Teoría y Aplicaciones*, vol. 30, no. 2, pp. 229-251, 2023, doi: 10.15517/rmta.v30i2.50545.
 - 15) L. Cveticanin, “Analytic solution of the system of two coupled differential equations with the fifth-order non-linearity,” *Physica A: Statistical Mechanics and its Applications*, vol. 317, nos. 1-2, pp. 83-94, 2003, doi: 10.1016/S0378-4371(02)01323-7.
 - 16) A. Breton, *Manifestoes of Surrealism*, R. Seaver and H. R. Lane, Trans. Ann Arbor, MI, USA: University of Michigan Press, 1969, doi: 10.3998/mpub.7558.
 - 17) S. D’Alessandro and M. Gale, Eds., *Surrealism Beyond Borders*. New York, NY, USA: The Metropolitan Museum of Art, 2021.
 - 18) M. Polizzotti, *Why Surrealism Matters*. New Haven, CT, USA: Yale University Press, 2024, doi: 10.2307/jj.10329835.
 - 19) Institut für Kulturaustausch in Tübingen and M. Letze, Eds., *Surrealism: Worlds in Dialogue*. Munich, Germany: Hirmer Publishers, 2024.
 - 20) The Museum of Modern Art, “Salvador Dalí. *The Persistence of Memory*. 1931,” collection record, New York, NY, USA.
 - 21) R. Magritte and H. Torczyner, *Le château de Magritte à Jérusalem [Magritte’s Castle in Jerusalem]*. Jerusalem, Israel: The Israel Museum, 1991.
 - 22) P. Kakou, S. K. Gupta, and O. Barry, “A nonlinear analysis of a Duffing oscillator with a nonlinear electromagnetic vibration absorber-inerter for concurrent vibration mitigation and energy harvesting,” *Nonlinear Dynamics*, vol. 112, pp. 5847-5862, 2024, doi: 10.1007/s11071-023-09163-6.
 - 23) The Museum of Modern Art, “Giorgio de Chirico. *Gare Montparnasse (The Melancholy of Departure)*. Paris, early 1914,” collection record, New York, NY, USA.
 - 24) G. Sani, M. Bednarek, K. Witkowski, and J. Awrejcewicz, “Dynamics and energy harvesting from parametrically coupled self-excited electromechanical oscillator,” *Nonlinear Dynamics*, vol. 112, pp. 11785-11802, 2024, doi: 10.1007/s11071-024-09698-2.
 - 25) The Metropolitan Museum of Art, “Yves Tanguy. *The Satin Tuning Fork*. 1940,” collection record, New York, NY, USA.