Comparative Physics Informed Neural Network Study of Tetra and Ternary Nano lubricant Systems in MHD Flow

Praveen Kumar U M[1], Venkata Sundaranand Putcha[2]

[1,2]Department of Mathematics, Rayalaseema University, Kurnool, 518007, Andhra Pradesh, India.

Abstract:

This study investigates heat and mass transfer characteristics of Casson hybrid nanofluids flowing over a stretching surface in the presence of a magnetic field, Joule heating, and thermal radiation. Two advanced nano-lubricant configurations are examined. The first is a ternary hybrid nanofluid consisting of Al_2O_3, ZnO, and SiC nanoparticles suspended in engine oil, while the second is a tetra hybrid nanofluid formed by adding graphene nanoplatelets (GNPs) to the ternary mixture. The objective is to compare how nanoparticle composition influences thermal conductivity, flow behaviour, and energy dissipation mechanisms. The mathematical formulation incorporates effects of chemical reaction, thermo-diffusion, Joule heating, and radiative heat flux. Through suitable similarity transformations, the governing nonlinear partial differential equations are reduced into a coupled system of ordinary differential equations. These equations are solved using a Physics Informed Neural Network (PINN) approach developed specifically for nanofluid lubrication systems. The proposed PINN architecture embeds the governing physical laws directly into the loss function, allowing simultaneous minimization of equation residuals and boundary condition errors. This two-way optimization enhances solution stability and computational efficiency. Numerical results indicate that the tetra hybrid nanofluid provides significantly higher thermal enhancement and reduced entropy generation compared to the ternary formulation. This performance improvement is mainly due to the exceptional thermal conductivity and large surface area of GNPs. However, the ternary hybrid nanofluid exhibits relatively stable viscous behavior with moderate temperature gradients. Overall, the PINN framework shows strong convergence and predictive accuracy, offering a reliable computational tool for complex nonlinear thermal-fluid problems in lubrication applications.

Keywords: PINN, Nanolubricants, MHD flow, Joule heating, Thermal radiation

1. Introduction

At present, there are continuous efforts by the researchers in dealing with new techniques of combining nanoparticles and base fluids such as engine oil for making nanofluids that conduct heat more efficiently. Investigations are conducted to optimize particle size, shape, concentration, and surface properties for enhanced thermal conductivity and stability. These engineered nanofluids may find applications in electronics cooling, heat exchangers, collectors, and automotive applications. In particular, hybrid nanofluids are advanced materials with improved heat transfer and thermal stability that can consider becoming strong contenders for thermal management in modern applications. Computer modeling or numerical simulations have proven indispensable to forecasting their practical behavior; they reveal how hybrid nanofluids transport heat and mass in complex conditions. Basically, these nanofluid systems, in particular, deal with lubrication under various physical effects, such as MHD, thermal radiation, and Joule heating [1,7]. These combined factors generate nonlinear PDEs that are difficult to handle by means of the usual techniques. The need for quicker and more accurate computational tools is therefore quite obvious. In this study, the MHD effect is modeled through a Lorentz force term in the momentum equation. This adds some resistive drag and reduces the velocity, which is in agreement with magnetohydrodynamic theory.

Nanofluid behavior depends on several physical phenomena that together shape heat and mass transfer. Understanding these effects is critiacal for optimizing advanced applications, especially lubrication systems where precise thermal control matters [5]. MHD explains the behavior of electrically conducting fluids in magnetic fields, such as Casson hybrid nanofluids ($MoS_2 + ZnO$) based on engine oil [1,5,8]. Magnetic fields impact motion and heat transfer, impacting everything from biomedical drug delivery to nuclear cooling and MHD generators [8]. The flow is resisted by the Lorentz force, which thickens the thermal boundary layer and slows velocity. Nusselt and Sherwood numbers decrease as velocity decreases and heat and mass transfer rates decrease as the magnetic parameter M rises. Fe_3O_4 and Al_2O_3 hybrid nanofluids exhibit additional performance improvements, which are essential for cooling and energy technologies. Thermal radiation becomes crucial in high-temperature applica- tions like aerospace, reactors, and power systems [9,10]. In nanofluids, increasing radiation raises velocity and temperature since it adds heat energy, though sometimes overall temperature may drop depending on conditions. It affects both heat and mass transfer rates usually decreasing the Nusselt number and raising the Sherwood number. In this study, radiation is modeled using the Rosseland approximation, applied in linear or nonlinear T^4 form within the energy equation. Joule (Ohmic) heating, during which the electric current generates internal heat, is one of the important factors in MHD systems. The rise of temperature and thickening boundary layer are important in many areas of electronics cooling, manufacturing, and energy systems [1]. Joule heating is modeled via the σB2u2 term in the energy equation. The flow through a porous medium is taken care of using the Darcy resistance term. Internal heat generation is described by a uniform heat source Q_0. Recent analyses demonstrate that tetra-hybrid nanofluids with four nanoparticles are superior to the ternary types. For instance, Adnan et al. reported the enhancement in heat and mass transport in magnetically and radiatively affected flow of a tetra nanofluid $[(Al_2O_3 - CuO - TiO_2 - Ag)/water]$ up to 7.4% and 4.3%, respectively [31,32,33]. Motivated by these findings, the present study investigates a hybrid ($Al_2O_3 + ZnO + SiC$) tetrabasic nanofluid and its counterpart including ternary GNP nanofluid through a Physics-Informed Neural Network (PINN) solver subject to the MHD, Joule heating, and thermal radiation effects toward high-fidelity mesh-free simulations.

1.1 Problems of Traditional Approaches

The nonlinear effect of MHD, radiation, Joule heating, chemical reactions, and Brownian motion combines to yield a complex PDE that is hard to solve [23]. Traditional methods like the FEM are indeed powerful yet slow; they involve several trial runs in order for it to manage variable boundary conditions, which cannot allow their use in real time. Many systems cannot reduce them into simplified ODEs, while handling unbounded conditions that is, zero at infinity remains problematic [23,25]. With the Runge-Kutta scheme in mind, even numerical schemes hardly get an exact solution out of them. In addition, creating labeled data with traditional solvers involves huge computations and requires much time. High-fidelity finite element tools are just too slow for creating the large datasets or real-time predictions needed in Industry 4.0 environments.

1.2 The Imperative for Advanced Computational PINNs

The advent of Machine Learning (ML) and Artificial Intelligence (AI) has presented an unprecedented opportunity to develop fast, near real-time simulation capabilities crucial for Industry 4.0 settings and for overcoming the limitations of conventional solvers [2,3]. PINNs emerge as a groundbreaking solution, capable of directly incorporating the governing PDEs into their training process by defining the loss function as the error in satisfying the physical laws [13].Unlike traditional supervised machine learning problems that require given labeled output data, PINNs are designed for scenarios where the actual outputs (solutions to equations) are unknown. This dilemma is resolved by leveraging prior knowledge of the underlying physics, elucidated by the governing equations themselves. The core idea is that the neural network's output is expected to satisfy the governing equations for every node within the domain (collocation points) in addition to the boundary nodes. The loss function in PINNs is therefore defined based on two primary terms:

i.Mean Squared Error for Boundary/Initial Conditions (MSEu): This term measures the discrepancy between the expected and estimated values at the boundaries and initial states, ensuring boundary conditions are met.

ii.Mean Squared Error for Residuals (MSEf): This term quantifies the error resulting from substituting the neural network's predicted output into the corresponding PDEs, thus enforcing the physical laws.

The sum of these losses constitutes the total loss, which the network optimizes by adjusting its weights and biases. This innovative approach offers several significant advantages over traditional numerical methods:

i.Elimination of Pre-Generated Training Data: A key advantage of PINNs is that they do not require extensive pre-generated training data from traditional Finite Element (FE) simulations [14].

ii.Faster Computational Times: Once trained, PINNs offer significantly faster computational times compared to FE models, making them highly suitable for industrial applications that require rapid predictions and real-time feedback loops [15].

iii.Versatility in Problem Solving: PINNs can solve both forward problems (pre-ducting solutions of governing equations) and inverse problems (inferring unknown parameters, material properties, or constitutive relationships from observed data, even if noisy or incomplete) [14,15].

iv.Generalization Capabilities: PINNs can predict outcomes even outside their initial training zone, a limitation often faced by purely data-driven models [3].

v.Automatic Differentiation: They leverage automatic differentiation to precisely enforce physical laws, enabling efficient calculation of gradients for complex nonlinear PDEs [17].

vi.Unbounded Boundary Conditions: PINNs are effective in solving problems with unbounded domains and boundary conditions at infinity, a challenge for many conventional numerical techniques.

1.3 PINN Architecture and Optimization

A typical PINN architecture involves a fully connected neural network (also known as a feedforward deep neural network or DNN) that approximates the unknown solution functions. The network consists of an input layer, several hidden layers (depth), and an output layer, with neurons in each layer (width). The weights and biases within these layers are the learnable parameters that are adjusted during the training process. Hyper parameter optimization (HPO) is a critical aspect of PINN performance, as the efficiency and accuracy of the solver are significantly impacted by the neural architecture's hyper parameters [19]. Key hyper parameters include:

- Width and Depth: The number of neurons in hidden layers (width) and the number of hidden layers (depth) directly influence the network's capacity to approximate complex functions. Empirically, DNN size needs to be carefully determined; smaller networks may lack accuracy, while excessively large ones can be difficult to train with limited data.
- Activation Functions: These functions introduce nonlinearity into the network, enabling it to learn complex relationships. Common choices include Tanh, Sigmoid, ReLU, and Swish. The selection of an appropriate activation function is problem- dependent and can significantly affect convergence rate and accuracy. Adaptive activation functions have been proposed to accelerate convergence and improve accuracy, especially in the early training period [17,18].
- Learning Rate: This parameter controls the step size during the optimization process. A learning rate that is too high can lead to instability, while one that is too low can result in slow convergence.

Optimizing these hyper parameters is challenging due to the large search space and difficulty in identifying a suitable search objective for PDEs. Advanced HPO methods, such as Bayesian optimization and Neural Architecture Search (NAS), are being explored [14,19]. Auto-PINN proposes automated HPO by decoupling hyper- parameter search spaces and using loss value as the search objective. Despite their strengths, PINNs can sometimes struggle with problems exhibiting high-frequency and/or multi-scale solutions [16]. To address this, advanced architectures like Finite Basis Physics-Informed Neural Networks (FBPINNs) and their extension, Multilevel FBPINNs, have been developed [20]. These methods combine PINNs with overlap- ping domain decomposition approaches, significantly outperforming standard PINNs in solving complex, high-frequency, and multi-scale problems by improving global communication between subdomains.

1.4 Applications of PINNs

PINNs are being used in a wide range of engineering and scientific contexts:

Heat Transfer and Manufacturing: PINNs have been successfully applied to heat transfer equations in advanced manufacturing processes, including phase trans- formations in additive manufacturing and convective curing of thermoset composites [9]. They can model complex heat conduction problems with convective boundary conditions.

- **Fluid Dynamics:** PINNs are effective for simulating complex fluid flows, including bioconvection, where they can model heat, mass, and micro-organism transfer phenomena in porous media under the influence of thermal radiation. They have been used to solve classical boundary layer flow equations, such as Blasius, Falkner-Skan, and Natural convection problems, particularly those with unbounded domains [3, 6].
- **Inverse Problems:** PINNs excel in solving inverse problems, such as inferring unknown material properties, constitutive relationships, or even the underlying differential equations themselves from limited or noisy data [12]. This is particularly valuable when direct measurements are difficult to obtain.
- **Multi-phase Flow:** PINNs, including variants like the physics-informed radial basis function neural network (PIRBFNN), are being used for efficiently modeling complex systems like oil-water two-phase Darcy flow in porous media, offering significant speed advantages over traditional methods [3].

1.5 Proposed PINN Framework for MHD Hybrid Nano lubricants:

To address the nonlinear coupling and complexity in thermal-fluid transport involving advanced tetra hybrid nano lubricants, this study proposes a PINN framework tailored to magnetohydrodynamic (MHD) flow systems under radiative and reactive conditions [6, 8, 10]. The model encodes governing laws including momentum, energy, mass trans- port, and electromagnetic effects directly into the loss function, enabling mesh-free and data-efficient resolution of the transformed differential equations [12, 13, 16].Two lubricant formulations are analyzed: a ternary nanofluid $\mathrm{Al_2O_3}$, ZnO, and SiC in engine oil) and a tetra variant enhanced with graphene nanoplatelets (GNP), chosen for their distinct thermal and viscous behaviors [2, 5, 7]. The PINN approach uses a two-way optimization strategy that minimizes both PDE residuals and deviation from boundary conditions and reference numerical benchmarks, ensuring convergence, physical consistency, and high accuracy [17,18,19]. This hybrid modeling approach offers new insights into entropy generation, flow stability, and heat transfer performance, providing a robust computational alternative to traditional solvers in real-world lubrication system design [4, 8, 9].

Section 3 introduces Physics-Informed Neural Networks (PINNs), activation functions, loss construction, and the two-stage optimization process along with implementation results. Section 4 presents a study on the interdependence of the magnetic parameter (M), porous medium parameter (Kp), Prandtl number (Pr), and other related parameters. It also includes a comparative analysis with recent works based on numerical methods. Finally, Section 5 concludes the paper with key observations.

Governing Equations

Let’s look at a mathematical expression for incompressible Casson hybrid nanofluids flowing across a long wall in two dimensions (2D). Figure 1 shows the problem’s design, which includes Cartesian coordinates (x, y) and velocity components (u, v) in a fluid flow arrangement. At the interface of a magnetic field, changing viscosity, and a porous material, scientists look at how mass moves and how heat moves. Along the x-axis, the magnetic field $\mathrm{B_0}$is applied.

The rules for this problem are as follows:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{1}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\frac{\mu_{qnf}}{\rho_{qnf}}\left(1+\frac{1}{\beta}\right)\frac{\partial^2 u}{\partial y^2}-\frac{\sigma B_0^2 u}{\rho_{qnf}}-\frac{\mu_{qnf}u}{\rho_{qnf}k^*}+g\beta_C(C-C_\infty)g+\beta_T(T-T_\infty) \tag{2}$$

$$u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=\frac{\mu_{qnf}}{(\rho C_p)_{qnf}}\left(\frac{\partial u}{\partial y}\right)^2+\frac{k_{qnf}}{(\rho C_p)_{qnf}}\frac{\partial^2 T}{\partial y^2}+\frac{\sigma B_0^2 u^2}{(\rho C_p)_{qnf}}+\frac{Q_0}{(\rho C_p)_{qnf}}(T-T_\infty)-\frac{1}{(\rho C_p)_{qnf}}\frac{\partial q_r}{\partial y} \tag{3}$$

$$u\frac{\partial C}{\partial x}+v\frac{\partial C}{\partial y}=\frac{D_m K_T}{T_m}\frac{\partial^2 T}{\partial y^2}+D_m\frac{\partial^2 C}{\partial y^2}-K_r(C-C_\infty) \tag{4}$$

with boundary conditions: At $y=0$: $u = 1$, $v = 0$, $T = 1$, $C = 1$

As $y \to \infty$: $u \to 0$, $T \to 0$, $C \to 0$

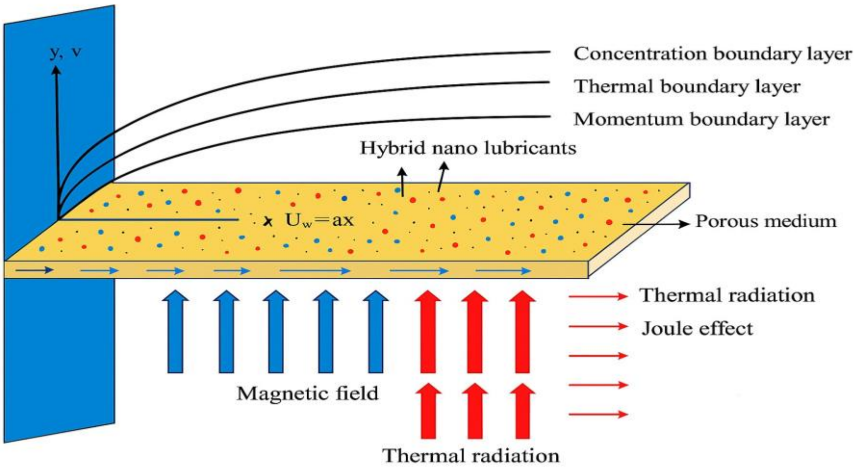

Fig. 1 Physical configuration and coordinate system of the problem.

Dimensionless Governing Equations

Introducing the following dimensionless variables:

$$x^*=\frac{x}{L},\quad y^*=\frac{y}{L},\quad u^*=\frac{u}{U_0},\quad v^*=\frac{v}{U_0},\qquad T^*=\frac{T-T_\infty}{T_w-T_\infty},\quad C^*=\frac{C-C_\infty}{C_w-C_\infty}$$

and the dimensionless parameters:

$$\mathrm{Re}^*=\frac{\rho_{qnf}U_0L}{\mu_{qnf}},\mathrm{Pr}^*=\frac{\mu_{qnf}C_{pqnf}}{k_{qnf}},\mathrm{Sc}^*=\frac{\mu_{nf}}{\rho_{qnf}D_m},\quad M^*=\frac{\sigma B_0^2L^2}{\mu_{qnf}},\qquad K_p^*=\frac{L^2}{k^*\nu_{qnf}},$$

$$Q^*=\frac{Q_0L^2}{\rho_{qnf}C_{pqnf}\nu_{qnf}},\quad S_r^*=\frac{K_T}{T_mSc^*},K_r^*=\frac{K_rL^2}{\nu_{qnf}},Ec^*=\frac{U_0^2}{C_{pqnf}(T_w-T_\infty)},$$

The dimensionless PDEs are:

$$\frac{\partial u^*}{\partial x^*}+\frac{\partial v^*}{\partial y^*}=0, \tag{5}$$

$$u^*\frac{\partial u^*}{\partial x^*}+v^*\frac{\partial u^*}{\partial y^*}=\frac{1}{Re^*}\left(1+\frac{1}{\beta}\right)\frac{\partial^2 u^*}{\partial y^{*2}}-M^*u^*-K_p^*u^*+Gr_T^*T^*+Gr_C^*C^*, \tag{6}$$

$$u^*\frac{\partial T^*}{\partial x^*}+v^*\frac{\partial T^*}{\partial y^*}=\frac{1}{Re^*Pr^*}\frac{\partial^2 T^*}{\partial y^{*2}}+Ec^*\left(\frac{\partial u^*}{\partial y^*}\right)^2+M_T^*(u^*)^2-Q_r^*\frac{\partial q_r^*}{\partial y^*}+Q^*T^* \tag{7}$$

$$u^*\frac{\partial C^*}{\partial x^*}+v^*\frac{\partial C^*}{\partial y^*}=\frac{1}{Re^*Sc^*}\frac{\partial^2 C^*}{\partial y^{*2}}+S_r^*\frac{\partial^2 T^*}{\partial y^{*2}}-K_r^*C^* \tag{8}$$

The associated boundary conditions are:

$$\begin{aligned}&\text{at}\quad y^*=0: u^*=1,\ v^*=0,\ T^*=1,\ C^*=1,\\ &\text{as}\quad y^*\to\infty:\ \ u^*\to 0,\ \ T^*\to 0,\ \ C^*\to 0.\end{aligned}$$

The flow behavior of the hybrid nanofluids in this study is thoroughly explained using the thermophysical characteristics of the nano lubricants. The volume fractions ϕ_1^*, ϕ_2^*, and ϕ_3^*represent the concentration of the nanoparticles in the hybrid nanofluids. In the case of the tetra nanofluid, nanoparticles such as Al_2O_3, ZnO, and $SiC$are mixed with base engine oil. For the tetra nanofluid, the volume fractions are set as $\phi_1^*=0.01$, $\phi_2^*=0.01$, and $\phi_3^*=0.01$for the Al_2O_3, ZnO, and SiCnanoparticles, respectively.In contrast, the ternary nanofluid system, which is enhanced with graphene nanoplatelets (GNP), includes the nanoparticles Al_2O_3, ZnO, SiC, and GNP, blended with engine oil. The volume fractions for this ternary system are $\phi_1^*=0.01$, $\phi_2^*=0.01$, $\phi_3^*=0.01$, and $\phi_4^*=0.01$for the Al_2O_3, ZnO, SiC, and GNP nanoparticles, respectively.For clarity, Table1 outlines the thermophysical characteristics of both the tetra and ternary hybrid nanofluids. These properties are sourced from various literature reviews to ensure the accuracy of the data. Additionally, Table2 and Table3 provides a detailed overview of the thermophysical properties of the engine oil used as the base fluid in both nanofluid formulations.

Table 1: Thermophysical properties of Al_2O_3, ZnO, SiC, and GNP nanoparticles with engine oil

$Property$	Al_2O_3	ZnO	SiC	GNP	$Engine\ Oil$
$Density, \rho\ (kg/m^3)$	3970	5600	3210	2200	884
$Specific\ heat, C_p\ (J/kg\cdot K)$	765	495.2	600	700	1910
$Thermal\ conductivity, k\ (W/m\cdot K)$	36	13	120	200	0.144

Table 2. Thermophysical property models for ternary nanolubricant (Al_2O_3 + ZnO + SiC + Engine Oil)

Property	**Ternary Nanolubricant Formula**
Viscosity	$\mu_{tnf} = \frac{\mu_f}{(1-\phi_3)^{2.5}(1-\phi_2)^{2.5}(1-\phi_1)^{2.5}}$
Density	$\rho_{tnf} = \rho_f(1-\phi_1-\phi_2-\phi_3)+\phi_1\rho_1+\phi_2\rho_2+\phi_3\rho_3$
Heat Capacity	$(\rho C_p)_{tnf} = (1-\phi_1-\phi_2-\phi_3)(\rho C_p)_f+\phi_1(\rho C_p)_1+\phi_2(\rho C_p)_2+\phi_3(\rho C_p)_3$
Thermal Conductivity	$\frac{k_{tnf}}{k_f} = \frac{2k_f+k_{eq}-2\phi_t(k_f-k_{eq})}{k_{eq}+2k_f+\phi_t(k_f-k_{eq})}$
Equivalent Conductivity (Nanoparticles)	$k_{eq} = \frac{\phi_1k_1+\phi_2k_2+\phi_3k_3}{\phi_1+\phi_2+\phi_3}$

Similarity ODEs for Casson Hybrid Nanofluid

Similarity transformations:

$$\left.\begin{aligned} u = axf'(\eta), v = \left(\sqrt{a\nu_f}\right)f(\eta), \eta = \sqrt{\frac{a}{\nu_f}}y, \\ \theta(\eta) = \frac{T-T_\infty}{T_w-T_\infty}, \phi(\eta) = \frac{C-C_\infty}{C_w-C_\infty} \end{aligned}\right\} \tag{9}$$

The transformed non-dimensional ordinary differential equations are:

$$\left(1+\frac{1}{\beta}\right)f'''(\eta) \quad -K_pf'(\eta) - S_4Mf'(\eta) -$$

$$S_1[(f'(\eta))^2 - f(\eta)f''(\eta)] + \lambda\,\theta(\eta) + N\,\phi(\eta) = 0(10) \tag{10}$$

$$(1+R)\,\theta''(\eta) + \frac{Pr}{S_6}\left[\ \frac{Ec}{S_5}(f''(\eta))^2 + M(f'(\eta))^2 + Q\,\theta(\eta) + f(\eta)\theta'(\eta)\right] = 0 \tag{11}$$

$$\phi''(\eta) - Sc\,K_r\,\phi(\eta) - Sc\,f(\eta)\phi'(\eta) - Sc\,Sr\,\theta''(\eta) = 0 \tag{12}$$

Boundary conditions

$$f(0) = 0, f'(0) = 1, \theta(0) = 1, \phi(0) = 1,$$

$$f'(\infty) \to 0, \theta(\infty) \to 0, \phi(\infty) \to 0.$$

Table 3. Thermophysical property models for tetra nanolubricant (Al_2O_3 + ZnO + SiC + GNP + Engine Oil)

Property	**Tetra Nanolubricant Formula**
Viscosity	$\mu_{qnf} = \frac{\mu_f}{(1-\phi_4)^{2.5}(1-\phi_3)^{2.5}(1-\phi_2)^{2.5}(1-\phi_1)^{2.5}}$
Density	$\rho_{qnf} = \rho_f(1-\phi_1-\phi_2-\phi_3-\phi_4) + \phi_1\rho_1 + \phi_2\rho_2 + \phi_3\rho_3 + \phi_4\rho_4$
Volumetric Heat Capacity	$(\rho C_p)_{qnf} = (1-\phi_1-\phi_2-\phi_3-\phi_4)(\rho C_p)_f + \phi_1(\rho C_p)_1 + \phi_2(\rho C_p)_2 + \phi_3(\rho C_p)_3 + \phi_4(\rho C_p)_4$
Thermal Conductivity	$\frac{k_{qnf}}{k_f} = \frac{k_{eq}+2k_f-2\phi_t(k_f-k_{eq})}{k_{eq}+2k_f+\phi_t(k_f-k_{eq})}$
Equivalent Nanoparticle Conductivity	$k_{eq} = \frac{\phi_1 k_1+\phi_2 k_2+\phi_3 k_3+\phi_4 k_4}{\phi_1+\phi_2+\phi_3+\phi_4}$

where the following parameters are represented:

M for the magnetic parameter, Sc for the Schmidt number,K_p for the porosity parameter, Sr for the Soret effect parameter, K_r for the chemical reaction parameter, Pr for the Prandtl number, Ec for the Eckert number, Q for the heat source, β for the Casson fluid parameter, Jh for the Joule heating parameter, and R for the thermal radiation parameter.

The dimensionless parameters are:

$$M = \frac{\sigma B_0^2}{\rho_f a}, \quad K_p = \frac{\nu_f}{ak^*}, \quad Pr = \frac{\nu_f(\rho C_p)_f}{k_f}, \quad Sc = \frac{\nu_f}{D_m}, Sr = \frac{D_m K_T}{T_m \nu_f}(T_w - T_\infty),$$

$$Q = \frac{Q_0}{(\rho C_p)_f(T_w-T_\infty)}, \quad Ec = \frac{a^2x^2}{(\rho C_p)_f(T_w-T_\infty)}, R = \frac{4\sigma^* T_\infty^3}{k^* k_{nf}}, \lambda = \frac{Gr}{Re^2}, \quad N = \frac{Gc}{Re^2}$$

Table 4: Thermophysical property ratios for ternary nanolubricant (Al_2O_3 + ZnO + SiC / Engine Oil)

Symbol	Property Ratio	Formula
S_1	Dynamic Viscosity	$\frac{\mu_{tnf}}{\mu_f} = \frac{1}{(1-\phi_1)^{2.5}(1-\phi_2)^{2.5}(1-\phi_3)^{2.5}}$
S_2	Schmidt Number	$\frac{Sc_{tnf}}{Sc_f} = \frac{\mu_{tnf}\rho_f}{\mu_f\rho_{tnf}}$
S_3	Kinematic	$\frac{\nu_{tnf}}{\nu_f} = \frac{\mu_{tnf}\rho_f}{\mu_f\rho_{tnf}}$

Symbol	Property Ratio	Formula
	Viscosity	
S₄	Density	$\frac{\rho_{tnf}}{\rho_f} = (1-\phi_1-\phi_2-\phi_3) + \frac{\phi_1\rho_1+\phi_2\rho_2+\phi_3\rho_3}{\rho_f}$
S₅	Heat Capacity	$\frac{(\rho C_p)_{tnf}}{(\rho C_p)_f} = (1-\phi_1-\phi_2-\phi_3) + \frac{\phi_1(\rho C_p)_1+\phi_2(\rho C_p)_2+\phi_3(\rho C_p)_3}{(\rho C_p)_f}$
S₆	Thermal Conductivity	$k_{eq} = \frac{\phi_1k_1+\phi_2k_2+\phi_3k_3}{\phi_1+\phi_2+\phi_3}, \frac{k_{tnf}}{k_f} = \frac{k_{eq}+2k_f+2\phi_t(k_{eq}-k_f)}{k_{eq}+2k_f-\phi_t(k_{eq}-k_f)},$ $\phi_t = \phi_1+\phi_2+\phi_3$

Physical quantities of interest

$$C_{fx} = \frac{\tau_w}{\rho_f U_w^2}, Nu_x = \frac{x\,q_w}{k_f(T_w-T_\infty)}, Sh_x = \frac{x\,q_m}{D_B(C_w-C_\infty)} \tag{13}$$

$$\tau_w = \mu_{hnf}\frac{\partial u}{\partial y}|_{y=0}, \ q_w = -k_{hnf}\frac{\partial T}{\partial y}|_{y=0}, \quad q_m = -D_B\frac{\partial C}{\partial y}|_{y=0}$$

Similarity transformed quantities

$$C_{fx}(2Re_x)^{1/2} = \frac{\mu_{hnf}}{\mu_f}f''(0), Nu_x(Re_x)^{-1/2} = -\frac{k_{hnf}}{k_f}\theta'(0), \ Sh_x(Re_x)^{-1/2} = -\phi'(0)$$

Where $Re_x = \frac{U_w x}{\nu_f}$ (14)

Table 5: Thermophysical property ratios for tetra nanolubricant (Al_2O_3+ZnO+SiC+GNP/Engine Oil)

Symbol	Property Ratio	*Formula*
S₁	Dynamic Viscosity	$\frac{\mu_{qnf}}{\mu_f} = \frac{1}{(1-\phi_1)^{2.5}(1-\phi_2)^{2.5}(1-\phi_3)^{2.5}(1-\phi_4)^{2.5}}$

Symbol	Property Ratio	*Formula*
S₂	Schmidt Number	$\frac{Sc_{qnf}}{Sc_f} = \frac{\mu_{qnf}\rho_f}{\mu_f\rho_{qnf}}$
S₃	Kinematic Viscosity	$\frac{\nu_{qnf}}{\nu_f} = \frac{\mu_{qnf}\rho_f}{\mu_f\rho_{qnf}}$
S₄	Density	$\frac{\rho_{qnf}}{\rho_f} = (1-\phi_1-\phi_2-\phi_3-\phi_4) + \frac{\phi_1\rho_1+\phi_2\rho_2+\phi_3\rho_3+\phi_4\rho_4}{\rho_f}$
S₅	Heat Capacity	$\frac{(\rho C_p)_{qnf}}{(\rho C_p)_f} = (1-\phi_1-\phi_2-\phi_3-\phi_4) +$ $\frac{\phi_1(\rho C_p)_1+\phi_2(\rho C_p)_2+\phi_3(\rho C_p)_3+\phi_4(\rho C_p)_4}{(\rho C_p)_f}$
S₆	Thermal Conductivity	$\frac{k_{qnf}}{k_f} = \frac{k_{eq}+2k_f+2\phi_t(k_{eq}-k_f)}{k_{eq}+2k_f-\phi_t(k_{eq}-k_f)}$, $k_{eq} = \frac{\phi_1k_1+\phi_2k_2+\phi_3k_3+\phi_4k_4}{\phi_1+\phi_2+\phi_3+\phi_4}$ $\phi_t = \phi_1+\phi_2+\phi_3+\phi_4$

3.Physics-informed neural networks (PINNs):

3.1 Discretization and node structure:

In the ODE setting, the computational domain is simplified to a single spatial variable, the transverse coordinate $y^* \in [0, y_{max}]$. All computations were performed at ηmax = 20; ηmax = 7 was shown sufficient for hydrodynamic and thermal fields, while still keeping the training problem manageable. Two different kinds of nodes are introduced. The first are boundary nodes, placed at $y^* = 0$ and $y^* = y_{max}$, which impose the wall values and far-field conditions directly into the loss function to ensure the network respects the essential physical limits. The second are collocation nodes, scattered throughout the interior of the interval, whose role is to enforce the residuals of the governing ODEs, making the learned solution consistent throughout the domain. Collocation nodes are generated by Latin hypercube sampling, which provides more even coverage than a uniform mesh and avoids undesirable clustering. A slightly denser concentration of nodes is maintained near the wall region ($y^* = 0$) because sharp gradients often appear there, and insufficient resolution could prevent the network from capturing them accurately. Let $N_f$be the number of collocation nodes and N_u the number of boundary nodes. Both sets are essential: the boundary nodes anchor the solution to the prescribed physics, while the collocation nodes drive the network to reduce the ODE residuals inside the domain. This balance forms the backbone of the PINN discretization strategy for the one-dimensional problem. Figure 2 illustrate the overall Physics-Informed Neural Network (PINN) framework used in this work, highlighting the formation of physics-based residual, boundary condition enforcement, and the two stage optimization strategy.

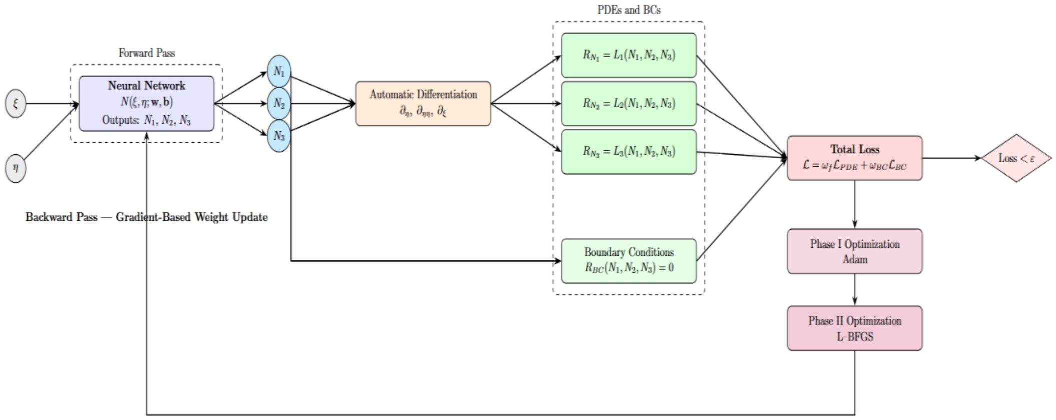
PDEs and BCs
Forward Pass
ξ
η
Neural Network
$N(\xi,\eta;\mathbf{w},\mathbf{b})$
Outputs: N_1, N_2, N_3
N_1
N_2
N_3
Automatic Differentiation
∂_η, $\partial_{\eta\eta}$, ∂_ξ
$R_{N_1} = L_1(N_1, N_2, N_3)$
$R_{N_2} = L_2(N_1, N_2, N_3)$
$R_{N_3} = L_3(N_1, N_2, N_3)$
Boundary Conditions
$R_{BC}(N_1, N_2, N_3) = 0$
Total Loss
$\mathcal{L} = \omega_f \mathcal{L}_{PDE} + \omega_{BC}\mathcal{L}_{BC}$
Loss $< \varepsilon$
Phase I Optimization
Adam
Phase II Optimization
L-BFGS
Backward Pass — Gradient-Based Weight Update

Figure 2: PINN architecture used for the present study

3.2 Function construction:

The purpose of the PINN model is to approximate the unknown functions $u^*(y^*)$, $T^*(y^*)$, and $C^*(y^*)$that satisfy the reduced one-dimensional equations. The network takes the coordinate y^*as input and produces the predicted physical quantities as output. Symbolically, the mapping can be written as

$$\widehat{\boldsymbol{U}}(y^*;\boldsymbol{w},\boldsymbol{b}) = [\hat{u}^*(y^*), \hat{T}^*(y^*), \hat{C}^*(y^*)],$$

where $\boldsymbol{w}$ and $\boldsymbol{b}$ are the weights and biases that evolve during training.

Automatic differentiation is applied to compute derivatives of $\widehat{\boldsymbol{U}}$ with respect to y^*. This is important because the governing equations involve first and second derivatives. Using exact gradients from the computational graph avoids numerical errors from finite differences. The resulting expressions are then inserted into the governing ODEs to construct residuals for momentum, thermal energy, and concentration.

The momentum residual is expressed as

$$R_U(y^*) = \frac{d^2\hat{u}^*(y^*)}{dy^{*2}} - \frac{Re^*}{1+1/\beta}\left(M^*\hat{u}^*(y^*) + K_p^*\hat{u}^*(y^*) - Gr_T^*\hat{T}^*(y^*) - Gr_C^*\hat{C}^*(y^*)\right),$$

the thermal residual is

$$R_T(y^*) = \frac{d^2\hat{T}^*(y^*)}{dy^{*2}} - Re^*Pr^*(-Ec^*(\frac{d\hat{u}^*(y^*)}{dy^*})^2 - M_T^*(\hat{u}^*(y^*))^2 + Q_r^*\frac{d\hat{q}_r^*(y^*)}{dy^*} - Q^*\hat{T}^*(y^*))\,,$$

and the concentration residual is

$$R_C(y^*) = \frac{d^2\hat{C}^*(y^*)}{dy^{*2}} - Re^*Sc^*(-S_r^*\frac{d^2\hat{T}^*(y^*)}{dy^{*2}} + K_r^*\hat{C}^*(y^*)).$$

Each residual must vanish across the domain for the predicted solution to satisfy the physics. In practice, they do not vanish exactly during training, but the network is optimized to minimize them. This is how the PINN combines learning with physical consistency: the residuals push the model toward the correct dynamics, while the boundary conditions anchor it to the prescribed values at $y^* = 0$and $y^* = y_{max}$.

3.3 Optimization process:

The optimization routine is the key step that allows the neural network to approximate the actual solution of the ODE system. The network parameters, denoted by $\boldsymbol{w}$ and $\boldsymbol{b}$, are not fixed; they are gradually updated in a way that reduces the loss, which measures how far the predictions deviate from the physics and the boundary conditions. Initially, the network often struggles to produce meaningful outputs, and the residuals are large. As training progresses, the optimizer slowly pushes the solution toward consistency with both the governing equations and the prescribed boundaries. In practice, a two-stage optimization is employed. The first stage uses the Adam algorithm, which is stochastic and adaptive, making it suitable for the early part of training when the gradients are noisy and unstable. Once the model reaches a region close to the correct solution, the second stage uses the L-BFGS method, a quasi-Newton algorithm. Unlike Adam, L-BFGS is deterministic and tends to provide more accurate convergence. Combining the two methods avoids slow progress at the start while preventing premature stagnation at the end.

The total loss is defined as the weighted sum of two contributions: the boundary mismatch and the residual mismatch, written as

$$\mathcal{L}(\boldsymbol{w},\boldsymbol{b}) = \omega_u\, MSE_u(\boldsymbol{w},\boldsymbol{b}) + \omega_f\, MSE_f(\boldsymbol{w},\boldsymbol{b}),$$

where $\omega_u$and $\omega_f$control the relative influence of each term. The boundary part is

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} \parallel \boldsymbol{U}(\,y_i^*) - \widehat{\boldsymbol{U}}(y_i^*;\boldsymbol{w},\boldsymbol{b})\parallel^2,$$

while the residual part is

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}(\mid R_U(y_i^*)\mid^2 + \mid R_T(y_i^*)\mid^2 + \mid R_C(y_i^*)\mid^2).$$

During training, the weights and biases are updated according to the gradient of the loss function. In the Adam phase, the step size (or learning rate) changes adaptively based on moving averages of past gradients, while in the L-BFGS phase, the algorithm uses curvature information to choose better search directions and step lengths. Optimization concludes when either the loss stops decreasing or falls below a specified tolerance, at which point the network has learned a set of parameters that minimize disagreement with the ODE system and the boundary data. This two-stage strategy is now standard in many PINN studies because it combines the robustness of stochastic training with the precision of second-order methods, producing stable and reliable results.

3.4 Loss functions:

The design of the loss function is what drives the PINN toward the correct solution of the ODE system. Without it, the network would simply approximate arbitrary curves, ignoring the underlying physics. By carefully constructing the loss, the training process forces the network to honor both the governing differential equations and the boundary conditions at the wall and at the far field.

The first component of the loss comes from the momentum equation, written as

$$MSE_{R_1} = \frac{1}{N_f}\sum_{i=1}^{N_f} \left| \frac{d^2\hat{u}^*}{dy^{*2}} \right|_{y_i^*} - \frac{Re^*}{1+1/\beta}\left(M^*\hat{u}^*(y_i^*) + K_p^*\hat{u}^*(y_i^*) - Gr_T^*\hat{T}^*(y_i^*) - Gr_C^*\hat{C}^*(y_i^*)\right) |^2.$$

The second contribution is obtained from the thermal energy equation,

$$MSE_{R_2} = \frac{1}{N_f}\sum_{i=1}^{N_f} \left| \frac{d^2\hat{T}^*}{dy^{*2}} \right|_{y_i^*} - Re^*Pr^*\left(-Ec^*\left(\frac{d\hat{u}^*}{dy^*}\right)^2_{y_i^*} - M_T^*(\hat{u}^*(y_i^*))^2 + Q_r^*\frac{d\hat{q}_r^*}{dy^*} \right|_{y_i^*} - Q^*\hat{T}^*(y_i^*)\right) |^2.$$

The final residual term comes from the concentration equation,

$$MSE_{R_3} = \frac{1}{N_f}\sum_{i=1}^{N_f} \left| \frac{d^2\hat{C}^*}{dy^{*2}} \right|_{y_i^*} - Re^*Sc^*\left(-S_r^*\frac{d^2\hat{T}^*}{dy^{*2}} \right|_{y_i^*} + K_r^*\hat{C}^*(y_i^*)\right) |^2.$$

By collecting all three residuals, the physics-informed part of the loss becomes

$$MSE_f = MSE_{R_1} + MSE_{R_2} + MSE_{R_3}.$$

In addition, the loss also accounts for the mismatch at the boundary nodes,

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u}\left(|\,\hat{u}^*(y_i^*) - u_{\mathrm{BC}}^*(y_i^*)\,|^2 + |\,\hat{T}^*(y_i^*) - T_{\mathrm{BC}}^*(y_i^*)\,|^2 + |\,\hat{C}^*(y_i^*) - C_{\mathrm{BC}}^*(y_i^*)\,|^2\right).$$

The total loss that drives the training is finally written as

$$\mathcal{L} = \omega_u\, MSE_u + \omega_f\, MSE_f,$$

where ω_u and ω_f serve as balancing weights. If either is chosen too large, the network may focus too much on one part and ignore the other. In practice, both are tuned so that the ODE residuals and the boundary conditions are treated with equal importance.

3.5 Implementation

This research utilizes a PINN framework to examine heat and mass transfer in Casson hybrid nanofluids subjected to Joule heating and thermal radiation in the presence of a stretching surface and magnetic field. We are looking at two different nano-lubricant systems: a tetra formulation with Al_2O_3, ZnO, and SiC nanoparticles mixed into engine oil, and a ternary formulation with graphene nanoplatelets (GNP) added. The aim is to examine how particle composition influences thermal transport, viscous stability, and entropy generation, while also testing the flexibility of PINNs in capturing strongly coupled nonlinear phenomena. Training happens in two steps. The Adam optimizer is used first to explore the parameter space on a large scale. This makes it easier to quickly lower the loss function in the early epochs. The L-BFGS optimizer comes next. It improves convergence by using quasi-Newton updates, which effectively smooths out the solution landscape. This two-step method uses the speed of stochastic optimization while keeping the accuracy of deterministic solvers, making sure that the governing equations are solved quickly and correctly.

In the tables 6 and 7, the activation function is a key factor in how quickly the PINN models converge. Among all tested functions, Tanh consistently gave the best results. For the ternary nanofluid, it reached a total loss of 1.07×10^{-6} after L-BFGS, far lower than the Adam baseline (8.32×10^{-5}), while requiring only 54 s of training. The tetra nanofluid showed a similar pattern, with Tanh achieving 1.08×10^{-6} in 75 s. Other functions did not perform as well. Mish offered competitive accuracy but needed more than twice the time, while Swish and Sigmoid stabilized at higher losses. GELU, Softplus, and Softmin struggled with convergence, especially in the tetra case where Softmin plateaued at 1.79×10^{-3}. These outcomes highlight that while several activations can produce acceptable solutions, Tanh provides the most reliable balance of speed and accuracy.

Table 6: Performance of different activation functions for the ternary nanolubricant using Adam and L-BFGS optimizers.

Activation	Loss BCs	Loss PDEs	Total Loss	Total Loss Adam	Time (s)
Tanh	4.55×10^{-10}	1.07×10^{-6}	1.07×10^{-6}	8.32×10^{-5}	54.21
Sigmoid	4.94×10^{-9}	2.70×10^{-6}	2.70×10^{-6}	4.30×10^{-4}	77.98
Mish	1.70×10^{-10}	2.11×10^{-6}	2.11×10^{-6}	1.06×10^{-4}	156.92
GELU	1.12×10^{-7}	3.28×10^{-5}	3.29×10^{-5}	3.31×10^{-5}	82.44

Activation	Loss BCs	Loss PDEs	Total Loss	Total Loss Adam	Time (s)
Softplus	1.04×10^{-8}	1.01×10^{-5}	1.01×10^{-5}	8.05×10^{-4}	95.62
Softmin	1.14×10^{-7}	6.19×10^{-5}	6.20×10^{-5}	1.57×10^{-3}	109.24
Softmax	1.14×10^{-7}	2.00×10^{-5}	2.01×10^{-5}	1.57×10^{-3}	105.28
Swish	3.04×10^{-10}	6.27×10^{-5}	6.28×10^{-5}	3.83×10^{-4}	107.97

In the figure 3, different learning rates, from 10^{-1} to 10^{-5}, were tested under the two-stage Adam to L-BFGS optimization. The results show clear differences in stability and accuracy. At the largest rate, 10^{-1}, the training became unstable, with strong oscillations that prevented effective loss reduction. At the opposite extreme, very small rates (10^{-4}, 10^{-5}) converged too slowly, failing to minimize residuals within the practical training window. The most favorable outcome was obtained with a learning rate of 10^{-2}. Both ternary and tetra nanofluids converged smoothly at this setting, with Adam driving a sharp early decline in loss and L-BFGS refining the solution to very low residuals. Compared to 10^{-3}, the 10^{-2} case reached lower final losses more efficiently, confirming it as the optimal choice.

Table 7: Performance of different activation functions for the tetra nanolubricant using Adam and L-BFGS optimizers.

Activation	Loss BCs	Loss PDEs	Total Loss	Total Loss Adam	Time (s)
Tanh	5.74×10^{-11}	1.08×10^{-6}	1.08×10^{-6}	1.57×10^{-4}	75.81
Sigmoid	2.15×10^{-10}	5.09×10^{-5}	5.09×10^{-5}	4.60×10^{-4}	77.84
Mish	2.47×10^{-9}	2.01×10^{-6}	2.91×10^{-6}	1.60×10^{-4}	161.22
GELU	2.37×10^{-9}	2.91×10^{-6}	2.91×10^{-6}	4.15×10^{-5}	98.73
Softplus	2.16×10^{-9}	4.34×10^{-6}	4.34×10^{-6}	4.22×10^{-4}	83.88
Softmin	2.61×10^{-5}	1.76×10^{-3}	1.79×10^{-3}	1.79×10^{-3}	74.65
Softmax	3.71×10^{-8}	1.36×10^{-5}	1.37×10^{-5}	1.15×10^{-3}	98.72

Activation	Loss BCs	Loss PDEs	Total Loss	Total Loss Adam	Time (s)
Swish	6.65×10^{-7}	8.96×10^{-5}	9.02×10^{-5}	9.17×10^{-5}	100.53

The influence of hidden neuron count was studied for 40, 60, 80, and 100 neurons per layer. Both ternary and tetra nanofluids showed similar trends. Smaller networks with 40 neurons converged, but the final residuals plateaued at higher values, suggesting underfitting. At the other extreme, 100 neurons initially drove a sharp loss reduction, but the curves flattened too early, leaving higher residuals and showing signs of instability. The most consistent and balanced performance was achieved with 60 neurons figure 4 in both fluids, this configuration produced smooth convergence under Adam and further refinement with L-BFGS, reaching the lowest final losses. Networks with 80 neurons performed reasonably well but required longer training and showed less stability in later epochs. Overall, 60 neurons per layer provided the optimal trade-off between learning capacity and stability, ensuring accurate and efficient convergence across both nanofluid systems.

3.6 Physical Accuracy Validation

To evaluate the prediction accuracy of the proposed PINN framework, the obtained solution was compared with a high accuracy numerical solution computed using a collocation-based boundary value solver (bvp) with tolerance of 10^{-9}. The accuracy of velocity, temperature, and concentration profile was measured using the L_2 error norm. In addition, the wall related quantities such as the skin friction coefficient (C_f) and the Nusselt number (Nu) were evaluated through relative error analysis. The result show that the errors in velocity and temperature profiles are in the range of 10^{-4} to 10^{-3}, while the concentration profile error remain below 1.2×10^{-2}. Moreover, the relative error for the skin friction coefficient is less than 0.07%, and for the Nusselt number is below 0.6%. These outcomes confirm that the proposed PINN model provide high numerical accuracy and successfully capture both wall shear stress and heat transfer behavior in the coupled nonlinear system.

Table 8: L_2 Profile Errors and Relative Wall Quantity Errors (PINN vs Reference)

Case	Pr	$L_2(f')$	$L_2(\theta)$	Rel. Error (C_f)	Rel. Error (Nu)
Ternary	7	6.93e-4	6.40e-4	6.41e-4	5.21e-3
Ternary	14	7.21e-4	3.84e-4	1.40e-4	1.49e-3
Ternary	21	7.52e-4	3.61e-4	1.13e-4	1.36e-3
Ternary	28	7.15e-4	4.34e-4	2.20e-5	1.38e-3
Tetra	7	3.74e-4	1.09e-4	1.28e-4	5.49e-4

Tetra	14	7.23e-4	3.78e-4	1.10e-5	1.48e-3
Tetra	21	6.61e-4	3.70e-4	1.17e-4	1.67e-3
Tetra	28	8.82e-4	5.70e-4	5.11e-4	3.28e-3

4. Results and Discussion

In this study, two nanolubricant formulations were investigated:

i. a tetra-component nanofluid composed of Al_2O_3, ZnO, and SiC dispersed in engine oil;
ii. a ternary nanofluid further enhanced with graphene nanoplatelets (GNP).

Table 9: Comparison of $-\theta'(0)$ with the results of Asmat Ullah et al., Shami A.M. et al., and P. Shreedevi et al. for various values of the Prandtl number (Pr), with $\phi_1^* = 0$, $\phi_2^* = 0$, $\phi_3^* = 0$, $\phi_4^* = 0$, and all other parameters set to zero.

Pr	**Asmat Ullah [30]**	**Shami A.M. [23]**	**P. Shreedevi [32]**	**Our Results**	
				(PINN)	**SciPy bvp**
2.0	0.9112	0.91138	0.911341	0.911287	0.91152
6.13	1.7597	1.75965	1.759676	1.759228	1.759357
7.0	1.8953	1.8955	1.895397	1.894204	1.894159
20.0	3.3540	--	3.353915	3.352994	3.352651

According to Table 9, the obtained results are validated by comparing them with previous Pr results in limiting cases. Figures 5–15 present the comparison for both ternary nanolubricant (Al_2O_3, ZnO, SiC in engine oil) and tetra nanolubricant with graphene nanoplatelets (GNPs). Figures 5 and 6 show the influence of magnetic parameter M and porous medium parameter Kp on the velocity profile f′(η). For both nanofluid systems, increasing M and Kp significantly suppress the flow. This occur because stronger magnetic field generate higher Lorentz force that oppose the motion, while lower porosity increase the resistance inside the medium. Figure 7 demonstrate the effect of Casson parameter on velocity. As the Casson parameter increase, the velocity profile decrease, indicating the shear thinning behavior of Casson type fluid. This decrease lead to thinner hydrodynamic boundary layer and the effect is more noticeable in the ternary formulation due to higher viscous resistance. Figure 8 show that increasing Pr reduce the thermal boundary layer thickness, resulting in steeper temperature gradient near the wall, which agree with classical boundary layer theory since higher Pr mean lower thermal diffusivity. Figure 9 analyze the heat source parameter Q. Increasing Q raise the temperature profile because of extra energy supplied to the system. Similarly, Figure 10 show the effect of Eckert number Ec, where higher Ec increase temperature due to viscous dissipation converting mechanical energy into heat. In both cases, the tetra nanofluid exhibit stronger temperature enhancement because GNP improve thermal conductivity. Figures 11 illustrate the solutal parameter effects on concentration. Increasing Schmidt number Sc reduce the concentration profile since higher Sc correspond to lower mass diffusivity and thinner concentration boundary layer. Figure 13 show that higher Soret number increase concentration due to stronger thermo diffusion driven by temperature gradient. Figure 15 present the concentration profile under varying radiation parameter R, where stronger radiation increase concentration by enhancing solutal diffusion indirectly. Figure 12 show the influence of chemical reaction parameter Kr. As Kr increase, concentration decrease because stronger reactions consume species faster. Figure 14 display the effect of radiation parameter R on thermal profile $\theta(\eta)$. Increasing R raise the temperature distribution and thicken the thermal boundary layer. This effect is more pronounced in the ternary nanofluid due to enhanced radiative absorption caused by graphene nanoplatelets.

Conclusion:

This study investigated the capability of Physics-Informed Neural Networks (PINNs) for modeling Casson hybrid nanofluids under magnetic field, Joule heating, thermal radiation, and chemical reaction effects. Two systems were analyzed, namely ternary nanofluid (Al_2O_3, ZnO, SiC in engine oil) and tetra nanofluid enhanced with graphene nanoplatelets. From the computational analysis, Tanh activation function produced the lowest residual loss, a learning rate of 0.01 ensured stable convergence, and 60 hidden neurons provided sufficient learning capacity. The two stage Adam–L-BFGS optimization significantly improved convergence and accuracy, achieving loss values around 10^{-6}.

The parametric study revealed that increasing heat source and Eckert number intensified temperature profiles, while radiation thickened the thermal boundary layer. Schmidt number and chemical reaction reduced concentration, whereas Soret number enhanced solutal diffusion. The tetra nanofluid showed superior thermal enhancement, while the ternary nanofluid maintained more stable viscous behavior. The proposed PINN framework demonstrates strong stability, accuracy, and adaptability for nonlinear thermal-fluid problems in lubrication systems, making it a promising tool for future engineering applications.

Declarations

Conflict of interest: The author(s) declare that there are no conflicts of interest regarding the publication of this article.

Data availability: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding: This research received no external funding.

Nomenclature

Symbol	Description
S_1	Dynamic viscosity ratio of nanofluid to base fluid
S_2	Schmidt number ratio
S_3	Ratio of kinematic viscosities
S_4	Density ratio
S_5	Effective heat capacity ratio
S_6	Thermal conductivity ratio
f(η)	Dimensionless stream function
ωf	Weight associated with physics-based residuals in PINN
x, y	Cartesian spatial coordinates
qm	Mass flux at the wall surface
β	Casson fluid parameter
σ*	Stefan–Boltzmann constant
U0	Reference velocity scale
φ(η)	Dimensionless concentration profile
Nf	Number of interior collocation points
k*	Mean absorption coefficient
λ	Mixed convection parameter
w, b	Neural network weights and biases
θ(η)	Dimensionless temperature distribution
B_0	Strength of the applied magnetic field
f′(η)	Dimensionless velocity function
τw	Shear stress at the wall
keq	Effective thermal conductivity of hybrid nanoparticles
ωu	Boundary-condition loss weighting factor
Rex	Local Reynolds number
u, v	Velocity components in the x and y directions
φt	Total nanoparticle volume fraction
Nu	Number of boundary training points
qw	Heat flux at the wall
η	Similarity coordinate

Subscripts

Subscript	Description
tnf	Ternary nanofluid
qnf	Tetra nanofluid

References

2. Sarthak, S., Srinivasacharya, D.: PINN modeling for MHD natural convection: Thermal radiation and cross-diffusion effects on vertical plate dynamics. Eur. J. Mech. B Fluids 114, 204307 (2025)

3. Kumar, P.B.A.: Comprehensive investigations for entropy generation on MHD flow of non-Newtonian tetra hybrid nanofluid flow: Numerical with neural networks approaches. Eur. Phys. J. Plus (in press)
4. Bashir, M., Al-Hussein, A.: On the application of physics-informed neural networks (PINN) to solve boundary layer thermal-fluid problems. Int. Commun. Heat Mass Transf. 132, 105890 (2024)
5. Dandu, S., Chitrapu, V.R.M., Kodi, R.: Impact of thermal radiation and chemical reactions on Casson hybrid nanofluid flow through porous media with unstable mixed convection. (preprint)
6. Madiwal, S., Naduvinamani, N.B.: Heat and mass transformation of Casson hybrid nanofluid (MoS_2 + ZnO) over a stretched wall with chemical reaction and thermo-diffusion. Lubricants 12(6), 221 (2024)
7. Kodi, R.: Heat and mass transfer of unsteady MHD nanofluid flow past a vertical porous plate with radiation and Soret effects. J. Nanofluids 12(4), 767–776 (2023)
8. Raghunath, K., Guedri, K., Ali, F., Alnuwaiser, M.A., Ibrahim, S.M., Fayz-Al-Asad, M., Kumar, B.C.P.: MHD Darcy–Forchheimer squeezing Casson fluid flow through porous media. Sci. Rep. 13, 1–18 (2023)
9. Bhargava, K., Ibrahim, S.M., Kodi, R.: Magnetohydrodynamic mixed convection radiating 3D hybrid nanofluid flow through porous media. Math. Model. Numer. Simul. Appl. 4(4), 495–513 (2024)
10. Zobeiry, N., Humfeld, K.D.: Physics-informed machine learning for solving heat transfer equations. Eng. Appl. Artif. Intell. 101, 104232 (2021)
11. Shanmugapriya, M., Sundareswaran, R., Senthil Kumar, P.: Heat and mass transfer enhancement of MHD hybrid nanofluid flow. Int. J. Chem. Eng. 2021, 1–12 (2021)
12. Kumar, Y.S., Hussain, S., Raghunath, K., Ali, F., Guedri, K., Eldin, S.M., Khan, M.I.: Numerical analysis of MHD Casson nanofluid flow with activation energy and radiation. Sci. Rep. 13, 29702 (2023)
13. Tartakovsky, A.M., Perdikaris, P., Karniadakis, G.E.: Physics-informed deep neural networks for hydrological systems. Water Resour. Res. 56(5), e2019WR026731 (2020)
14. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks for nonlinear PDEs. J. Comput. Phys. 378, 686–707 (2019)
15. Wang, Y., Han, X., Chang, C.Y., Zha, D., Braga-Neto, U., Hu, X.: Auto-PINN: Understanding and optimizing physics-informed neural architectures. (preprint)
16. Belbute-Peres, F.A., Sha, F., Chen, Y.-F.: HyperPINN: Learning parameterized differential equations with hypernetworks. (preprint)
17. Al-Hussein, A.: Physics-informed neural network framework for 2D and 3D obstacle problems. Ph.D. Thesis (2023)
18. Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E.: Adaptive activation functions accelerate convergence in PINNs. J. Comput. Phys. 404, 109136 (2020)
19. Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E.: Hyperparameter tuning of PINNs with Helmholtz applications. (preprint)
20. Karl, A., Pielok, T.: Multi-objective hyperparameter optimization in machine learning. (preprint, 2023)
21. Dolean, V., Heinlein, A., Mishra, S., Moseley, B.: Multilevel domain decomposition architectures for PINNs. (preprint)

22. Subrahmanyam, P. V., Kathyayani, G., Venkata Ramudu, G., Venkatadri, K. The stagnation point flow of the MHD Casson polymeric nanofluid flows toward a wavy circular cylinder saturated with a porous medium under convective Nield conditions and thermal radiation. East European Journal of Physics, 3, 168–179 (2025). DOI 10.26565/2312-4334-2025-3-15
23. Zienkiewicz, O.C., Taylor, R.L.: The Finite Element Method, Vol. 1: The Basis. Butterworth-Heinemann, Oxford (2000)
24. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods: Fundamentals in Single Domains. Springer, Berlin (2007)
25. Kharazmi, E., Zhang, Z., Karniadakis, G.E.: Variational physics-informed neural networks. arXiv preprint arXiv:1912.00873 (2019)
26. Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E.: XPINNs: A domain decomposition framework for PINNs. Commun. Comput. Phys. 28(5), 2002–2041 (2020). DOI 10.4208/cicp.OA-2020-0164
27. Wang, S., Teng, Y., Perdikaris, P.: Understanding and mitigating gradient pathologies in PINNs. SIAM J. Sci. Comput. 43(5), A3055–A3081 (2021). DOI 10.1137/20M1318043
28. Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F.: Scientific machine learning through PINNs. J. Sci. Comput. 92(3), 88 (2022). DOI 10.1007/s10915-022-01939-z
29. Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural networks for solving PDEs. IEEE Trans. Neural Netw. 9(5), 987–1000 (1998). DOI 10.1109/72.712178
30. Adnan, M., Aman, S., Hussain, A., Khan, U., Khan, A.: MHD tetra-hybrid nanofluid flow with entropy generation. Heat Transfer 52(6), 5373–5392 (2023). DOI 10.1002/htj.22823
31. Ullah, A., Yahya, N.S., Huang, W.-H., Siddique, I., Abdal, S., Hussain, S.: Thermal characteristics of Williamson hybrid nanofluid flow. Case Stud. Therm. Eng. 26, 101196 (2021). DOI 10.1016/j.csite.2021.101196
32. Alsallami, S.A.M., Abbas, T., Al-Zubaidi, A., Khan, S.U., Saleem, S.: Analytical assessment of Williamson hybrid nanofluid heat transfer. Case Stud. Therm. Eng. 44, 103593 (2023). DOI 10.1016/j.csite.2023.103593
33. Sreedevi, P., Sudarsana Reddy, P., Chamkha, A.: Heat and mass transfer in unsteady hybrid nanofluid flow. SN Appl. Sci. 2, 1222 (2020). DOI 10.1007/s42452-020-3011-x
34. Venkata Ramudu, G., Kathyayani, G., Venkatadri, K. Magnetohydrodynamic double-diffusive convection of Casson fluid in an irregular pentagonal cavity with radiation and chemical reaction effects. Computational Mathematics and Modeling (2025). DOI 10.1007/s10598-025-09631-1
35. Alsallami, S. A. M., Abbas, T., Al-Zubaidi, A., Khan, S. U., Saleem, S. (2023). Analytical assessment of heat transfer due to Williamson hybrid nanofluid (MoS_2 + ZnO) with engine oil base material due to stretched sheet. Case Studies in Thermal Engineering, 44, 103593. https://doi.org/10.1016/j.csite.2023.103593.

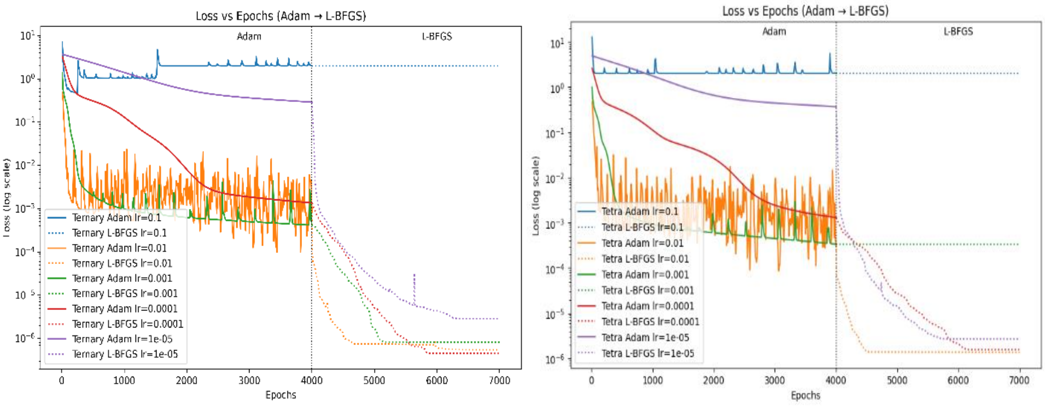

Figure 3: Loss versus epochs for different learning rates (Adam → L-BFGS).

Right: Ternary nanolubricant. Left: Tetra nanolubrican

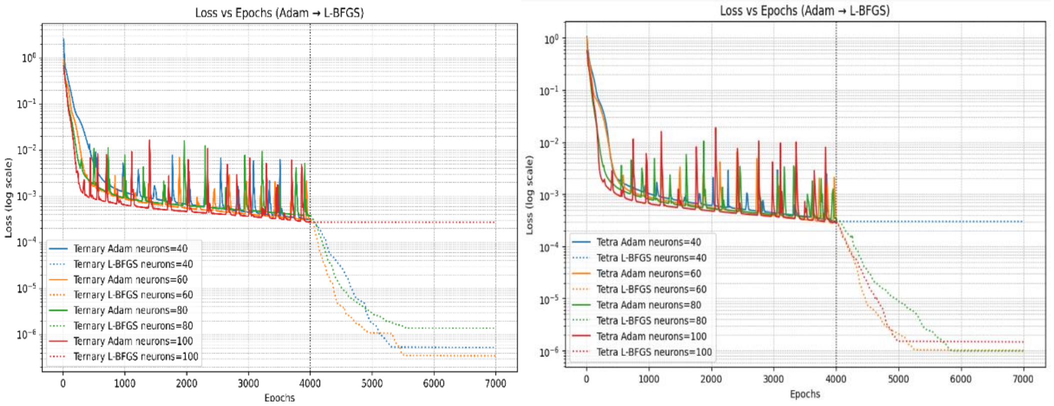

Figure 4 Loss versus epochs for different numbers of neurons (Adam → L-BFGS).
Right: Ternary nanolubricant. Left: Tetra nanolubricant.

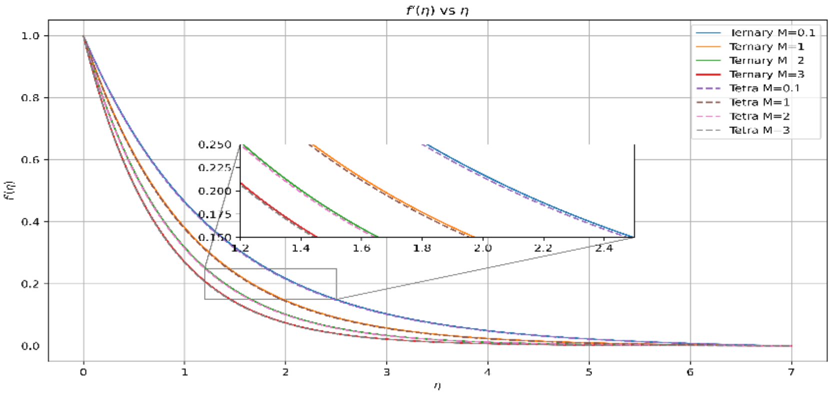

Fig. 5 f' impact on M.

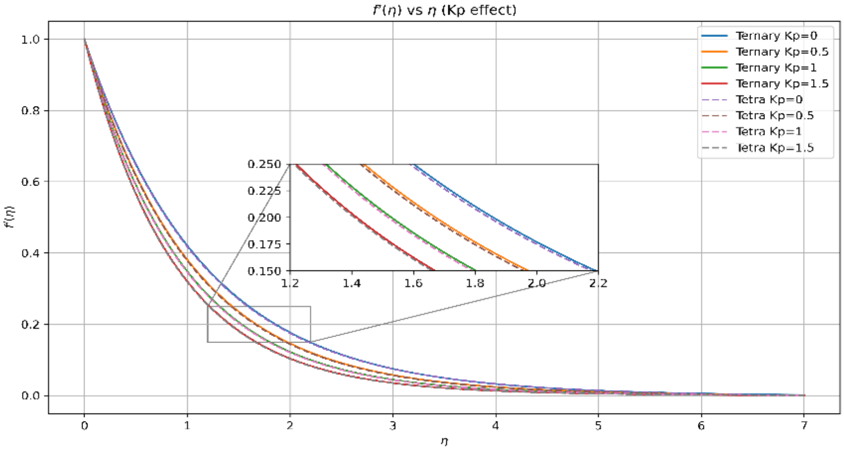

Fig. 6 f′ impact on Kp

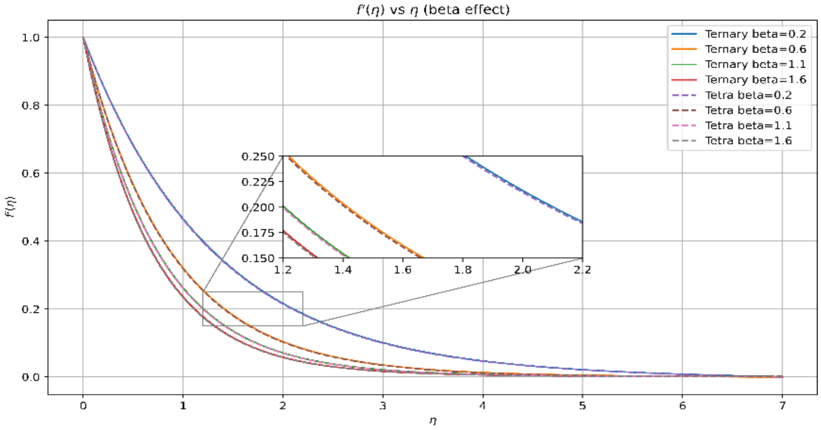

Fig. 7 f′ impact on β

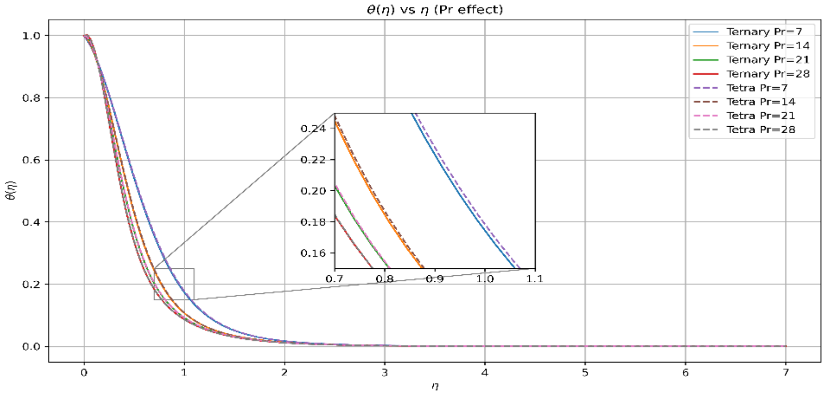

Fig. 8 θ impact on Pr

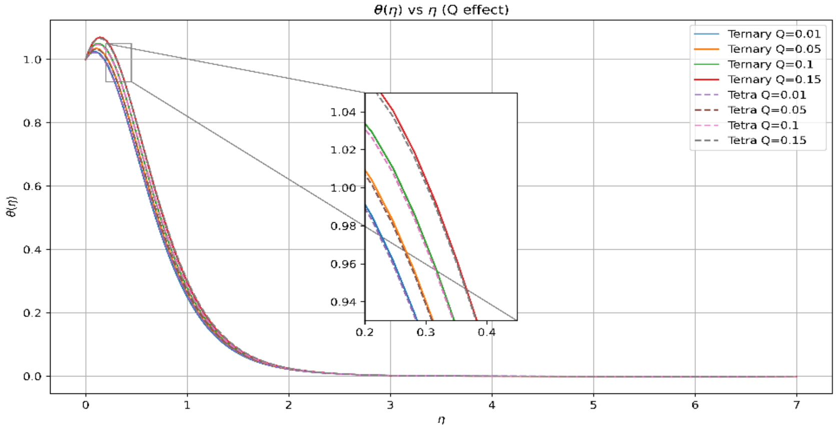

Fig. 9 θ impact on Q

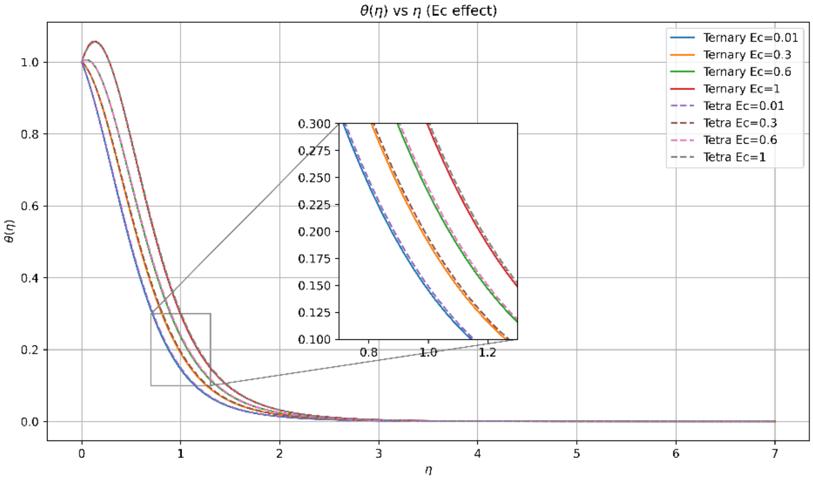

Fig. 10 θ impact on Ec

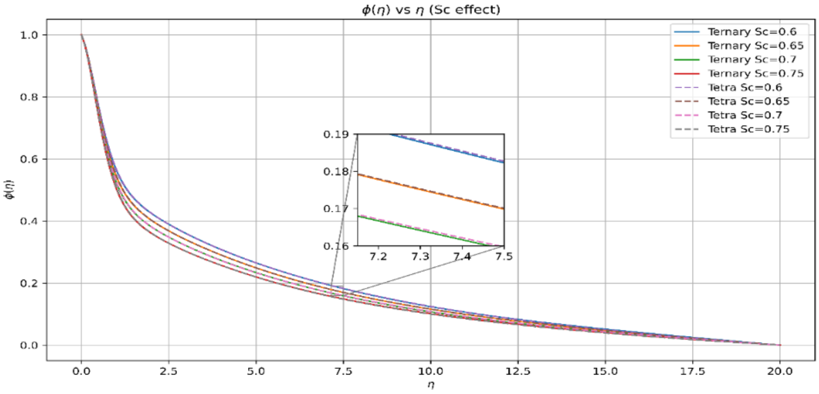

Fig. 11 ϕ impact on Sc

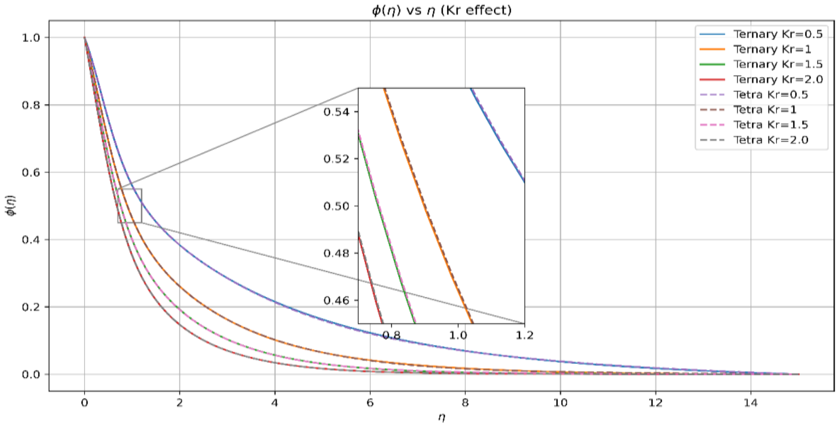

Fig. 12 ϕ impact on Kr

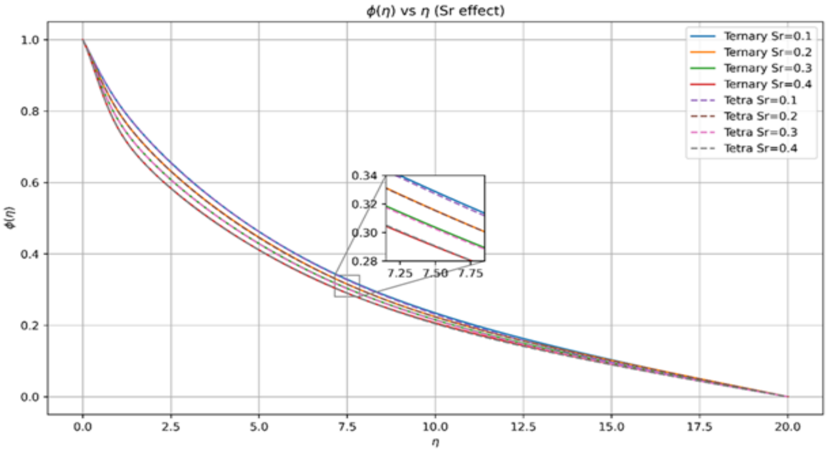

Fig. 13 ϕ impact on Sr

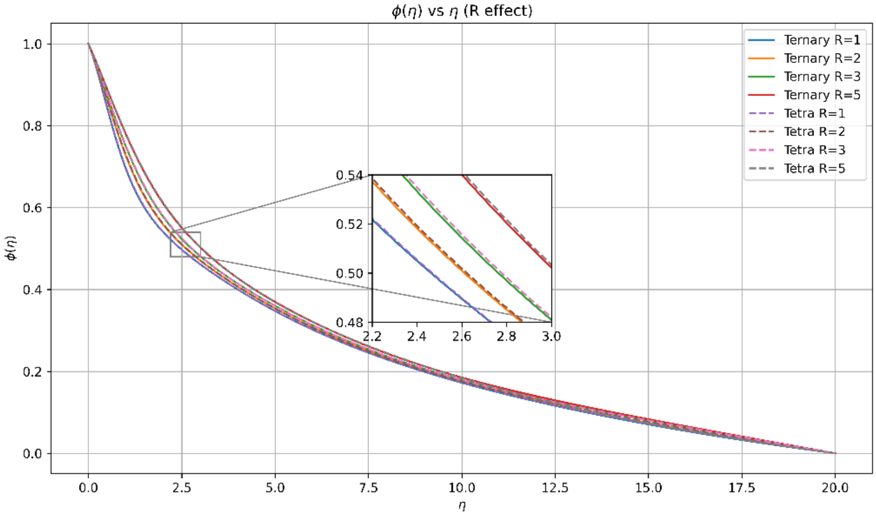

Fig. 14 ϕ impact on R

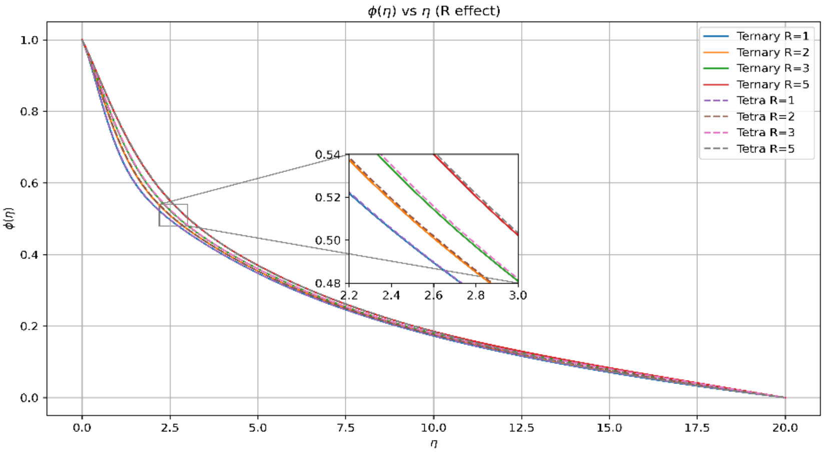

Fig. 15 ϕ impact on R