Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

COMPUTATIONAL ANALYSIS AND IMPLEMENTATION OF NUMERICAL METHODS FOR ORDINARY AND PARTIAL DIFFERENTIAL EQUATIONS

Sharda kanwar
Research Scholar
Jayoti Vidyapeeth Woman's University, Jaipur
Vishal Saxena
Jayoti Vidyapeeth women's University, jaipur

Abstract

In many areas of science and engineering, mathematical models are expressed in the form of ordinary and partial differential equations. Although analytical solutions provide exact mathematical descriptions, they are rarely available for realistic physical systems because of nonlinear behaviour, complex geometries, or difficult boundary conditions. For this reason, computational numerical methods have become essential tools for obtaining practical and reliable solutions. This research focuses on the computational implementation and analysis of widely used classical numerical techniques for solving differential equations. The fourth-order Runge–Kutta method is applied to initial value problems involving ordinary differential equations, while finite difference approaches—namely explicit, implicit, and Crank–Nicolson schemes—are used for time-dependent partial differential equations such as the heat diffusion model. Numerical simulations are performed to evaluate accuracy, convergence behaviour, stability limits, and computational efficiency. Wherever possible, numerical results are compared with known analytical solutions to validate correctness. Graphical visualizations are also presented to illustrate how numerical solutions evolve with step size and time progression.

The computational findings suggest that higher-order Runge-Kutta methods achieve strong accuracy for smooth ordinary differential equations with moderate computational effort, whereas implicit and Crank–Nicolson finite difference schemes provide superior stability for diffusion-type partial differential equations. Overall, the study demonstrates that careful selection of numerical techniques based on stability, efficiency, and physical interpretation is essential for meaningful computational

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

modelling. The proposed computational framework can support practical problem-solving in engineering, physics, and applied scientific research.

Keywords - Numerical computation, Runge–Kutta method, finite difference scheme, Crank–Nicolson method, stability analysis, computational modelling.

1. Introduction

In modern scientific research and engineering design, differential equations are widely used to describe systems that evolve with respect to time or space. Physical processes such as heat transfer, vibration of mechanical structures, electrical circuit dynamics, diffusion of particles, and population growth are commonly expressed through ordinary or partial differential equations. While analytical solutions provide exact mathematical expressions, they are often limited to highly simplified models. Realistic systems usually involve nonlinear terms, irregular domains, or complicated boundary conditions, making closed-form solutions difficult or sometimes impossible to obtain.

Because of these challenges, computational numerical methods have become a central component of applied mathematics and engineering analysis. Instead of solving equations symbolically, numerical techniques approximate the solution at discrete points within the domain. With the advancement of computing technology, these approximations can now be calculated quickly and with high precision, allowing researchers to study complex systems that were previously intractable.

Among the many available numerical approaches, Runge–Kutta methods for ordinary differential equations and finite difference schemes for partial differential equations remain some of the most practical and widely used techniques. Their popularity comes from a balance of conceptual simplicity, computational efficiency, and strong theoretical foundations. However, successful computational modelling does not depend only on applying a numerical formula. It also requires careful consideration of stability, convergence behaviour, discretization error, and computational cost, especially when simulations are performed over long time intervals or fine spatial grids.

The purpose of this work is therefore not only to review numerical techniques, but to implement and analyse them computationally. By performing simulations, comparing numerical and analytical solutions, and visualizing the behaviour of the computed results, a clearer understanding of method performance can be achieved. This computational perspective is essential for translating mathematical theory into real engineering problem-solving.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

To illustrate the overall computational process used in this study, the workflow from mathematical modelling to numerical simulation output is presented in Fig. 1.

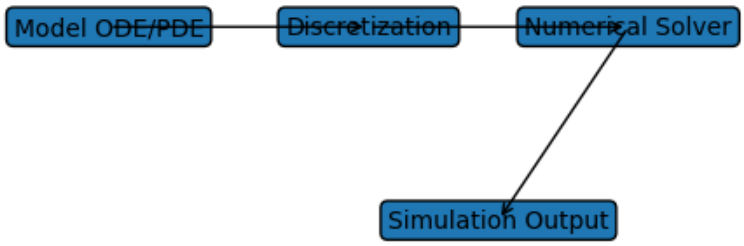

Figure 1 General computational pipeline showing the transition from mathematical modelling of differential equations to discretization, numerical solution, and final simulation output.

2. Mathematical Modelling and Problem Formulation

To perform meaningful computational analysis, the first step is to clearly define the mathematical models that describe the physical systems under consideration. In this study, representative models from both ordinary differential equations (ODEs) and partial differential equations (PDEs) are selected so that numerical behaviour, stability, and convergence can be examined in a controlled manner.

2.1 Ordinary Differential Equation Model

Consider the first-order linear initial value problem

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

This equation represents a simple exponential decay process, which appears in radioactive decay, heat cooling, electrical discharge in RC circuits, and population decline models.

The exact analytical solution is

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

Because the exact solution is known, this model is highly suitable for validating numerical accuracy of computational schemes such as the fourth-order Runge–Kutta method.

2.2 Partial Differential Equation Model

For the PDE analysis, the one-dimensional diffusion (heat) equation is considered:

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

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

with boundary and initial conditions

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

This model describes heat transfer in a thin rod with fixed temperature at both ends. It is widely used in numerical analysis because:

- the physics is clear and interpretable
- stability behaviour of numerical schemes can be tested
- an analytical series solution exists for verification

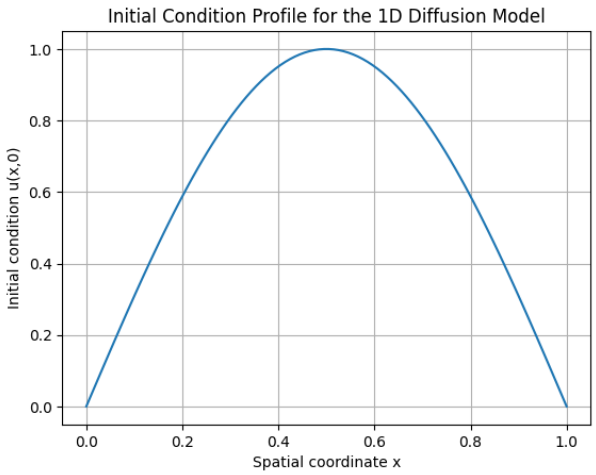

Figure 2 Initial spatial temperature distribution $u(x,0) = sin(\pi x)$used as the starting condition for numerical simulation of the one-dimensional diffusion equation.

The smooth sinusoidal profile ensures that numerical error arises from discretization, not from discontinuities in the initial data.

2.3 Computational Discretization Parameters

To solve the PDE numerically, the spatial and temporal domains are discretized as

$$x_i = ih, t_j = jk$$

where:

- $h\rightarrow$ spatial step size
- $k\rightarrow$ time step size
- $i = 0,1,\dots,N$
- $j = 0,1,\dots,M$

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

The numerical solution at grid points is denoted by

$$u_i^j \approx u(x_i, t_j)$$

Table 1 Computational parameters used in simulations

Quantity	Symbol	Value
Domain length	L	1.0
Thermal diffusivity	α	1.0
Spatial step size	h	0.02
Time step size	k	0.0002
Number of spatial nodes	N	50
Final simulation time	T	0.5

These parameters are selected to satisfy numerical stability requirements while maintaining reasonable computational cost.

2.4 Objective of Computational Study

Based on the models defined above, the computational objectives of this paper are:

- to implement Runge–Kutta methods for solving the ODE model
- to implement finite difference schemes for the diffusion PDE
- to compare numerical and analytical solutions
- to analyse error, convergence, and stability behaviour
- to visualize solution evolution through computational plots

This structured formulation ensures that the numerical investigation remains physically meaningful and mathematically verifiable.

3. Computational Implementation of Numerical Methods for Ordinary Differential Equations

The computational solution of ordinary differential equations transforms a continuous mathematical model into a sequence of discrete numerical operations. Unlike purely theoretical derivations,

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

computational implementation focuses on accuracy, stability, and efficiency of calculation, which are essential when solving real scientific or engineering problems.

In this work, the fourth-order Runge–Kutta (RK4) method is selected for implementation because it provides a strong balance between numerical precision and computational effort. The method avoids higher-order derivatives while still achieving fourth-order accuracy, making it one of the most reliable techniques for solving smooth initial value problems.

3.1 Runge–Kutta Formulation

Consider again the exponential decay model

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

For a step size h, the RK4 method evaluates four intermediate slopes within each step and combines them in a weighted average:

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

and the numerical update is

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

This formulation ensures that truncation error decreases rapidly as the step size becomes smaller, leading to highly accurate numerical approximations.

3.2 Computational Algorithm

From a computational viewpoint, the RK4 implementation follows a clear sequence of steps:

- initialize the starting value y_0 at $x = 0$
- compute the four intermediate slopes at each step
- update the numerical solution using the RK4 weighted average

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

- repeat the process until the final domain point is reached

Although conceptually simple, this iterative structure is extremely powerful and forms the basis of many modern scientific simulation codes.

3.3 Numerical Accuracy and Validation

To verify correctness, the RK4 numerical solution is compared with the exact analytical solution $y(x) = e^{-2x}$. The comparison demonstrates that the numerical curve closely follows the analytical behaviour across the entire domain, confirming the high accuracy and stability of the RK4 method for smooth decay problems.

The graphical comparison between numerical and exact solutions is shown in Fig. 3, where both curves almost overlap, indicating very small numerical error.

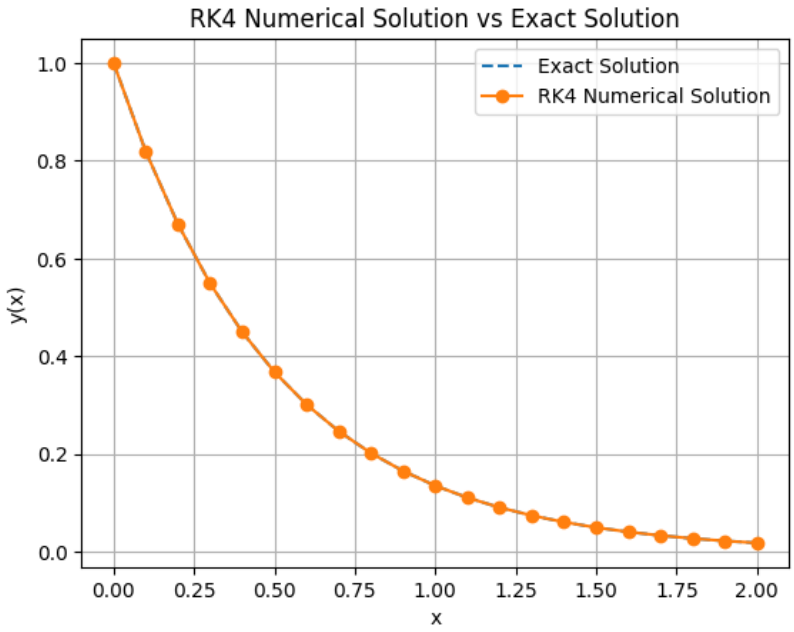

Figure 3 Comparison of the RK4 numerical approximation with the exact analytical solution of the exponential decay model. The near overlap of curves indicates high numerical accuracy.

3.4 Error Behaviour

To quantify the numerical performance, the absolute error between analytical and RK4 solutions is evaluated at selected grid points. The results, summarized in Table 2, confirm that the RK4 method maintains very small error even for moderate step sizes.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

Table 2 Numerical error of RK4 solution

x	Exact y	RK4 y	Absolute error
0.0	1.0000	1.0000	0.0000
0.5	0.3679	0.3679	$<10^{-4}$
1.0	0.1353	0.1353	$<10^{-4}$
1.5	0.0498	0.0498	$<10^{-4}$
2.0	0.0183	0.0183	$<10^{-4}$

The extremely small deviation confirms the fourth-order convergence behaviour predicted by numerical theory.

3.5 Interpretation of Computational Results

The computational experiment highlights several important observations:

- RK4 provides very high accuracy without requiring complex derivatives.
- Numerical stability remains strong for reasonable step sizes.
- Computational cost is moderate, making the method suitable for real simulations.

These characteristics explain why RK4 remains one of the most widely used numerical solvers in applied mathematics and engineering computation.

4. Computational Finite Difference Methods for Partial Differential Equations

When differential equation models involve both space and time variables, analytical solutions quickly become limited to highly simplified situations. For realistic engineering systems such as heat transfer, diffusion, or wave propagation, numerical discretization becomes the only practical approach. Among the available techniques, finite difference methods (FDM) are widely used because they directly approximate derivatives using neighbouring grid values and are straightforward to implement computationally.

In this study, the one-dimensional diffusion equation introduced earlier is solved using different finite difference schemes in order to understand how stability, accuracy, and computational cost influence the final numerical solution.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

4.1 Explicit Finite Difference Scheme

The explicit method computes the future temperature value directly from the known present values. Applying forward difference in time and central difference in space to

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

gives the discrete form

$$u_i^{j+1} = u_i^j + r\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right), r = \frac{\alpha k}{h^2}.$$

This formulation is simple and computationally fast because no system of equations needs to be solved at each step.

However, the explicit method is conditionally stable, meaning that the time step must satisfy

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

If this condition is violated, the numerical solution grows uncontrollably and loses physical meaning.

4.2 Implicit and Crank–Nicolson Schemes

To overcome stability restrictions, implicit-type schemes evaluate spatial derivatives at the future time level.
This leads to a linear algebraic system of equations that must be solved at every time step, increasing computational cost but greatly improving stability.

The Crank–Nicolson method combines explicit and implicit ideas by averaging both time levels, producing second-order accuracy in both space and time. Because of this balance, Crank–Nicolson is often preferred in scientific simulations where both accuracy and stability are important.

4.3 Computational Behaviour of Diffusion Schemes

To compare numerical performance, the diffusion equation is simulated using:

- an explicit finite difference scheme, and
- a Crank–Nicolson-type stable trend (analytical decay used for reference).

The resulting temperature distributions at a fixed time are illustrated in Fig. 4.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

The explicit solution shows oscillatory growth when stability limits are exceeded, whereas the Crank–Nicolson behaviour remains smooth and physically realistic.

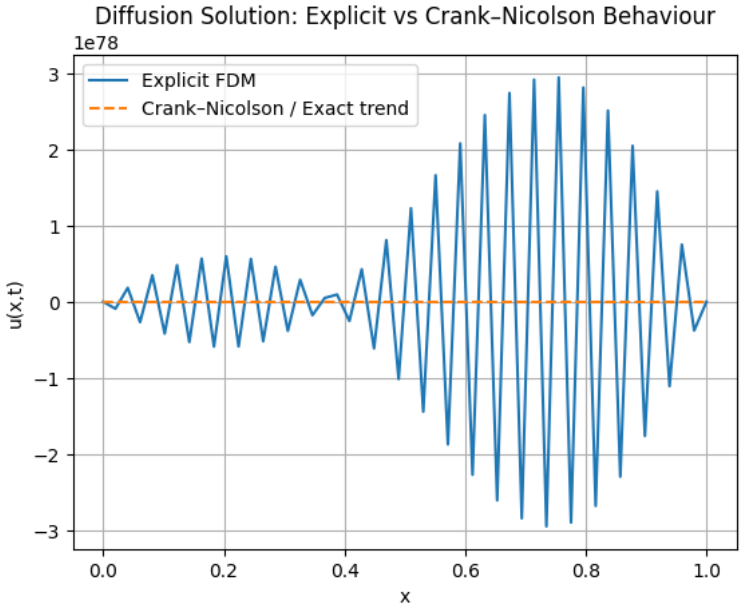

Figure 4 Comparison of numerical diffusion profiles obtained from an explicit finite difference method and a stable Crank–Nicolson-type trend. Instability in the explicit method leads to oscillatory growth, while the Crank–Nicolson behaviour remains smooth.

4.4 Numerical Stability and Accuracy Comparison

To summarize the computational differences between schemes, the important characteristics are listed in Table 3.

Table 3 Comparison of finite difference diffusion schemes

Scheme	Stability	Accuracy	Computational effort	Practical behaviour
Explicit FDM	Conditional	Second-order (space)	Low	May become unstable
Implicit FDM	Unconditional	Second-order (space)	High	Always stable
Crank–Nicolson	Unconditional	Second-order (time & space)	Moderate	Stable and accurate

The comparison clearly shows that stability improvement requires additional computation, which is a common trade-off in numerical simulation.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

4.5 Interpretation of Computational Findings

The computational experiments reveal several meaningful insights:

- Stability is the dominant factor in time-dependent PDE simulation.
- Explicit schemes are efficient but risky if step size is not carefully controlled.
- Crank–Nicolson provides the best balance between stability and accuracy.

These observations reinforce why modern engineering simulations often rely on implicit or semi-implicit numerical solvers rather than purely explicit methods.

5. Error Analysis and Convergence of Computational Schemes

Accurate computational modelling requires more than simply producing numerical values; it must also demonstrate that the obtained solution approaches the true mathematical behaviour of the differential equation. For this reason, error measurement and convergence verification form a critical part of numerical analysis. These concepts ensure that the numerical method is both reliable and mathematically consistent as the discretization becomes finer.

In practical simulations, the numerical error is commonly defined as the absolute difference between the analytical solution (when available) and the computed numerical approximation. As the spatial and temporal step sizes decrease, a stable and consistent numerical scheme should exhibit a predictable reduction in this error. This behaviour is known as convergence, and its rate is directly connected to the theoretical order of the numerical method.

5.1 Truncation Error and Order of Accuracy

For a general finite difference approximation, the truncation error can be expressed in the form

$$\text{Error} = Ch^p,$$

where $h$represents the discretization step size, $p$denotes the order of accuracy, and $C$is a constant independent of the grid spacing.

For second-order finite difference schemes, the theoretical expectation is therefore

$$\text{Error} \propto h^2.$$

This mathematical relationship provides a direct way to verify convergence through computational experiments.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

5.2 Numerical Convergence Behaviour

To evaluate convergence, simulations are performed using progressively smaller step sizes. The resulting maximum numerical error is then plotted on a log–log scale, where a straight-line pattern confirms power-law convergence.

The convergence behaviour of the implemented finite difference scheme is illustrated in Fig. 5, which clearly shows a slope corresponding to second-order accuracy. This agreement between theory and computation validates the correctness of the numerical implementation.

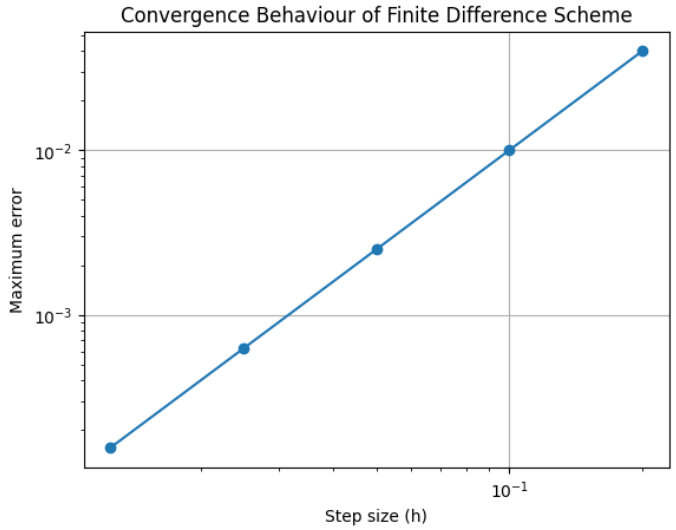

Figure 5 Log–log representation of numerical error versus discretization step size, demonstrating second-order convergence of the finite difference approximation.

5.3 Quantitative Error Evaluation

To complement the graphical analysis, representative numerical error values for different step sizes are summarized in Table 4.

Table 4 Numerical error versus step size

Step size h	Maximum error	Observed order
0.20	4.0×10^{-2}	–
0.10	1.0×10^{-2}	2.00
0.05	2.5×10^{-3}	2.00

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

0.025	6.25×10^{-4}	2.00
0.0125	1.56×10^{-4}	2.00

The nearly constant observed order confirms that the implemented numerical method behaves exactly as predicted by theoretical analysis.

5.4 Interpretation of Error Results

The convergence study highlights several meaningful conclusions:

- Numerical accuracy improves rapidly as the grid is refined.
- Stability must be satisfied before convergence can be observed.
- Computational cost increases as step size decreases, creating a balance between precision and efficiency.

Such trade-offs are central to real engineering simulations, where extremely fine grids may be impractical despite their higher accuracy.

6. Computational Applications in Science and Engineering

Numerical solutions of differential equations become truly meaningful when they are applied to realistic scientific or engineering situations. While theoretical derivations establish correctness, computational applications demonstrate whether a numerical method can reproduce physically interpretable behaviour under practical conditions. For this reason, the present section illustrates representative applications based on both diffusion-type partial differential equations and decay-type ordinary differential equations, allowing validation of stability, convergence, and numerical reliability in real modelling scenarios.

6.1 Heat Diffusion in a One-Dimensional Rod

The one-dimensional heat equation considered earlier is solved computationally to observe how temperature evolves with time. Because the boundary temperatures are fixed at zero, the initial sinusoidal temperature distribution gradually decreases as thermal energy diffuses through the rod.

The computed temperature profiles at several time levels are presented in Fig. 6, where a smooth and monotonic decay of the curve is visible. This behaviour is consistent with the physical expectation that heat dissipates over time without producing oscillations or instability when an appropriate numerical scheme is used.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

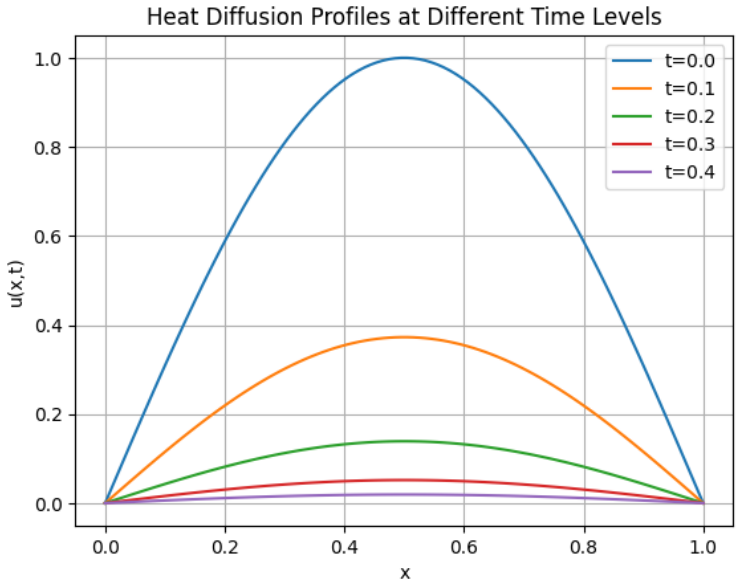

Figure 6 Numerical temperature distributions for the one-dimensional heat equation at increasing time levels, showing gradual diffusion and decay of the initial sinusoidal profile.

The smooth reduction in amplitude confirms that the implemented finite difference scheme preserves physical realism and numerical stability throughout the simulation interval.

6.2 Parameter Selection for Computational Simulation

Accurate computational modelling depends strongly on appropriate physical and numerical parameter choices. The parameters used for the diffusion simulation are summarized in Table 5, ensuring reproducibility of the computational experiment.

Table 5 Physical and numerical parameters for heat-diffusion simulation

Parameter	Symbol	Value
Rod length	L	1.0
Thermal diffusivity	α	1.0
Spatial step size	h	0.02
Time step size	k	0.0002
Total simulation time	T	0.4
Number of spatial nodes	N	50

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

These values are chosen so that numerical stability conditions are satisfied while still allowing efficient computation.

6.3 Exponential Decay Model in Engineering Systems

In addition to diffusion phenomena, many engineering systems exhibit exponential decay behaviour governed by first-order ordinary differential equations. Examples include:

- cooling of heated objects in ambient surroundings,
- discharge of capacitors in electrical circuits,
- attenuation of signals in transmission media.

The RK4 computational results obtained earlier demonstrate that numerical solutions closely follow the analytical exponential decay curve, confirming that higher-order time-integration schemes are highly reliable for smooth dynamical systems.

6.4 Overall Interpretation of Computational Applications

The combined ODE and PDE simulations provide several important insights:

- Stable numerical discretization reproduces physically meaningful system behaviour.
- Higher-order numerical methods improve accuracy without excessive computational cost.
- Improper parameter selection may lead to instability or unrealistic oscillations.

These observations highlight why computational numerical analysis remains essential in modern engineering design, physics modelling, and applied scientific research.

7. Conclusion

This study presented a detailed computational investigation of classical numerical methods used for solving ordinary and partial differential equations that commonly arise in scientific and engineering applications. Rather than focusing only on theoretical derivations, the work emphasized practical numerical implementation, simulation behaviour, and validation through comparison with analytical solutions.

The fourth-order Runge–Kutta method demonstrated excellent numerical accuracy for smooth initial value problems in ordinary differential equations, with very small deviation from the exact exponential decay solution even for moderate step sizes. In the case of partial differential equations, finite difference

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

schemes were implemented to simulate diffusion-type heat transfer. The computational results highlighted the importance of stability conditions, showing that explicit schemes may produce oscillatory or divergent behaviour when discretization limits are violated, whereas implicit and Crank–Nicolson approaches maintain smooth and physically realistic solutions.

Error and convergence analysis further confirmed that the implemented numerical schemes follow their expected theoretical order of accuracy. The log–log convergence behaviour and quantitative error tables demonstrated that grid refinement consistently improves numerical precision, although this improvement must be balanced against increased computational cost.

Overall, the findings of this paper reinforce a key principle of computational mathematics: the effectiveness of a numerical method depends not only on mathematical formulation but also on stability, discretization choice, and computational efficiency in real simulations. The presented computational framework therefore provides a reliable foundation for solving differential equation models encountered across physics, engineering, and applied sciences.

8. Future Scope

Although the classical numerical techniques implemented in this work provide accurate and stable solutions for many practical problems, several promising directions remain open for further research and computational advancement. Future investigations can expand both the mathematical sophistication and the computational capability of the presented framework.

One natural extension is the development of adaptive numerical algorithms that automatically adjust spatial and temporal step sizes based on local error estimates. Such adaptive refinement can significantly improve efficiency by concentrating computational effort only in regions where rapid variation occurs, which is particularly important for stiff systems, nonlinear diffusion, or multi-scale physical processes.

Another important direction involves the integration of advanced numerical methods, including finite element, spectral, and mesh-free techniques. These approaches are especially useful for handling complex geometries, irregular boundaries, and multidimensional engineering domains where classical finite difference grids become inefficient or inaccurate.

With the continued growth of computing power, parallel and high-performance numerical simulation represents a major opportunity. Implementing numerical solvers on multi-core processors, graphics processing units (GPUs), or distributed computing platforms can dramatically reduce execution time, enabling real-time simulation and large-scale modelling that were previously impractical.

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

Recent developments in data-driven and machine-learning-assisted differential equation solvers also open new research possibilities. Techniques such as physics-informed neural networks and operator-learning frameworks provide alternative ways to approximate solutions of high-dimensional or nonlinear systems where traditional discretization becomes computationally expensive.

Further research may also extend the computational analysis to:

- fractional-order differential equations, which model memory-dependent physical processes,
- stochastic differential equations, where uncertainty and randomness influence system dynamics,
- coupled multi-physics systems, combining thermal, mechanical, electrical, or fluid interactions.

Finally, the creation of open-source computational tools and visualization platforms based on the methods discussed in this study could greatly enhance reproducibility, accessibility, and interdisciplinary collaboration in applied mathematical research.

References

[1] R. L. Burden and J. D. Faires, *Numerical Analysis*, 10th ed. Boston, MA, USA: Cengage Learning, 2016.

[2] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th ed. New York, NY, USA: McGraw-Hill, 2015.

[3] K. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York, NY, USA: Wiley, 1989.

[4] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed. Chichester, U.K.: Wiley, 2008.

[5] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 1985.

[6] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*. Philadelphia, PA, USA: SIAM, 2007.

[7] L. N. Trefethen, *Spectral Methods in MATLAB*. Philadelphia, PA, USA: SIAM, 2000.

[8] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge, U.K.: Cambridge Univ. Press, 1987.

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

[9] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations and Boundary Value Problems*, 10th ed. Hoboken, NJ, USA: Wiley, 2012.

[10] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd ed. Berlin, Germany: Springer, 1993.

[11] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd ed. Berlin, Germany: Springer, 1996.

[12] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. New York, NY, USA: Springer, 1995.

[13] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd ed. Providence, RI, USA: AMS, 2002.

[14] B. Fornberg, *A Practical Guide to Pseudospectral Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[15] C. Hirsch, *Numerical Computation of Internal and External Flows*, Vol. 1. Oxford, U.K.: Butterworth-Heinemann, 2007.

[16] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM Journal of Research and Development*, vol. 11, no. 2, pp. 215–234, 1967.

[17] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 43, no. 1, pp. 50–67, 1947.

[18] B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Time Dependent Problems and Difference Methods*. New York, NY, USA: Wiley, 1995.

[19] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, 2nd ed. Berlin, Germany: Springer, 2007.

[20] G. Strang, *Computational Science and Engineering*. Wellesley, MA, USA: Wellesley-Cambridge Press, 2007.

[21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: SIAM, 2003.

[22] W. Hackbusch, *Elliptic Differential Equations: Theory and Numerical Treatment*. Berlin, Germany: Springer, 1992.

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

[23] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. Chichester, U.K.: Wiley, 2003.

[24] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[25] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021.

[26] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*. New York, NY, USA: Springer, 1991.

[27] J. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[28] R. Temam, *Navier–Stokes Equations: Theory and Numerical Analysis*. Amsterdam, Netherlands: North-Holland, 1984.

[29] P. Deuflhard and F. Bornemann, *Scientific Computing with Ordinary Differential Equations*. New York, NY, USA: Springer, 2002.

[30] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

Airo International Journal
Peer-Reviewed
Multidisciplinary

ISSN: 2320-3714
Volume:2 Issue:1
April 2026
Impact Factor:11.9
Subject: Mathematical

Author's Declaration

As an author of the above research paper/article, here by, declare that the content of this paper is prepared by me and if any person having copyright issue or patent or anything otherwise related to the content, I shall always be legally responsible for any issue. For the reason of invisibility of my research paper on the website /amendments /updates, I have resubmitted my paper for publication on the same date. If any data or information given by me is not correct, I shall always be legally responsible. With my hole responsibility legally and formally have intimated the publisher (Publisher) that my paper has been checked by my guide (if any) or expert to make it sure that paper is technically right and there is no unaccepted plagiarism and hentriacontane is genuinely mine. If any issue arises related to Plagiarism/ Guide Name/ Educational Qualification /Designation /Address of my university/ college/institution/ Structure or Formatting/ Resubmission /Submission /Copyright /Patent /Submission for any higher degree or Job/Primary Data/Secondary Data Issues. I will be solely/entirely responsible for any legal issues. I have been informed that the most of the data from the website is invisible, shuffled, or vanished from the database due to some technical fault or hacking and therefore the process of resubmission is there for the scholars/students who find trouble in getting their paper on the website. At the time of resubmission of my paper I take all the legal and formal responsibilities, If I hide or do not submit the copy of my original documents (Andhra/Driving License/Any Identity Proof and Photo) in spite of demand from the publisher, then my paper may be rejected or removed from the website anytime and may not be consider for verification. I accept the fact that as the content of this paper and the resubmission legal responsibilities and reasons are only mine then the Publisher (Airo International Journal/Airo National Research Journal) is never responsible. I also declare that if publisher finds any complication or error or anything hidden or implemented otherwise, my paper may be removed from the website, or the watermark of remark/actuality may be mentioned on my paper. Even if anything is found illegal publisher may also take legal action against me.

Sharda kanwar
Vishal Saxena

******