

Retraction

Retracted: Indifference Computer Experiment for Mathematical Identification of Two Variables

Wireless Communications and Mobile Computing

Received 18 July 2023; Accepted 18 July 2023; Published 19 July 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] J. Yu, "Indifference Computer Experiment for Mathematical Identification of Two Variables," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 9181840, 7 pages, 2022.

Research Article

Indifference Computer Experiment for Mathematical Identification of Two Variables

Jianxi Yu 

Henan Institute of Economics and Trade, Zhengzhou, Henan 450018, China

Correspondence should be addressed to Jianxi Yu; 31115314@njau.edu.cn

Received 1 June 2022; Revised 21 June 2022; Accepted 1 July 2022; Published 29 July 2022

Academic Editor: Balakrishnan Nagaraj

Copyright © 2022 Jianxi Yu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to understand the two types of nonlinear differential equation problems in engineering dynamics, the author proposes a numerical analysis method for the two types of nonlinear differential equations based on computer simulation. This method establishes the MATLAB algorithm structure of the numerical solution of the fourth-order fixed-step Runge-Kutta and Lorenz models, discusses the error control in the case of variable step size, and plots the numerical solutions of the Lorenz system based on MATLAB in two-dimensional and three-dimensional space graphics. The x -direction displacement and y -direction displacement data are extracted from the Lorenz equation as iterative samples of the model, the regression curve obtained after iteration has a slope of 0.996, and the iterative regression model reflects the basic characteristics of the data well. This method presents the basic idea of numerical solution verification within acceptable error limits. For solving engineering problems with differential equations as mathematical models, an effective numerical solution method is provided, and further discussion on the numerical solutions of partial differential equations is of great significance.

1. Introduction

We know that the dynamic changes of many systems in the real world can be expressed by differential equations; in order to study the evolution law of the system, the solution of differential equations has become a very important problem. The theory and method of differential equations have always been one of the keys to the development of modern science and technology and also one of the key issues of mathematical research. However, most differential equations cannot be solved analytically, and even some differential equations that can be solved require very high skills; differential equations are a compilation of tricks [1]. Therefore, while people are looking for analytic solutions, discussing and studying the analytical features of the solutions have also become one of people's research methods. Especially with the rapid development of nonlinear scientific research, people not only need to discuss individual solutions of differential equations but also try to know the general trend and structure of a large class of solutions or all solutions; it is not only satisfied with the discussion of the local properties of the solution but also cares about the topological struc-

ture of the orbital formed by a solution in the corresponding space and even cares whether this structure will be destroyed due to small perturbations, in the case of perturbation and no perturbation, whether there will be orbital structures that are different from each other. People hope to study differential equations from a higher perspective, with newer viewpoints, and with more advanced methods, so as to obtain more and more profound results, which leads to the rise of the research upsurge of differential dynamical systems [2]. To put it simply, what the differential dynamic system pursues is to analyze the differential equations from the perspective of topology, so as to reveal the essential laws of the change and development of things.

The world we live in and face is an evolutionary system that is extremely complex. Complexity is everywhere, and complex systems are everywhere, such as cosmic celestial bodies, biological systems, and social systems. Currently, there is no unified understanding of complexity. It is generally believed that complexity can be summed up as the multilevel, multifactor, and multivariability of the system, the interaction between various factors or subsystems and between the system and the environment, and the ensuing

overall behavior and evolution [3]. The so-called complexity research is to study the composition, structure, function, and interaction of the system, study the overall behavior and evolution of the system and the mechanisms that control them, and then establish models, conduct simulation experiments, and further influence, manage, and control them.

2. Literature Review

For this research question, Tu et al. based on the probability representation of linear backward stochastic differential equations proposed an approximate solution method for linear partial differential equations [4]. Leonov proposed an approximate solution to nonlinear parabolic partial differential equations based on second-order backward stochastic differential equations [5]. Zhu gave a numerical algorithm based on the probabilistic representation of the branch-diffusion process, which exploits the fact that the solutions of semilinear PDEs with a polynomial nonlinear form can be expressed as the expectation of the branch-diffusion process functional; although this method does not have the curse of dimensionality problem, its applicability is still limited due to the instability of the approximate solution in finite time [6]. For the high-dimensional case of such equations, Nafil et al. developed an algorithm based on compressed sensing and the Hopf formulation of the Hamilton-Jacobi equation, which achieved good numerical performance in high-dimensional cases [7]. Abdulwahid et al. proposed a general algorithm for this type of equation based on the Feynman-Kac formula, Bismut-Elworthy-Li formula, and Picard iterative multilevel decomposition, which has been proven for some applications in finance and physics very effective. For semilinear parabolic PDEs, the computational complexity of the algorithm is $O(d\epsilon^{-4})$, where d is the dimension of the equation and ϵ is the desired precision [8]. Baothman and Edhah exploited recent advances in probabilistic machine learning to infer governing equations represented by parametric linear operators [9]. This method modifies the Gaussian process prior according to the form of the equation and is used to infer the parameters of linear PDEs from observations. Liu et al. proposed the use of deep neural networks to approximate the solutions of high-dimensional partial differential equations; the network is iterated to satisfy differential operators, initial conditions, and boundary conditions; the neural network operates on a randomly sampled set of time and space points, iteratively; the solution is approximated by a neural network [10]. Parise et al. by using different neural network structures and different parameterizations improved machine learning algorithms for solving semilinear partial differential equations; the proposed algorithm is compared with several algorithms that utilize deep learning techniques to solve semilinear PDE problems [11]. Ly et al. reviewed methods for solving partial differential equations using physics-informed neural networks (PINNs), a novel residual-based adaptive optimization method proposed to improve the iterative efficiency of PINNs. This paper also provides a Python program library DeepXDE for the implementation of PINNs; the program library DeepXDE can

solve the forward problem of given initial value conditions and boundary value conditions and can also solve the inverse problem given additional measurement results [12].

“Numerical analysis” is the main content of computational mathematics, and its research scope covers almost all branches of mathematical science; in turn, many branches of mathematics need to explore numerical methods applicable to computers. In the theory of numerical calculation, the analytical solution of the general nonlinear differential equation does not exist and can only be solved numerically. The numerical solution problem and algorithm realization of nonlinear differential equations have very important applications in engineering dynamic control, computer simulation, mathematical modeling, etc. Figure 1 shows the nonlinear simulation technology. The author will study the algorithm, error, step size, and other problems of numerical solution of general nonlinear differential equation problems, the fourth-order fixed step size Runge-Kutta and its improved algorithm, and the numerical solution of the Lorenz model, which is the graphical interface for the results of MATLAB real variable operations.

3. Research Methods

3.1. Overview of Algorithms for Differential Equation Problems. As an important branch of mathematics, nonlinear differential equations and equation systems are traditional mathematical tools to describe or solve nonlinear dynamical system problems and have been widely used in many fields [13].

The numerical solutions of general differential equations are mostly numerical solutions to the initial value problems of differential equations; such problems can be described as the following formula (1) with a first-order explicit differential equation system:

$$\dot{x}(t) = f(t, x(t)). \quad (1)$$

Among them, $x^T(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ is called the state vector, and $f^T(\cdot) = [f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot)]$ can be any nonlinear function.

In the initial value problem, if the initial state $x_0 = [x_1(0), \dots, x_n(0)]^T$ is known, the numerical solution method is used to obtain the value of each variable state $x(t)$ in a certain interval $t \in [0, t_f]$; t_f is also called the terminal variable [14].

3.2. Numerical Algorithm of Multivariate Nonlinear Differential Equations and Problems of Error and Step Size

3.2.1. Numerical Algorithm for Initial Value Problem. The multivariate nonlinear extremum problem is a very common mathematical problem; in theory, the multivariate nonlinear extremum problem is often reduced to the root-finding problem of the multivariate nonlinear equation system, and then, the Newton iteration method is used to calculate it. But in fact, the root-finding problem of multivariate nonlinear equations is more difficult to solve than the extremum problem of multivariate nonlinear functions. In

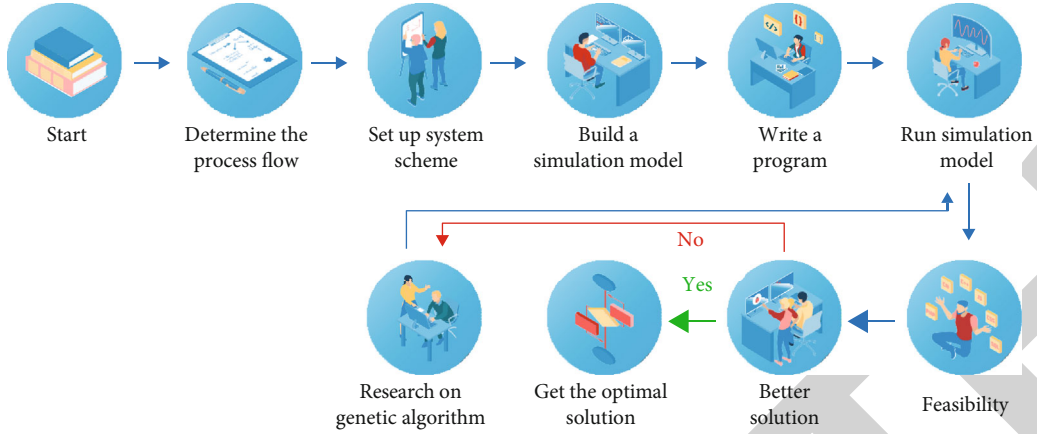


FIGURE 1: Nonlinear simulation technology.

computational mathematics, in order to solve the multivariate extreme value problem, many new methods have been introduced, such as the steepest descent method, genetic algorithm, and edge detection algorithm [15].

For the initial value problem of multivariate nonlinear differential equations, Euler's algorithm is a class that is easier to understand and control than other algorithms and is widely used in the numerical solution of complex differential equations; the following will give a numerical algorithm for the initial value problem of nonlinear differential equations based on MATLAB, taking Euler's algorithm as an example.

Assuming that the value of the system state vector at time t_0 is known to be $x(t_0)$, if a smaller calculation step h is selected, the derivative on the left side of the differential equation can be approximated as $x(t_0 + h) - x(t_0)/(t_0 + h) - t_0$, and the approximate solution of the equation at time $t_0 + h$ can be obtained by substituting it into the differential equation in the following:

$$\hat{x}(t_0 + h) = x(t_0) + hf(t_0, x(t_0)). \quad (2)$$

Due to the existence of the approximate solution error, the value of the system state vector at time $t_0 + h$ can be strictly expressed as the following:

$$x(t_0 + h) = x(t_0) + hf(t, x(t_0)) + R_0. \quad (3)$$

Or denoted as $x_1 = x(t_0 + h)$, then $\hat{x}_1 = \hat{x}(t_0 + h)$ is the approximate value of the system state vector at $t_0 + h$ time, that is, the numerical solution, where R_0 is the rounding error of the numerical solution.

More generally, assuming that the state vector of the system at time t_k is x_k , the numerical solution of Euler's algorithm at time $x_k + h$ can be written as

$$x_{k+1} = x_k + hf(t, x_k). \quad (4)$$

For the above general numerical solution expression, the iterative method can be used to gradually obtain the given initial value problem within a certain time period $t \in [0, T]$, the original numerical solution at each time instant $t_0 + kh$ ($k = 1, 2, \dots, n$) [16].

3.2.2. Accuracy Control of Numerical Solution and Analysis of Influencing Factors of Variable Step Size. For the effect of variable step size on accuracy and speed, an effective way to improve the accuracy of numerical solutions is to reduce the value of the step size h , but the continuous decrease of the step size value will cause the following two problems:

Reduce operation speed. For the chosen solution time limit, reducing the step size is equivalent to increasing the number of computational points in a fixed time interval, which directly leads to a sharp drop in computational speed [17].

Expand the cumulative error. Due to the inherent error of the numerical solution, even if a small step size is chosen, the obtained numerical solution will be accompanied by a rounding error; decreasing the step size increases the number of loops, which increases the number of passes for the rounding error stacking, resulting in a larger cumulative error.

A schematic diagram of the relationship between rounding error, accumulated error, and total error is shown in Figure 2.

For an effective way of error control, in multivariate nonlinear differential equations, step size and error are a pair of contradictions; according to the algorithm and accuracy requirements of the actual problem, the following ways can be used to choose:

Choose an appropriate step size. When using a simple algorithm like Euler, the step size should be moderate and follow the principle of rather small rather than large.

Improve the accuracy of the approximation algorithm. Since the Euler algorithm only converts the original integral problem into a trapezoidal method for the approximate solution, it cannot effectively approximate the original problem due to its low accuracy.

The composite Simpson method or the more accurate spline interpolation method can be used to replace the Euler algorithm, among which the Runge-Kutta method and the Adams method are the most common [18, 19].

In using the variable step size method, in the numerical solution of multivariate differential equations, many methods can be solved by variable step size; if the error is

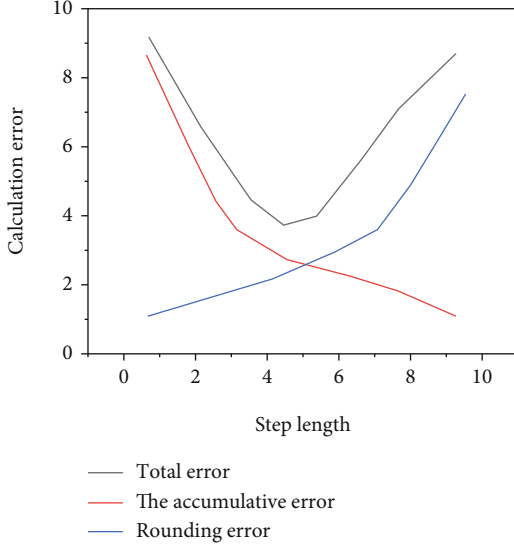


FIGURE 2: Error diagram.

small, the step size will be increased, and if the error is large, the step size will be automatically reduced, so as to solve the initial value problem of differential equations accurately and effectively.

The principle of the general variable step size algorithm is shown in Figure 3. Knowing the state variable x_k at time t_k , calculate the state variable \tilde{x}_{k+1} under the step size h and $h/2$, respectively; if the error $\varepsilon = \|\tilde{x}_{k+1} - \tilde{x}_{k+1}\|$ under the two step sizes is less than the given error limit, the step size can be used. If the error is large, gradually reduce the step size until the error is reduced to the allowable range [20]. This adaptive variable step size algorithm can solve two problems at the same time, namely, operation speed and calculation accuracy.

3.3. Algorithms and Functions for Numerical Solutions of Two Types of First-Order Differential Equations. The key to solving a system of first-order explicit differential equations is to write a function in MATLAB language and describe the system of differential equations to be solved.

In nonlinear differential equation solving, it is sometimes necessary to further set the algorithm and control conditions, which can be modified by the options variable in the solving process, and the initial variable can be obtained by the `odeset()` function.

For practical applications, it is often necessary to define some additional parameters, which can be represented by m_1, m_2, \dots, m_n . The following two types of important algorithms in numerical analysis will be analyzed, and a general program structure will be established.

3.3.1. Fourth-Order Fixed-Step Runge-Kutta Algorithm and Its Improvement. For a complex function, it is always prohibitive to use Taylor expansion to find its derivatives; Runge-Kutta uses the idea of the Taylor series method but avoids the analytical derivation of the original function process. Therefore, in essence, Runge-Kutta is based on the Taylor series method; it abandons the drawbacks of the Taylor series method for derivatives and uses the idea of compound

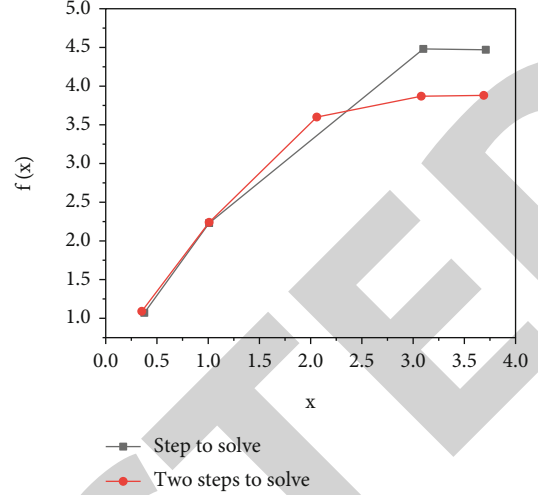


FIGURE 3: Schematic diagram of variable step size.

functions to achieve derivatives [21]. The Runge-Kutta method is a high-precision single-step algorithm widely used in engineering, including the well-known Euler method, which numerically solves differential equations. Due to the high precision of this algorithm, measures are taken to suppress the error. The numerical solution is obtained by the fourth-order Runge-Kutta method; its accuracy may not be as good as the improved Euler method. In the actual calculation, the appropriate algorithm should be selected according to the specific characteristics of the problem. For solutions with poor smoothness, it is better to use a low-order algorithm with a small step size.

The fourth-order fixed-step Runge-Kutta algorithm is an important algorithm in numerical analysis and system simulation theory. Its mathematical algorithm needs to define 4 additional vectors first, as shown in the following:

$$\begin{cases} k_1 = hf(t_k, x_k), \\ k_2 = hf\left(t_k + \frac{h}{2}, x_k + \frac{k_1}{2}\right), \\ k_3 = hf\left(t_k + \frac{h}{2}, x_k + \frac{k_2}{2}\right), \\ k_4 = hf(t_k + h, x_k + k_3). \end{cases} \quad (5)$$

Among them, Simpson is the calculation step, which is a constant in practical applications, and then, the state variable value of the next step is calculated by the Runge-Kutta algorithm as the following:

$$x_{k+1} = x_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (6)$$

For the above expression, the iterative method is used to obtain the numerical solution at each time point $t_0 + h, t_0 + 2h, \dots$ [22].

Based on mathematical algorithms, in MATLAB, the solution can be achieved through a set of loop statements.

In the above MATLAB algorithm, the variable span has two construction methods, namely, the equidistant time vector and the variable step vector based on t_0, t_f, h . After the function call ends, the matrix formed by the time vector and the state variable is returned by tout and yout in the program, respectively.

The abovementioned algorithm is characterized by simple structure and fast operation speed, but its limitation lies in its low precision. In order to ensure higher accuracy and numerical stability, we improve the above algorithm in terms of step size increment: that is, the $f_i(\bullet)$ function is evaluated 6 times in each calculation step, and this algorithm is called the fourth-order five-level RKF algorithm [23].

For the current step size h_k , six variables can be defined as follows:

$$k_i = h_k f \left(t_k + \alpha_i h_k, x_k + \sum_{j=1}^{i-1} \beta_{ij} k_j \right) \quad (i = 1, 2, \dots, 6). \quad (7)$$

In the formula, $x_{k+1} = x_k + \sum_{i=1}^6 \gamma_i k_i$ is the current calculation time, and the intermediate parameters α_i and β_{ij} can be given by the RKF algorithm coefficient table. At this time, the state vector can be expressed as the following:

$$x_{k+1} = x_k + \sum_{i=1}^6 \gamma_i k_i. \quad (8)$$

In the above algorithm, if an error vector $\varepsilon_k = \sum_{i=1}^6 (\gamma_i - \gamma_i^*) k_i$ is defined, the step size can be changed by the size of ε_k , thereby transforming into an adaptive variable step size algorithm, which makes the whole program easier to control [24].

3.3.2. Numerical Algorithm of Lorenz Model. The Lorenz model is a well-known equation describing chaotic phenomena, and the Lorenz system is worthy of detailed study from the point of view of both mathematics and physics [25].

Suppose the state equation of the Lorenz model is the following:

$$\begin{cases} \dot{x}_1(t) = -\beta x_1(t) + x_2(t)x_3(t), \\ \dot{x}_2(t) = -\rho x_2(t) + \rho x_3(t), \\ \dot{x}_3(t) = -x_1(t)x_2(t) + \sigma x_2(t) - x_3(t). \end{cases} \quad (9)$$

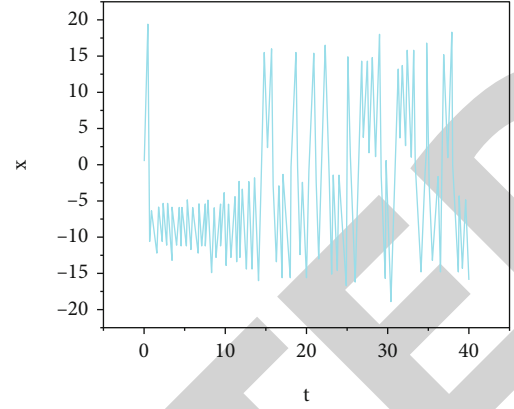
Among them, β, ρ, σ are fixed values, $x_1(0) = x_2(0) = 0$, $x_3(0) = \varepsilon$ is the initial value, and for $\varepsilon = 10^{-10}$, the numerical solution of the system of equations is obtained.

Since this equation is a nonlinear differential equation and there is no analytical solution, we can complete the programming of the numerical algorithm based on MATLAB according to the following steps.

First, use the Lorenz.m function to describe the dynamic model of the system.

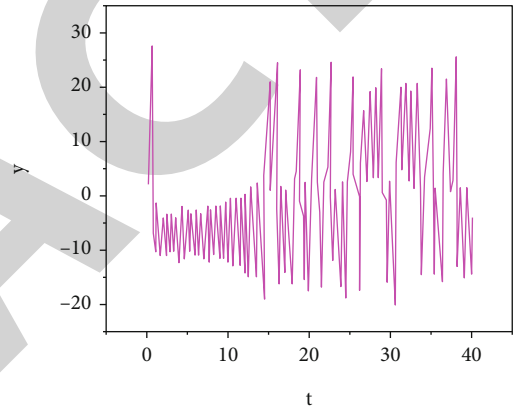
Then, call the numerical solution function ode45() to numerically solve the system described by the Lorenz.m function, and display the result graphically.

Among them, t_final is the set simulation termination time, x0 is the initial state, and the two commands draw



Phase space curve

FIGURE 4: Phase space plot of the x state.



Phase space curve

FIGURE 5: Phase space plot of the y state.

the two-dimensional curve graph of the time relationship of each state of the system (as shown in Figures 4–6) and the phase of the three state space graphs.

In the above program, the comet3() function can also be used to draw an animated trajectory for observing the phase space trajectory; just rewrite the last statement appropriately. In order to establish a description function based on the MATLAB differential equation system, ode45() must be called, so writing the MATLAB function to describe the differential equation is the key to the numerical solution of the initial value problem, especially for the first-order nonlinear differential equation.

4. Result Analysis

In the process of numerical solution, if the simulation algorithm and control parameters are not selected properly, unreliable results or even wrong results may be obtained; therefore, the verification of the numerical solution can be carried out in two ways. A feasible method is to modify the simulation control parameters, such as the acceptable error limit, such as setting the RelTol option to a smaller value, and observe the obtained results to see if they are consistent

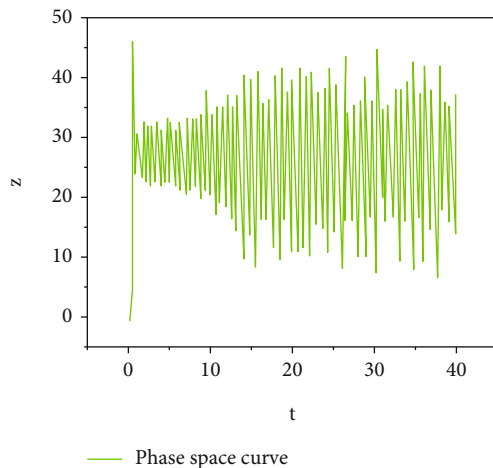


FIGURE 6: Phase space plot of z state.

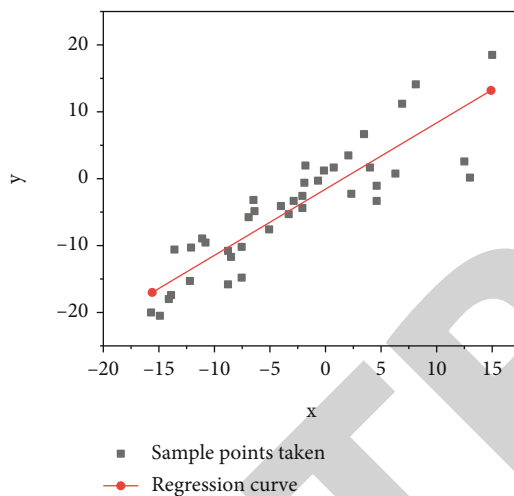


FIGURE 7: Lorenz equation x, y data image regression curve.

with the results of the first run; if there are no unacceptable differences, further reductions in the error limits should be considered. In addition, choosing a differential equation solving algorithm for the same problem can also check the correctness of the operation result.

The x -direction displacement and y -direction displacement data are extracted from the Lorenz equation as an iterative sample of the model, and the iterative results are shown in Figure 7. The dots in Figure 7 represent the extracted sample points, and the straight line represents the regression curve obtained after the iteration of the sample data, with a slope of 0.996; the iterative regression model reflects the basic characteristics of the data well.

5. Conclusion

The solution and simulation of scientific or engineering problems often boil down to the solution of mathematical models and the fitting of experimental data, while nonlinear differential equations, as an important mathematical terminal model, can only be solved numerically. Through the systematic analysis of the fourth-order fixed-step Runge-Kutta

algorithm and the Lorenz model algorithm, the author established a variable-step adaptive algorithm based on nonlinear differential equations and the MATLAB program structure under the explicit function description; the key elements such as step size, error, and numerical solution verification are analyzed; and the influencing factors are analyzed, and relevant conclusions are drawn. It provides an effective numerical solution method for solving engineering problems with differential equations as mathematical models, and further discussion on the numerical solutions of partial differential equations is also of great significance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that they have no conflicts of interest.

References

- [1] H. Jamshidi and A. A. Jafari, "Predicting unbalance asymmetric rotor vibration behavior based on sensitivity analysis and using response surface methodology method considering parallel misalignment," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 24, pp. 7430–7444, 2021.
- [2] Q. Liu, S. Zhong, and L. Li, "Investigation of riblet geometry and start locations of herringbone riblets on pressure losses in a linear cascade at low Reynolds numbers," *Journal of Turbomachinery*, vol. 142, no. 10, pp. 1–20, 2020.
- [3] Y. H. Ni, X. Li, and J. F. Zhang, "Indefinite mean-field stochastic linear-quadratic optimal control: from finite horizon to infinite horizon," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3269–3284, 2016.
- [4] X. Tu, C. Mu, and P. Zheng, "On effects of the nonlinear signal production to the boundedness and finite-time blow-up in a flux-limited chemotaxis model," *Mathematical Models and Methods in Applied Sciences*, vol. 32, no. 4, pp. 647–711, 2022.
- [5] A. S. Leonov, "Source recovery with a posteriori error estimates in linear partial differential equations," *Journal of Inverse and Ill-Posed Problems*, vol. 28, no. 5, pp. 677–692, 2020.
- [6] B. Zhu, "On the well-posedness of a parametric spectral estimation problem and its numerical solution," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1089–1099, 2020.
- [7] R. Q. Nafil, H. T. Khamees, and M. S. Majeed, "Identification the internal parameters for mono-crystalline solar module using matlab -simulation and experimental ascertainment," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 716–723, 2021.
- [8] M. M. Abdulwahid and N. Basil, "Implementation of micro-strip patch antenna using matlab," *Informatica*, vol. 2, no. 1, pp. 29–35, 2021.
- [9] F. A. Baothman and B. S. Edhah, "Toward agent-based lsb image steganography system," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 903–919, 2021.

- [10] R. Liu, J. Liu, J. Wang, Z. Liu, and R. Guo, "Seismic simultaneous inversion using a multidamped subspace method," *Geophysics*, vol. 85, no. 1, pp. R1–R10, 2020.
- [11] F. Parise, B. Gentile, and J. Lygeros, "A distributed algorithm for almost-Nash equilibria of average aggregative games with coupling constraints," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 770–782, 2020.
- [12] H. B. Ly, H. Vu, L. S. Ho, and B. T. Pham, "Dimensionality reduction and prediction of soil consolidation coefficient using random forest coupling with relief algorithm," *Frontiers of Structural and Civil Engineering*, vol. 16, no. 2, pp. 224–238, 2022.
- [13] H. Dida, F. Charif, and A. Benchabane, "Registration of computed tomography images of a lung infected with covid-19 based in the new meta-heuristic algorithm hpsgwo," *Multimedia Tools and Applications*, vol. 81, no. 13, pp. 18955–18976, 2022.
- [14] O. A. Vanli and C. N. Taylor, "Robust error estimation based on factor-graph models for non-line-of-sight localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, Article ID 3, 26 pages, 2022.
- [15] F. S. Rizi and A. Falahati, "Ergodic capacity and symbol error rate analysis of a wireless system with α - μ composite fading channel," *Wireless Personal Communications*, vol. 124, no. 1, pp. 103–118, 2022.
- [16] F. Zhi, Y. Zhang, X. Zhou, J. Zhang, and W. Lu, "New model-based method to improve the moving performance of the planar near-field scanning frame," *Journal of Beijing Institute of Technology*, vol. 30, pp. 92–102, 2021.
- [17] M. Bahrami and N. Faraji, "Minimum mean square error estimator for speech enhancement in additive noise assuming Weibull speech priors and speech presence uncertainty," *International Journal of Speech Technology*, vol. 24, no. 1, pp. 97–108, 2021.
- [18] Z. M. Alaofi, T. S. El-Danaf, and S. S. Dragomir, "Comparing solutions to the nonlinear dissipative wave equation," *Journal of Applied Mathematics and Physics*, vol. 10, no. 4, pp. 1281–1296, 2022.
- [19] Z. Liao and S. Li, "Solving nonlinear equations systems with an enhanced reinforcement learning based differential evolution," *Complex System Modeling and Simulation*, vol. 2, no. 1, pp. 78–95, 2022.
- [20] K. R. Aida-Zade, "Numerical solution of linear differential equations with nonlocal nonlinear conditions," *Computational Mathematics and Mathematical Physics*, vol. 60, no. 5, pp. 808–816, 2020.
- [21] A. Sharma, R. Kumar, M. Talib, S. Srivastava, and R. Iqbal, "Network modelling and computation of quickest path for service-level agreements using bi-objective optimization," *International Journal of Distributed Sensor Networks*, vol. 15, no. 10, Article ID 155014771988111, 2019.
- [22] M. Bradha, N. Balakrishnan, A. Suvitha et al., "Experimental, computational analysis of butein and lanceoletin for natural dye-sensitized solar cells and stabilizing efficiency by IoT," *Environment, Development and Sustainability*, vol. 24, no. 6, pp. 8807–8822, 2022.
- [23] X. Zhao, X. Liu, J. Liu, J. Chen, F. Shihong, and F. Zhong, "The effect of ionization energy and hydrogen weight fraction on the non-thermal plasma volatile organic compounds removal efficiency," *Journal of Physics D: Applied Physics*, vol. 52, no. 14, article 145201, 2019.
- [24] R. Huang, S. Zhang, W. Zhang, and X. Yang, "Progress of zinc oxide-based nanocomposites in the textile industry," *IET Collaborative Intelligent Manufacturing*, vol. 3, no. 3, pp. 281–289, 2021.
- [25] Z. Guo and Z. Xiao, "Research on online calibration of lidar and camera for intelligent connected vehicles based on depth-edge matching," *Nonlinear Engineering*, vol. 10, no. 1, pp. 469–476, 2021.