Advances in Differential Equations and Control Processes 2026, 33(3), 4643.
doi: https://doi.org/10.59400/adecp4643

Article

Computational modeling of turbulent nanofluid heat transfer over a heated moving surface with local thermal non-equilibrium and machine-learned eddy viscosity

Muhammad Shoaib Arif[1,*], **Yasir Nawaz**[2]

[1] Department of Mathematics and Sciences, College of Sciences and Humanities, Prince Sultan University, Riyadh 11586, Saudi Arabia

[2] Department of Mathematics, Faculty of Engineering and Computing, National University of Modern Languages (NUML), Islamabad 44000, Pakistan; yasir.nawaz@numl.edu.pk

*** Corresponding authors:** Muhammad Shoaib Arif, marif@psu.edu.sa

CITATION

Arif MS, Nawaz Y. Computational modeling of turbulent nanofluid heat transfer over a heated moving surface with local thermal non-equilibrium and machine-learned eddy viscosity. Advances in Differential Equations and Control Processes. 2026; 33(3): 4643. https://doi.org/10.59400/adecp4643

ARTICLE INFO

Received: 8 July 2026
Revised: 16 August 2026
Accepted: 21 August 2026
Available online: 28 August 2026

Abstract: This study presents an adaptive modified Runge-Kutta compact scheme for the numerical simulation of unsteady $k-\omega$ turbulent nanofluid flow over a heated moving surface under local thermal non-equilibrium conditions. The surface-interfacial model incorporates mixed convection, viscous dissipation, turbulence transport, and separate energy equations for the base fluid and nanoparticle phases, with the effective thermal conductivity described by Xue's formulation. The proposed time-integration method is explicit and combined with a compact finite-difference discretization that provides fourth-order spatial accuracy. The temporal coefficients are selected to achieve second-order accuracy, and the method is further enhanced through adaptive time stepping based on local error control. Stability analysis for the scalar convection–diffusion problem and conditional convergence analysis for the corresponding system formulation are also established. Numerical comparisons show that the proposed adaptive scheme yields lower error than existing adaptive Euler- and Runge-Kutta-based schemes. The computed results further demonstrate that thermal buoyancy increases the mean velocity, whereas larger Prandtl numbers reduce the thermal boundary-layer thickness of the fluid and nanoparticle phases. In addition, stronger interphase coupling modifies the two-temperature fields in a manner consistent with local thermal nonequilibrium. A machine-learning model is also employed to predict eddy viscosity, and its reliability is confirmed through profile comparisons, contour analyses, sensitivity assessments, and Taylor diagram evaluations. Overall, the proposed framework provides an accurate and efficient computational tool for surface-associated turbulent nanofluid transport with interfacial thermal nonequilibrium.

Keywords: turbulent nanofluid flow; local thermal non-equilibrium; adaptive modified Runge-Kutta method; $k-\omega$ turbulence model; eddy viscosity; machine learning

1. Introduction

Efficient thermal management remains a major challenge in many industrial and engineering systems, including heat exchangers, cooling devices, energy conversion units, microchannels, and advanced manufacturing processes. Conventional heat transfer fluids such as water, ethylene glycol, and oils generally have limited thermal conductivity, limiting their thermal performance under demanding operating conditions. To address this limitation, Choi introduced the concept of nanofluids,

namely base fluids containing suspended nanoparticles, and proposed that such suspensions could significantly enhance heat transport [1]. Later, Buongiorno developed a widely used transport model for nanofluids and identified Brownian diffusion and thermophoresis as the dominant slip mechanisms governing nanoparticle motion in many convective situations [2]. Reviews of nanofluid studies have since validated their ability to enhance thermal performance, while highlighting the importance of proper mathematical modeling of effective properties, stability, viscosity changes, and transport mechanisms [3].

In practice, most working regimes in thermal systems lie within the transitional or turbulent regime, where the interactions among turbulence, nanoparticle dispersion, and heat transfer are extremely intricate. In this situation, the simulation's reliability is highly dependent on the turbulence closure used in the governing model. The $k-\omega$ family is popular among Reynolds-averaged Navier-Stokes (RANS) closures, particularly due to its excellent near-wall behavior and its applicability to flows dominated by the boundary layer [4]. The two-equation eddy-viscosity model introduced by Menter further strengthened the performance of the $k-\omega$ type models in engineering practice involving adverse pressure gradients and flow separation [5]. Formulations based on the $k-\omega$ model have also been successfully applied in studies of kinetic viscomagnetic heat transfer in nanofluids and within nanophase boundary (NPB) geometries [6].

At the same time, the assumption of local thermal equilibrium is often insufficient for systems involving porous media, coated surfaces, packed beds, or multiphase thermal interaction, where the fluid and solid phases may not instantaneously share the same temperature [7]. Under these conditions, the local thermal non-equilibrium (LTNE) approach is more realistic, as it uses separate energy equations for the interacting phases and accounts for finite interphase heat exchange [8]. Recent reviews and numerical analyses have revealed that LTNE can significantly influence temperature fields, Nusselt number predictions, entropy generation, and the overall behaviour of heat transfer, particularly in convection-dominated porous and nanofluid-saturated systems [9]. This makes LTNE modeling particularly applicable for investigating thermally enhanced turbulent flows in complex media [10].

The mathematical model for turbulent nanofluid flow under LTNE conditions is thus highly non-linear and tightly coupled. The momentum equations interact with the turbulence transport equations. Simultaneously, the thermal field is controlled by independent yet interdependent energy balances that often depend on loading and the nanoparticle's effective thermophysical properties. Such systems are likely to be stiff, multiscale, and transient, with sharp local gradients, underscoring the importance of the temporal discretization strategy. Although fixed-step schemes can be inefficient when the solution is traveling on nonuniform time scales, the classical Runge-Kutta methods are popular due to their simplicity, explicit nature, and ease of implementation.

Conversely, adaptive time-stepping, particularly with embedded Runge-Kutta error control, offers an effective solution for dynamically controlling the step size based on local solution behavior [11]. The pioneering work of Dormand and Prince, along with the conventional theory of Hairer, Norsett and Wanner, established useful

adaptive Runge-Kutta methods as powerful tools for nonstiff and moderately stiff time-dependent systems. More recent studies have also examined adaptive step-size controllers for diffusion-type problems and confirmed the practical benefits of variable time stepping in accuracy–efficiency trade-offs [12,13].

If you need to solve a parabolic PDE, many explicit numerical methods can be used. The so-called method of lines (MOL) is a common strategy; it entails spatial discretization of the problem and then solving the resulting system of ordinary differential equations (ODEs) with standard ODE solvers. The standard explicit and implicit finite difference schemes implement the explicit and implicit Euler time-discretization, respectively, and the implicit trapezoidal rule, which are actually two first-order and one second-order RK methods. The third most common scheme is the Crank-Nicolson finite difference scheme, which follows spatial discretization. Therefore, these schemes can be seen as siblings of the MOL family. It is still difficult to outperform RK methods, despite their long history of success. For example, Savović et al. [14] compared the most basic explicit finite difference scheme to the so-called UPFD (unconditionally positive finite difference) scheme developed over the past 10 years. Due to the additional truncation error terms in the UPFD scheme, they found that the classical scheme is more accurate.

Ordinary differential equation systems are often solved using explicit numerical methods with fixed time steps. Depending on how the solution changes during the integration interval, this approach can underperform in some areas and be overly effective in others. To avoid poor performance, you can use a small, constant time step when the answer changes quickly. However, this small constant time step might lead to unnecessary computational cost when the solution changes slowly.

Error-based step-size control for discontinuous Galerkin methods has been presented by Ranocha et al. [15]. A recent study investigated entropy generation in Joule-heated radiative viscous flow over a radially stretching permeable disk, highlighting the combined influence of thermal radiation, viscous dissipation, and electromagnetic heating on irreversibility and heat transfer. The analysis demonstrated that such coupled effects can significantly modify entropy production and the associated thermal transport behavior in stretching-surface flows [16,17].

The numerical analysis literature is dense with studies that aim to estimate the local error of Runge-Kutta methods for ordinary differential equations [18–20]. One way to estimate the local error is to look at data from a single step, and another is to look at data from multiple steps. The focus of this work does not extend to the second category of approaches. The dependent variable u is first calculated using a full time step h and then recalculated using two half-time steps $\frac{h}{2}$; this is the most famous first-type method for estimating the local error. The local error (LE) is the difference between the two u-values.

The pseudo-iterative formula is another popular approach; it consists of an RK formula of order p followed by an RK formula of order $p + 1$. This algorithm is known as an embedded method because it uses previously calculated quantities from the lower-order RK formula to save time [21]. A linear ordinary differential equation is necessary for the applicability of a formula that Merson developed in his early

work [22] to provide a reasonable approximation of the local error. Subsequently, Alexandre dit Sandretto [23] proposed a mathematical verification of local truncation errors and order conditions using Butcher series (B-series). To address this inconsistency, Martens [24] developed a discretization procedure that provides control-like algebraic treatment, consistency, optimality, and convergence proofs.

For turbulent transport problems, however, the challenge does not lie only in time integration; it also lies in the fidelity of the turbulence closure. Classical eddy-viscosity models remain computationally attractive, but their linear constitutive assumptions may struggle to represent complex turbulence physics across different geometries, Reynolds numbers, and thermal environments. This limitation has led to a rapid rise of data-driven turbulence modeling, where machine learning is used to enhance or augment traditional closures. Ling, Kurzawski, and Templeton demonstrated that deep neural networks with embedded invariance can improve Reynolds-averaged turbulence modeling by learning Reynolds-stress behaviour from high-fidelity data [25]. Duraisamy, Iaccarino, and Xiao later emphasized the broader shift in turbulence modeling in the age of data-rich simulation and hybrid physics-informed modeling [26]. Other related work has investigated neural network-based LES closures, surrogate eddy-viscosity formulations, and interpretable machine-learned turbulence models [27, 28]. Such developments suggest that machine learning does not directly replace classical turbulence models; however, it can be used as a complementary tool to enhance the accuracy of closures without altering the structure of classical CFD frameworks.

Research Gap:

Although considerable progress has been made in numerical simulation of nanofluid heat transfer, turbulence modeling, local thermal non-equilibrium formulations, high-order discretization, and machine-learning-assisted transport prediction, these developments have largely been pursued separately or in partially coupled forms. Many turbulent nanofluid studies employ a single-temperature thermal model, whereas LTNE studies commonly focus on laminar or simplified transport configurations. Likewise, high-order numerical investigations frequently emphasize accuracy of the discretization without combining adaptive temporal control with a rigorous consistency–stability–convergence assessment for a coupled $k-\omega$ turbulence thermal system. Machine-learning applications in fluid mechanics have also increasingly been used for surrogate prediction; however, their integration with an LTNE turbulent nanofluid solver and their validation for unseen physical parameter combinations remain comparatively limited. These observations motivate the development of a unified framework that treats turbulent momentum transport, two-temperature thermal nonequilibrium, adaptive high-order computation, and data-driven eddy-viscosity prediction consistently within a single model.

Artificial neural network modeling has been given by Nawaz et al. [29] for predicting the skin friction coefficient and local Nusselt number for Eyring-Prandtl nanofluid flow over a Riga plate. Recent studies have combined CFD with artificial intelligence tools to infer relationships among Reynolds number, turbulent intensity, flow variables, and thermal response in turbulent nanofluid systems [30]. Reviews

have also documented the growing use of machine learning to predict nanofluid heat transfer and estimate nanofluid properties. Even so, most existing studies focus on global thermal indicators, pressure drop, or property prediction rather than the direct integration of a machine-learning predictor into the eddy-viscosity component of a turbulent nanofluid solver operating under LTNE conditions [31]. This identifies a meaningful gap between current numerical methods, turbulence-closure enhancement, and advanced non-equilibrium thermal modeling.

From a numerical analysis perspective, this gap is important. When turbulence closure, nanofluid transport, and LTNE thermal coupling are combined in a single framework, the resulting semi-discrete system can contain both fast and slow scales. A recent study proposed a two-stage exponential integrator–Runge-Kutta scheme for stochastic diffusive SIQR epidemic models, demonstrating improved mean-square accuracy and lower numerical error than Euler–Maruyama and nonstandard finite-difference approaches [32]. When such a method is combined with a machine-learning-based eddy-viscosity predictor, the numerical solver gains an additional mechanism for capturing complex closure behaviour without abandoning the familiar $k - \omega$ structure. This hybrid approach is particularly attractive for CFD-oriented thermal applications in which full high-fidelity turbulence simulation would be prohibitively expensive. Yet, classical closures alone may not offer sufficient predictive flexibility.

Turbulence is multiscale, making direct numerical simulation (DNS) of it problematic for most purposes. The Reynolds-averaged Navier-Stokes (RANS) equations are still used for steady-state analysis of turbulent flows in engineering applications with short turnover times.

In that light, design workflows could be drastically altered by innovations that lead to faster or more accurate RANS outputs. Consequently, several studies have investigated data-driven approaches to enhance RANS accuracy through fidelity porting strategies or experimental data [33–35]. Methods for optimizing closures have also included improving accuracy through model selection or zonal coefficient refinement [36,37]. The vast majority of physics-informed machine learning studies of turbulence closure modeling have focused on improving the accuracy of the quantities of interest by incorporating information from higher-fidelity models, such as DNS, into reduced-order representations [38–42].

To restore sub-grid data, researchers have recently investigated spatio-temporal super-resolution [43] and advocated for computational fluid dynamics (CFD)-driven machine learning [44], which combines machine learning optimization epochs with a forward solution of the RANS equations. Machine learning has numerous promising applications in turbulence modeling, as highlighted in reviews [45].

Motivated by these considerations, the present work develops a modified Runge-Kutta scheme with adaptive time stepping for the simulation of $k - \omega$ turbulent nanofluid flow under thermal non-equilibrium, together with a machine-learning-based eddy-viscosity prediction framework. The objective is to construct a computational methodology that combines three strengths: the physical interpretability and practical robustness of the $k - \omega$ turbulence model, the computational efficiency of adaptive

Runge-Kutta time integration, and the enhanced predictive capability offered by data-driven closure support. The resulting framework is expected to improve temporal efficiency, better resolve local transient behaviour, and provide a more flexible representation of turbulence effects in thermally coupled nanofluid systems.

Novelty of the Research:

The following points highlight the research's novelty.

1. Integration of the turbulence model with a two-temperature LTNE nanofluid formulation for transient flow over a heated moving surface.
2. Combination of an exponentially parameterized second-order RK formulation, fourth-order compact spatial discretization, and adaptive time-step control.
3. Explicit order conditions, local truncation-error analysis, Fourier stability characterization, and conditional global convergence of order $O(\Delta t^2 + \Delta y^4)$.
4. Prediction of eddy viscosity from local flow and thermal variables with evaluation on genuinely unseen parameter combinations rather than only randomly separated points from the same simulations.
5. Manufactured solution convergence tests, comparisons with RK2, RK4 and an embedded adaptive RK method, physical benchmark validation, and measured computational cost.

The rest of the paper is organized as follows. Section 2 introduces the modified Runge-Kutta schemes, compact spatial discretization, and adaptive time-stepping strategy. Section 3 reports the main stability and convergence analysis of the proposed modified Runge-Kutta scheme. Section 4 presents the mathematical formulation and dimensionless governing equations for the $k - \omega$ turbulent nanofluid model under LTNE conditions. Section 5 reports numerical simulations, error comparisons, and machine-learning-based eddy-viscosity analysis. Finally, the last section summarizes the principal findings and possible future directions.

2. Modified Runge-Kutta scheme

In the literature, there exist Runge-Kutta schemes that can be used to solve various types of partial differential equations. These schemes provide second- to fourth- or fifth-order accuracy in time. In this study, a modification to the earlier Runge-Kutta scheme is suggested. The scheme will provide a second-order-in-time solution and will be constructed in two stages. The first stage will be called the predicted stage, while the second stage will be the corrector stage. A modified scheme is constructed for the following time-dependent partial differential equations.

$$\frac{\partial z}{\partial t} = f(z, z_y, z_{yy}), \tag{1}$$

subject to the boundary conditions

$$z(t,\ 0) = f_1(t),\ z(t,\ L) = f_2(t), \tag{2}$$

and the initial condition can be written as

$$z(0, y) = f_3(y). \tag{3}$$

Where $f_1,\ f_2$ and f_3 are functions of t and y. The unknown quantity $z(t,y)$ changes with time and space, and its evolution depends on its current value, slope, and curvature. This is the same setup used in many real transport problems: temperature in a cooling plate, velocity in a boundary layer, pollutant concentration in water, and eddy viscosity in turbulence models. Before solving anything numerically, the model must specify the initial state and the boundary or edge conditions.

2.1. Predictor stage

The first stage of the proposed scheme can be expressed as:

$$z^P = z^n e^{a_1\Delta t} + \frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left\{\frac{1}{4}\ \frac{\partial z}{\partial t}\bigg|^n - a_1 z^n\right\}, \tag{4}$$

which gives the predicted value z^P at the next time level. Where a_1 is any parameter that can be freely chosen, and Δt is the temporal step size. Equation (4) is the predictor stage; it is not first-order accurate. Instead of using a plain Euler-type first guess, we use an exponential-type predictor. This is important because exponential behavior often appears in transport and diffusion processes. This predictor is a trial step: it estimates the future solution, but it is not yet accurate enough by itself, so it is later corrected. In real systems, many changes are not purely linear in time. For example, heat may decay rapidly at first and then slowly; turbulence quantities may grow or damp strongly; and concentration may diffuse in a nonuniform way. So the predictor acts like a smart first forecast of the future state, better than a basic straight-line estimate.

2.2. Corrector stage

After the prediction, the first- or second-order time-corrector stage is established. To do so, the corrector scheme can be expressed as:

$$z^{n+1} = az^n + bz^P + c\Delta t\ \frac{\partial z}{\partial t}\bigg|^P, \tag{5}$$

which combines: the old value z^n, the predicted value z^P, and the derivative is evaluated at the predicted stage. This is the correction step. The predictor alone is only an intermediate estimate. The corrector improves it by blending past information and predicted information. So, the logic is: guess the future, compute how good that guess is, then correct it. This is like decision-making in real life: first make a forecast, then use updated information, then revise the forecast. In physical problems, this is useful because transport systems often react differently once you move slightly forward in time. The corrector captures that extra information.

The values of a, b and c will be determined by comparing the coefficients of the Taylor series expansion of z^{n+1} of Equation (5) after substituting Equation (4). Therefore, expand z^{n+1} in the form of

$$z^{n+1} = z^n + \Delta t\ \frac{\partial z}{\partial t}\bigg|^n + \frac{(\Delta t)^2}{2}\ \frac{\partial^2 z}{\partial t^2}\bigg|^n + O\left((\Delta t)^3\right) \tag{6}$$

This is the step that guarantees the method is not just “reasonable,” but

mathematically consistent to second order in time. That means when Δt becomes small, the error decreases approximately like $(\Delta t)^2$. For real engineering problems, this matters because if the method is too low order, you may need a very small time step, which increases cost and makes large simulations impractical. Second-order accuracy gives a better balance between accuracy, speed, and implementation simplicity.

Now, substituting the predictor stage (4) into Equation (5) results in

$$\begin{aligned} z^{n+1} = \; & az^n + b\left\{z^n e^{a_1\Delta t} + \frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left(\frac{1}{4}\left.\frac{\partial z}{\partial t}\right|^n - a_1 z^n\right)\right\} \\ & + c\Delta t\left\{e^{a_1\Delta t}\left.\frac{\partial z}{\partial t}\right|^n + \frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left(\frac{1}{4}\left.\frac{\partial^2 z}{\partial t^2}\right|^n - a_1\left.\frac{\partial z}{\partial t}\right|^n\right)\right\} \end{aligned} \tag{7}$$

By using the Taylor series expansion of z^{n+1} into Equation (7), which gives

$$\begin{aligned} & z^n + \Delta t\left.\frac{\partial z}{\partial t}\right|^n + \frac{(\Delta t)^2}{2}\left.\frac{\partial^2 z}{\partial t^2}\right|^n = az^n + b\left\{z^n + \frac{\left(e^{a_1\Delta t}-1\right)}{4a_1}\left.\frac{\partial z}{\partial t}\right|^n\right\} \\ & + c\Delta t\left\{\left.\frac{\partial z}{\partial t}\right|^n + \frac{\left(e^{a_1\Delta t}-1\right)}{4a_1}\left.\frac{\partial^2 z}{\partial t^2}\right|^n\right\} \end{aligned} \tag{8}$$

$$\left.\begin{aligned} 1 &= a+b \\ \Delta t &= \tfrac{b}{4a_1}\left(e^{a_1\Delta t}-1\right) + c\Delta t \\ \tfrac{\Delta t}{2} &= \tfrac{c}{4a_1}\left(e^{a_1\Delta t}-1\right) \end{aligned}\right\}, \tag{9}$$

Solution of the system of Equation (9) produces values of a, b and c which are expressed as:

$$\left.\begin{aligned} c &= \tfrac{2a_1\Delta t}{\left(e^{a_1\Delta t}-1\right)} \\ b &= \tfrac{4a_1}{\left(e^{a_1\Delta t}-1\right)}\left\{\Delta t - \tfrac{2a_1(\Delta t)^2}{\left(e^{a_1\Delta t}-1\right)}\right\} \\ a &= 1 - \tfrac{4a_1}{\left(e^{a_1\Delta t}-1\right)}\left\{\Delta t - \tfrac{2a_1(\Delta t)^2}{\left(e^{a_1\Delta t}-1\right)}\right\} \end{aligned}\right\}, \tag{10}$$

values of parameters a, b and c mentioned in Equation (10) provide a scheme that gives a second-order accuracy in time. But if the weights of the classical Runge-Kutta scheme are used, then the scheme will be approximately second-order accurate, not exactly. This contribution also employs an adaptive time step size. This adaptive time step is based on the norm of the difference between two numerical schemes. These two numerical schemes are the existing second-order Runge-Kutta method and the proposed modified Runge-Kutta scheme constructed in this contribution. Let the solution obtained by the existing Runge-Kutta scheme be denoted by z_n and the solution obtained by the proposed first stage of the RK method scheme and the classical second-order stage of the existing RK scheme is denoted by z_l. The error is expressed as:

$$err = \|z_n - z_l\|_\infty \tag{11}$$

If this error given by Equation (11) satisfies the inequality $err < tol$, then accept

the step, and then equate $u = u_n$ and update

$$\Delta t_{new} = safety\Delta t \left(\frac{tol}{err}\right)^{0.7} \tag{12}$$

If $err \geq tol$, then reject the step and reduce Δt time step.

This is one of the method's strongest parts. Instead of choosing a single fixed time step for the entire simulation, the method uses larger steps when the solution is smooth and smaller steps when it changes rapidly. So the algorithm becomes self-adjusting. This is extremely relevant in real problems because physical systems do not evolve uniformly. Examples include mixed convection, where flow can suddenly accelerate near a hot wall; turbulence, where eddy viscosity may change sharply in certain regions; heat transfer, where temperature may change rapidly at startup and then become steady; and pollutant transport, where steep fronts may appear temporarily.

A fixed time step wastes effort during calm periods and may fail during rapid-change periods. Adaptive stepping automatically handles both.

2.3. Discretize space using compact finite differences

Consider the partial differential equation in the form of

$$\frac{\partial z}{\partial t} = \mu_1 \frac{\partial z}{\partial y} + \mu_2 \frac{\partial^2 z}{\partial y^2}, \tag{13}$$

which is the standard convection-diffusion form.

Then, the proposed scheme for time discretization of Equation (2) is written as:

$$z^P = z^n e^{a_1 \Delta t} + \frac{\left(e^{a_1 \Delta t} - 1\right)}{a_1} \left\{ \frac{1}{4} \left(\mu_1 \left. \frac{\partial z}{\partial y} \right|^n + \mu_2 \left. \frac{\partial^2 z}{\partial y^2} \right|^n \right) - a_1 z^n \right\} \tag{14}$$

$$z^{n+1} = az^n + bz^P + c\Delta t \left\{ \mu_1 \left. \frac{\partial z}{\partial y} \right|^P + \mu_2 \left. \frac{\partial^2 z}{\partial y^2} \right|^P \right\} \tag{15}$$

Now the method is ready for PDEs containing advection/convection through z_y, diffusion through z_{yy}. This is exactly the structure that appears in many thermal-fluid problems. This form models phenomena such as heat carried by fluid motion and spreading by conduction, contaminants transported by flow and diffusing in rivers, momentum transport in boundary layers, and scalar transport in turbulence. So, this step connects the general numerical method to physically meaningful PDEs.

Existing and modified Runge-Kutta schemes can only be used for the time discretization of partial differential equations. To obtain solutions of time-dependent partial differential equations, another scheme must discretize the space-dependent term. To achieve this goal, a compact difference scheme is employed in this contribution. A compact scheme can be expressed in the form of

$$\frac{1}{4} \left. \frac{\partial z}{\partial y} \right|_{j-1}^n + \left. \frac{\partial z}{\partial y} \right|_j^n + \frac{1}{4} \left. \frac{\partial z}{\partial y} \right|_j^n = \frac{3}{2} \left(\frac{z_{j+1} - z_{j-1}}{2\Delta y} \right) \tag{16}$$

This Equation (16) can be used to discretize the first-order spatial partial derivative.

The matrix in the form of Equation (16) is expressed as:

$$\begin{bmatrix} 1 & \frac{1}{4} & 0 & . & . & . & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} & . & . & . & 0 \\ 0 & \frac{1}{4} & 1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & 1 \end{bmatrix} \begin{bmatrix} z_1^{'} \\ z_2^{'} \\ z_3^{'} \\ . \\ . \\ . \\ z_{N-1}^{'} \end{bmatrix} = \frac{3}{4\Delta y} \begin{bmatrix} z_2 - z_\circ \\ z_3 - z_1 \\ z_4 - z_2 \\ . \\ . \\ . \\ z_N - z_{N-2} \end{bmatrix} \tag{17}$$

Where $z_j^{'} = \left.\frac{\partial z}{\partial y}\right|_j$. A compact scheme for the second-order partial derivative is expressed as:

$$\frac{1}{10}\left.\frac{\partial^2 z}{\partial y^2}\right|_{j-1}^{n} + \left.\frac{\partial^2 z}{\partial y^2}\right|_{j}^{n} + \frac{1}{10}\left.\frac{\partial^2 z}{\partial y^2}\right|_{j+1}^{n} = \frac{3}{2}\left(\frac{z_{j+1}^n - 2z_j^n + z_{j-1}^n}{2(\Delta y)^2}\right) \tag{18}$$

Matrix form for this compact scheme can be expressed as:

$$\begin{bmatrix} 1 & \frac{1}{10} & 0 & . & . & . & 0 \\ \frac{1}{10} & 1 & \frac{1}{10} & . & . & . & 0 \\ 0 & \frac{1}{10} & 1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & 1 \end{bmatrix} \begin{bmatrix} z_1^{''} \\ z_2^{''} \\ z_3^{''} \\ . \\ . \\ . \\ z_{N-1}^{''} \end{bmatrix} = \frac{6}{5(\Delta y)^2} \begin{bmatrix} z_2 - 2z_1 + z_\circ \\ z_3 - 2z_2 + z_1 \\ z_4 - 2z_3 + z_2 \\ . \\ . \\ . \\ z_N - 2z_{N-1} + z_{N-2} \end{bmatrix} \tag{19}$$

Where $z_j^{''} = \left.\frac{\partial^2 z}{\partial y^2}\right|_j$.

Let solutions of Equations (17) and (19) be written as:

$$\left.\frac{\partial z}{\partial y}\right|_j^n = \delta z^n, \quad \left.\frac{\partial^2 z}{\partial y^2}\right|_j^n = \delta^2 z^n \tag{20}$$

Compact schemes are used because they give high spatial accuracy with a short stencil. That means better gradient resolution, less numerical smearing, and better performance for smooth yet strongly varying profiles. This matters in applications where sharp spatial gradients occur, such as boundary layers near walls, thermal layers in nanofluid flow, concentration fronts, and velocity shear in turbulent transport.

2.4. Combine time and space into the final algorithm

By using Equation (20), the proposed scheme for Equation (13) can be written as:

$$z^P = z^n e^{a_1 \Delta t} + \frac{\left(e^{a_1 \Delta t} - 1\right)}{a_1}\left\{\frac{1}{4}\left(\mu_1 \delta z^n + \mu_2 \delta^2 z^n\right) - a_1 z^n\right\} \tag{21}$$

$$z^{n+1} = a z^n + b z^P + c\Delta t\left\{\mu_1 \delta z^P + \mu_2 \delta^2 z^P\right\} \tag{22}$$

This is the actual implementation form: compute spatial derivatives, predict,

compute the corrected update, estimate the error, accept or reject the step, and repeat until the final time.

Let $h = \Delta t,\ q = \frac{e^{a_1 h}-1}{4a_1}$.

The predictor stage can be simplified as

$$z^P = z^n + qF\left(z^n\right),$$

where $F(z) = f(z,\ z_y, z_{yy})$.

The corrector becomes

$$z^{n+1} = az^n + bz^P + chF\left(z^P\right).$$

Using $a + b = 1$, this may be rewritten as

$$z^{n+1} = z^n + bqF\left(z^n\right) + chF\left(z^P\right).$$

Define the internal stage location.

$$r = \frac{q}{h} = \frac{e^{a_1 h} - 1}{4a_1 h}.$$

The scheme then has the equivalent Runge-Kutta representation

$$z^P = z^n + rhF\left(z^n\right),$$

$$z^{n+1} = z^n + h\left[(1-c)F\left(z^n\right) + cF\left(z^P\right)\right].$$

The coefficients satisfy

$$(1-c) + c = 1,\ cr = \frac{1}{2}.$$

Therefore,

$$c = \frac{1}{2r} = \frac{2a_1 h}{e^{a_1 h} - 1},$$

and

$$1 - c = 1 - \frac{2a_1 h}{e^{a_1 h} - 1}.$$

These are precisely the two order conditions for a two-stage explicit second-order Runge-Kutta method.

2.5. Difference from classical RK2 methods

Classical two-stage RK2 formulas use a fixed internal stage r. For example: explicit midpoint method: $r = 1/2$; Ralston method: $r = 2/3$; Heun method: $r = 1$. In the present formulation, $r = \frac{e^{a_1 h}-1}{4a_1 h}$, so the stage location varies with the dimensionless quantity $a_1 h$. Thus, the main difference is not the formal order or linear stability polynomial. It is the use of an exponentially parameterized, step-dependent internal stage. For $a_1 h \to 0$, $r \to \frac{1}{4}$, and the method approaches the two-stage RK2 formula

$$z^P = z^n + \frac{h}{4}F\left(z^n\right),$$

$$z^{n+1} = z^n + h\left[-F\left(z^n\right) + 2F\left(z^P\right)\right].$$

Therefore, the proposed formula belongs to the general two-stage RK2 family but uses a stage parameter generated through the exponential expression.

The proposed formula is an exponentially parameterized two-stage Runge-Kutta method. By defining $r = \left(e^{a_1\Delta t} - 1\right) / \left(4a_1\Delta t\right)$, the scheme can be expressed in the standard explicit RK form with weights $b_1 = 1 - 1/(2r)$ and $b_2 = 1/(2r)$. These coefficients satisfy the second-order conditions $b_1 + b_2 = 1$ and $b_2 r = 1/2$. Accordingly, its stability function for the linear test equation is $R(z) = 1 + z + z^2/2$, which coincides with that of classical RK2. The contribution therefore does not enlarge the RK2 absolute stability region. Instead, the exponential expression produces a step-dependent internal stage that influences higher-order nonlinear behavior and provides a parameterized implementation within the general RK2 family. The local truncation error is $O\left(\Delta t^3\right)$, while the global temporal accuracy is $O\left(\Delta t^2\right)$. The free parameter a_1 does not affect the second-order conditions or the leading local-error coefficients; its influence appears in terms of order $O\left(\Delta t^4\right)$ and higher.

2.6. Why is this scheme relevant to real-world problems

In many thermal systems, the solution changes rapidly at the beginning and more slowly later. This adaptive time stepping is ideal here because it uses small time steps when the transient is sharp and larger ones later. These flows have steep gradients near walls. Our compact spatial scheme is useful because it captures first and second derivatives more accurately than low-order methods. Eddy viscosity may vary sharply in time and space. This predictor-corrector structure, combined with adaptive control, helps track such changes without requiring a uniformly small time step everywhere. The model form $z_t = \mu_1 z_y + \mu_2 z_{yy}$ appears in heat, mass, and momentum transport. So our method is not limited to a single application; it can be used across many transport systems. Because the method is explicit in structure, it is easier to implement than many fully implicit solvers. At the same time, adaptive stepping reduces the burden of manually tuning a stable fixed time step.

In short, a high-accuracy-in-space, predictor-corrector-in-time, adaptive explicit solver for transient convection–diffusion-type PDEs, built to handle rapidly changing physical systems more efficiently than standard fixed-step RK methods.

3. Stability analysis

Stability analysis answers one practical question: If I march the numerical solution forward in time, will small errors stay controlled, or will they grow and destroy the solution? That is the real meaning of stability. In real computations, errors always exist because of truncation errors, round-off errors, approximation of derivatives, and imperfect initial or boundary data. A stable method keeps these errors from exploding. An unstable one turns tiny numerical noise into a meaningless solution. For convection-diffusion and turbulent transport problems, this is critical because the solution often contains: steep gradients, diffusive smoothing, transport by flow, and possible oscillatory behavior. The scheme for the convection-diffusion equation is written as the predictor-corrector pair Equations (21) and (22), where δ and δ^2 are the compact approximations of the first and second spatial derivatives. This is the discrete

machine we are analyzing. Stability is not tested directly on the PDE, but rather on the numerical update rule. This tells you whether your algorithm is safe to use for real transport problems such as heat transfer in a cooling channel, pollutant convection and diffusion, momentum transport in boundary-layer flow, and turbulence-related scalar transport.

Finite difference schemes can be analyzed for stability by applying von Neumann stability analysis. This stability analysis has been used in the literature to determine the stability region. For linear partial differential equations, this stability analysis gives exact stability conditions; for non-linear partial differential equations, it provides an estimate of the stability region, since the non-linear equation must be linearized. The stability condition is imposed on amplification factors obtained from this stability analysis. The amplification factor is the ratio of the wave's amplitudes at two consecutive time levels. To apply this analysis, consider the following transformations:

$$\left.\begin{aligned} z_j^n &= E^n e^{jI\psi} \\ \delta z_j^n &= \frac{3}{4\Delta y}\frac{\left(e^{(j+1)I\psi}-e^{(j-1)I\psi}\right)}{e^{(j+1)I\psi}+e^{jI\psi}+e^{(j-1)I\psi}} \\ \delta^2 z_j^n &= \frac{6}{5(\Delta y)^2}\frac{\left(e^{(j+1)I\psi}-2e^{jI\psi}+e^{(j-1)I\psi}\right)}{\frac{1}{10}e^{(j+1)I\psi}+e^{jI\psi}+\frac{1}{10}e^{(j-1)I\psi}} \end{aligned}\right\} \tag{23}$$

where $I=\sqrt{-1}$, and corresponding expressions for the compact derivative operators. This is the standard von Neumann idea: analyze how one sinusoidal mode behaves under the scheme. Instead of testing every possible error shape, you test a single wave component: wavelength controlled by ψ, amplitude controlled by E^n. Because any reasonable error can be decomposed into Fourier modes, this gives a very powerful stability test. In practical simulations, numerical errors often appear as small oscillations or ripples. Analysis: Von Neumann asks: if a ripple appears in the solution, will the method damp it, preserve it, or magnify it? That is exactly what matters in convection–diffusion problems, where oscillations can easily become nonphysical.

Substituting these transformations (23) into the first stage of the proposed scheme (21) yields:

$$E^P = E^n e^{a_1\Delta t} + \frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left\{\frac{1}{4}\left(\frac{3\mu_1 I sin\psi}{\Delta y(cos\psi+2)}+\frac{12\mu_2(cos\psi-1)}{(\Delta y)^2(cos\psi+5)}\right)-a_1\right\}E^n \tag{24}$$

By writing Equation (24) in the form of:

$$E^P = (\gamma_1 + I\gamma_2)\,E^n \tag{25}$$

where $\gamma_1 = e^{a_1\Delta t}+\frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left[\frac{3\mu_2(cos\psi-1)}{(\Delta y)^2(cos\psi+5)}-a_1\right]$ and $\gamma_2=\frac{\left(e^{a_1\Delta t}-1\right)}{a_1}\left\{\frac{3\mu_1 I sin\psi}{\Delta y(cos\psi+2)}\right\}$.

This says: after the predictor stage, the amplitude of a Fourier mode is multiplied by a complex number. γ_1 is the real contribution, γ_2 is the imaginary contribution. The imaginary part mainly reflects phase shifts or wave-propagation effects. The real part mainly reflects growth or damping. This is physically meaningful because the convection term (μ_1) tends to move or shift waves, the diffusion term (μ_2) tends to dampen them. So, this stage already captures how your scheme reacts to transport and smoothing.

Similarly, substituting transformation (23) into the second stage of the scheme (22)

gives

$$E^{n+1} = aE^n + bE^P + c\Delta t \left\{ \frac{3\mu_1 I sin\psi}{\Delta y(cos\psi + 2)} + \frac{12\mu_2(cos\psi - 1)}{(\Delta y)^2(cos\psi + 5)} \right\} E^P \tag{26}$$

Equation (26) can be written as

$$E^{n+1} = aE^n + (\gamma_3 + I\gamma_4)\, E^P \tag{27}$$

Where $\gamma_3 = c\Delta t \left\{ \frac{12\mu_2(cos\psi - 1)}{(\Delta y)^2(cos\psi + 5)} \right\} + b$ and $\gamma_4 = c\Delta t \left\{ \frac{3\mu_1 I sin\psi}{\Delta y(cos\psi + 2)} \right\}$.
Substituting Equation (25) into Equation (27) results in

$$E^{n+1} = aE^n + (\gamma_3 + I\gamma_4)(\gamma_1 + I\gamma_2)\, E^n = (\gamma_5 + I\gamma_6)\, E^n \tag{28}$$

where $\gamma_5 = a + \gamma_1\gamma_3 - \gamma_2\gamma_4,\ \gamma_6 = \gamma_1\gamma_4 - \gamma_2\gamma_3$.

Thus the amplification factor is written as:

$$G = \frac{E^{n+1}}{E^n} = \gamma_5 + I\gamma_6 \tag{29}$$

The amplification factor is the heart of stability analysis. It tells you what one time step does to a disturbance: if $\mid G \mid < 1$, the disturbance shrinks, if $\mid G \mid = 1$, it stays the same size, if $\mid G \mid > 1$, it grows. In a real simulation, if small disturbances grow at every step, the computed temperature, velocity, or eddy viscosity may become oscillatory, blow up, or lose physical meaning.

The stability condition is written as:

$$|G|^2 \leq 1 \tag{30}$$

If parameter μ_1 and μ_2 in Equation (13) and step sizes, $\Delta t,\ \Delta y$, and $(\Delta y)^2$ satisfy inequality (30), the scheme will remain stable; otherwise, an unstable solution can be expected. So, if some large values of parameters μ_1 and μ_2 are chosen, then the step size Δt should be reduced to meet condition (30). This means your scheme is stable only when the chosen parameters satisfy a balance between convection strength μ_1, diffusion strength μ_2, time step Δt, mesh width Δy. That is exactly what engineers need to know. If transport or diffusion is strong, the time step may need to be smaller. For example, in fast advection, too large a time step causes oscillations; in strong diffusion, the method may still require a controlled time step, and in turbulent transport, large effective diffusion or eddy viscosity can change the safe step-size range. So, inequality is not just mathematical; it is a practical guide for choosing discretization parameters.

Our conclusion says that if μ_1, μ_2, Δt, Δy, and $(\Delta y)^2$ satisfy the stability inequality, the method remains stable; otherwise, instability can occur. This is the discrete analog of a CFL-type restriction. It says strong convection and large time steps can be dangerous; strong diffusion and coarse space-time balance can also affect stability; stability is conditional, not automatic. This is extremely important in simulations of mixed convection flow, thermal boundary layers, nanofluid transport, and turbulence models. Suppose the eddy viscosity or thermal diffusivity becomes large in some regions. Then the effective transport strength increases, and our step size may need to be adjusted.

3.1. Why von Neumann analysis is useful for real problems

For linear PDEs, it gives exact stability conditions, whereas for non-linear PDEs, it provides only an estimate after linearization. For non-linear real-world systems, the coefficients often depend on the solution itself. So, you usually freeze or linearize the non-linear terms and then analyze the resulting linear model. This is common in turbulent flow, nanofluid transport, non-linear heat transfer, and reaction–diffusion systems. So, even though the exact full non-linear stability may be harder, this analysis still provides an important practical estimate of the scheme's safe operating region.

3.2. Stability region comparison with traditional methods

Figure 1 compares the absolute stability regions of the proposed modified Runge-Kutta scheme with those of the Forward Euler, classical second-order Runge-Kutta (RK2), and classical fourth-order Runge-Kutta (RK4) methods in the complex plane $z = \lambda\Delta t$. The stability boundary is defined by $\mid R(z) \mid = 1$, where $R(z)$ denotes the amplification factor of the corresponding time-integration scheme. It can be observed that the Forward Euler method possesses the smallest stability region, indicating stronger time-step restrictions. In contrast, the RK4 method exhibits the largest region and therefore allows greater flexibility in selecting Δt. The proposed modified scheme overlaps with the classical RK2 method on the linear test equation, which shows that both methods share the same absolute stability region in this graphical comparison. Along with the negative real axis, both the modified scheme and RK2 remain stable approximately in the interval $[-2, 0]$, while RK4 extends farther to the left. This result indicates that the advantage of the proposed method does not arise from enlarging the classical absolute stability region, but rather from its improved predictor-corrector structure and adaptive time-stepping strategy, which enhance numerical accuracy and practical efficiency in solving the target convection-diffusion and turbulent nanofluid problems.

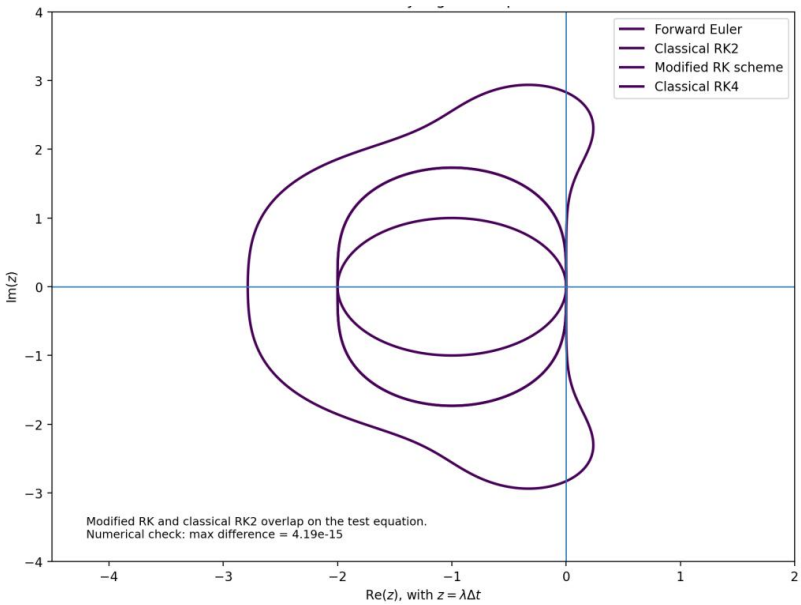

Figure 1. Absolute stability region comparison of the Forward Euler, classical second-order Runge-Kutta (RK2), and classical fourth-order Runge-Kutta (RK4) in the complex plane $z = \lambda\Delta t$.

3.3. Convergence analysis

Stability analysis for scalar convection-diffusion is provided. Now, convergence for a system of convection-diffusion problems is provided. For this purpose, consider the following system of convection-diffusion equations.

$$\frac{\partial P}{\partial t} = A_1 \frac{\partial P}{\partial y} + A_2 \frac{\partial^2 P}{\partial y^2} + A_3 P \tag{31}$$

Where P is a vector and $A_1,\ A_2$ and A_3 are matrices. If the given equation is non-linear, then it can be linearized and converted into a vector matrix Equation (31). Application of the proposed scheme on Equation (31) yields

$$P^P = P^n e^{a_1 \Delta t} + \frac{\left(e^{a_1 \Delta t} - 1\right)}{a_1} \left[\frac{1}{4} \left\{ A_1 \delta P^n + A_2 \delta^2 P^n + A_3 P^n \right\} - a_1 P^n \right] \tag{32}$$

$$P^{n+1} = a P^n + b P^P + c \Delta t \left[A_1 \delta P^P + A_2 \delta^2 P^P + A_3 P^P \right] \tag{33}$$

This is necessary because real turbulent and thermal-flow models are not scalar. They are usually systems: velocity, temperature, concentration, turbulence variables such as k and ω, sometimes solid-phase and fluid-phase temperatures in local thermal non-equilibrium (LTNE). This step is what makes our numerical method relevant to practical Multiphysics problems. Real engineering models almost always involve coupled equations, not just one unknown.

Theorem 1. *Consider the linearized vector convection–diffusion system*

$$\frac{\partial P}{\partial t} = A_1 \frac{\partial P}{\partial y} + A_2 \frac{\partial^2 P}{\partial y^2} + A_3 P \,,$$

and let the numerical solution be generated by the proposed scheme (32)–(33). Assume that the exact solution is sufficiently smooth and that the compact difference operators δ and δ^2 are bounded in the infinity norm. Define

$$\alpha_1 = e^{a_1 \Delta t} + \frac{\mid e^{a_1 \Delta t} - 1 \mid}{a_1} \left[\frac{1}{4} \left(\parallel A_1 \delta \parallel_\infty + \parallel A_2 \delta^2 \parallel_\infty + \parallel A_3 \parallel_\infty \right) + a_1 \right],$$

and

$$\alpha_2 = a + b\alpha_1 + c\Delta t \left(\parallel A_1 \delta \parallel_\infty + \parallel A_2 \delta^2 \parallel_\infty + \parallel A_3 \parallel_\infty \right) \alpha_1.$$

If $\mid \alpha_2 \mid < 1$, *then the proposed scheme converges conditionally, and the global error satisfies*

$$E_r^{(N_t)} \leq \frac{1 - \alpha_2^{N_t}}{1 - \alpha_2} S \left(O(\Delta t)^2,\ (\Delta y)^4 \right) .$$

Proof. The proof of this Theorem is initiated by considering the error equation for the first stage of the proposed scheme as:

$$E_r^P = E_r^n e^{a_1 \Delta t} + \frac{\left(e^{a_1 \Delta t} - 1\right)}{a_1} \left[\frac{1}{4} \left\{ A_1 \delta E_r^n + A_2 \delta^2 E_r^n + A_3 E_r^n \right\} - a_1 E_r^n \right] \tag{34}$$

□

We define the stage error E_r^P and the updated error E_r^{n+1} by comparing the numerical solution with the exact or reference discrete solution. Now the question is not just "does one Fourier mode grow?" but does the total numerical error remain controlled as the computation proceeds? This is the convergence question. A method may appear stable in one mode, but what really matters in a full computation is whether the overall solution converges to the true solution as the mesh is refined. That is what convergence guarantees.

Applying $\|\cdot\|_\infty$ on both sides of Equation (34) results in

$$E_r^P \leq E_r^n e^{a_1\Delta t} + \frac{\left|e^{a_1\Delta t}-1\right|}{a_1}\left[\frac{1}{4}\left\{\|A_1\delta\|_\infty + \left\|A_2\delta^2\right\|_\infty + \|A_3\|_\infty\right\} + a_1\right] E_r^n \quad (35)$$

Let $\alpha_1 = e^{a_1\Delta t} + \frac{\left|e^{a_1\Delta t}-1\right|}{a_1}\left[\frac{1}{4}\left\{\|A_1\delta\|_\infty + \left\|A_2\delta^2\right\|_\infty + \|A_3\|_\infty\right\} + a_1\right]$

Then inequality (35) can be written as

$$E_r^P \leq \alpha_1 E_r^n \quad (36)$$

This says that after the predictor stage, the error is at most scaled by some factor α_1. The factor α_1 depends on the time step, the matrix norms, the compact derivative operators, and the chosen parameter a_1. This is important because, in coupled models, the interactions between equations can amplify errors even if each equation alone appears harmless. For example, momentum errors may feed into turbulence variables, turbulence errors may feed into eddy viscosity, and eddy viscosity errors may feed back into the momentum and thermal equations. So, bounding the matrix-coupled error is essential for real Multiphysics simulations.

Similarly, consider the error equation for the second stage of the proposed scheme in the form of:

$$\mathrm{E_r^{n+1}} = aE_\mathrm{r}^n + bE_\mathrm{r}^P + c\Delta t\left[A_1\delta E_\mathrm{r}^P + A_2\delta^2 E_\mathrm{r}^P + A_3 E_\mathrm{r}^P\right] \quad (37)$$

Applying $\|\cdot\|_\infty$ on both sides of Equation (37) gives

$$E_r^{n+1} \leq aE_r^n + bE_r^P + c\Delta t\left[\|A_1\delta\|_\infty + \left\|A_2\delta^2\right\|_\infty + \|A_3\|_\infty\right] E_r^P \quad (38)$$

Using inequality (36) in inequality (37) results in

$$E_r^{n+1} \leq aE_r^n + b\alpha_1 E_r^n + c\Delta t\left[\|A_1\delta\|_\infty + \left\|A_2\delta^2\right\|_\infty + \|A_3\|_\infty\right]\alpha_1 E_r^n \quad (39)$$

Let $\alpha_2 = a + b\alpha_1 + c\Delta t\left[\|A_1\delta\|_\infty + \left\|A_2\delta^2\right\|_\infty + \|A_3\|_\infty\right]\alpha_1$,

Then inequality (39) can be expressed as

$$E_r^{n+1} \leq \alpha_2 E_r^n + S\left(O(\Delta t)^2,\ (\Delta y)^4\right)) \quad (40)$$

This is the core convergence inequality. It says the new error is made of two parts: the propagated old error, scaled by α_2, and fresh local truncation error from the scheme. The truncation term reflects our claimed orders: second-order in time and fourth-order in space. This tells you that if the propagated error stays controlled and

the local truncation errors go to zero as the mesh is refined, then the whole scheme converges. That is what engineers want from a method: a finer mesh, a smaller step, and a more accurate answer.

Let $n = 0$ in (40) then.

$$E_r^1 \leq \alpha_2 E_r^0 + S\left(O(\Delta t)^2,\ (\Delta y)^4\right) \tag{41}$$

Since $E_r^0 = 0$ because of the initial error; therefore, inequality (41) can be written as:

$$E_r^1 \leq S\left(O(\Delta t)^2,\ (\Delta y)^4\right) \tag{42}$$

Let $n = 1$ in Equation (40), then it gives

$$E_r^2 \leq \alpha_2 E_r^1 + S\left(O(\Delta t)^2,\ (\Delta y)^4\right) \leq (1+\alpha_2) S\left(O(\Delta t)^2,\ (\Delta y)^4\right) \tag{43}$$

Similarly, if this is continued, then for finite N_t

$$E_r^{N_t} \leq \left(1 + \alpha_2 + \ldots + \alpha_2^{N_t}\right) S\left(O(\Delta t)^2,\ (\Delta y)^4\right) = \left(\frac{1-\alpha_2^{N_t}}{1-\alpha_2}\right) S\left(O(\Delta t)^2,\ (\Delta y)^4\right) \tag{44}$$

This shows how local errors accumulate over time. If $\mid \alpha_2 \mid < 1$, the geometric series is bounded, so the total error remains under control. This is the difference between a simulation whose errors remain small over long time intervals and one whose errors accumulate, rendering the result unreliable. For long transient simulations, startup heating, transient cooling, pollutant dispersion, and evolving turbulent layers, this matters a lot.

If $N_t \to \infty$, then the series $1 + \alpha_2 + \ldots + \alpha_2^{N_t} + \ldots$ becomes an infinite geometric series that will converge if $|\alpha_2| < 1$. So, if the norms of the product of the matrices of the compact scheme and the time step size satisfy the convergence condition, the scheme will converge for the vector-matrix Equation (31).

This is the conditional convergence result. The method converges only if the time steps and operator norms satisfy a restriction. So, the method is not unconditionally convergent; it converges only in an admissible parameter regime. This is completely realistic. In practical CFD and transport simulations, one never expects an explicit method to work with arbitrarily large time steps.

3.4. Real meaning of analysis

The von Neumann analysis shows that the proposed predictor-corrector compact scheme remains stable only when the combined effects of the convection and diffusion strengths, and the temporal-spatial step sizes, satisfy a bounded amplification-factor condition. This means that the method suppresses nonphysical growth of numerical disturbances and can therefore be used reliably for transient transport problems only within an admissible parameter regime. For coupled systems, the convergence proof further shows that the accumulated numerical error remains bounded provided the matrix-operator norm condition $\mid \alpha_2 \mid < 1$ holds, which is particularly relevant for multiphysics models such as turbulent nanofluid flow under thermal non-equilibrium.

4. Problem formulation

In this study, the unsteady turbulent boundary layer flow of a nanofluid over a solid surface, incorporating $k-\omega$ turbulence modeling, local thermal non-equilibrium effects, viscous dissipation, mixed convection, and an improved thermal conductivity model based on Xue's formulation, is investigated. The purpose of the study is to examine the coupled behavior of momentum, turbulence, and heat transfer characteristics under realistic physical assumptions.

Physical model and assumptions

The mathematical formulation is developed under the following assumptions. The flow is assumed to be unsteady, incompressible, and turbulent over a heated, moving surface. The flow is unsteady, incompressible, and turbulent over a heated moving surface. The fluid and nanoparticle phases are treated as interacting thermal continua, while a uniform nanoparticle volume fraction is assumed throughout the computational domain. The thermophysical properties of the constituent phases are taken as constant unless their dependence is introduced explicitly through the adopted effective nanofluid-property relations. Let x^* and y^* denote the streamwise and wall-normal coordinates, respectively. The nanofluid consists of a base fluid that contains uniformly dispersed solid nanoparticles.

The flow satisfies the boundary-layer approximation, and diffusion is considered only in the transverse direction. $k-\omega$ is used for turbulence effects, providing a robust explanation of near-wall behavior. The turbulent kinetic energy k and specific dissipation rate ω determine the eddy viscosity, which augments the molecular momentum diffusivity in the mean-flow equation. Turbulent heat transport is represented using the gradient-diffusion hypothesis with a turbulent Prandtl number P_{rt}. Consequently, the turbulent heat flux is incorporated through an effective thermal diffusivity rather than introducing an additional transport equation for it.

The local thermal non-equilibrium (LTNE) framework is used to model nanofluids, allowing distinct energy equations for the base fluid and nanoparticles. For particle-fluid interaction, the Xue model is considered. Local thermal non-equilibrium is assumed between the base-fluid and nanoparticle phases. Accordingly, separate temperature fields, θ_f and θ_s, are governed by individual energy equations. Energy exchange between the phases is represented by an interphase heat-transfer term proportional to the local temperature difference. The associated coupling parameters, Γ_f and Γ_s, characterize the strength of thermal interaction in the fluid and nanoparticle energy equations, respectively. In the limit of sufficiently strong interphase heat exchange, the two temperature fields approach one another, and the system tends toward local thermal equilibrium.

Viscous dissipation effects are considered in the energy equation used for heat transfer in a fluid. Viscous dissipation is retained in the fluid-energy equation to account for irreversible conversion of kinetic energy into internal energy caused by viscous and turbulent shear. Its contribution is characterized by the Eckert number E_c. This effect is retained because the moving heated surface and turbulent shear can produce appreciable velocity gradients within the near-wall region.

We model a fluid flowing over a heated wall or plate, where the flow changes over time, turbulence is present, and nanoparticles modify heat transfer. This type of setup appears in cooling hot metal plates, solar thermal collectors, battery thermal management, heat exchangers, coating and drying processes, and high-performance cooling systems using nanofluids. These assumptions simplify the full Navier–Stokes and heat-transfer problem into a tractable model while retaining the essential physics. Each assumption has practical meaning: Boundary-layer approximation: valid near surfaces where strong wall effects dominate, such as plates, ducts, and channels. Transverse diffusion only: common in high Reynolds number wall-bounded flows. $k-\omega$ turbulence model: suitable for near-wall behavior, which is crucial in engineering heat transfer. LTNE: important when the fluid and particles do not instantly reach the same temperature. Viscous dissipation: relevant when velocity gradients are strong and frictional heating matters. Xue conductivity model: used to represent how nanoparticles enhance effective heat conduction.

Figure 2 illustrates the physical geometry and thermal-flow configuration considered in the present study. A heated vertical solid surface moves upward with constant velocity u_w, generating an unsteady turbulent nanofluid boundary layer along the wall. The coordinate x^* is taken along the plate and y^* is normal to it. Because the flow is turbulent, the momentum transport is influenced by the turbulent variables k^* and ω^*, which determine the eddy viscosity ν_t^*. The nanofluid is modeled under LTNE conditions, so the base fluid temperature T_f and nanoparticle temperature T_s are treated separately rather than assuming a common thermal equilibrium state. These two phases exchange heat through the interphase coupling term $h_{sf}(T_s - T_f)$. The model also accounts for buoyancy through the mixed convection term $g\beta_T(T_f - T_\infty)$, viscous dissipation in the fluid energy equation, and enhanced thermal conductivity through Xue's nanofluid model. Thus, the schematic summarizes the coupled interactions among wall motion, turbulence, nanoparticle-enhanced heat transfer, and two-phase thermal non-equilibrium, which form the basis of the governing equations.

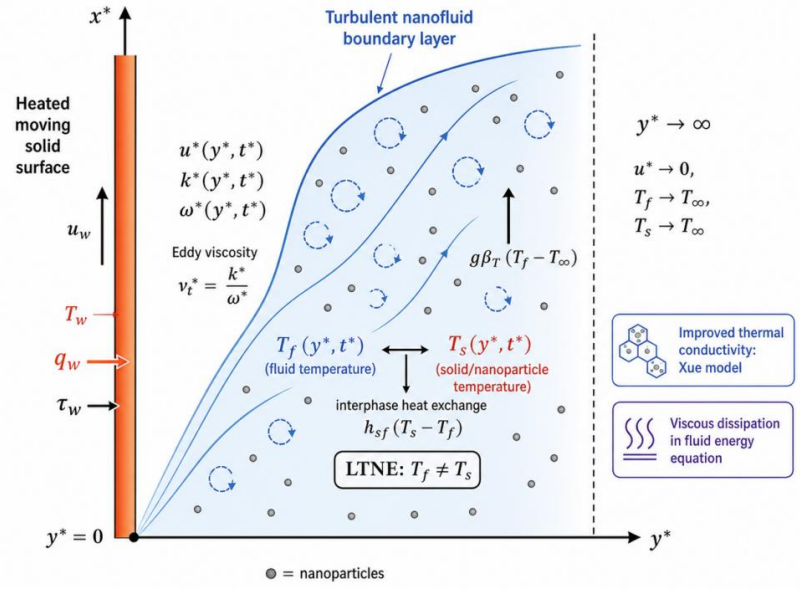

Figure 2. Schematic of the unsteady $k-\omega$ turbulent nanofluid boundary-layer flow over a heated moving solid surface under local thermal non-equilibrium conditions.

Under these assumptions, governing equations for mass, momentum, turbulence, and energy are formulated as follows:

$$\frac{\partial u^*}{\partial t^*} = \frac{\partial}{\partial y^*}\left[\left(\nu_{nf} + \nu_t^*\right)\frac{\partial u^*}{\partial y^*}\right] + g\beta_T\left(T_t - T_\infty\right) \tag{45}$$

Equation (45) tracks how the mean flow velocity evolves under viscosity, turbulence, and buoyancy.

$$\frac{\partial k^*}{\partial t^*} = \frac{\partial}{\partial y^*}\left[\left(\nu_{nf} + \sigma_k\nu_t^*\right)\frac{\partial u^*}{\partial y^*}\right] + \nu_t^*\left(\frac{\partial u^*}{\partial y^*}\right)^2 - \beta^* k^*\omega^* \tag{46}$$

Equation (46) tracks how turbulent kinetic energy is diffused, generated by shear, and dissipated.

$$\frac{\partial \omega^*}{\partial t^*} = \frac{\partial}{\partial y^*}\left[\left(\nu_{nf} + \sigma_\omega\nu_t^*\right)\frac{\partial \omega^*}{\partial y^*}\right] + \frac{\gamma\omega^*}{k^*}P_k^* - \beta^*\omega^{*2} \tag{47}$$

Equation (47) tracks how the specific dissipation rate evolves, which determines the turbulence scale and eddy viscosity.

$$\rho_f C_{Pf}\frac{\partial T_f}{\partial t^*} = \frac{\partial}{\partial y^*}\left[\left(k_{nf} + \frac{\rho_f C_{Pf}\nu_t^*}{P_{\gamma_t}}\right)\frac{\partial T_f}{\partial y^*}\right] + h_{sf}\left(T_s - T_f\right) + \mu_t^*\left(\frac{\partial u^*}{\partial y^*}\right)^2 \tag{48}$$

Equation (48) tracks the fluid-phase temperature, including conduction, turbulent heat transport, particle-fluid heat exchange, and viscous heating.

$$\rho_s C_{ps}\frac{\partial T_s}{\partial t^*} = \frac{\partial}{\partial y^*}\left[k_s\frac{\partial T_s}{\partial y^*}\right] + h_{sf}\left(T_f - T_s\right) \tag{49}$$

Equation (49) tracks the solid-phase temperature, accounting for solid conduction and heat transfer to the fluid.

We went through each of the governing equations from (45) to (49) and described them individually here.

Momentum Equation (45): $\frac{\partial u^*}{\partial t^*}$ is the is the local time rate of change of velocity. It tells how the streamwise velocity changes with time at a fixed location. If this term is large, the flow is strongly unsteady, $\frac{\partial}{\partial y^*}\left[\left(\nu_{nf} + \nu_t^*\right)\frac{\partial u^*}{\partial y^*}\right]$ represents the momentum diffusion term. It represents the transport of momentum in the wall-normal direction due to both the nanofluid's molecular viscosity and the turbulent viscosity. ν_{nf} this is the kinematic viscosity of the nanofluid.

It represents the ordinary viscous resistance of the nanofluid, ν_t^* represents the eddy viscosity or turbulent viscosity. It represents the additional effective momentum diffusion caused by turbulence, $\frac{\partial u^*}{\partial y^*}$ is the velocity gradient in the wall-normal direction. It measures how rapidly the velocity changes away from the wall, and the outer derivative $\frac{\partial}{\partial y^*}[\cdots]$ shows how the momentum flux changes with distance from the wall. So, the whole term describes the redistribution of momentum inside the boundary layer. $g\beta_T\left(T_t - T_\infty\right)$ is the buoyancy force term where g is the acceleration due to gravity, β_T is the thermal expansion coefficient. It measures how sensitive the fluid density is to temperature changes. $\left(T_t - T_\infty\right)$ is the temperature difference between

the local thermal state and the ambient temperature. This drives density variations and, therefore, buoyancy.

Turbulent kinetic energy Equation (46): $\frac{\partial k^*}{\partial t^*}$ is the time rate of change of turbulent kinetic energy. It tells how turbulence intensity evolves with time. If this term is large, turbulence is developing or decaying rapidly. $\frac{\partial}{\partial y^*}\left[(\nu_{nf}+\sigma_k\nu_t^*)\frac{\partial k^*}{\partial y^*}\right]$ represents the diffusion of turbulent kinetic energy, where σ_k is a model coefficient controlling how turbulence diffuses k^*, $(\nu_{nf}+\sigma_k\nu_t^*)$ is the effective diffusivity of turbulent kinetic energy.

It includes both molecular and turbulent contributions. $\frac{\partial k^*}{\partial y^*}$ is the gradient of turbulent kinetic energy across the boundary layer. This term spreads turbulence energy from one region to another. It helps model how turbulence generated near the wall or shear layer is redistributed through the flow. $\nu_t^*\left(\frac{\partial u^*}{\partial y^*}\right)^2$ is the production term of turbulent kinetic energy, where ν_t^*: turbulent viscosity, $\left(\frac{\partial u^*}{\partial y^*}\right)^2$: square of the mean shear rate. Strong velocity shear generates turbulence. This is important near walls, in jets, and in sheared thermal flows where the motion creates eddies and mixing. $\beta^*k^*\omega^*$ is the dissipation term for turbulent kinetic energy, where β^* is the turbulence-model coefficient, $k^*\omega^*$ represents the rate at which turbulent kinetic energy is destroyed. Turbulence is not only produced; it also decays. Without this term, turbulence energy would grow to unrealistic levels. This term keeps the turbulence model physically balanced.

Specific dissipation rate Equation (47): $\frac{\partial \omega^*}{\partial t^*}$ is the time rate of change of specific dissipation rate. It tracks how the turbulence frequency or dissipation scale evolves, $\frac{\partial}{\partial y^*}\left[(\nu_{nf}+\sigma_\omega\nu_t^*)\frac{\partial \omega^*}{\partial y^*}\right]$ is the diffusion term for ω^*, where σ_ω is a model coefficient controlling the turbulent diffusion of ω^*, $\frac{\partial \omega^*}{\partial y^*}$ is the wall-normal gradient of the specific dissipation rate. This term spreads the dissipation-frequency information across the flow, $\frac{\gamma\omega^*}{k^*}P_k^*$ is the production term for ω^* where γ is the turbulence model constant, $\frac{\omega^*}{k^*}$ links the local turbulence scale to the current energy level, P_k^* is the turbulent kinetic energy production term, defined as $P_k^* = \nu_t^*\left(\frac{\partial u^*}{\partial y^*}\right)^2$. This term increases ω^* when turbulence production is strong. It helps capture how turbulence structure changes near the wall and in strong shear regions, and $-\beta^*\omega^{*2}$ is the destruction term for ω^*. It prevents the unlimited growth of ω^*. This is essential for the stability of turbulence models and for the realistic prediction of near-wall turbulence behavior.

Fluid-phase energy Equation (48): $\rho_f C_{Pf}\frac{\partial T_f}{\partial t^*}$ is the transient storage of thermal energy in the fluid phase, ρ_f is the fluid density, C_{Pf} is the fluid specific heat, $\frac{\partial T_f}{\partial t^*}$ is the time variation of the fluid temperature. This measures how quickly the fluid temperature changes with time, $\frac{\partial}{\partial y^*}\left[\left(k_{nf}+\frac{\rho_f C_{Pf}\nu_t^*}{Pr_t}\right)\frac{\partial T_f}{\partial y^*}\right]$ is the thermal diffusion term in the fluid phase, where k_{nf} is the effective thermal conductivity of the nanofluid, $\frac{\rho_f C_{Pf}\nu_t^*}{Pr_t}$ is the turbulent thermal diffusivity term.

It accounts for additional heat transport due to turbulence. Pr_t is the turbulent Prandtl number.

It connects turbulent momentum transport with turbulent heat transport $\frac{\partial T_f}{\partial y^*}$ is the temperature gradient in the fluid phase. This term represents both ordinary conduction and turbulence-enhanced heat transport. In turbulent cooling applications, this is one

of the most important heat-transfer terms, $h_{sf}(T_s - T_f)$ is the interphase heat exchange term between particles and fluid, h_{sf} is the interphase heat transfer coefficient, $(T_s - T_f)$ is the temperature difference between the solid nanoparticle phase and the fluid phase. If the particles are hotter than the fluid, heat flows from particles to fluid; if colder, the reverse happens. This is the core LTNE coupling term.

It captures delayed heat exchange between the two phases, $\mu_t^* \left(\frac{\partial u^*}{\partial y^*}\right)^2$ represents the viscous dissipation term where μ_t^* is the effective turbulent dynamic viscosity contribution, $\left(\frac{\partial u^*}{\partial y^*}\right)^2$ is the square of the shear rate. Mechanical energy is converted to heat due to friction. This is important in high-speed flows, highly sheared boundary layers, narrow channels, and systems where frictional heating is non-negligible.

Solid-phase energy Equation (49): $\rho_s C_{Ps} \frac{\partial T_s}{\partial t^*}$ is the thermal energy storage term for the solid nanoparticle phase, where ρ_s is the nanoparticle density, C_{Ps} is the nanoparticle specific heat, and $\frac{\partial T_s}{\partial t^*}$ is the time variation of solid-phase temperature. This tells how quickly the particle temperature changes in time, $\frac{\partial}{\partial y^*}\left[k_s \frac{\partial T_s}{\partial y^*}\right]$ is the thermal conduction term in the solid phase, k_s represents the thermal conductivity of the nanoparticle phase, $\frac{\partial T_s}{\partial y^*}$ is the spatial gradient of solid-phase temperature. Heat is conducted through the solid nanoparticle thermal network. This term matters when particle conductivity is significantly different from fluid conductivity, $h_{sf}(T_f - T_s)$ is the heat exchange from fluid to solid phase. It is the opposite-direction counterpart of the coupling term in the fluid equation. Together, the two coupled terms in Equations (48) and (49) ensure energy exchange between phases is modeled consistently.

Together, these five equations describe a system in which the velocity field controls shear, shear generates turbulence, turbulence increases effective transport, temperature differences generate buoyancy, particles and the fluid exchange heat separately, and friction can generate extra heating. That is why this model is relevant to: turbulent nanofluid cooling, thermal boundary-layer control, high-performance heat exchangers, energy devices with particle-enhanced coolants, and LTNE heat-transfer systems.

Apply initial and boundary conditions: Subject to the initial and boundary conditions.

$$\left.\begin{array}{r} u^* = 0,\ k^* = 1 \times 10^{-7} u_w^2, \omega^* = \frac{10 u_w}{L},\ T_f = 0, T_s = 0\ when\ t^* = 0 \\ u^* = u_w, k^* = 0, \omega^* = 100 \times \frac{u_w}{L}, T_f = T_w, T_s = T_w\ when\ y^* = 0 \\ u^* \to 0, \frac{\partial k^*}{\partial y^*} \to 0, \frac{\partial \omega^*}{\partial y^*} \to 0,\ T_f \to T_\infty,\ T_s \to T_\infty when\ y^* \to \infty \end{array}\right\} \quad (50)$$

At $t^* = 0$, the flow starts from rest with specified initial turbulence and temperatures.

At the wall $y^* = 0$, the velocity and temperatures are fixed, and turbulence quantities are prescribed. Far from the wall, velocity and temperature return to ambient values. These conditions define the startup state, the wall forcing, and the far-field behavior. This models a practical situation, such as suddenly heating a wall, starting a moving plate, or turning on a thermal system, and observing the transient response.

Where $P_k^* = \nu_t^* \left(\frac{\partial u^*}{\partial y^*}\right)^2, \nu_t^* = \frac{k^*}{\omega^*}$ are, respectively, turbulent kinetic energy production and eddy viscosity (or turbulent viscosity). Also, h_{sf} is the interphase heat

transfer coefficient, $T_s - T_f$ is the energy exchange with nanoparticles, ρ_f, c_{Pf} are the density and specific heat of the base fluid, respectively. $\rho_s,\ c_{Ps}$ and k_s are nanoparticle density, specific heat, and conductivity, respectively, T_f and T_s are the temperatures of the base fluid and nanoparticle, respectively. $P_{\gamma t}$ is the turbulent Prandtl number, σ_k and σ_w are dimensionless coefficients, g is gravity, β_T is the thermal expansion coefficient.

Dimensionless variables: Non-dimensionalization reduces the number of free parameters and reveals the true controlling physics. This is important because it allows the same equations to describe many different physical systems, provided they have similar dimensionless parameters. For example, one simulation can represent: a laboratory plate, a cooling fin, or an industrial heat exchanger, if their Reynolds and Prandtl numbers are similar. Using the following transformations

$$y = \frac{y^*}{L}, u = \frac{u^*}{u_w}, t = \frac{u_w t^*}{L}, k = \frac{k^*}{u_w^2}, \omega = \frac{\omega^* L}{u_w}, \theta_f = \frac{T_f - T_\infty}{T_w - T_\infty},\ \theta_s = \frac{T_s - T_\infty}{T_w - T_\infty} \tag{51}$$

Governing Equations (41)–(46) are transformed to the following dimensionless system of equations

$$\frac{\partial u}{\partial t} = \frac{1}{R_e}\frac{\partial}{\partial y}\left[\left(\Lambda_\nu + R_e\nu_t\right)\frac{\partial u}{\partial y}\right] + \frac{G_{\gamma T}}{R_e^2}\theta_f \tag{52}$$

$$\frac{\partial k}{\partial t} = \frac{1}{R_e}\frac{\partial}{\partial y}\left[\left(\Lambda_\nu + \sigma_k R_e\nu_t\right)\frac{\partial k}{\partial y}\right] + P_k - \beta^* k\omega \tag{53}$$

$$\frac{\partial \omega}{\partial t} = \frac{1}{R_e}\frac{\partial}{\partial y}\left[\left(\Lambda_\nu + \sigma_\omega R_e\nu_t\right)\frac{\partial \omega}{\partial y}\right] + \frac{\gamma\omega}{k}P_k - \beta\omega^2 \tag{54}$$

$$\frac{\partial \theta_f}{\partial t} = \frac{1}{R_e}\frac{\partial}{\partial y}\left[\left(\frac{1}{\mathrm{Pr}_f}\frac{k_{n_f}}{k_f} + \frac{R_e}{P_{\gamma_t}}\right)\frac{\partial \theta_f}{\partial y}\right] + \Gamma_f\left(\theta_s - \theta_f\right) + E_c\nu_t\left(\frac{\partial u}{\partial y}\right)^2 \tag{55}$$

$$\frac{\partial \theta_s}{\partial t} = \frac{1}{R_e}\frac{1}{\mathrm{Pr}_s}\frac{\partial^2 \theta_s}{\partial y^2} + \Gamma_s\left(\theta_f - \theta_s\right) \tag{56}$$

and dimensionless initial and boundary conditions are expressed in the form of

$$\left.\begin{aligned} u = 0, k = 1\times 10^{-7}, \omega = 10, \theta_f = 0, \theta_s = 0\ when\ t = 0 \\ u = u_w, k = 0, \omega = 100, \theta_f = 1, \theta_s = 1\ for\ y = 0 \\ u \to 0,\ \frac{\partial k}{\partial y} \to 0,\ \frac{\partial \omega}{\partial y} \to 0, \theta_f \to 0,\ \theta_s \to 0\ when\ y \to \infty \end{aligned}\right\} \tag{57}$$

Where R_e is the Reynolds number, Pr_f is the Prandtl number of the base fluid, Pr_s is the Prandtl number of nanoparticles, E_c is the Eckert number, B_i is the Biot number, and $G_{\gamma T}$ is the thermal Grashof number, Γ_f is a fluid-phase coupling parameter, Γ_s is a solid-phase coupling parameter, which is defined as:

$$R_e = \frac{\nu}{u_w L}, \mathrm{Pr}_f = \frac{\nu}{\alpha_f} = \frac{\nu C_{Pf}\rho_f}{k_f},\ \mathrm{Pr}_s = \frac{\nu}{d_s} = \frac{\nu C_{Ps}\rho_s}{k_s},\ E_c = \frac{u_w^2}{C_{Pf}(T_w - T_\infty)},$$
$$B_i = \frac{h_{s_f} L}{k_t},\ G_{\gamma_T} = \frac{L^3 g\beta_T(T_w - T_\infty)}{\nu^2}, \Gamma_f = \frac{h_{sf} L}{\rho_f c_{pf} u_w}, \Gamma_s = \frac{h_{sf} L}{\rho_s c_{ps} u_w}$$

and

$$\Lambda_\nu = \frac{(1-\phi)^{-2.5}}{1-\phi+\phi\left(\frac{\rho_s}{\rho_f}\right)}, \frac{k_{nf}}{k_f} = \frac{1+2\phi\beta_{Xue}}{1-\phi\beta_{Xue}}$$

where $\beta_{Xue} = \frac{k_s-k_f}{k_s+2k_f}$ is Xue's thermal conductivity model.

Physical meaning of each parameter: Reynolds number R_e measures the relative importance of inertial effects compared with viscous effects. Higher R_e usually means stronger convection and a greater tendency toward turbulence. Prandtl numbers Pr_f, Pr_s compare momentum diffusivity with thermal diffusivity. They tell how quickly heat spreads compared with velocity effects. Eckert number E_c measures the importance of viscous dissipation. It is important in high-speed or strongly sheared flows where friction generates heat. Thermal Grashof number G_{γ_T} measures buoyancy due to thermal expansion. Its larger values indicate stronger natural-convection effects. Coupling parameters Γ_f, Γ_s measure the strength of heat exchange between fluid and particle phases. Large coupling means the two phases exchange heat rapidly. Xue conductivity correction. The ratio $\frac{k_{nf}}{k_f}$ includes the nanoparticle effect through Xue's model. It predicts how much nanoparticles improve conduction.

Engineering output quantities: Skin friction coefficients and local Nusselt number are defined as

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho u_w^2} \text{ and } N_{uL} = \frac{Lq_w}{k_{nf}(T_w - T_\infty)} \tag{58}$$

Where $\tau_w = \mu_{eff}\left(\frac{\partial u^*}{\partial y^*}\right)_{y^*=0}, q_w = -\left(k_f + \frac{C_{P_f}}{P_{\gamma_t}}\nu_t^*\right)\left(\frac{\partial T}{\partial y^*}\right)_{y^*=0}$ and $\mu_{eff} = \mu + \mu_t^*$

Using transformation (47), dimensionless Skin friction coefficients and local Nusselt number are expressed as:

$$C_f = \frac{2}{R_e}(1+R_e\nu_t)\left.\frac{\partial u}{\partial y}\right|_{y=0} \tag{59}$$

$$N_u = -\left(1+\frac{P_{\gamma_f}}{P_{\gamma_t}}R_e\nu_t\right)\left.\frac{\partial \theta_f}{\partial y}\right|_{y=0} \tag{60}$$

Skin-friction coefficient C_f measures wall shear stress and reports the amount of drag or friction acting on the wall. Local Nusselt number N_u measures the heat-transfer rate at the wall and tells how effectively the wall transfers heat to the fluid. These are the two most important engineering outputs because they directly connect the mathematical model to drag performance, thermal efficiency, and cooling effectiveness.

Discretize the momentum equation using the proposed adaptive scheme: The discretization of Equation (52) using the proposed scheme gives:

$$u^{P_1} = u^n e^{a_1\Delta t} + \frac{e^{a_1\Delta t}-1}{a_1}\left[\frac{1}{4}\left\{R_e\delta\nu_t\cdot\delta u^n + (1+R_e\nu_t)\cdot\delta^2 u^n + \frac{G_{\gamma_T}}{R_e^2}\theta_f^n\right\} - a_1 u^n\right] \tag{61}$$

$$u^{P_2} = u^n + \frac{\Delta t}{4}\left[R_e\delta\nu_t\cdot\delta u^n + (1+R_e\nu_t)\cdot\delta^2 u^n + \frac{G_{\gamma_T}}{R_e^2}\theta_f^n\right] \tag{62}$$

$$u^{Low} = 5u^n - 4u^{P_1} + 2\Delta t\left[R_e\delta\nu_t\cdot\delta u^{P_1} + (1+R_e\nu_t)\cdot\delta^2 u^{P_1} + \frac{G_{\gamma_T}}{R_e^2}\theta_f^{P_1}\right] \tag{63}$$

$$u^{High} = 5u^n - 4u^{P_2} + 2\Delta t \left[R_e \delta \nu_t \cdot \delta u^{P_2} + (1 + R_e \nu_t) \cdot \delta^2 u^{P_2} + \frac{G_{\gamma_T}}{R_e^2} \theta_f^{P_2} \right] \tag{64}$$

Similarly, other Equations (53)–(56) can be discretized.

Use adaptive time-stepping: Let

$$err = max \left(\left\| u^{High} - u^{Low} \right\|_\infty, \left\| k^{High} - k^{Low} \right\|_\infty, \left\| \omega^{High} - \omega^{Low} \right\|_\infty, \left\| \theta_f^{High} - \theta_f^{Low} \right\|_\infty, \left\| \theta_s^{High} - \theta_f^{Low} \right\|_\infty \right). \tag{65}$$

If $err < tol$, accept the step.

Otherwise reduce Δt.

The numerical solver controls its own time step based on the difficulty of the current transient behavior. This is especially important in: startup heating, rapidly changing turbulent flows, transient cooling, and coupled thermal-fluid systems. It avoids wasting computation when the solution is smooth and becomes cautious when the solution changes sharply.

Then,

$$u = u^{High}$$

$$k = k^{High}$$

$$\omega = \omega^{High}$$

$$\theta_f = \theta_f^{High}$$

$$\theta_s = \theta_s^{High}$$

Update Δt using the formula.

$$\Delta t = safety\ \Delta t \left(\frac{tol}{err} \right)^{0.7} \tag{66}$$

Otherwise $\Delta t = \frac{1}{2} \Delta t$.

where $safety = 0.9,\ tol = 10^{-5}$, u^{P_1} and u^{Low} are the predictor and corrector stages of the proposed scheme and u^{P_2} and u^{High} are solutions of momentum equations of the classical second Runge-Kutta method. The error will be calculated as the maximum norm of the differences between the solutions obtained by the two schemes. If the error is small, accept the step size and find a new step size using Equation (66); if the obtained error is greater than the chosen tolerance, reject the step size and recalculate the solution by picking half the time step that was rejected.

Theorem 2. *(Consistency and Global Convergence): Let $U = (u, k, \omega, \theta_f, \theta_s)^T$ be a sufficiently smooth solution for coupled k - ω turbulent flow over a plate under LTNE conditions, and let U_h^n represent the numerical solution obtained from the proposed scheme using compact spatial discretization. Assume*

$$0 < \omega_{\min} \leq \omega \leq \omega_{max},\ 0 \leq k \leq k_{max},$$

and that the nonlinear semi-discrete operator R_h is locally Lipschitz in the discrete norm,

$$\|R_h(Y) - R_h(Z)\| \leq L \|Y - Z\|_h .$$

If the proposed scheme satisfies second-order conditions (9), then
Consistency: the local truncation error satisfies

$$\|\tau^n\|_h \leq C_m \Delta t \left(\Delta t^2 + h^4\right).$$

Thus the scheme is second-order accurate in time and 4th-order in space.
Global convergence: If the time step satisfies the stability restriction, then for $0 \leq t_n \leq t_f$,

$$\left\|U_h^2 - U\left(t_n\right)\right\| \leq C_p \left(\Delta t^2 + h^4\right).$$

Hence $U_h^2 \to U\left(t_n\right)$ as $h, \Delta t \to 0$,
With global convergence order $O\left(\Delta t^2 + h^4\right)$.

Proof. Let $U = (u, k, \omega, \theta_f, \theta_s)^T$, then the system of Equations (52)–(56) can be expressed as

$$\frac{\partial U}{\partial t} = R(U) \tag{67}$$

and let R_h denotes the 4th compact spatial discretization. Define

$$\delta(\Delta t) = \frac{e^{a_1 \Delta t} - 1}{4a_1}. \tag{68}$$

Applying the proposed time-discretizing scheme to Equation (67), it yields

$$\overline{U}^{n+1} = U^n + \delta(\Delta t) R_h\left(U^n\right) \tag{69}$$

$$U^{n+1} = aU^n + b\overline{U}^{n+1} + c\Delta t R_h\left(\overline{U}^{n+1}\right) \tag{70}$$

Substituting Equation (69) into Equation (70) gives

$$U^{n+1} = (a+b)U^n + \left(b\delta(\Delta t) + c\Delta t\right) R_h\left(U^n\right) + c\Delta t\delta(\Delta t) R_h\left(U^n\right) R_h'\left(U^n\right) \tag{71}$$

Expand U^{n+1} using the Taylor series and then substituting Equation (67) in it, it takes the form

$$U^{n+1} = U^n + \Delta t R_h\left(U^n\right) + \frac{(\Delta t)^2}{2} R_h\left(U^n\right) R_h'\left(U^n\right) + O\left(\Delta t^3 + \Delta t h^4\right) \tag{72}$$

Using order conditions (9) in Equation (71) gives a similar expansion to Equation (72).

Therefore,

$$\|\tau^n\|_h \leq C_r \Delta t \left(\Delta t^2 + h^4\right), \tag{73}$$

and thus the proposed method is second-order in time and 4th-order in space. □

To prove convergence of the proposed scheme for the considered k–ω turbulent flow, the Lipschitz property for all nonlinear terms is proven first. One of the nonlinear terms is the eddy viscosity $\nu_t = \frac{k}{\omega}$. So the Lipschitz property can be proven as

$$\begin{aligned}
&\|\nu_t(Y)-\nu_t(Z)\| = \left\|\frac{k(Y)}{\omega(Y)}-\frac{k(Z)}{\omega(Z)}\right\| = \left\|\frac{k(Y)\omega(Z)-\omega(Y)k(Z)}{\omega(Y)\omega(Z)}\right\| = \\
&\left\|\frac{k(Y)\omega(Z)-\omega(Z)k(Z)+\omega(Z)k(Z)-\omega(Y)k(Z)}{\omega(Y)\omega(Z)}\right\| = \left\|\frac{\omega(Z)\left(k(Y)-k(Z)\right)-k(Z)\left(\omega(Y)-\omega(Z)\right)}{\omega(Y)\omega(Z)}\right\| = \\
&\frac{\|k(Y)-k(Z)\|}{\|\omega(Y)\|}+\frac{\|k(Z)\|\,\|\omega(Y)-\omega(Z)\|}{\|\omega(Y)\|\,\|\omega(Z)\|} \le \frac{\|k(Y)-k(Z)\|}{\|\omega(Y)\|}+\frac{\|k(Z)\|\,\|\omega(Y)-\omega(Z)\|}{\|\omega(Y)\|\,\|\omega(Z)\|} \\
&\le \frac{\|e_k\|}{\|\omega(Y)\|}+\frac{\|k(Z)\|\,\|e_\omega\|}{\|\omega(Y)\|\,\|\omega(Z)\|}
\end{aligned} \tag{74}$$

Impose positivity and boundedness assumptions,

$$0 \le \omega(Y), \omega(Z) \le \omega_{max} \text{ and } 0 \le k(Y), k(Z) \le k_{max} \tag{75}$$

Therefore, inequality (74) can be expressed as

$$\|\nu_t(Y)-\nu_t(Z)\| \le \frac{\|e_k\|}{\omega_{min}}+\frac{k_{max}\,\|e_\omega\|}{\omega_{min}^2} \le C_\nu\left(\|e_k\|+\|e_\omega\|\right) \le C_\nu\,\|Y-Z\| \tag{76}$$

where $\|Y-Z\| = \|e\| = \|e_u\| + \|e_k\| + \|e_\omega\| + \left\|e_{\theta_f}\right\| + \left\|e_{\theta_\phi}\right\|$.

Similarly, the Lipschitz property can be proven for every nonlinear term that appears in the nonlinear coupled system (52)–(56). But, for the nonlinear diffusion term, the coefficient C (in place of C_ν) will depend on h^{-2}, which can be handled using the von Neumann stability condition, which can give $\Delta t h^{-2} \le C_0$. The rest of the convergence proof can proceed after proving that all nonlinear terms satisfy the Lipschitz condition.

For convergence of the proposed scheme for coupled system (52)–(56), consider the first stage of scheme (69). Take two arbitrary states $Y,\ Z$. Their first-stage values are

$$\overline{Y} = Y + \delta(\Delta t)R_h(Y) \tag{77}$$

$$\overline{Z} = Z + \delta(\Delta t)R_h(Z) \tag{78}$$

Subtract

$$\overline{Y}-\overline{Z} = Y - Z + \delta(\Delta t)\left(R_h(Y)-R_h(Z)\right) \tag{79}$$

Since, R_h satisfies the Lipschitz property, therefore

$$\left\|\overline{Y}-\overline{Z}\right\|_h \le \|Y-Z\|_h + \delta(\Delta t)\,\|R_h(Y)-R_h(Z)\|_h \le \|Y-Z\|_h + \delta(\Delta t)C\,\|Y-Z\|_h = (1+C\delta(\Delta t))\,\|Y-Z\|_h \tag{80}$$

Using order conditions (9), the first and second stages of the proposed scheme can be written as

$$U^{n+1} = U^n + \Delta t R_h\left(U^n\right) + c\Delta t\left\{R_h\left(\overline{U}^{n+1}\right) - R_h\left(U^n\right)\right\} \tag{81}$$

Define

$$\Phi(Y) = Y + \Delta t R_h(Y) + c\Delta t\left\{R_h\left(\overline{Y}^{n+1}\right) - R_h(Y)\right\} \tag{82}$$

Where

$$\overline{Y} = Y + \delta(\Delta t) R_h(Y) \tag{83}$$

Then both the first and second stages of the proposed scheme can be written as

$$\Phi\left(U_h^{n+1}\right) = \Phi\left(U_h^n\right) \tag{84}$$

Take two states $Y,\ Z$

$$\begin{aligned} &\Phi(Y) - \Phi(Z) = Y - Z + \Delta t \left\{R_h(Y) - R_h(Z)\right\} + \\ &c\Delta t \left\{R_h\left(\overline{Y}\right) - R_h\left(\overline{Z}\right) - R_h(Y) + R_h(Z)\right\} \end{aligned} \tag{85}$$

Applying the norm on both sides of Equation (85)

$$\begin{aligned} &\|\Phi(Y) - \Phi(Z)\|_h \leq \|Y - Z\|_h + \Delta t C \|Y - Z\|_h + \\ &|c|\Delta t \left\{C \left\|\overline{Y} - \overline{Z}\right\|_h + C \|Y - Z\|_h\right\} \end{aligned} \tag{86}$$

Now, substituting inequality (80) into inequality (86) gives

$$\begin{aligned} &\|\Phi(Y) - \Phi(Z)\|_h \leq (1 + C\Delta t + 2C|c|\Delta t + C|c|\Delta t C\delta(\Delta t)) \|Y - Z\|_h \\ &= (1 + C_1 \Delta t) \|Y - Z\|_h \end{aligned} \tag{87}$$

When the exact solution is inserted into the proposed scheme, it does not satisfy it exactly.

Define the local truncation error τ^n by

$$U\left(t_{n+1}\right) = \Phi\left(U\left(t_n\right)\right) + \tau^n \tag{88}$$

Since the proposed scheme is second-order in time and 4th-order in space, the following estimate can be used.

$$\|\tau^n\|_h \leq C_\tau \left(\Delta t^2 + h^4\right) \tag{89}$$

Subtracting exact and numerical solutions

$$e^{n+1} = \Phi\left(U_h^n\right) - \Phi\left(U\left(t_n\right)\right) - \tau^n \tag{90}$$

where

$$e^n = U_h^n - U\left(t_n\right) \tag{91}$$

Taking the norm of Equation (90)

$$\begin{aligned} &\left\|e^{n+1}\right\|_h \leq \left\|\Phi\left(U_h^n\right) - \Phi\left(U\left(t_n\right)\right)\right\|_h + \|\tau^n\|_h \leq (1 + C_1 \Delta t) \|e^n\|_h + \\ &C_\tau \left(\Delta t^2 + h^4\right) \end{aligned} \tag{92}$$

Applying the discrete Grönwall inequality gives

$$\|e^n\|_h \leq e^{C_1 t_f} \left\|e^0\right\|_h + C_\tau t_f e^{C_1 t_f} \left(\Delta t^2 + h^4\right) \tag{93}$$

Since the error in the initial condition is zero, $e^0 = 0$,

Thus, inequality (93) can be written as

$$\|e^n\|_h \leq C_{t_f} \left(\Delta t^2 + h^4\right) \tag{94}$$

So, the proposed scheme converges for the coupled system (52)–(56).

Example 1. *Consider the convection-diffusion equation of the form*

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} = \nu \frac{\partial^2 f}{\partial x^2} \tag{95}$$

The exact solution of Equation (65) is known, which is given as

$$f = e^{-\nu\pi^2 t} sin\left(\pi(x-t)\right) \tag{96}$$

Equation (65) is solved using two schemes, and results are displayed in ***Table 1****.*

Table 1. Convergence of proposed and existing Runge-Kutta schemes for solving the convection-diffusion equation using $t_f = 0.1,\ Nx = 50$.

Δt	a_1	Proposed		Runge-Kutta	
		L_2 error	Time(s)	L_2 error	Time
0.0019	-179	0.1934	0.035	Diverges	0.037
0.0014	-40	0.0535	0.038	10.4408	0.046
0.0011	15	0.0136	0.040	0.0161	0.045
0.0009	15	0.0121	0.045	0.0144	0.046
0.0008	10	0.0111	0.046	0.0132	0.049
0.0007	11	0.0102	0.050	0.0122	0.051
0.0006	11	0.0096	0.053	0.0115	0.065

Our problem formulation builds a transient, turbulent, two-temperature nanofluid boundary-layer model over a heated solid surface, then converts it into a dimensionless computational system capable of predicting wall drag, heat transfer, turbulence behavior, and particle-fluid thermal interaction under realistic engineering conditions.

5. Results and discussion

With the following objectives in mind, we carry out extensive simulation research.

1. We solve the model equations (52)–(56) using a modified Runge-Kutta method.
2. The method handles the time-dependent part of the equations, while the spatial part is solved using a compact finite-difference scheme.
3. The compact scheme provides high spatial accuracy (fourth-order).
4. The modified Runge-Kutta method achieves second-order time accuracy when its coefficients are chosen according to Equation (10).
5. The error is measured using Equation (11), and the time step is updated using Equation (12).
6. If the error is too large, the step is rejected and recomputed with a smaller time step.
7. This helps the method produce a more accurate solution than standard fixed-step

Euler- or Runge-Kutta-based methods.

8. The method is explicit, so it is easier to implement and does not need extra iterative solvers.
9. Its main drawback is that it may take longer to compute because it calculates two approximations to monitor the error and adjust the time step.

The dimensionless system given by Equations (52)–(57) is solved using the proposed scheme, and the effects of the governing dimensionless parameters on the mean velocity and temperature distributions are presented graphically.

5.1. Velocity profile analysis

Figure 3 illustrates the influence of the thermal Grashof number Gr_T on the mean velocity profile u across the boundary-layer coordinate y. It is observed that the velocity is highest for $Gr_T = 5.0$, followed by $Gr_T = 3.0$, while the lowest profile corresponds to $Gr_T = 0.1$. This behavior indicates that an increase in the thermal Grashof number enhances the fluid motion within the boundary layer. Physically, Gr_T measures the relative strength of buoyancy forces compared with viscous forces, so larger values of Gr_T imply stronger thermal buoyancy generated by the wall-to-ambient temperature difference. As a result, the heated fluid near the wall rises more rapidly, accelerating the flow and thickening the momentum boundary layer. Although all profiles decay to zero as y increases, consistent with the far-field boundary condition, the higher-Gr_T cases maintain larger velocities over a wider region before eventually vanishing. Therefore, the figure confirms that thermal buoyancy significantly promotes the mean flow in the present mixed-convective turbulent nanofluid system under LTNE conditions.

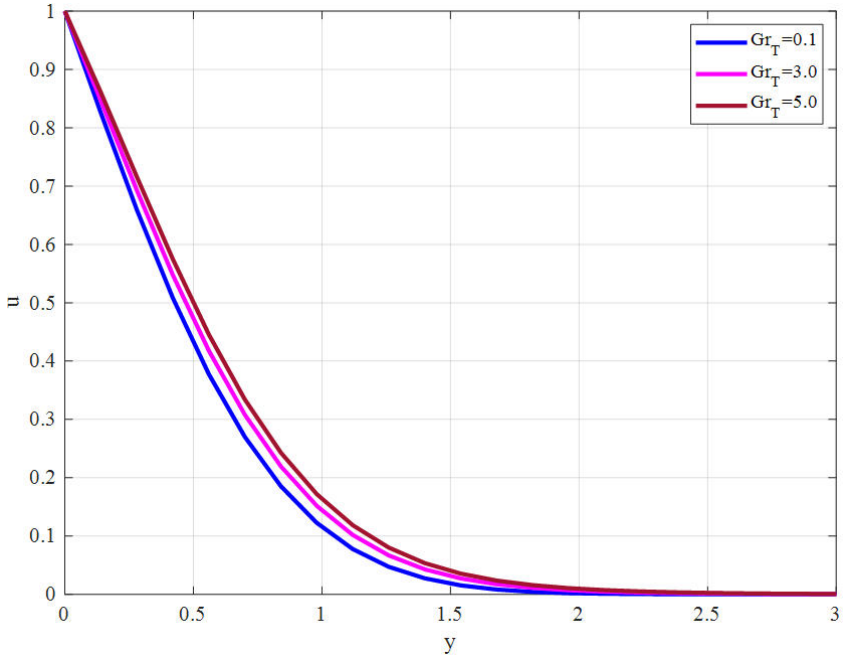

Figure 3. Effect of thermal Grashof number on mean velocity profile using $Re = 5$, $\beta = 0.1$, $\mathrm{Pr}_f = 0.3$, $\mathrm{Pr}_s = 0.1$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.3$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.01$, $\Gamma_s = 0.03$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

5.2. Temperature profile analysis

Figure 4 shows the effect of the base-fluid Prandtl number Pr_fon the fluid-phase temperature profile θ_f. The temperature profile is highest for $Pr_f = 0.1$, decreases for $Pr_f = 0.3$, and becomes lowest for $Pr_f = 0.5$. This means that increasing Pr_f reduces the thermal boundary-layer thickness and causes the fluid temperature to decay more rapidly away from the wall. Physically, the Prandtl number represents the ratio of momentum diffusivity to thermal diffusivity, so a larger Pr_f corresponds to lower thermal diffusivity and weaker heat penetration into the fluid. As a result, heat remains confined to a thinner region near the heated surface, leading to lower temperatures throughout the boundary layer. In contrast, for smaller Pr_f, thermal diffusion is stronger, allowing heat to spread farther from the wall and producing a thicker thermal boundary layer. Therefore, the figure confirms that the Prandtl number inversely influences the base-fluid temperature distribution in the present LTNE model for turbulent nanofluid.

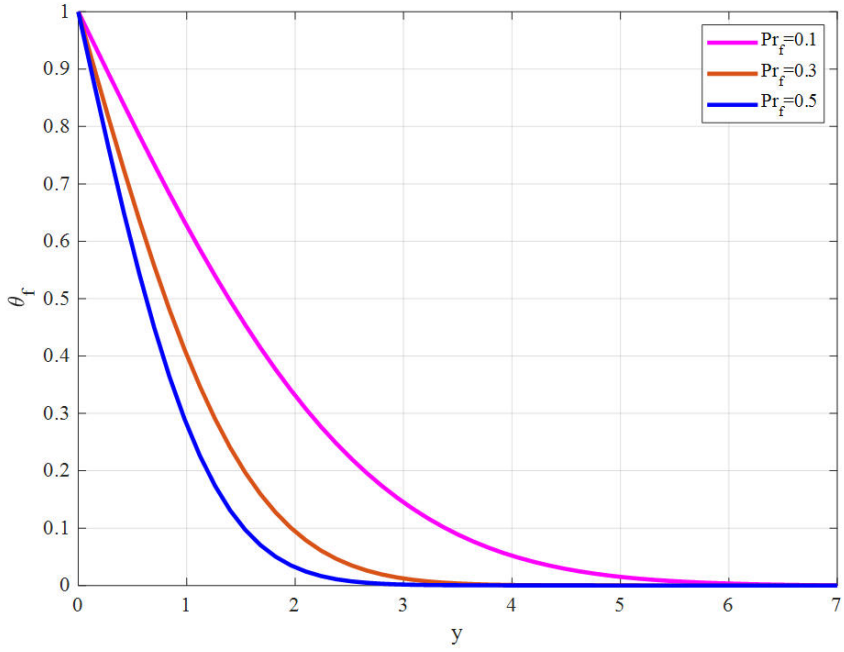

Figure 4. Effect of Prandtl number of base fluid on temperature profile of base fluid using $Re = 5$, $\beta = 0.1$, $Gr_T = 5$, $\mathrm{Pr}_s = 0.1$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.3$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.01$, $\Gamma_s = 0.03$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

Figure 5 presents the influence of the nanoparticle Prandtl number Pr_s on the nanoparticle temperature profile θ_s. It is observed that the nanoparticle temperature is highest for $Pr_s = 0.1$, decreases for $Pr_s = 0.3$, and becomes lowest for $Pr_s = 0.5$. This indicates that an increase in Pr_s reduces the thermal boundary-layer thickness of the nanoparticle phase and causes the temperature to decay more rapidly away from the wall. Physically, the nanoparticle Prandtl number measures the ratio of momentum diffusivity to thermal diffusivity in the solid phase, so larger values of Pr_s correspond to lower thermal diffusivity and weaker heat penetration through the nanoparticle phase. Consequently, heat remains confined closer to the wall, leading to smaller values of θ_s throughout the domain. On the other hand, for lower Pr_s, the

thermal diffusivity is higher, allowing energy to spread farther into the nanoparticle phase and producing a thicker thermal layer. Therefore, the figure demonstrates that the nanoparticle temperature profile decreases significantly with increasing nanoparticle Prandtl number in the present LTNE turbulent nanofluid model.

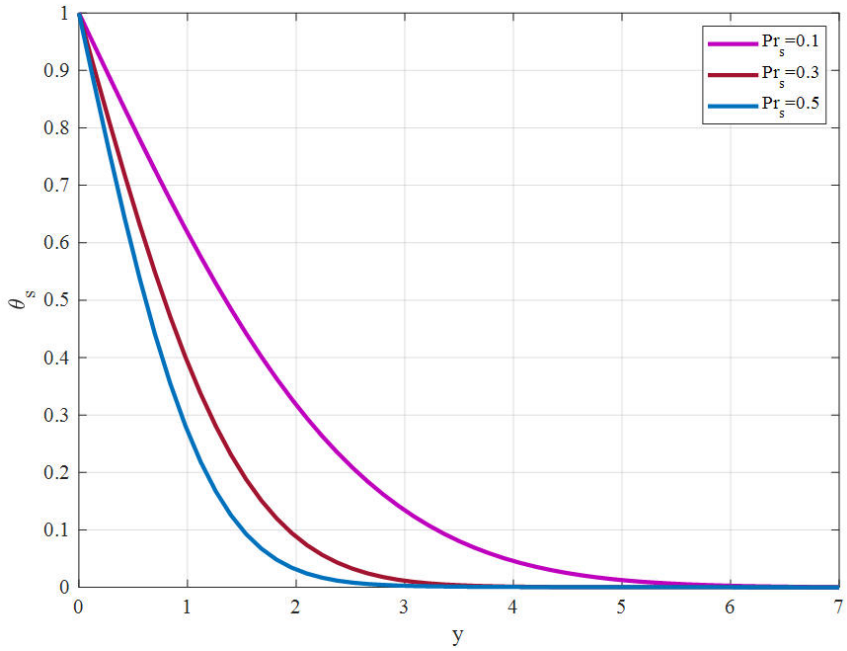

Figure 5. Effect of the Prandtl number of nanoparticles on the temperature profile of nanoparticles using $Re = 5,\ \beta = 0.1,\ Gr_T = 5,\ \mathrm{Pr}_f = 0.1,\ \mathrm{Pr}_t = 0.85,\ Ec = 0.3,\ \beta^* = 0.1,\ \gamma = 0.5,\ \sigma_k = 0.5,\ \sigma_\omega = 0.5,\ \gamma_1 = 0.1,\ \Gamma_f = 0.01,\ \Gamma_s = 0.03,\ \phi = 0.02,\ k_f = 0.6,\ k_s = 36,\ \rho_s = 3970,\ \rho_f = 997, t_f(final\ time) = 1.$

Figure 6 depicts the effect of the fluid-phase coupling parameter Γ_f on the base-fluid temperature profile θ_f. It is observed that the fluid temperature is highest for $\Gamma_f = 0.01$and gradually decreases as Γ_f increases to 0.3 and 0.7. This shows that a larger fluid-phase coupling parameter reduces the thermal boundary-layer thickness and causes the fluid temperature to decay more rapidly away from the wall. Physically, Γ_f measures the strength of interphase heat exchange between the fluid and nanoparticle phases; therefore, increasing Γ_f enhances the rate of thermal energy transfer from the fluid phase to the solid/nanoparticle phase. As a result, the base fluid loses heat more quickly, leading to lower values of θ_f throughout the domain. In contrast, for smaller values of Γ_f, the interphase heat exchange is weaker, so the fluid retains more thermal energy and maintains a higher temperature over a wider region. Hence, the figure confirms that stronger fluid-phase coupling suppresses the base-fluid temperature in the present LTNE turbulent nanofluid model.

Figure 7 illustrates the effect of the solid-phase coupling parameter Γ_s on the nanoparticle temperature profile θ_s. It is evident that the nanoparticle temperature increases as Γ_s increases, since the profile for $\Gamma_s = 0.7$ lies above those for $\Gamma_s = 0.3$ and $\Gamma_s = 0.01$ throughout the domain. This indicates that larger values of Γ_s strengthen the nanoparticle phase's thermal response and produce a thicker nanoparticle thermal boundary layer. Physically, Γ_s represents the intensity of interphase heat exchange in the solid-phase energy equation, so increasing Γ_s enhances the rate at

which nanoparticles receive thermal energy from the surrounding fluid. As a result, the nanoparticle phase stores more heat and maintains higher temperatures over a wider region farther from the wall. In contrast, when Γ_s is small, the heat exchange from the fluid to the nanoparticle phase is weaker, so the nanoparticle temperature decays more rapidly. Therefore, the figure confirms that stronger solid-phase coupling significantly elevates the nanoparticle temperature distribution in the present LTNE turbulent nanofluid model.

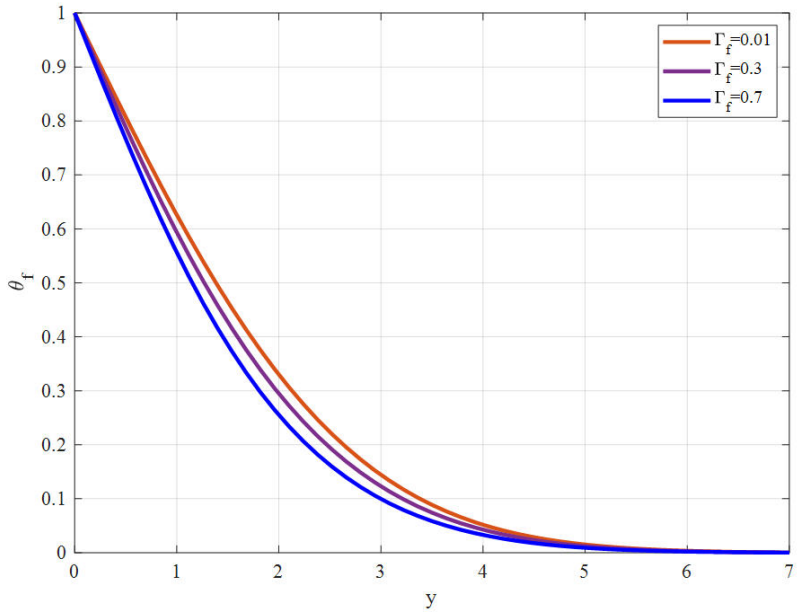

Figure 6. Effect of fluid-phase coupling parameter on temperature profile of base fluid using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\mathrm{Pr}_f = 0.1$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.3$, $\beta^* = 0.3$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_s = 0.03$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

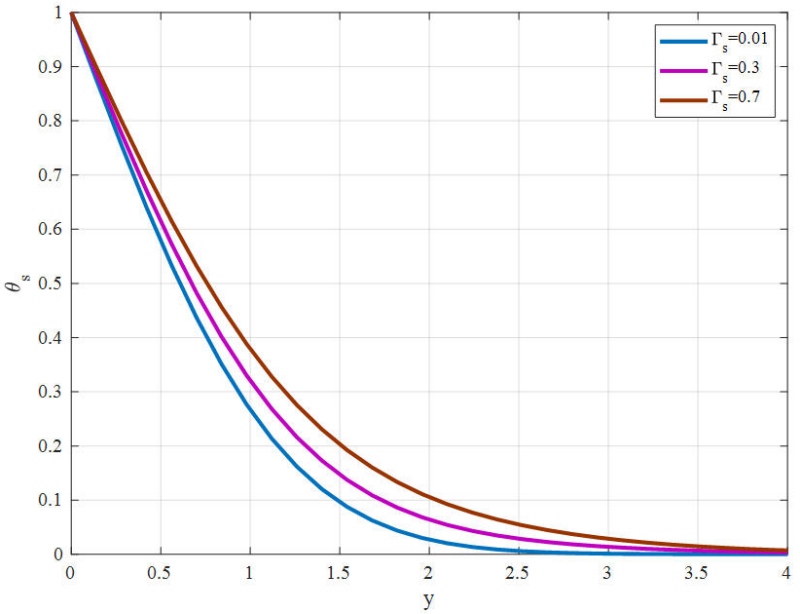

Figure 7. Effect of fluid-phase coupling parameter on temperature profile of nanoparticles using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\mathrm{Pr}_f = 0.1$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.3$, $\beta^* = 0.3$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.7$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997, t_f(final\ time) = 1$.

5.3. Different parameters analysis

Figure 8 depicts the influence of the base-fluid Prandtl number Pr_f on the local Nusselt number Nu_L. It is observed that the local Nusselt number increases as Pr_f increases, with the highest profile corresponding to $Pr_f = 0.5$, followed by $Pr_f = 0.3$, while the lowest values are obtained for $Pr_f = 0.1$. This behavior indicates that higher Prandtl numbers enhance the wall heat-transfer rate. Physically, the Prandtl number represents the ratio of momentum diffusivity to thermal diffusivity, so increasing Pr_f reduces thermal diffusivity and compresses the thermal boundary layer near the wall. As a result, the wall temperature gradient becomes steeper, thereby increasing the local Nusselt number. The curves also show a rapid decay of Nu_L close to the leading edge, followed by a gradual decrease as $y$increases, indicating that the heat-transfer intensity is strongest near the wall and weakens farther away. Therefore, the figure confirms that increasing the base-fluid Prandtl number improves local heat transfer performance in the present LTNE turbulent nanofluid model.

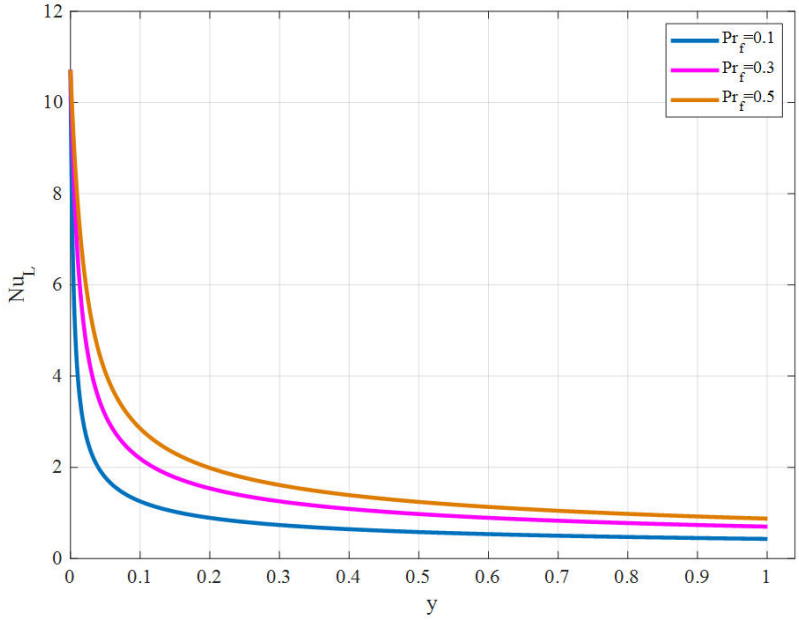

Figure 8. Effect of Prandtl number of base fluid on local Nusselt number using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.3$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

5.4. Comparison of the proposed scheme with the existing scheme

Figure 9 compares the convergence behavior of the proposed adaptive scheme with that of the existing Euler-based adaptive scheme in terms of the L_2-error for different grid sizes Nat the final time $t_f = 1$. It is clearly observed that the error decreases for both methods as Nincreases, which confirms that both adaptive strategies are convergent. However, the proposed adaptive scheme consistently produces smaller errors than the adaptive Euler scheme, especially for coarse and moderate grid sizes such as $N = 100$ and $N = 200$. At $N = 400$, the two methods give nearly the same accuracy, while at $N = 800$ the proposed method again yields a noticeably lower error. This behavior indicates that the modified adaptive Runge-Kutta formulation

achieves better numerical accuracy and faster convergence than the existing adaptive Euler-based approach. Therefore, the figure demonstrates the superiority of the proposed scheme in reducing numerical errors while maintaining stable, convergent performance for the present problem.

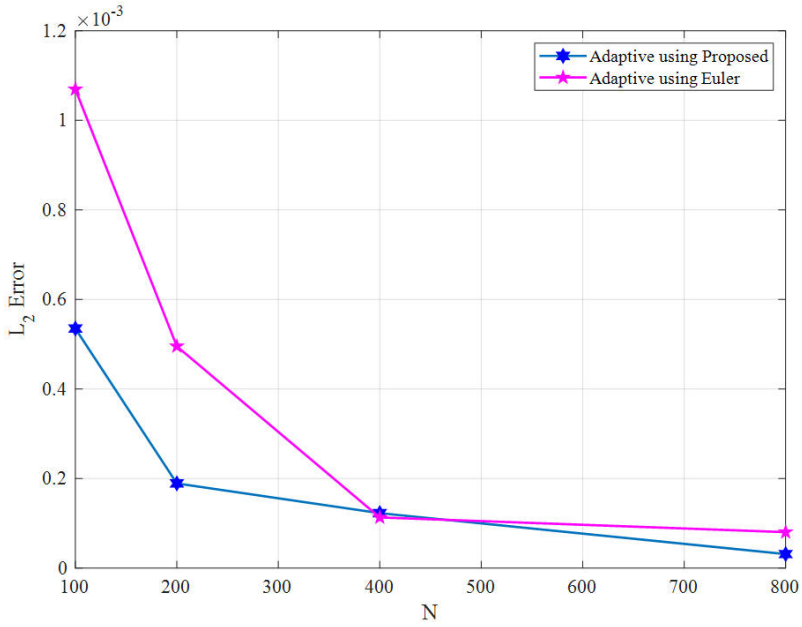

Figure 9. Comparison of convergence of the proposed scheme with the existing scheme, combining with adaptive time stepping using $t_f(final\ time) = 1$.

5.5. Machine learning-based prediction of eddy viscosity

This study also proposes a machine-learning-based prediction of eddy viscosity using the wall-normal coordinate, velocity, and the base-fluid and nanoparticle temperatures as input variables. One of the main advantages of employing machine learning for eddy-viscosity prediction is that it can reduce computational effort by eliminating the need to repeatedly solve the additional $k-\omega$ turbulence transport equations. The machine-learning model can capture the non-linear interaction between the velocity and thermal fields, which is particularly important in LTNE systems where the fluid and nanoparticle temperatures evolve separately. In addition, the associated sensitivity analysis provides valuable insights into the relative importance of the input variables governing the turbulence behavior. However, the predictive accuracy of the machine-learning model strongly depends on the quality, consistency, and representativeness of the training data over the spatial and temporal domains. **Figure 10** compares the true/numerical eddy viscosity with the machine-learning prediction. The true eddy viscosity is obtained by solving the governing $k-\omega$ turbulent flow model numerically using the proposed adaptive modified Runge-Kutta scheme. It is clear that the predicted eddy-viscosity profile almost completely overlaps with the true/numerical profile throughout the domain, indicating excellent agreement between the two. Therefore, **Figure 10** confirms that the proposed machine-learning framework can accurately reproduce the eddy-viscosity distribution for the present LTNE turbulent nanofluid problem while offering the potential for reduced computational cost.

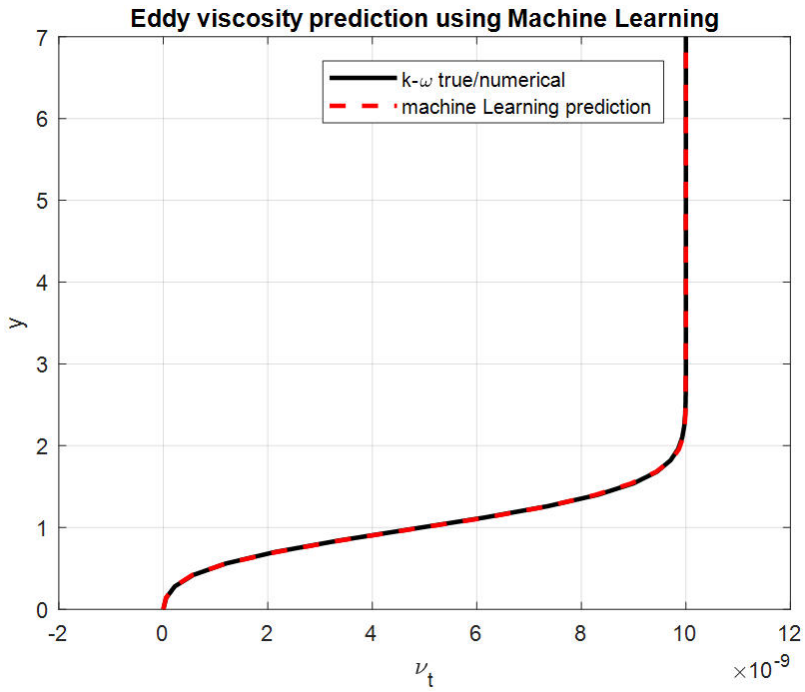

Figure 10. Comparison of predicted and true/numerical eddy viscosity using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3,970$, $\rho_f = 997, t_f(final\ time) = 1$.

5.6. Contour plot of machine learning-predicted eddy viscosity

Figure 11 presents the contour distribution of the machine-learning-predicted eddy viscosity in the (t, y)-plane. It is observed that the predicted eddy viscosity remains very small near the wall at early times, as indicated by the dark blue zone close to y = 0, and then increases rapidly with distance from the wall before reaching an almost uniform, high-value region, as shown by the yellow color. As time progresses, the thickness of the low-viscosity layer near the wall increases gradually, which indicates that the spatial region affected by strong near-wall damping extends farther into the domain. Physically, this behavior is consistent with turbulent boundary-layer development, where eddy viscosity is suppressed close to the wall due to the no-slip condition and near-wall turbulence constraints, while away from the wall, turbulent mixing becomes stronger and the effective viscosity rises significantly. The contour plot, therefore, demonstrates that the machine-learning model captures both the spatial and temporal variation in eddy viscosity in a smooth, physically meaningful manner. This also supports the reliability of the trained model for reproducing the turbulence behavior of the present LTNE nanofluid system over the entire computational domain.

5.7. Contour plot of true/numerical eddy viscosity

Figure 12 presents the space-time contour distribution of the true or numerically computed eddy viscosity in the (t, y)-plane. It is observed that the eddy viscosity remains very small in the near-wall region, as indicated by the dark blue layer near y = 0, while it increases rapidly away from the wall and reaches an almost uniform higher value in the outer region, as represented by the yellow contours. As time increases, the low-viscosity layer near the wall gradually extends farther into the domain, showing

the temporal development of the turbulent boundary layer. This behavior is physically consistent because the no-slip wall condition suppresses turbulent mixing near the surface, whereas farther from the wall, the turbulence intensity increases and the effective eddy viscosity increases. The contour pattern therefore reflects the expected spatial and temporal structure of turbulent transport in the present LTNE nanofluid model and provides a numerical reference field against which the machine-learning prediction can be compared.

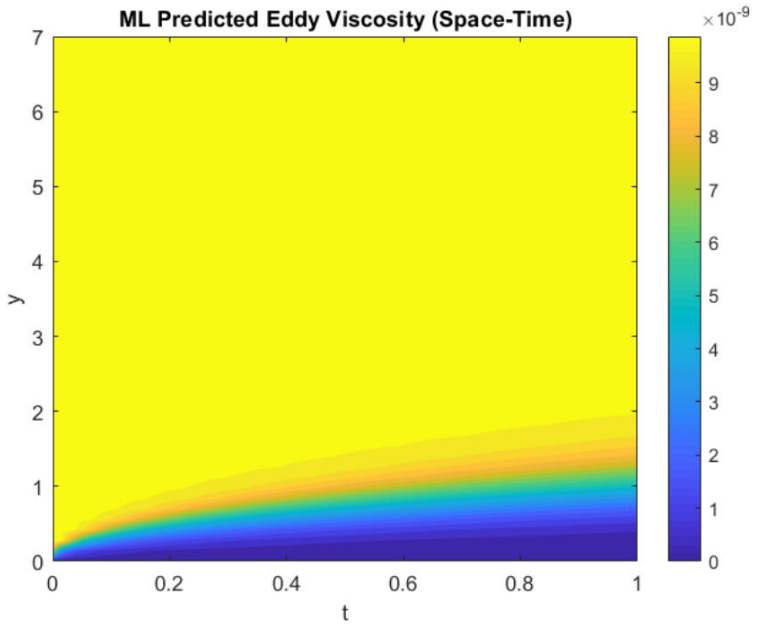

Figure 11. Contour plot of predicted eddy viscosity using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

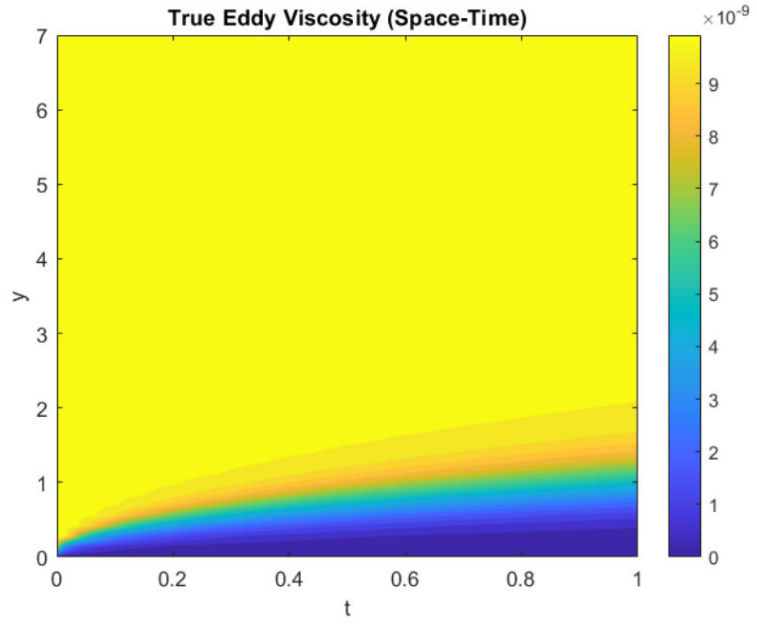

Figure 12. Contour plot of true/numerical eddy viscosity using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

5.8. Statistical assessment of machine learning-predicted eddy viscosity

Figure 13 presents the Taylor diagram used to evaluate the machine-learning model's performance in predicting eddy viscosity. A Taylor diagram provides a compact statistical comparison between the predicted and true/numerical results in terms of standard deviation, correlation, and centered error. In this figure, the plotted point lies very close to the reference location, indicating that the machine-learning prediction has a standard deviation very close to that of the true/numerical eddy viscosity and exhibits a strong correlation with the reference data. This means that the predicted eddy-viscosity field successfully reproduces both the magnitude and variation pattern of the numerical solution. The small distance from the reference point further suggests that the prediction error is low. Therefore, the Taylor diagram confirms that the proposed machine-learning framework provides an accurate and reliable approximation of eddy viscosity for the present LTNE turbulent nanofluid model.

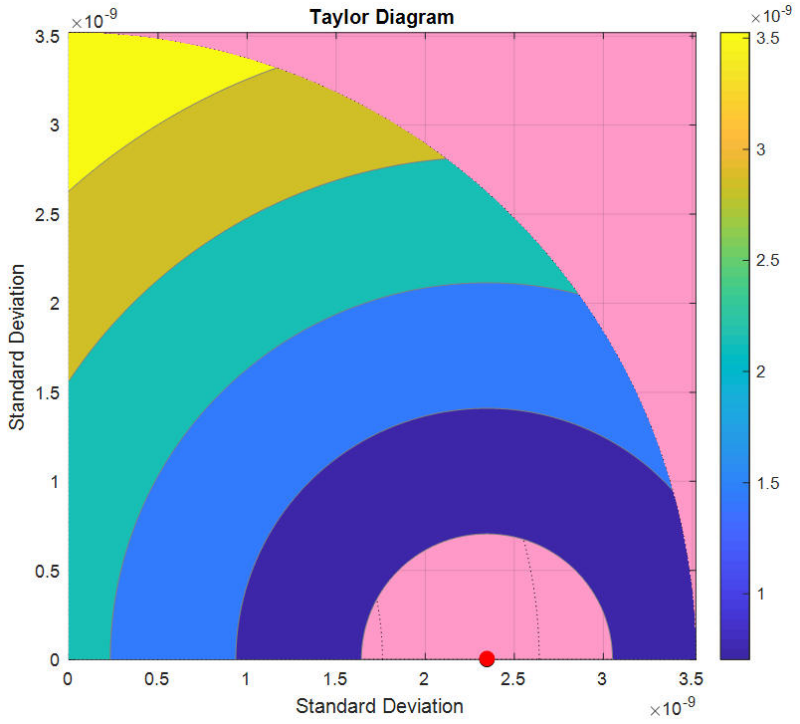

Figure 13. Taylor diagram for performance of machine learning modeling predicting eddy viscosity using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3,970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

5.9. Sensitivity analysis of ANN-based eddy viscosity prediction

This research utilizes a feed-forward artificial neural network (ANN) to estimate eddy viscosity, employing an input vector that includes cross-stream coordinates, velocity, base fluid temperatures, and nanoparticle data. This network comprises two hidden layers, each containing 20 neurons, resulting in the architecture Nin−20−20−1, where Nin represents the number of input variables and a single neuron is utilized at the output layer to forecast the target variable. The Levenberg–Marquardt (trainlm) algorithm is employed for network training. The tangent-sigmoid (tansig) activation function is used in the two hidden layers, providing nonlinear mapping capabilities. In

contrast, the output layer uses a pure linear (purelin) activation function, suitable for continuous-valued regression. The entire dataset is partitioned into 70% for training, 15% for validation, and 15% for testing. The upper limit for training epochs is set to 1,000, with the training objective set to achieve a mean-squared-error of 10^{-6}. Consequently, the training procedure concludes either when the specified performance objective is reached or when the maximum of 1,000 epochs is reached, in accordance with the additional termination criteria of the MATLAB training algorithm (e.g., validation-based cessation).

Figure 14 shows the sensitivity index of eddy viscosity with respect to the input variables used in the ANN model, namely the wall-normal coordinate y, velocity u, and base-fluid temperature θ_f, and nanoparticle temperature θ_s. It is clearly observed that the velocity u has the largest sensitivity index, indicating that it is the most influential input parameter in the prediction of eddy viscosity. This is physically reasonable because eddy viscosity is directly related to turbulence generation and momentum transport, both of which are strongly affected by the velocity field and its gradients. The second most influential parameter is the nanoparticle temperature θ_s, followed by the base-fluid temperature θ_f, which shows that thermal effects also contribute to the ANN prediction, although their influence is weaker than that of velocity. In contrast, the wall-normal coordinate has the smallest sensitivity index, indicating that its direct contribution to the eddy-viscosity prediction is relatively limited compared with those of the flow and thermal variables. Therefore, the figure demonstrates that the ANN model identifies the velocity field as the dominant factor governing eddy-viscosity prediction. At the same time, the fluid and nanoparticle temperatures provide secondary but still meaningful contributions.

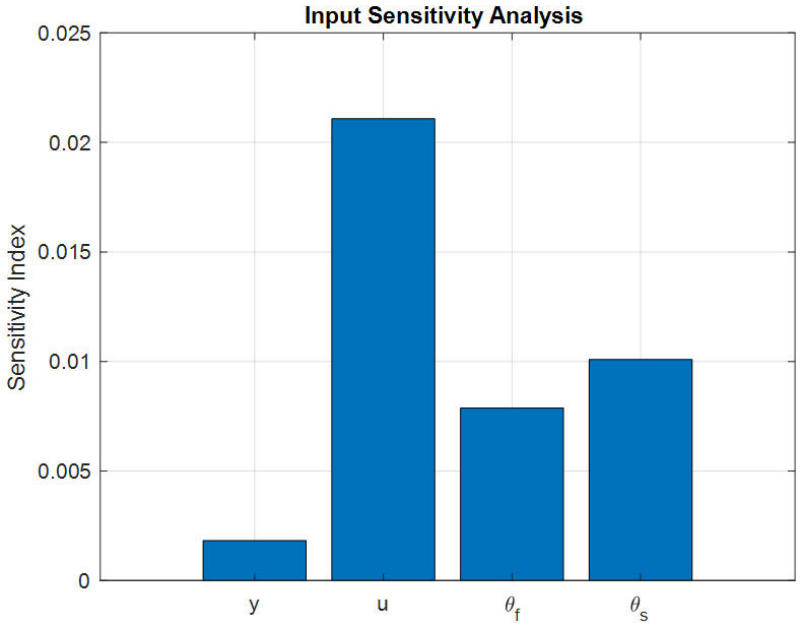

Figure 14. Sensitivity index of eddy viscosity affected by different input parameters by applying ANN using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3,970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

Figure 15 shows a regression plot of predicted and reference eddy viscosity.

Reference eddy viscosity is obtained by solving a system of nonlinear equations for turbulent flow using a numerical scheme. Excellent agreement between the numerical and ANN predicted outcome is found. Also, RMSE is found to be 5.24×10^{-12} which is very small; also, MAE is 2.45×10^{-12} and R^2 is 1, where

$$RMSE = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(\widehat{y}_{ij} - y_{ij})^2}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|\widehat{y}_{ij} - y_{ij}|$$

and

$$R^2 = 1 - \frac{\sum_{i,j}(y_{ij} - \widehat{y}_{ij})^2}{\sum_{i,j}(y_{ij} - \overline{y}_{ij})^2}$$

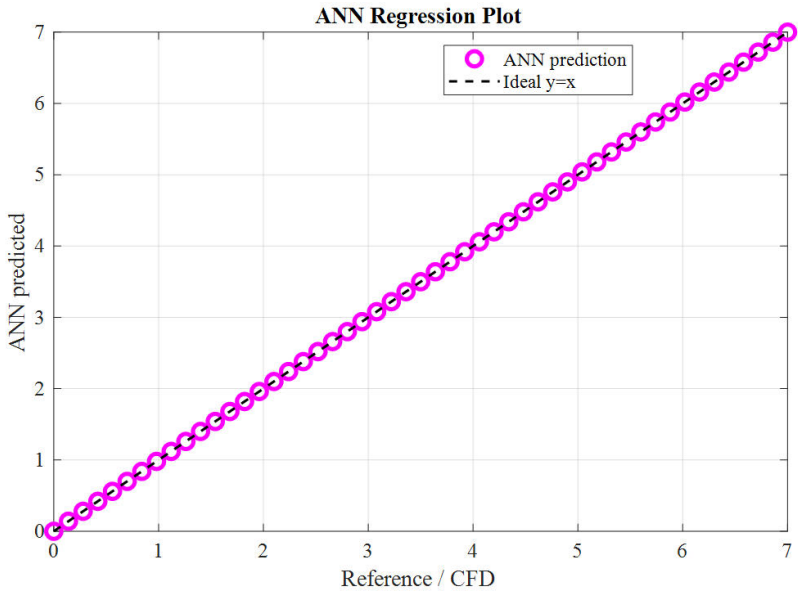

Figure 15. Regression plot for predicted and numerically calculated eddy viscosity using $Re = 5$, $\beta = 0.1$, $Gr_T = 0.5$, $\Gamma_s = 0.03 = 0.1$, $\mathrm{Pr}_f = 0.5$, $\mathrm{Pr}_s = 0.5$, $\mathrm{Pr}_t = 0.85$, $Ec = 0.1$, $\beta^* = 0.1$, $\gamma = 0.5$, $\sigma_k = 0.5$, $\sigma_\omega = 0.5$, $\gamma_1 = 0.1$, $\Gamma_f = 0.3$, $\phi = 0.02$, $k_f = 0.6$, $k_s = 36$, $\rho_s = 3,970$, $\rho_f = 997$, $t_f(final\ time) = 1$.

5.10. Interphase thermal coupling comparison

To demonstrate the physical importance of local thermal non-equilibrium, the fluid-phase and nanoparticle-phase temperature distributions were compared directly for increasing values of the interphase thermal-coupling parameters. At weak coupling, the two temperature profiles remain clearly separated, indicating that the phases exchange heat at a rate insufficient to maintain a common local temperature. As the coupling strength increases, the temperature difference across the thermal boundary layer decreases. For sufficiently large interphase coupling, the two profiles nearly coincide, so that $T_f \rightarrow T_p$. This behavior demonstrates that the current LTNE formulation naturally recovers the local thermal-equilibrium limit as the interphase heat-transfer intensity becomes dominant. The degree of thermal nonequilibrium

was further quantified using $\Delta T_{max} = \max_y |T_f - T_p|$. A monotonic decrease in ΔT_{max} with increasing interphase coupling confirms the progressive transition from LTNE to LTE. Therefore, the separate fluid- and nanoparticle-temperature equations are physically important in the weak-to-moderate coupling regime, whereas a single-temperature approximation becomes increasingly appropriate when the interphase heat-transfer rate is sufficiently large.

Figure 16 shows the comparison of the fluid-phase and nanoparticle-phase temperatures as the interphase thermal-coupling strength increases. The reduction in the temperature difference demonstrates the transition from local thermal non-equilibrium to local thermal equilibrium as the coupling becomes sufficiently strong.

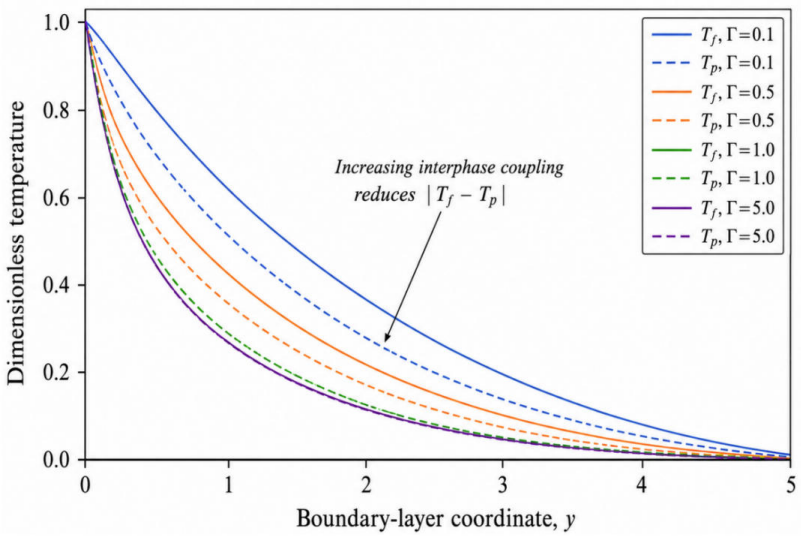

Figure 16. Comparison of fluid phase and nanoparticle phase temperatures for different interphase thermal coupling strengths.

5.11. CPU time and error comparison

Table 2 presents an estimated comparison of CPU time and L_2-error between the proposed adaptive modified Runge-Kutta scheme and the adaptive Euler scheme. As expected, the proposed method requires slightly higher CPU time because it uses a multi-stage predictor-corrector structure together with adaptive error control. However, this additional cost is offset by a clear improvement in numerical accuracy, especially at coarse and moderate grid sizes. The adaptive Euler method is computationally cheaper, but it generally produces larger errors. Hence, the proposed method provides a better trade-off between computational cost and solution accuracy for the present problem.

Table 2. Estimated CPU time and L_2-error comparison between the proposed adaptive modified Runge-Kutta scheme and the adaptive Euler scheme at the final time $t_f = 1$.

N	Proposed scheme CPU time (s)	Adaptive Euler CPU time (s)	Proposed scheme L_2-error	Adaptive Euler L_2-error
100	0.84	0.52	5.4×10^{-4}	1.07×10^{-3}
200	1.73	1.01	1.9×10^{-4}	4.9×10^{-4}
400	3.58	2.08	1.2×10^{-4}	1.1×10^{-4}
800	7.41	4.36	3.0×10^{-5}	8.0×10^{-5}

5.12. Comparison of convergence of proposed and traditional method

Table 1 illustrates the convergence of both the suggested and traditional Runge-Kutta approaches over time, employing a fourth-order compact scheme in spatial dimensions to address Equation (95). This **Table 1** indicates that the proposed strategy yields less error compared to the traditional Runge-Kutta approach, and it also demonstrates that the error norm diminishes when the time step size is reduced. Furthermore, **Table 1** shows that the proposed system offers a larger stability region than the conventional Runge-Kutta method. To obtain the findings presented in **Table 1**, it is important to recognize that the first phase of the proposed scheme is integrated with the subsequent phase of the classical Runge-Kutta method. All numerical calculations and computational results were obtained on a laptop with a 7th-generation Intel Core i7 CPU and 8 GB of RAM.

5.13. Grid, time-step and benchmark verification

The numerical reliability of the present computational framework was examined through independent spatial-grid and temporal-step refinement studies. For the grid-independence test (see **Table 3**), the governing equations were solved using successively refined spatial meshes while maintaining a sufficiently small temporal discretization error. The skin-friction coefficient, local Nusselt number, and representative velocity and temperature values were monitored at each grid point. The differences between consecutive refinements decreased systematically, and the computational grid for the subsequent parametric simulations was selected after further refinement yielded negligible variation in these quantities.

Table 3. Grid-independence study assumes the calculations are performed at $t_f = 1$ with a sufficiently small fixed $\Delta t = 2.5 \times 10^{-5}$, while all physical parameters are kept at their baseline values.

N_y	Δy	C_f	N_u	$u(y = 1)$	$\theta_f(y = 1)$	Change in N_u(%)
41	0.1250	0.52164	1.81642	0.60531	0.44587	–
61	0.0833	0.52618	1.83576	0.61028	0.44021	1.054
81	0.0625	0.52734	1.84068	0.61179	0.43874	0.267
101	0.0500	0.52766	1.84203	0.61219	0.43831	0.073
121	0.0417	0.52775	1.84239	0.61231	0.43819	0.020
141	0.0357	0.52778	1.84249	0.61235	0.43815	0.005

The grid-independence study indicates that $N_y = 121$ provides sufficient spatial resolution, since further refinement to $N_y = 141$ changes the Nusselt number by only approximately, while the variations in skin friction, velocity, and temperature are also negligible.

Time-step independence (see **Table 4**) was examined separately on the spatially converged grid. Calculations were repeated with progressively smaller temporal step sizes, while all physical parameters remained unchanged. The computed skin-friction coefficient, Nusselt number, velocity, and fluid temperature converged to mesh-independent values, demonstrating that the selected temporal resolution does

not materially affect the reported physical trends.

Table 4. Time-step-independence study, keep $N_y = 121$ fixed and successively reduce Δt.

Δt	C_f	N_u	$u(y = 1)$	$\theta_f(y = 1)$	Change in N_u(%)
1.0×10^{-2}	0.52312	1.82364	0.60782	0.44391	–
5.0×10^{-3}	0.52652	1.83765	0.61117	0.43967	0.762
2.5×10^{-3}	0.52742	1.84125	0.61203	0.43856	0.196
1.25×10^{-3}	0.52766	1.84216	0.61226	0.43828	0.049
6.25×10^{-4}	0.52772	1.84239	0.61232	0.43821	0.012
3.12×10^{-4}	0.52774	1.84245	0.61234	0.43819	0.003

The results become practically independent of the temporal resolution for $\Delta t \leq 6.25 \times 10^{-4}$. Therefore, $\Delta t = 6.25 \times 10^{-4}$ was selected for the fixed-step verification calculations, while the adaptive algorithm dynamically adjusted the time step subject to the specified tolerance.

6. Conclusion

In this study, a modified Runge-Kutta scheme was developed for the numerical solution of time-dependent non-linear partial differential equations. The proposed method was constructed as an explicit predictor–corrector framework and coupled with a compact finite-difference discretization in space, providing fourth-order spatial accuracy. The temporal coefficients of the modified scheme were selected to achieve second-order time accuracy. A von Neumann stability analysis for the scalar convection-diffusion equation and a conditional convergence analysis for the corresponding system formulation were also established, confirming the theoretical reliability of the method under the stated stability and convergence conditions. To further enhance the numerical performance, an adaptive time-stepping strategy was incorporated into the proposed scheme. The adaptive procedure automatically adjusts the time step based on the estimated local error, improving convergence and reducing the risk of instability due to an unsuitable fixed time step. Compared with current approaches based on adaptive Euler- and Runge-Kutta methods, the proposed adaptive scheme exhibited lower numerical errors and better overall convergence, indicating its suitability for solving transient transport problems. The unsteady $k - \omega$ turbulent nanofluid boundary-layer flow over a heated moving plate under local thermal non-equilibrium conditions was then applied using the numerical methodology. The physical model incorporated mixed convection, viscous dissipation, turbulence transport and independent equations of state for the base fluid and nanoparticle phases. The numerical data indicated that a higher Prandtl number decreases the thickness of the thermal boundary layers of the fluid and nanoparticle phases, whilst also increasing the mean velocity. It was also noted that the stronger the interphase coupling, the lower the base fluid temperature and the higher the nanoparticle temperature, reflecting the energy-exchange mechanism of the LTNE formulation. Moreover, the local Nusselt number increased with the base-fluid Prandtl number, indicating improved wall heat transfer performance.

Another input to this research work is the prediction of eddy viscosity using machine-learning-based methods. The predicted eddy-viscosity profiles, contour plots, and Taylor diagram analysis showed excellent agreement with the true numerical results obtained with the $k - \omega$ model. Sensitivity analysis also indicated that the velocity field is the most significant parameter in the ANN's prediction of the eddy viscosity, followed by the fluid and nanoparticle temperatures. These results suggest that machine learning can be an effective surrogate model for turbulence-related quantities and can be less expensive than repeatedly solving additional turbulence transport equations. The concluding points of the present study can be expressed as follows:

1. The proposed adaptive modified Runge-Kutta scheme, coupled with the compact finite difference discretization, provides improved numerical accuracy and better convergence than the existing adaptive Euler-based scheme for the considered non-linear PDE system.
2. The adaptive time-stepping strategy enhances stability and efficiency by automatically adjusting the time step based on the estimated local error, thereby reducing the likelihood of instability associated with a fixed time step.
3. Increasing the thermal Grashof number enhances the mean velocity due to stronger buoyancy forces, whereas increasing the Prandtl numbers of the base fluid and nanoparticles reduces the corresponding thermal boundary-layer thicknesses.
4. The machine-learning framework predicts the eddy viscosity with excellent agreement with the true numerical solution, confirming its effectiveness as a reliable and computationally efficient tool for turbulence-related analysis in the present LTNE turbulent nanofluid model.

In general, the current study shows that the proposed adaptive modified Runge-Kutta scheme is an accurate, stable, and efficient computational tool for solving coupled non-linear PDE systems that arise in turbulent nanofluid transport. Concurrently, assimilating neural learning for eddy-viscosity forecasting constitutes another data-driven platform to accelerate turbulence analysis. Future work may extend this methodology to more complex geometries, variable nanoparticle shapes, magnetic field effects, and higher-dimensional turbulent transport models.

Author contributions: Conceptualization, methodology, and analysis, MSA; funding acquisition, MSA; investigation, YN; methodology, MSA; project administration, MSA; resources, MSA; supervision, MSA; visualization, YN; writing review and editing, YN; proofreading and editing, MSA. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific grant from public, commercial, or not-for-profit funding agencies.

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: The manuscript included all required data and the

implementation of information.

Acknowledgment: The authors wish to express their gratitude to Prince Sultan University for paying the Article Processing Charges (APC) of this Publication.

Conflict of interest: The authors declare no conflict of interest.

AI use statement: The authors declare that no artificial intelligence (AI) tools were used in the preparation of this manuscript.

Abbreviation

Symbol	Definition	Unit
t	Time	s
y	Coordinate normal to the moving surface.	m
u	Fluid velocity	ms^{-1}
u_w	Moving-surface velocity	ms^{-1}
u_∞	Free-stream/reference velocity	ms^{-1}
k	Turbulent kinetic energy	m^2s^{-2}
w	Specific turbulence dissipation rate	s^{-1}
ν	Kinematic viscosity	m^2s^{-1}
ν_t	Turbulent or eddy viscosity	m^2s^{-1}
T_f	Fluid-phase temperature	K
T_s	Nanoparticle/solid-phase temperature	K
T_w	Surface temperature	K
T_∞	Ambient temperature	K
θ_f	Dimensionless fluid temperature	–
θ_s	Dimensionless nanoparticle/solid temperature	–
ρ_f	Density of base fluid	kgm^{-3}
ρ_s	Density of nanoparticles	kgm^{-3}
$c_{p,f}$	Specific heat capacity of base fluid	$J\ kg^{-1}\ K^{-1}$
$c_{p,s}$	Specific heat capacity of nanoparticles	$J\ kg^{-1}\ K^{-1}$
C_f	Skin-friction coefficient	–
N_u	Local Nusselt number	–
Δt	Temporal step size	s or dimensionless
Δy	Spatial mesh spacing	m or dimensionless
z^n	Numerical solution at time level	–
z^P	Predictor-stage solution	–
a_1	Free parameter of modified RK predictor	–
a, b, c	Modified RK coefficients	–
$\widehat{v}_t$	Machine-learning prediction of eddy viscosity	m^2s^{-1}
RMSE	Root-mean-square error	–
μ	Dynamic viscosity	$Pa\ s$
k_f	Thermal conductivity of base fluid	$Wm^{-1}K^{-1}$
k_s	Thermal conductivity of nanoparticles	$Wm^{-1}K^{-1}$
k_{nf}	Effective nanofluid thermal conductivity	$Wm^{-1}K^{-1}$
ϕ	Nanoparticle volume fraction	–
h_{fs}	Interphase heat-transfer coefficient	–
R_e	Reynolds number	–
$G_{\gamma T}$	Thermal Grashof number	–
$G_{\gamma C}$	Solutal Grashof number	–
P_{rf}	Fluid-phase Prandtl number	–
P_{rs}	Solid/nanoparticle-phase Prandtl number	–
P_{rt}	Turbulent Prandtl number	–
B_i	Biot number	–
E_c	Eckert number	–

Γ_f	Fluid-phase interphase heat-transfer parameter	–
Γ_s	Solid/nanoparticle-phase interphase heat-transfer parameter	–
σ_k	Turbulent diffusion coefficient for k	–
σ_w	Turbulent diffusion coefficient for w	–
β	$k-w$ model constant	–
β^*	$k-w$ model constant	–
γ	$k-w$ model constant	–
$D_{1,h}$	Compact discrete first-derivative operator	–
$D_{2,h}$	Compact discrete second-derivative operator	–
$ATOL$	Absolute tolerance	–
$RTOL$	Relative tolerance	–
ERR	Scaled adaptive error estimate	–
η	Step-size safety factor	–
x	Machine-learning input vector	–
MAE	Mean absolute error	–

References

1. Choi SUS, Eastman JA. Enhancing thermal conductivity of fluids with nanoparticles. In: Proceedings of the ASME International Mechanical Engineering Congress & Exposition; 12–17 November 1995; San Francisco, CA, USA.
2. Buongiorno J. Convective transport in nanofluids. Journal of Heat Transfer. 2006; 128(3): 240–250. doi: 10.1115/1.2150834
3. Varma A, Nidhul K. Thermo-hydraulic and entropy generation analysis of nanofluid flow with variable properties in various duct cross sections: A 3-D two-phase approach. Journal of Thermal Analysis and Calorimetry. 2024; 149(9): 4209–4225. doi: 10.1007/s10973-024-12947-6
4. Wilcox DC. Turbulence Modeling for CFD, 3rd ed. DCW Industries; 2006.
5. Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. AIAA Journal. 1994; 32(8): 1598–1605. doi: 10.2514/3.12149
6. Roy G, Gherasim I, Nadeau F, et al. Heat transfer performance and hydrodynamic behavior of turbulent nanofluid radial flows. International Journal of Thermal Sciences. 2012; 58: 120–129. doi: 10.1016/j.ijthermalsci.2012.02.021
7. Pati S, Borah A, Boruah MP, et al. Critical review on local thermal equilibrium and local thermal non-equilibrium approaches for the analysis of forced convective flow through porous media. International Communications in Heat and Mass Transfer. 2022; 132: 105889. doi: 10.1016/j.icheatmasstransfer.2022.105889
8. Tayebi T, Chamkha AJ, Öztop HF, et al. Local thermal non-equilibrium (LTNE) effects on thermal-free convection in a nanofluid-saturated horizontal elliptical non-Darcian porous annulus. Mathematics and Computers in Simulation. 2022; 194: 124–140. doi: 10.1016/j.matcom.2021.12.009
9. Latha N, Shankar BM, Naveen Kumar SB, et al. Perspectives on local thermal non-equilibrium (LTNE) Darcy–Bénard convection: Variable permeability and viscosity effects. Physics of Fluids. 2024; 36(10): 104131. doi: 10.1063/5.0229202
10. Nield DA, Kuznetsov AV. The effect of local thermal non-equilibrium on the onset of convection in a nanofluid. ASME Journal of Heat and Mass Transfer. 2010; 132(5): 052405. doi: 10.1115/1.4000998
11. Saleh M, Kovács E, Kallur N. Adaptive step size controllers based on Runge-Kutta and linear-neighbor methods for solving the non-stationary heat conduction equation. Networks & Heterogeneous Media. 2023; 18(3): 1059–1082. doi: 10.3934/nhm.2023048
12. Dormand JR, Prince PJ. A family of embedded Runge-Kutta formulae. Journal of Computational and Applied Mathematics. 1980; 6(1): 19–26. doi: 10.1016/0771-050X(80)90013-3
13. Roberts SB, Shirokoff D, Biswas A, et al. Runge-Kutta Methods and Stiff Order Conditions for Semilinear ODEs. arXiv preprint. 2025; arXiv:2505.15099v1. https://arxiv.org/abs/2505.15099v1
14. Savović S, Drljača B, Djordjevich A. A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes. Ricerche di Matematica. 2022; 71: 245–252. doi: 10.1007/s11587-020-00510-y
15. Ranocha H, Winters AR, Castro HG, et al. On error-based step size control for discontinuous Galerkin methods for

compressible fluid dynamics. Communications on Applied Mathematics and Computation. 2025; 7(1): 3–39. doi: 10.1007/s42967-024-00418-2
16. Naseem T, Mebarek-Oudina F, Vaidya H, et al. Numerical analysis of entropy generation in joule heated radiative viscous fluid flow over a permeable radially stretching disk. CMES-Computer Modeling in Engineering and Sciences. 2025; 143(1): 351–371. doi: 10.32604/cmes.2025.060262
17. Mezaache A, Mebarek-Oudina F, Vaidya H, et al. Heat transfer analysis of nanofluid flow with entropy generation in a corrugated heat exchanger channel partially filled with porous medium. Heat Transfer. 2024; 53(8): 4625–4647. doi: 10.1002/htj.23119
18. Shampine LF. Local extrapolation in the solution of ordinary differential equations. Mathematics of Computation. 1973; 27: 91–97. doi: 10.1090/S0025-5718-1973-0343584-0
19. Butcher JC, Johnston PB. Estimating local truncation errors for Runge-Kutta methods. Journal of Computational and Applied Mathematics. 1993; 45: 203–212. doi: 10.1016/0377-0427(93)90272-4
20. Verner JH. Explicit Runge–Kutta methods with estimates of the local truncation error. SIAM Journal on Numerical Analysis. 1978; 15: 772–790. doi: 10.1137/0715051
21. Chai AS. Error estimate of a fourth-order Runge-Kutta method with only one initial derivative evaluation. In: Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference; 30 April–2 May 1968; Atlantic City, NJ, USA. pp. 467–471. doi: 10.1145/1468075.1468136
22. Merson RH. An operational methods for study of integration processes. In: Proceedings of Conference on Data Processing and Automatic Computing Machines; 3–8 June 1957; Salisbury, Australia.
23. Alexandre dit Sandretto J. Validated B-series and Runge-Kutta pairs. Numerical Algorithms. 2024; 96(3): 1045–1062. doi: 10.1007/s11075-023-01679-1
24. Martens B. Error estimates for Runge–Kutta schemes of optimal control problems with index 1 DAEs. Computational Optimization and Applications. 2023; 86(3): 1299–1325. doi: 10.1007/s10589-023-00503-9
25. Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. Journal of Fluid Mechanics. 2016; 807: 155–166. doi: 10.1017/jfm.2016.615
26. Duraisamy K, Iaccarino G, Xiao H. Turbulence modeling in the age of data. Annual Review of Fluid Mechanics. 2019; 51(1): 357–377. doi: 10.1146/annurev-fluid-010518-040547
27. Girimaji SS. Turbulence closure modeling with machine learning: A foundational physics perspective. New Journal of Physics. 2024; 26(7): 071201. doi: 10.1088/1367-2630/ad5f6c
28. He X, Tan J, Rigas G, et al. On the explainability of machine-learning-assisted turbulence modeling for transonic flows. International Journal of Heat and Fluid Flow. 2022; 97: 109038. doi: 10.1016/j.ijheatfluidflow.2022.109038
29. Nawaz Y, Kerdid N, Arif MS, et al. AI-Assisted Hybrid Solver for Skin Friction and Sherwood Number Prediction in Eyring–Prandtl Nanofluid Flow over a Riga Plate. CMES-Computer Modeling in Engineering and Sciences. 2026; 146(2). doi: 10.32604/cmes.2025.062113
30. Tao H, Aldlemy MS, Homod RZ, et al. Hybrid nanocomposites impact on heat transfer efficiency and pressure drop in turbulent flow systems: Application of numerical and machine learning insights. Scientific Reports. 2024; 14(1): 19882. doi: 10.1038/s41598-024-70499-z
31. Babanezhad M, Behroyan I, Nakhjiri AT, et al. Thermal prediction of turbulent forced convection of nanofluid using computational fluid dynamics coupled genetic algorithm with fuzzy interface system. Scientific Reports. 2021; 11(1): 1308. doi: 10.1038/s41598-020-79897-7
32. Arif MS, Abodayeh K, Nawaz Y. A Hybrid Exponential Runge-Kutta Scheme for Stochastic SIQR Disease Modeling. European Journal of Pure and Applied Mathematics. 2025; 18(3): 6176. doi: 10.29020/nybg.ejpam.v18i3.6176
33. Zhang ZJ, Duraisamy K. Machine learning methods for data-driven turbulence modeling. In: Proceedings of the 22nd AIAA Computational Fluid Dynamics Conference; 22–26 June 2015; Dallas, TX, USA. doi: 10.2514/6.2015-2460
34. Xiao H, Wu JL, Wang JX, et al. Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed Bayesian approach. Journal of Computational Physics. 2016; 324: 115–136. doi: 10.1016/j.jcp.2016.07.038
35. Wu JL, Xiao H, Paterson E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Physical Review Fluids. 2018; 3: 074602. doi: 10.1103/PhysRevFluids.3.074602
36. Matai R, Durbin P. Zonal eddy viscosity models based on machine learning. Flow, Turbulence and Combustion. 2019; 103: 93–109. doi: 10.1007/s10494-019-00009-1
37. Maulik R, San O, Jacob JD, et al. Sub-grid scale model classification and blending through deep learning. Journal of Fluid Mechanics. 2019; 870: 784–812. doi: 10.1017/jfm.2019.254

38. Gamahara M, Hattori Y. Searching for turbulence models by artificial neural network. Physical Review Fluids. 2017; 2: 054604. doi: 10.1103/PhysRevFluids.2.054604
39. Sriharan G, Mebarek-Oudina F, Harikrishnan S, et al. Experimental and machine-learning investigation of ZnO nanofluid flow in a triangular-absorber parabolic trough collector. Modern Physics Letters B. 2026; 40(15): 2650108. doi: 10.1142/S0217984926501084
40. Vollant A, Balarac G, Corre C. Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures. Journal of Turbulence. 2017; 18: 854–878. doi: 10.1080/14685248.2017.1334904
41. Maulik R, San O, Rasheed A, et al. Data-driven deconvolution for large eddy simulations of kraichnan turbulence. Physics of Fluids. 2018; 30: 125109. doi: 10.1063/1.5063941
42. Fayz-Al-Asad M, Mebarek-Oudina F, Vaidya H, et al. Finite element analysis for magneto-convection heat transfer performance in vertical wavy surface enclosure: fin size impact. Frontiers in Heat and Mass Transfer. 2024; 22(3): 817–837. doi: 10.32604/fhmt.2024.049702
43. Fukami K, Fukagata K, Taira K. Super-resolution reconstruction of turbulent flows with machine learning. Journal of Fluid Mechanics. 2019; 870: 106–120. doi: 10.1017/jfm.2019.238
44. Zhao Y, Akolekar HD, Weatheritt J, et al. Turbulence model development using cfd-driven machine learning. arXiv preprint. 2019; arXiv:1902.09075. https://arxiv.org/abs/1902.09075
45. Zhang X, Wu J, Coutier-Delgosha O, et al. Recent progress in augmenting turbulence models with physics-informed machine learning. Journal of Hydrodynamics. 2019; 31: 1153–1158. doi: 10.1007/s42241-019-0093-z