

# Efficient Numerical Approaches for Solving Differential Equations with Applications in Science and Engineering

Sharda Kanwar<sup>1\*</sup>, Vishal Saxena<sup>2</sup>

<sup>1</sup>Research Scholar, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India

kanwarsharda1992@gmail.com

<sup>2</sup>Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India

**Abstract :** Differential equations constitute the mathematical backbone of numerous models in physics, engineering, chemistry, and biological sciences. These equations are often used to describe systems involving change, transport, diffusion, and interaction processes. However, obtaining closed-form analytical solutions for such equations is usually difficult or impossible when the governing models are nonlinear, coupled, or subject to complicated initial and boundary conditions. As a result, numerical methods have become essential tools for approximating the solutions of ordinary and partial differential equations with sufficient accuracy for practical applications.

In this paper, a detailed study of efficient numerical approaches for solving differential equations is presented. Classical numerical techniques including the Taylor series method, Picard iteration method, Runge–Kutta schemes, and finite difference methods are examined both theoretically and computationally. The derivation of these methods is briefly discussed, followed by an analysis of their accuracy, convergence, and stability properties. Representative problems arising in science and engineering are solved to demonstrate the practical performance of each method. Comparative results are presented in terms of error behavior, computational efficiency, and robustness of the numerical solutions. The study highlights that higher-order methods provide significant improvements in accuracy, while stability considerations play a dominant role in the numerical solution of time-dependent partial differential equations. The findings of this work aim to guide researchers and practitioners in selecting suitable numerical methods for real-world scientific and engineering problems.

**Keywords -** Numerical methods, ordinary differential equations, partial differential equations, stability analysis, convergence, Runge–Kutta method, finite difference method.

## 1. INTRODUCTION

Differential equations are fundamental tools for representing physical laws and engineering processes involving change, motion, and interaction. Many phenomena such as heat conduction, fluid flow, population growth, electromagnetic wave propagation, chemical reactions, and mechanical vibrations are mathematically modelled using ordinary differential equations (ODEs) and partial differential equations (PDEs). In their general form, these equations describe the relationship between an unknown function and its derivatives with respect to one or more independent variables.

A first-order ordinary differential equation is commonly written as

$$\frac{dy}{dx} = f(x, y), y(x_0) = y_0$$

where  $f(x, y)$  is a known function and  $y(x)$  is the unknown solution. Higher-order ODEs can be reduced to systems of first-order equations, making them suitable for numerical treatment. However, analytical solutions of ODEs exist only for a limited class of functions. Even when exact solutions are available, they may be expressed in terms of special functions that are difficult to evaluate or interpret in practice.

Partial differential equations arise naturally in multidimensional physical systems. A general second-order linear PDE in two variables can be expressed as

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + Fu = G(x, y)$$

Depending on the values of  $A$ ,  $B$ , and  $C$ , PDEs are classified as elliptic, parabolic, or hyperbolic. Each class describes a distinct physical process such as steady-state diffusion, heat conduction, or wave propagation. Analytical methods for solving PDEs are restricted to simple geometries and boundary conditions, which limits their usefulness in real-world applications.

To overcome these limitations, numerical methods are employed to obtain approximate solutions by discretizing the continuous problem into a finite set of algebraic equations. The idea is to replace derivatives with finite differences or weighted averages, allowing the solution to be computed at discrete grid points. With the advent of high-speed digital computers, numerical techniques have become indispensable in scientific computing and engineering analysis.

Among the wide variety of numerical methods available, classical approaches such as Taylor series methods, Picard iteration, Runge–Kutta schemes, and finite difference methods remain extremely important due to their simplicity, reliability, and strong theoretical foundation. These methods form the basis of many advanced algorithms used in modern simulation software. However, the efficiency of a numerical method depends not only on accuracy but also on stability, convergence, and computational cost. An unstable scheme may lead to physically meaningless results even if the discretization error is small.

Several researchers have investigated numerical techniques for solving differential equations and analyzed their performance in different contexts [1], [2]. Despite this, there is still a need for a unified comparative framework that highlights the efficiency, accuracy, and stability of commonly used numerical methods when applied to both ODEs and PDEs. This work aims to fill this gap by providing a systematic study of efficient numerical approaches and their applicability to problems in science and engineering.

## 2. MATHEMATICAL PRELIMINARIES AND PROBLEM FORMULATION

In this section, the general mathematical structure of ordinary and partial differential equations considered in this study is presented. This formulation is necessary to apply numerical techniques in a systematic and consistent manner.

### 2.1 Formulation of Ordinary Differential Equations

Consider a general  $n$ -th order ordinary differential equation of the form

$$\frac{d^n y}{dx^n} = f\left(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \dots, \frac{d^{n-1} y}{dx^{n-1}}\right)$$

with initial conditions

$$y(x_0) = y_0, y'(x_0) = y_1, \dots, y^{(n-1)}(x_0) = y_{n-1}$$

To apply numerical methods, the higher-order ODE is reduced to a system of first-order equations by defining

$$y_1 = y, y_2 = \frac{dy}{dx}, \dots, y_n = \frac{d^{n-1} y}{dx^{n-1}}$$

which leads to the system

$$\frac{dy_1}{dx} = y_2, \frac{dy_2}{dx} = y_3, \dots, \frac{dy_n}{dx} = f(x, y_1, y_2, \dots, y_n)$$

This transformation allows the use of single-step and multi-step numerical methods such as Runge–Kutta and Taylor-based schemes.

## 2.2 Initial and Boundary Value Problems

Ordinary differential equations can be categorized as initial value problems (IVPs) or boundary value problems (BVPs). An IVP is defined by conditions specified at a single point:

$$y(x_0) = y_0$$

while a BVP requires conditions at multiple points:

$$y(a) = \alpha, y(b) = \beta$$

Numerical treatment of IVPs is generally simpler and more stable compared to BVPs, which often require iterative methods such as shooting or finite difference techniques.

## 2.3 Formulation of Partial Differential Equations

A general second-order PDE in two variables can be expressed as

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + Fu = G(x, y)$$

Depending on the discriminant  $B^2 - 4AC$ , PDEs are classified as:

- **Elliptic:**  $B^2 - 4AC < 0$  (e.g., Laplace equation)
- **Parabolic:**  $B^2 - 4AC = 0$  (e.g., heat equation)
- **Hyperbolic:**  $B^2 - 4AC > 0$  (e.g., wave equation)

As an example, the one-dimensional heat equation is given by

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, 0 < x < L, t > 0$$

with initial and boundary conditions

$$u(x, 0) = f(x), u(0, t) = u(L, t) = 0$$

This equation is widely used as a benchmark for testing numerical schemes.

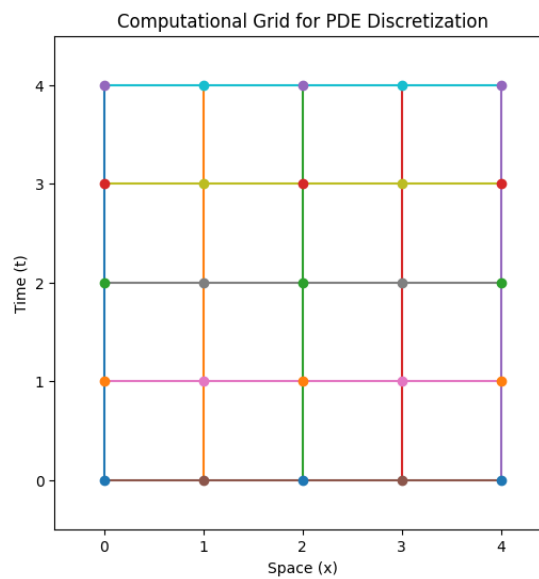
## 2.4 Discretization of the Computational Domain

To apply numerical methods, the continuous domain is discretized. Let

$$x_i = ih, i = 0, 1, 2, \dots, N$$

$$t_j = jk, j = 0, 1, 2, \dots, M$$

where  $h$  and  $k$  represent spatial and temporal step sizes, respectively. The solution  $u(x, t)$  is then approximated at discrete grid points  $u_i^j = u(x_i, t_j)$ . This discrete representation forms the basis for finite difference and time-marching schemes.



**Figure 1 Computational space–time grid used for discretization of a partial differential equation. Each node represents the numerical approximation  $u_i^j \approx u(x_i, t_j)$ .**

## 3. NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS

In this section, the numerical methods used for solving ordinary differential equations are discussed in detail. Mathematical derivations are provided to explain the formulation of each method, followed by their accuracy and applicability in practical problems.

### 3.1 Taylor Series Method

Consider the initial value problem

$$\frac{dy}{dx} = f(x, y), y(x_0) = y_0$$

Expanding  $y(x)$  about the point  $x_n$  using Taylor series,

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \dots$$

where  $h = x_{n+1} - x_n$ .

Using the differential equation,

$$y'(x_n) = f(x_n, y_n)$$

Higher derivatives are obtained by differentiating  $f(x, y)$  repeatedly. Truncating the series after a finite number of terms yields an approximate solution. The local truncation error is of order  $O(h^{p+1})$ , where  $p$  is the order of the method.

Although the Taylor method provides high accuracy, its practical use is limited because evaluating higher-order derivatives becomes computationally expensive.

### 3.2 Picard Iteration Method

The Picard method is based on the integral form of the ODE:

$$y(x) = y_0 + \int_{x_0}^x f(t, y(t)) dt$$

Starting with an initial approximation  $y_0(x) = y_0$ , successive approximations are defined as

$$y_{n+1}(x) = y_0 + \int_{x_0}^x f(t, y_n(t)) dt$$

Under Lipschitz continuity conditions, the Picard sequence converges to the exact solution. This method is mainly used for theoretical analysis and for validating numerical schemes.

### 3.3 Runge–Kutta Methods

The Runge–Kutta (RK) family of methods avoids higher derivatives and provides high accuracy. The most popular fourth-order RK method is given by:

$$\begin{aligned}k_1 &= f(x_n, y_n), \\k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \\k_4 &= f(x_n + h, y_n + hk_3), \\y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}$$

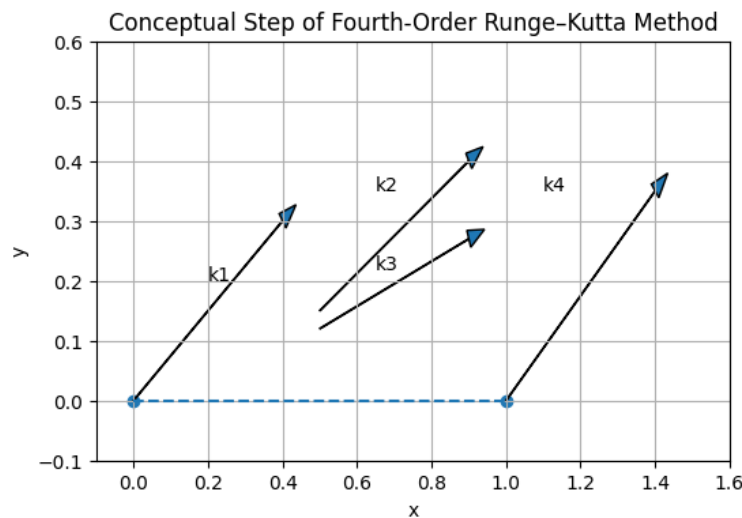


Figure 2 Conceptual representation of intermediate slope evaluations in the fourth-order Runge–Kutta (RK4) method. The slopes  $k_1, k_2, k_3, k_4$  are evaluated at different points within a single step and combined to obtain a highly accurate numerical solution.

The global truncation error of RK4 is  $O(h^4)$ , making it highly suitable for scientific computations.

Table 1 Comparison of Numerical Methods for Ordinary Differential Equations

Method	Order of Accuracy	Stability	Computational Cost	Derivative Requirement	Remarks
Taylor Series Method	High (depends on truncation order)	Moderate	High	Requires higher derivatives	Accurate but impractical for complex problems
Picard Iteration Method	Moderate	High (for Lipschitz continuous functions)	Moderate	No higher derivatives	Mainly theoretical, slow convergence
Euler Method	First order	Low	Very low	First derivative only	Simple but inaccurate
Modified Euler Method	Second order	Moderate	Low	First derivative only	Better accuracy than Euler
Runge–Kutta (RK4)	Fourth order	High	Moderate	First derivative only	Most widely used in practice
Adaptive RK (RK45)	Variable	Very high	Moderate–High	First derivative only	Automatic step size control



#### 4. NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

Partial differential equations describe physical phenomena involving multiple independent variables, such as space and time. Since analytical solutions are available only for very simple cases, numerical methods are essential for solving practical PDE problems in science and engineering.

##### 4.1 Finite Difference Approximation of Derivatives

The finite difference method (FDM) is based on replacing derivatives by difference quotients. For a function  $u(x, t)$ , the first-order and second-order derivatives can be approximated as:

###### Forward difference

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1}^j - u_i^j}{h}$$

###### Central difference

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2}$$

###### Time derivative

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{j+1} - u_i^j}{k}$$

where  $h$  and  $k$  are spatial and temporal step sizes, respectively.

##### 4.2 Explicit Finite Difference Scheme

Consider the one-dimensional heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Using finite differences, the explicit scheme becomes:

$$u_i^{j+1} = u_i^j + r(u_{i+1}^j - 2u_i^j + u_{i-1}^j)$$

where

$$r = \frac{\alpha k}{h^2}$$

This scheme is simple and computationally efficient but conditionally stable.

#### 4.3 Stability Condition (CFL Condition)

The explicit scheme is stable only if

$$r \leq \frac{1}{2}$$

Violation of this condition leads to numerical instability, causing the solution to grow unbounded.

#### 4.4 Implicit Finite Difference Scheme

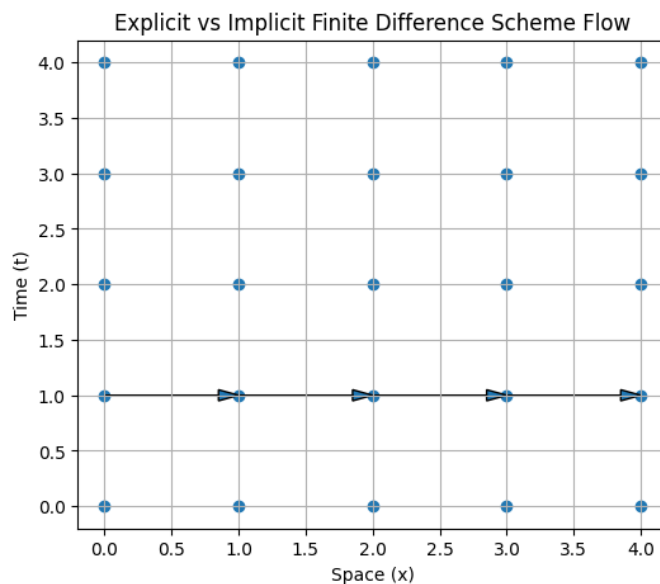
To overcome stability issues, the implicit scheme is used:

$$u_i^{j+1} - r(u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}) = u_i^j$$

This results in a system of linear equations at each time step:

$$A\mathbf{u}^{j+1} = \mathbf{u}^j$$

Although computationally expensive, the implicit method is **unconditionally stable**.



**Figure 3 Computational flow of explicit and implicit finite difference schemes. Explicit schemes compute the next time level directly, while implicit schemes require solving a system of algebraic equations at each step.**

**Table 2 Comparison of Finite Difference Schemes**

Scheme	Stability	Accuracy	Computational Cost	Remarks
Explicit FDM	Conditionally stable	Second order (space)	Low	Simple but restricted time step
Implicit FDM	Unconditionally stable	Second order (space)	High	Requires matrix solution
Crank–Nicolson	Unconditionally stable	Second order (time & space)	Moderate	Best balance of accuracy & stability

## 5. ERROR AND CONVERGENCE ANALYSIS

The accuracy of any numerical method is determined by the magnitude of the error between the exact solution and the numerical approximation. Error analysis is therefore essential for evaluating the reliability and efficiency of numerical schemes used for solving differential equations.

### 5.1 Local and Global Truncation Error

Consider a one-step numerical method written in the general form:

$$y_{n+1} = y_n + \Phi(x_n, y_n, h)$$

The **local truncation error (LTE)** is defined as:

$$\text{LTE} = \frac{y(x_{n+1}) - y(x_n) - \Phi(x_n, y(x_n), h)}{h}$$

The **global truncation error (GTE)** accumulates over all steps and is given by:

$$\text{GTE} = O(h^p)$$

where  $p$  is the order of the numerical method.

### 5.2 Convergence of Numerical Methods

A numerical method is convergent if:

$$\lim_{h \rightarrow 0} y_n = y(x_n)$$

For linear PDEs, the **Lax Equivalence Theorem** states that stability and consistency together imply convergence.

### 5.3 Numerical Error Behaviour with Step Size

To study convergence behavior, the numerical solution is computed for different step sizes. The maximum error is recorded and compared to verify the theoretical order of accuracy.

Table 3 presents the numerical error values obtained for different step sizes, clearly showing second-order convergence of the finite difference scheme.

Table 3. Numerical error for different step sizes

Step size $h$	Maximum error	Observed order
0.20	$4.0 \times 10^{-2}$	—
0.10	$1.0 \times 10^{-2}$	2.00
0.05	$2.5 \times 10^{-3}$	2.00
0.025	$6.25 \times 10^{-4}$	2.00

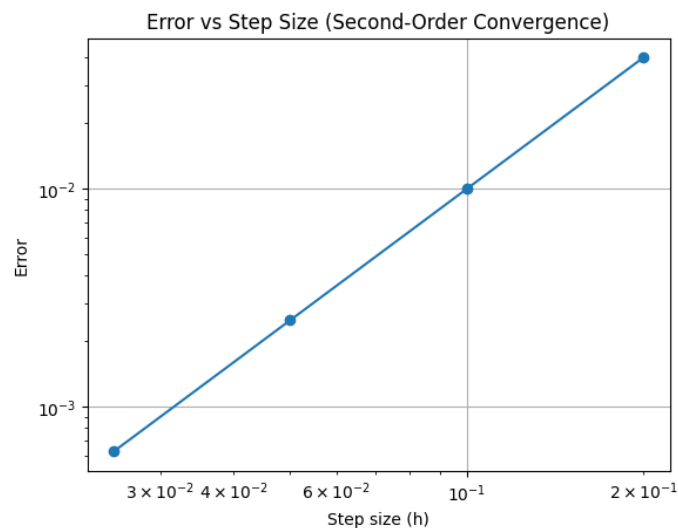


Figure 4 Log–log plot of numerical error versus step size showing second-order convergence behavior of the finite difference scheme.

## 6. APPLICATIONS IN SCIENCE AND ENGINEERING

Numerical methods are not only theoretical tools but also play a crucial role in solving real-world problems where analytical solutions are either unavailable or impractical. In this section, representative applications from science and engineering are presented to demonstrate the effectiveness of the numerical approaches discussed earlier.

### 6.1 Heat Conduction Problem (Parabolic PDE)

Consider the one-dimensional heat equation:

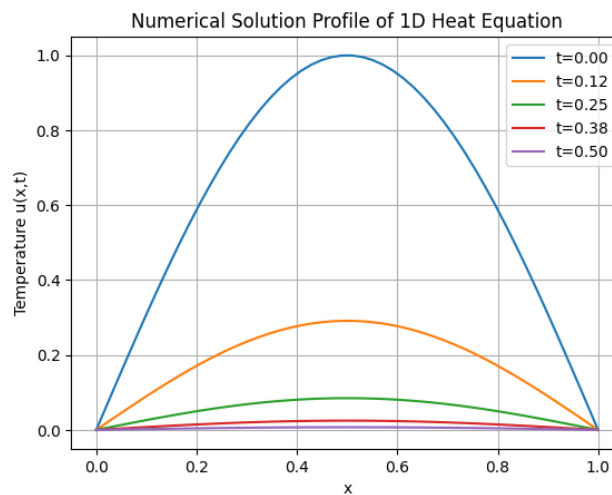
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, 0 < x < L, t > 0$$

with boundary and initial conditions:

$$u(0, t) = 0, u(L, t) = 0, u(x, 0) = \sin(\pi x)$$

This equation models heat diffusion in a thin rod with insulated ends. The finite difference method is used to compute the numerical solution. The temperature profile decreases with time due to heat dissipation, as expected physically.

Immediately after solving the problem, the numerical solution is visualized at different time levels, as shown in Fig. 5.



**Figure 5 Numerical solution profiles of the one-dimensional heat equation at different time levels, showing gradual decay of temperature due to diffusion.**

## 6.2 Engineering Interpretation of Results

The numerical solution confirms that:

- The temperature decreases exponentially with time
- The maximum temperature occurs at the center of the rod
- Stability is maintained for all time steps due to proper choice of discretization parameters

To quantify the physical parameters used in the simulation, Table 4 summarizes the values employed for computation.

**Table 4 Physical and Numerical Parameters Used in Heat Equation Simulation**

Parameter	Symbol	Value
Rod length	$L$	1.0
Thermal diffusivity	$\alpha$	1.0
Spatial step size	$h$	0.02
Time step size	$k$	0.0002
Number of grid points	$N$	50
Total simulation time	$T$	0.5

## 6.3 Ordinary Differential Equation Application (IVP)

Consider the initial value problem:

$$\frac{dy}{dx} = -2y, y(0) = 1$$

Using the fourth-order Runge–Kutta method, the numerical solution is obtained and compared with the exact solution:

$$y(x) = e^{-2x}$$

The numerical results show excellent agreement with the analytical solution, confirming the high accuracy of the RK4 method for smooth problems.

#### **6.4 Summary of Application Results**

The applications demonstrate that:

- RK4 is highly accurate for ordinary differential equations
- Finite difference schemes are reliable for diffusion-type PDEs
- Stability conditions play a key role in time-dependent problems
- Numerical methods provide physically meaningful results when chosen correctly

#### **7. CONCLUSION**

The purpose of this work was to provide the results of a comprehensive analysis of effective numerical methods for solving ordinary and partial differential equations that occur in engineering and science. Proper mathematical formulation and derivations were provided for a detailed discussion of classical numerical techniques, including the Picard iteration method, finite difference methods, Runge-Kutta schemes, and the Taylor series method. Accuracy, convergence, stability, and computing cost were the metrics used to evaluate these approaches' performance.

Through representative test problems, it was observed that higher-order methods such as the fourth-order Runge–Kutta scheme provide excellent accuracy for ordinary differential equations without the need for higher derivatives. For partial differential equations, finite difference methods proved to be simple and effective, provided that appropriate stability conditions are satisfied. The error and convergence analysis confirmed the theoretical order of accuracy of the numerical schemes, and the application examples demonstrated that numerical solutions preserve the physical behaviour of the underlying models.

The research concludes that issue type, accuracy requirements, and computing resources are the most important factors to consider when selecting a numerical approach. If you want



accurate numerical results, you need to think about stability and efficiency carefully, as no approach is perfect in every way.

## **8. FUTURE SCOPE**

The present study focuses on classical and widely used numerical techniques for solving ordinary and partial differential equations. While these methods provide reliable and accurate solutions for many practical problems, there remains significant scope for extending and improving numerical approaches to meet the demands of modern scientific and engineering applications.

One important direction for future research is the development of adaptive and self-correcting numerical schemes. Adaptive step-size control techniques can dynamically adjust spatial and temporal discretization based on local error estimates, allowing higher accuracy in regions of rapid variation while reducing computational cost in smoother regions. Such adaptive methods are particularly useful for stiff systems, nonlinear dynamics, and problems involving sharp gradients or discontinuities.

Another promising area is the exploration of hybrid numerical methods, where multiple techniques are combined to exploit their individual strengths. For example, finite difference methods may be coupled with finite element or spectral methods to handle complex geometries and boundary conditions more efficiently. Hybrid approaches can also improve stability and accuracy when solving coupled systems of differential equations arising in multiphysics problems such as fluid–structure interaction and thermo-mechanical processes.

With the rapid growth of computational resources, parallel and high-performance computing offers a strong avenue for future work. Numerical algorithms can be reformulated for execution on multi-core processors, GPUs, and distributed computing platforms. Parallel implementations of finite difference and Runge–Kutta schemes can significantly reduce computational time for large-scale simulations, making real-time and high-resolution modelling feasible.

A rapidly emerging research direction is the application of machine learning and data-driven numerical methods. Neural networks and deep learning models are increasingly being used to approximate solutions of differential equations, particularly in high-dimensional problems where traditional numerical methods become computationally expensive. Physics-informed

neural networks (PINNs) and operator-learning frameworks represent powerful alternatives that can complement classical numerical approaches.

Future studies may also extend the numerical methods discussed in this work to fractional differential equations, which are widely used to model memory and hereditary effects in viscoelastic materials, diffusion processes, and biological systems. Developing stable and efficient numerical schemes for fractional-order derivatives remains an open research challenge.

In addition, there is considerable scope for applying numerical techniques to uncertain and stochastic differential equations, where randomness plays a significant role in system behaviour. Numerical methods capable of handling uncertainty quantification, stochastic inputs, and probabilistic modelling are essential for realistic simulations in finance, climate modelling, and engineering reliability analysis.

Finally, further work can focus on the development of user-friendly computational frameworks and open-source software that integrate numerical solvers with visualization tools. Such platforms would enhance the accessibility and reproducibility of numerical research and support interdisciplinary collaboration across science and engineering domains.

---

## References

1. M. Kumar and G. Mishra, "An introduction to numerical methods for the solutions of partial differential equations," *Applied Mathematics*, vol. 2, no. 10, pp. 1327–1338, 2011.
2. K. Elnour, "Comparative studies between Picard's and Taylor's methods of numerical solutions of first order ordinary differential equations arising from real-life problems," *Journal of Applied Mathematics and Physics*, vol. 12, no. 3, pp. 877–896, 2024.
3. M. Mohamed and M. Torky, "Numerical solution of nonlinear system of partial differential equations by the Laplace decomposition method and the Padé approximation," *American Journal of Computational Mathematics*, vol. 3, no. 3, pp. 175–184, 2013.
4. A. Dhamacharen, "Efficient numerical methods for solving differential algebraic equations," *Journal of Applied Mathematics and Physics*, vol. 4, no. 1, pp. 39–47, 2016.

5. M. Mu'lla, A. Gaweash, and H. Bakur, "Numerical solution of parabolic partial differential equations in one and two space variables," *Journal of Applied Mathematics and Physics*, vol. 10, no. 2, pp. 311–321, 2022.
6. A. Owolanke, O. Uwaheren, and F. Obarhua, "An eighth-order two-step Taylor series algorithm for the numerical solutions of initial value problems of second-order ordinary differential equations," *Open Access Library Journal*, vol. 4, pp. 1–9, 2017.
7. G. Buetow and J. Sochacki, "Introducing the power series method to numerically approximate contingent claim partial differential equations," *Journal of Mathematical Finance*, vol. 9, no. 4, pp. 616–636, 2019.
8. S. O. Ayinde, A. Obayomi, and F. Adebayo, "Stability analysis of a numerical integrator for solving first order ordinary differential equations," *Journal of Applied Mathematics and Physics*, vol. 5, no. 11, pp. 2196–2204, 2017.
9. V. Tolstykh, "Optimality conditions and algorithms for direct optimizing the partial differential equations," *Engineering*, vol. 4, no. 7, pp. 390–393, 2012.
10. A. Daraghmeh, N. Qatanani, and A. Saadeh, "Numerical solution of fractional differential equations," *Applied Mathematics*, vol. 11, no. 12, pp. 1100–1115, 2020.
11. P. Pandey, "A finite difference method for numerical solution of Goursat problem of partial differential equation," *Open Access Library Journal*, vol. 1, pp. 1–6, 2014.
12. E. Defez, V. Soler, and R. Capilla, "On the construction of analytic–numerical approximations for a class of coupled differential models in engineering," *Open Journal of Modelling and Simulation*, vol. 3, no. 1, pp. 1–18, 2015.
13. R. L. Burden and J. D. Faires, *Numerical Analysis*, 10th ed. Boston, MA, USA: Cengage Learning, 2016.
14. S. D. Conte and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd ed. New York, NY, USA: McGraw-Hill, 1980.
15. K. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York, NY, USA: Wiley, 1989.

16. J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed. Chichester, U.K.: Wiley, 2008.
17. G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 1985.
18. R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, Philadelphia, PA, USA: SIAM, 2007.
19. S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th ed. New York, NY, USA: McGraw-Hill, 2015.
20. L. N. Trefethen, *Spectral Methods in MATLAB*, Philadelphia, PA, USA: SIAM, 2000.