International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Physics-Informed Neural Network Framework for MHD Casson Ternary and Tetra Hybrid Nanolubricant Flow

Praveen Kumar U M[1]* and Venkata Sundaranand Putcha[2]

Submitted: 10/12/2025 **Revised**: 17/01/2026 **Accepted**: 13/02/2026

Abstract: The heat and mass transport properties of Casson hybrid nanofluids flowing across a stretched surface in the presence of thermal radiation, Joule heating, and a magnetic field are examined in this work. We look at two sophisticated nano-lubricant arrangements. Al_2O_3, ZnO, and SiC nanoparticles suspended in engine oil make up the first ternary hybrid nanofluid. Graphene nanoplatelets (GNPs) are added to the ternary mixture to create the second tetra hybrid nanofluid. Comparing the effects of nanoparticle composition on energy dissipation mechanisms, flow behavior, and thermal conductivity is the aim. Joule heating, radiative heat flux, thermo-diffusion, and chemical reaction effects are all included in the mathematical formulation. The controlling nonlinear partial differential equations are reduced to a linked system of ordinary differential equations by means of appropriate similarity transformations. A Physics Informed Neural Network (PINN) method designed especially for nanofluid lubrication systems is used to solve these equations. By directly integrating the governing physical laws into the loss function, the suggested PINN architecture enables the simultaneous elimination of boundary condition errors and equation residuals. Computational efficiency and solution stability are improved by this two-way optimization. Also wed did Numerical Validation of the PINN Solver Comparing the tetra hybrid nanofluid to the ternary formulation, numerical results show that the former offers noticeably greater thermal enhancement and lower entropy generation. GNPs' remarkable heat conductivity and enormous surface area are primarily responsible for this performance enhancement. On the other hand, the ternary hybrid nanofluid shows moderate temperature gradients and comparatively constant viscosity behavior. For complicated nonlinear thermal-fluid problems in lubrication applications, the PINN framework provides a dependable computational tool with good convergence and prediction accuracy overall.

Keywords: *PINN, Nanolubricants, MHD Flow, Joule Heating, Thermal Radiation.*

1. Introduction

Researchers have been investigating novel ways to improve heat transmission in recent years by distributing nanoparticles into common base fluids like engine oil. It has been demonstrated that altering the particle's size, shape, concentration, and surface characteristics can enhance its long-term stability and heat conductivity. These nanofluids are being explored more and more for use in automotive lubricating systems, heat exchangers, collectors, and electronic cooling.

[1,2]Department of Mathematics, Rayalaseema University, Kurnool, 518007, Andhra Pradesh, India.*
E-mail:[1] praveenumkumar5@gmail.com

Because of their superior thermal performance and resilience to degradation in challenging environments, hybrid nanofluids are more appealing than the others. A number of physical mechanisms, including as magnetohydrodynamics (MHD), thermal radiation, Joule heating, and porous medium resistance, typically influence hybrid nanofluid flow simultaneously [1,7]. The combination of these effects results in coupled partial differential equations that are extremely nonlinear and challenging to solve analytically.

In MHD flow, the presence of a magnetic field creates a Lorentz force that acts against the fluid motion, adding drag force that slows the flow and inhibits the formation of the boundary layer. This behavior is consistent with the theory of classical magnetohydrodynamics. Heat and mass transfer are

less effective as the magnetic parameter rises because the thermal boundary layer thickens and the Nusselt and Sherwood numbers drop. Magnetic fields have a significant impact on the behavior of electrically conducting fluids, such as Casson hybrid nanofluid made from motor oil [1,5,8]. These effects are significant in energy devices, nuclear cooling, biomedical transport, and lubricating systems. In high-temperature environments, such as nuclear reactors and aerospace systems, thermal radiation becomes important [1,9,10]. Radiation often raises the temperature in nanofluid flow by adding thermal energy, while the overall effect varies depending on the operating conditions. The Rosseland approximation is used in this study to describe radiation, and the $\sigma B_0^2 u^2$ term is used to account for Joule heating. Darcy resistance is used to simulate the porous media effect, and uniform internal heat generation is assumed.

According to recent research, tetra-hybrid nanofluids outperform ternary systems in terms of thermal performance. Adnan et al., for example, observed improvements of around 4.3% in mass transport and 7.4% in heat transfer for tetra nanofluid that is impacted by radiation and magnetic fields [31–33]. Motivated by these results, the current study uses a Physics-Informed Neural Network (PINN) framework to investigate a hybrid $(Al_2O_3 + ZnO + SiC)$, nano lubricant and its graphene nanoplatelet augmented tetra counterpart under MHD, Joule heating, and radiation effect. The governing equations created by the simultaneous presence of MHD force, nonlinear radiation, Joule heating, chemical reaction, and Brownian motion are extremely challenging to solve using traditional methods [23]. Although computationally costly and requiring frequent mesh modification when parameters change, the Finite Element Method (FEM) is dependable. Serious problems are also caused by unbounded boundary conditions, particularly those defined at infinity [23, 25]. For severely nonlinear systems, Runge-Kutta and other classical numerical schemes frequently fail. Furthermore, it takes a lot of time to generate high-quality labeled datasets using this solution.

2. Motivation for Physics-Informed Neural Networks:

Recent advancements in artificial intelligence (AI) and machine learning (ML) offer fresh opportunities to get beyond these restrictions [2,3]. The mesh-free framework provided by Physics-Informed Neural Networks (PINNs) allows the governing equations to be directly incorporated into the loss function [13]. PINNs do not require labeled data, in contrast to conventional supervised learning. The two main components of the loss function are the residual of the governing differential equation and the boundary condition error. When combined, these terms guarantee that the physical laws be upheld throughout the entire realm. With little data, this method enables PINNs to handle nonlinear systems and unbounded domains more effectively. PINNs can produce solutions quickly after training, which is helpful for inverse problems and real-time prediction [14,15]. Accurate gradient computation for complex nonlinear equations is aided by automatic differentiation [17]. As a result, PINNs are frequently more adaptable and effective than many conventional numerical solutions.

PINN Architecture and Optimization: A fully connected deep neural network structure is commonly used in physics-informed neural networks. The network approximates the unknown solution field with trainable weight and bias. The chosen hyperparameter, such as network depth, width, activation function, and learning rate, has a significant impact on the model's accuracy [1,19]. While a very deep network may experience sluggish or unstable training, a network that is too shallow will not be able to adequately capture nonlinear behavior. Convergence speed is significantly influenced by activation functions. Although Tanh, Sigmoid, ReLU, and Swish are popular options, no single function is optimal for every issue. In stiff and nonlinear PDE systems, adaptive activation functions promote convergence, particularly in the early training stages, according to recent studies [17, 18]. The choice of learning rate is similarly crucial; small values slow down training, while big values produce divergence. The wide and problem-dependent search space makes hyperparameter tuning challenging. To automate this process, sophisticated techniques such as Bayesian optimization, Neural Architecture Search (NAS), and the Auto-PINN framework have been developed [14,15,19]. These methods lessen the need for human adjustment while increasing training robustness.

When dealing with multiscale or high frequency problems, standard PINNs may not function well. By breaking down the computing domain and enhancing global information sharing, Finite Basis PINNs (FBPINNs) and multilayer FBPINNs solve

this problem [20]. For Casson type ternary and tetra hybrid nanolubricants under magnetic, radiative, and reactive effects, a tailored PINN framework is created in this work [6,8,10]. The loss function directly incorporates the equations regulating momentum, energy, mass transport, and electromagnetic interaction. By doing this, the solution is guaranteed to adhere to physical laws across the domain rather than relying solely on data. Two systems of lubricants are examined. The first is an engine oil-suspended ternary hybrid nanofluid $(Al_2O_3 + ZnO + SiC)$, while the second is a tetra hybrid formulation boosted with graphene nanoplatelet. These nanoparticles were selected because of their potent rheological and thermal conductivity [5,7]. Changes in particle composition cause discernible changes in heat transfer behavior and flow resistance. In order to enhance convergence and physical consistency, a two-stage optimization technique is used during training, where PDE residual and boundary error are simultaneously minimized [17,19]. The suggested approach provides insight into the mechanisms of thermal enhancement, flow stability, and entropy formation while capturing intricate magneto-thermal interactions at a reduced computational cost. As a result, the approach offers a useful substitute for traditional numerical solvers in the analysis and design of lubrication systems [4,8,9].

The majority of current research on MHD hybrid nanofluid flows with thermal radiation, Joule heating, and porous media effects relies on traditional numerical solvers and focuses mostly on mono- or binary nanofluid systems, despite the expanding body of work in this area [21,33]. Even though recent studies show that ternary and tetra-hybrid nanofluids perform better thermally, systematic comparisons under the same physical conditions are still scarce, especially for Casson-type nanolubricants, and they are rarely handled with data-efficient learning frameworks [29,32]. However, there is still a lack of application for Physics-Informed Neural Networks to highly coupled MHD Casson ternary and graphene-enhanced tetra hybrid nanolubricants, despite their demonstrated significant ability to solve nonlinear thermal-fluid issues [13,17]. In order to fill these gaps and gain a better understanding of magneto-thermal transport behavior, the current work creates a tailored PINN framework for comparing the flows of ternary and tetra hybrid nanolubricants under magnetic, radiative, and Joule heating effects.

Physics-Informed Neural Networks (PINNs), activation functions, loss creation, the two-stage optimization approach, and implementation outcomes are all covered in Section 3. A study on the dependency of the Prandtl number (Pr), porous media parameter (Kp), magnetic parameter (M), and other associated factors is presented in Section 4. A comparison examination of current numerically based efforts is also included. The paper is finally concluded with important observations in Section 5.

A. Governing Equations

Let's look at a mathematical expression for incompressible Casson hybrid nanofluids flowing across a long wall in two dimensions (2D). Fig. 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 $B_0$is applied.

The rules for this problem are as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (1)$$

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

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

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

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

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

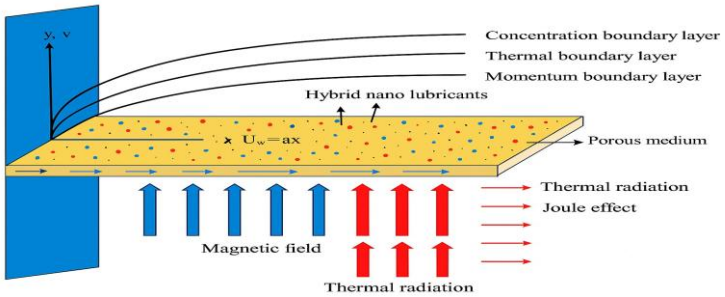


Fig. 1 Physical Configuration and Coordinate System of the Problem

B. Dimensionless Governing Equations

Introducing the following dimensionless variables:

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

and the dimensionless parameters:

$$\mathrm{Re}^* = \frac{\rho_{qnf} U_0 L}{\mu_{qnf}}, \mathrm{Pr}^* = \frac{\mu_{qnf} C_{pqnf}}{k_{qnf}}, \mathrm{Sc}^* = \frac{\mu_{nf}}{\rho_{qnf} D_m}, \quad M^* = \frac{\sigma B_0^2 L^2}{\mu_{qnf}},$$
$$K_p^* = \frac{L^2}{k^* \nu_{qnf}},$$
$$Q^* = \frac{Q_0 L^2}{\rho_{qnf} C_{pqnf} \nu_{qnf}}, \quad S_r^* = \frac{K_T}{T_m Sc^*}, K_r^* = \frac{K_r L^2}{\nu_{qnf}}, Ec^* = \frac{U_0^2}{C_{pqnf}(T_w - T_\infty)},$$

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:

$$at \quad y^* = 0: u^* = 1, v^* = 0, T^* = 1, C^* = 1,$$
$$as \quad y^* \to \infty: \quad u^* \to 0, \quad T^* \to 0, \quad C^* \to 0.$$

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, ρ (kg/m^3)	3970	5600	3210	2200	884
Specific heat, C_p (J/kg · K)	765	495.2	600	700	1910
Thermal conductivity, k (W/m · 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)	$k_{eq} = \frac{\phi_1 k_1 + \phi_2 k_2 + \phi_3 k_3}{\phi_1 + \phi_2 + \phi_3}$

C. Similarity ODEs for Casson Hybrid Nanofluid

Similarity transformations:

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

The transformed non-dimensional ordinary differential equations are:

$$\left(1+\frac{1}{\beta}\right)f'''(\eta) \quad -K_p f'(\eta) - S_4 M f'(\eta) - S_1[(f'(\eta))^2 - f(\eta)f''(\eta)] + \lambda\,\theta(\eta) + N\,\phi(\eta) = 0 \quad (10)$$

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

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

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	$k_{eq} = \frac{\phi_1 k_1 + \phi_2 k_2 + \phi_3 k_3 + \phi_4 k_4}{\phi_1 + \phi_2 + \phi_3 + \phi_4}$

D. 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:

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

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

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 Viscosity	$\frac{\nu_{tnf}}{\nu_f} = \frac{\mu_{tnf}\rho_f}{\mu_f \rho_{tnf}}$
S_4	Density	$\frac{\rho_{tnf}}{\rho_f} = (1-\phi_1-\phi_2-\phi_3) + \frac{\phi_1\rho_1 + \phi_2\rho_2 + \phi_3\rho_3}{\rho_f}$
S_5	Heat Capacity	$\frac{(\rho C_p)_{tnf}}{(\rho C_p)_f} = (1-\phi_1-\phi_2-\phi_3) + \frac{\phi_1(\rho C_p)_1 + \phi_2(\rho C_p)_2 + \phi_3(\rho C_p)_3}{(\rho C_p)_f}$
S_6	Thermal Conductivity	$k_{eq} = \frac{\phi_1 k_1 + \phi_2 k_2 + \phi_3 k_3}{\phi_1+\phi_2+\phi_3}, \frac{k_{tnf}}{k_f} = \frac{k_{eq}+2k_f+2\phi_t(k_{eq}-k_f)}{k_{eq}+2k_f-\phi_t(k_{eq}-k_f)}, \phi_t = \phi_1+\phi_2+\phi_3$

E. Physical Quantities of Interest

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

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

F. Similarity Transformed Quantities

$$C_{fx}(2Re_x)^{1/2} = \frac{\mu_{hnf}}{\mu_f} f''(0), Nu_x(Re_x)^{-1/2} = -\frac{k_{hnf}}{k_f}\theta'(0), \quad Sh_x(Re_x)^{-1/2} = -\phi'(0) \tag{14}$$

Where $Re_x = \frac{U_w x}{\nu_f}$

Table 5: Thermophysical Property Ratios for Tetra Nanolubricant (Al_2O_3+Zno+Sic+GNP/Engine Oil)

Symbol	Property Ratio	Formula
S_1	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
S_2	Schmidt Number	$\frac{Sc_{qnf}}{Sc_f} = \frac{\mu_{qnf}\rho_f}{\mu_f\rho_{qnf}}$
S_3	Kinematic Viscosity	$\frac{\nu_{qnf}}{\nu_f} = \frac{\mu_{qnf}\rho_f}{\mu_f\rho_{qnf}}$
S_4	Density	$\frac{\rho_{qnf}}{\rho_f} = (1-\phi_1-\phi_2-\phi_3-\phi_4) + \frac{\phi_1\rho_1+\phi_2\rho_2+\phi_3\rho_3+\phi_4\rho_4}{\rho_f}$
S_5	Heat Capacity	$\frac{(\rho C_p)_{qnf}}{(\rho C_p)_f} = (1-\phi_1-\phi_2-\phi_3-\phi_4) + \frac{\phi_1(\rho C_p)_1+\phi_2(\rho C_p)_2+\phi_3(\rho C_p)_3+\phi_4(\rho C_p)_4}{(\rho C_p)_f}$
S_6	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:

The transverse coordinate $y^* \in [0, y_{max}]$. is the only spatial variable that remains in the computational domain when using the ODE configuration. The training problem is kept manageable by truncating the semi-infinite region at $y_{max} = 7.0$, which is large enough to capture the gradual decrease of temperature, concentration, and velocity. The domain uses two different kinds of nodes. To ensure that the network adheres to the primary physical boundaries, the initial set of boundary nodes, which are situated at $y^* = 0$ and $y^* = y_{max}$, directly impose the wall value and far-field condition in the loss function. Collocation nodes, which are dispersed throughout the domain, are the second kind. Their function is to enforce the residual of the governing ODEs, ensuring that the learned solution stays constant throughout the interval. Latin hypercube sampling is used to create the collocation nodes, which prevents undesired point clustering and provides more uniform coverage than a standard grid. Near the wall region ($y^* = 0$), a slightly higher node density is maintained because abrupt gradients frequently occur there and insufficient resolution may prevent the network from accurately capturing them. Let Nu represent the number of boundary nodes and Nf represent the number of collocation nodes. Both sets of nodes are significant; the collocation nodes compel the network to minimize the ODE residual inside the domain, while the boundary nodes fix the solution to the specified physics. The PINN discretization technique for the one-dimensional issue is based on this balancing. The whole Physics-Informed Neural Network (PINN) structure utilized in this study is depicted in Fig. 2, which also includes the two-stage optimization technique, boundary condition enforcement, and the development of a physics-based residual.

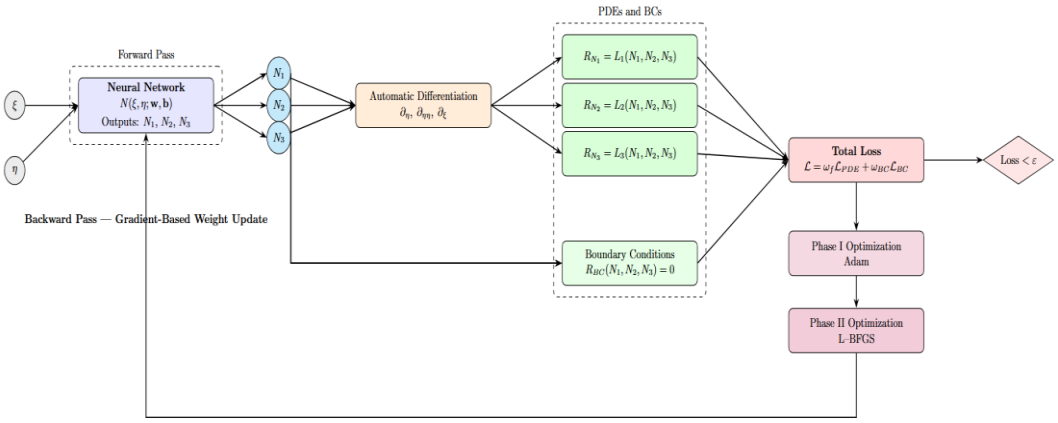


Fig. 2: PINN Architecture Used for the Present Study

3.2 Function Construction:

The main aim of the PINN model is to approximate the unknown function u*(y*), T*(y*), and C*(y*) that satisfy the reduced one-dimensional equation. The network takes the coordinate y* as input and give the predicted physical quantity as output. In symbolic form, the mapping can be written as

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

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

Automatic differentiation is used to compute the derivative of $\widehat{\boldsymbol{U}}$ with respect to y*. This is very important because the governing equation involve first and second order derivative. Using exact gradient from the computational graph avoid numerical error that usually occur in finite difference method. The obtained expressions are then substituted into the governing ODEs to form the residual for momentum, energy, and concentration equation.

The momentum residual is expressed as

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

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 over the domain for the predicted solution to fully satisfy the physics. In practice, they does not vanish exactly during training, but the network is trained to minimize them as much as possible. This is the way the PINN combine learning with physical consistency: the residual terms push the model toward the correct dynamic behavior, while the boundary condition anchor the solution to the prescribed value at $y^* = 0$ and $y^* = y_{\max}$.

3.3 Optimization Process:

The primary process that enables the neural network to approximate the actual solution of the ODE system is the optimization procedure. The loss function, which measures how much the prediction deviates from the physics and boundary condition, is decreased by gradually updating the network parameter, represented by w and b. The network typically struggles to provide significant output in the beginning, and the residuals are quite high. The optimizer gradually moves the solution toward conformity with the governing equation and the specified boundary as training progresses. In actuality, a two-phase optimization approach is employed. When the gradient is noisy and unstable in the early training phase, the Adam algorithm which is stochastic and adaptive is used in the first step. The L-BFGS method, a quasi-Newton algorithm, is used in the second stage once the model has approached the correct solution region. L-BFGS is deterministic and typically provides more accurate convergence than Adam. Combining these two approaches avoids both early stagnation at a later stage and slow initial growth.

The total loss function is defined as the weighted sum of two parts: the boundary mismatch and the residual mismatch, and can be written as

$$\mathcal{L}(\mathbf{w}, \mathbf{b}) = \omega_u \,\mathrm{MSE}_u(\mathbf{w}, \mathbf{b}) + \omega_f \,\mathrm{MSE}_f(\mathbf{w}, \mathbf{b}),$$

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

$$\mathrm{MSE}_u = \frac{1}{N_u}\sum_{i=1}^{N_u} \| \mathbf{U}(y_i^*) - \hat{\mathbf{U}}(y_i^*;\mathbf{w},\mathbf{b}) \|^2,$$

while the residual part is

$$\mathrm{MSE}_f = \frac{1}{N_f}\sum_{i=1}^{N_f}(| R_U(y_i^*) |^2 + | R_T(y_i^*) |^2 + | R_C(y_i^*) |^2).$$

During the training process, the weight and bias are updated based on the gradient of the loss function. In the Adam stage, the step size or learning rate change adaptively according to moving average of previous gradient, while in the L-BFGS stage, the algorithm uses curvature information to select better search direction and step length. The optimization process stop when the loss no longer decreases or reach a specified tolerance value, which indicate that the network has learned a set of parameters that minimize the disagreement with the ODE system and boundary data. This two stage approach is now commonly used in many PINN studies because it combine the robustness of stochastic optimization with the accuracy of second order method, resulting in stable and reliable solution.

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

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

The second contribution is obtained from the thermal energy equation,

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

The final residual term comes from the concentration equation,

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

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

$$\mathrm{MSE}_f = \mathrm{MSE}_{R_1} + \mathrm{MSE}_{R_2} + \mathrm{MSE}_{R_3}.$$

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

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

The total loss that drives the training is finally written as

$$\mathcal{L} = \omega_u\,\mathrm{MSE}_u + \omega_f\,\mathrm{MSE}_f,$$

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

Heat and mass transfer in Casson hybrid nanofluids exposed to Joule heating and thermal radiation in the presence of a stretched surface and magnetic field are investigated in this work using a PINN framework. We are examining two distinct nano-lubricant systems: a ternary formulation that incorporates graphene nanoplatelets (GNP) and a tetra formulation that incorporates Al_2O_3, ZnO, and SiC nanoparticles into engine oil. In addition to assessing the adaptability of PINNs in capturing tightly coupled nonlinear phenomena, the primary goal is to investigate the effects of particle composition on heat transport, viscous stability, and entropy generation. There are two steps involved in the training. In order to rapidly lower the loss function during the early era, the Adam optimizer is first used to broadly explore the parameter space. Following that, quasi-Newton updates are applied using the L-BFGS optimizer to further enhance convergence by efficiently smoothing the solution surface. This two-step approach ensures that the governing equations are solved effectively and accurately by combining the accuracy of a deterministic solver with the quick convergence of stochastic optimization.

The activation function is a key factor in deciding how the PINN model converges in Tables 6 and 7. Tanh constantly performs the best out of all the

evaluated activation functions. After L-BFGS optimization, it obtained a total loss of 1.07×10^{-6} for the ternary nanofluid case, which is significantly less than the Adam baseline value of 8.32×10^{-5} and only required 54 s of training time. Tanh achieved a loss of 1.08×10^{-6} in 75 s for the tetra nanofluid, showing a similar pattern. The performance of the other activation function is lower. Swish and Sigmoid converged to larger loss values, but Mish offered similar accuracy without requiring more than twice as much training time. Poor convergence behavior was displayed by GELU, Softplus, and Softmin, especially in the case of tetra nanofluid, where Softmin halted at 1.79×10^{-3}. These findings unequivocally show that Tanh offers the best dependable balance between prediction accuracy and computational economy, even though a number of activation functions can yield acceptable solutions.

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
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 Fig. 3, different learning rate from 10^{-1} to 10^{-5} were tested using the two stage Adam to L-BFGS optimization scheme. The result show clear variation in stability and accuracy. At the highest learning rate of 10^{-1}, the training become unstable with strong oscillation, which prevent effective reduction of the loss value. On the other extreme, very small learning rate (10^{-4} and 10^{-5}) converge too slowly and fail to minimize the residual within a reasonable training time. The best performance is obtained at a learning rate of 10^{-2}. Both ternary and tetra nanofluid cases converge smoothly at this value, where Adam produce a rapid initial decrease in the loss and L-BFGS further refine the solution to very small residual. When compared with the 10^{-3} case, the 10^{-2} learning rate reach lower final loss in a more efficient way, confirming it as the optimal selection

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
Swish	6.65×10^{-7}	8.96×10^{-5}	9.02×10^{-5}	9.17×10^{-5}	100.53

For 40, 60, 80, and 100 neurons in each layer, the impact of the concealed neuron number was investigated. Tetra and ternary nanofluids behave similarly. The final residual remained at a greater level, suggesting that the model was underfitted, even though a smaller network with 40 neurons was able to converge. On the other side, networks with 100 neurons caused a rapid drop in loss at the beginning, but the curve flatten too early, resulting in higher residual and signs of instability. As seen in Fig. 4 for both fluid scenarios, 60 neurons produced the most reliable and consistent results. In order to achieve the lowest final loss values, this design provides smooth convergence throughout the Adam stage and additional enhancement utilizing L-BFGS. Although they needed more training time and had less stability in later stages, networks with 80 neurons also did fairly well. The optimal balance between model capacity and training stability is achieved by employing 60 neurons per layer, which results in precise and effective convergence for both nanofluid systems.

3.6 Physical Accuracy Validation: Numerical Validation of the PINN Solver

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 (Cf) 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.

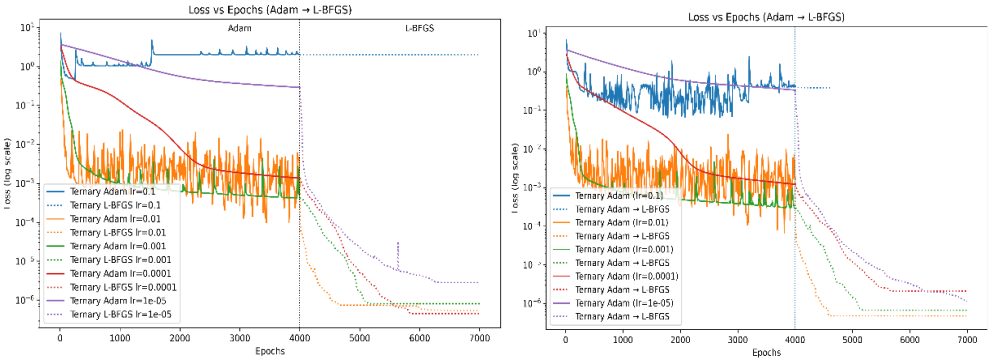
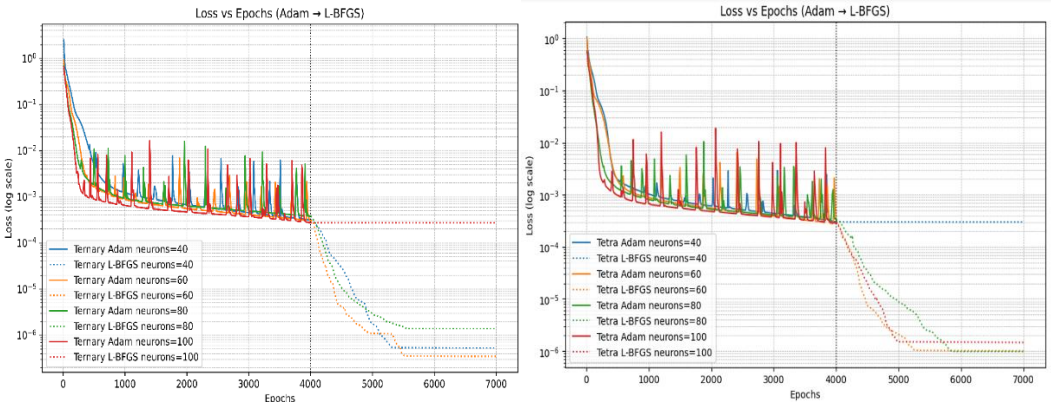


Fig. 3: Loss Versus Epochs for different Learning Rates (Adam → L-BFGS).



Right: Ternary Nanolubricant. Left: Tetra Nanolubricant

Fig. 4 Loss Versus Epochs for Different Numbers of Neurons (Adam → L-BFGS).
Right: Ternary Nanolubricant. Left: Tetra Nanolubricant.

Table 9: Full Coupled Model Validation ($\varphi_i \neq 0$, M/Ec/R/Q/Kr/Sr/Sc Active) Vs Bvp Baseline

Case	Pr	f″(0) PINN	f″(0)	θ′(0) PINN	θ′(0)	Cf PINN	Cf	Nu PINN	Nu
Ternary	7	-0.961342	-0.961958	0.589809	0.592900	-2.766414	-2.768188	-0.644103	-0.647479
Ternary	14	-0.963360	-0.963494	1.499105	1.501342	-2.772221	-2.772608	-1.637103	-1.639547
Ternary	21	-0.964019	-0.963910	2.374375	2.377613	-2.774117	-2.773805	-2.592946	-2.596481
Ternary	28	-0.964008	-0.964029	3.211710	3.216139	-2.774087	-2.774147	-3.507360	-3.512197
Tetra	7	-0.970260	-0.970136	0.563063	0.562754	-2.865358	-2.864992	-0.633104	-0.632756
Tetra	14	-0.971718	-0.971728	1.441216	1.443355	-2.869663	-2.869694	-1.620492	-1.622898

Tetra	21	-0.972059	-0.972173	2.291670	2.295513	-2.870669	-2.871005	-2.576736	-2.581057
Tetra	28	-0.971812	-0.972309	3.102075	3.112268	-2.869941	-2.871408	-3.487949	-3.499410

Table 10: L_2 Profile Errors and Relative Wall Quantity Errors (PINN Vs Reference)

Case	Pr	$L_2(f')$	$L_2(\theta)$	$L_2(\phi)$	Rel. Error (C_f)	Rel. Error (Nu)
Ternary	7	6.93e-4	6.40e-4	8.54e-3	6.41e-4	5.21e-3
Ternary	14	7.21e-4	3.84e-4	8.74e-3	1.40e-4	1.49e-3
Ternary	21	7.52e-4	3.61e-4	8.34e-3	1.13e-4	1.36e-3
Ternary	28	7.15e-4	4.34e-4	6.28e-3	2.20e-5	1.38e-3
Tetra	7	3.74e-4	1.09e-4	5.28e-3	1.28e-4	5.49e-4
Tetra	14	7.23e-4	3.78e-4	6.83e-3	1.10e-5	1.48e-3
Tetra	21	6.61e-4	3.70e-4	8.55e-3	1.17e-4	1.67e-3
Tetra	28	8.82e-4	5.70e-4	1.19e-2	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 11: 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

Table 11 indicates that if the aforementioned results fairly compare with earlier Pr results in limited circumstances, they can be confirmed. Two results for the ternary nanolubricant Al_2O_3, ZnO, and SiC / engine oil and the tetra nanolubricant (containing graphene nanoplatelets (GNP) / engine oil) are shown in Fig. s 5–15. Fig. s 5 and 6 show how the velocity distribution f ′(η) is affected by the magnetic parameter (M) and the porous media parameter (Kp). Higher values of M and Kp considerably reduce the flow field for both nanofluid systems. In physical terms, this phenomenon happens because a stronger magnetic field produces a larger Lorentz force that opposes fluid motion, whereas a medium with reduced porosity increases resistance to flow. The effect of the Casson parameter on velocity is depicted in Fig. 7. The velocity profile decreases as the Casson parameter rises, indicating the Casson

type fluid's shear thinning tendency. Because the ternary formulation has a larger viscous resistance, the effect is more pronounced. This decrease in velocity corresponds to a thinner hydrodynamic boundary layer. A greater temperature gradient close to the wall results from rising Pr, which decreases the thickness of the thermal boundary layer (Fig. 8). Given that a greater Pr value denotes a lower thermal diffusivity, this result is consistent with traditional boundary layer theory. The impact of the heat source parameter Q is examined in Fig. 9. The temperature profile rises as Q increases because the system is receiving more energy. Similarly, the effect of Eckert number Ec is shown in Fig. 10, where a larger Ec value raises the temperature due to the conversion of mechanical energy into thermal energy by viscous dissipation. Because of the greater thermal enhancement brought on by GNP, the tetra nanofluid exhibits more noticeable temperature increases in both situations. The impact of the solutal parameter on the concentration distribution is shown in Fig. 11. Since a higher Schmidt number Sc is associated with a smaller concentration boundary layer and lower mass diffusivity, it lowers the concentration profile in Fig. 11. The effect of Soret number is depicted in Fig. 13, where greater values result in a concentration profile that is stronger due to thermodiffusion coupling induced by temperature gradient. Lastly, the concentration profile $\phi(\eta)$ for various radiation parameters R is shown in Fig. 15. Higher concentration levels are a result of stronger radiation, indicating that radiative heating indirectly promotes solutal diffusion. When compared to the ternary nanofluid, the tetra nanofluid shows superior thermal enhancement because of the graphene nanoplatelet. The effect of the chemical reaction parameter Kr is seen in Fig. 12. Because a greater reaction rate consumes species more quickly, reducing the thickness of the solutal boundary layer, increasing Kr decreases the concentration distribution. The impact of radiation parameter R on the temperature profile $\theta(\eta)$ is seen in Fig. 14. A thicker thermal boundary layer results from an increase in R, which also causes an increase in the temperature distribution. In the ternary nanofluid, where the addition of graphene nanoplatelets enhances thermal transport and radiative heat absorption, this effect is more noticeable.

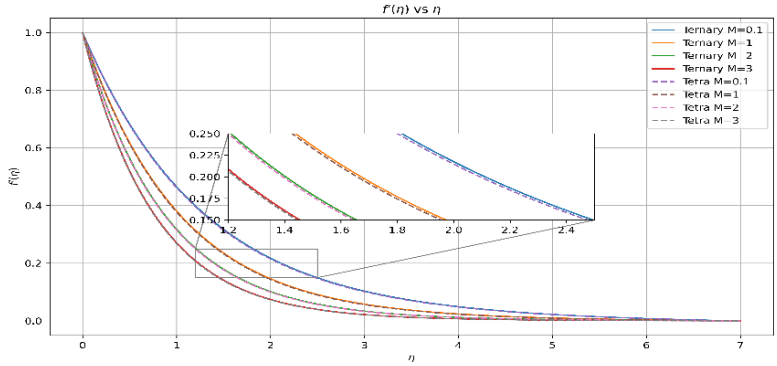


Fig. 5 f' Impact on M.

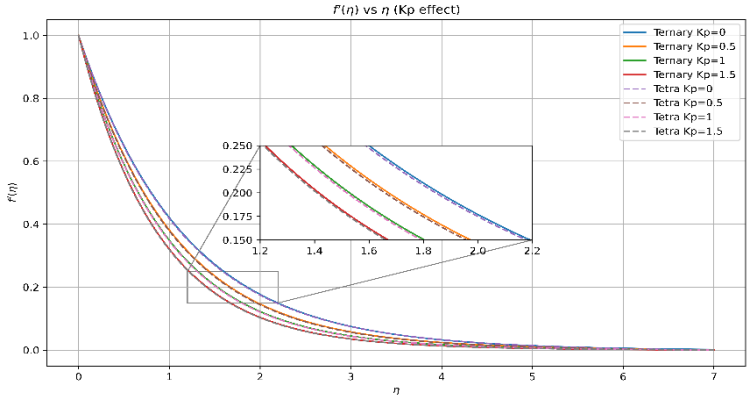


Fig. 6 f' Impact on KP

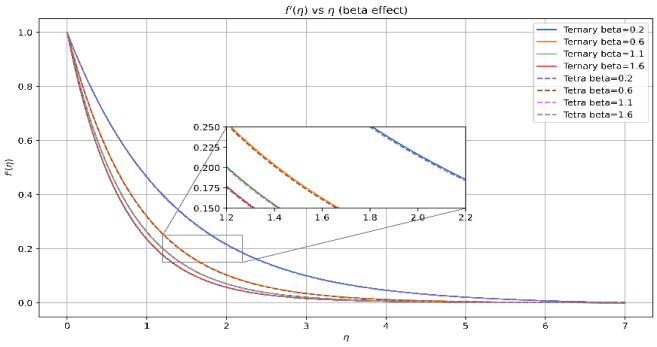


Fig. 7 f′ Impact on β

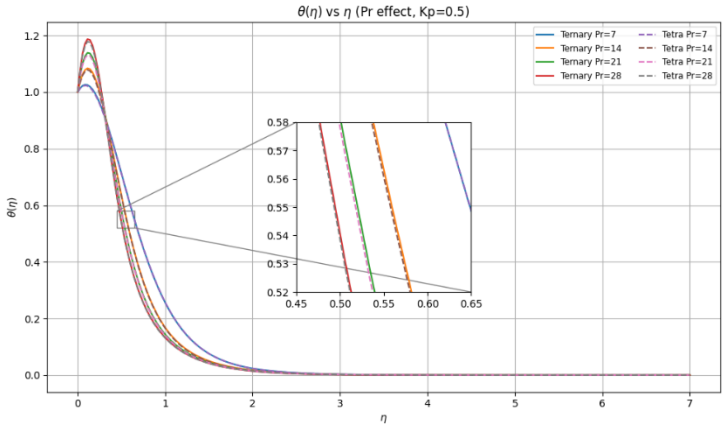
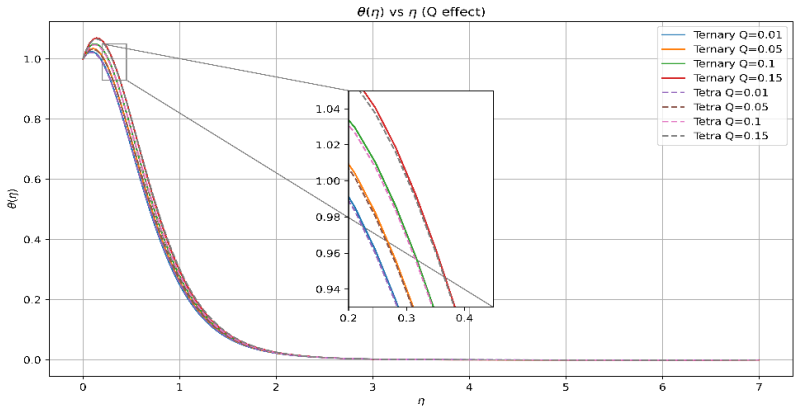


Fig. 8 θ Impact on PR



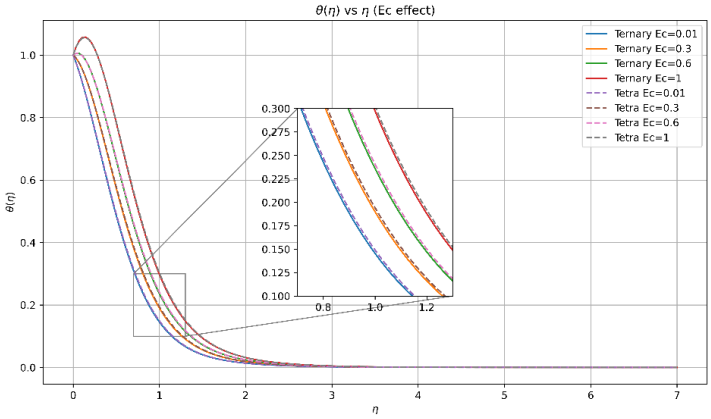


Fig. 9 θ Impact on Q

Fig. 10 θ Impact on EC

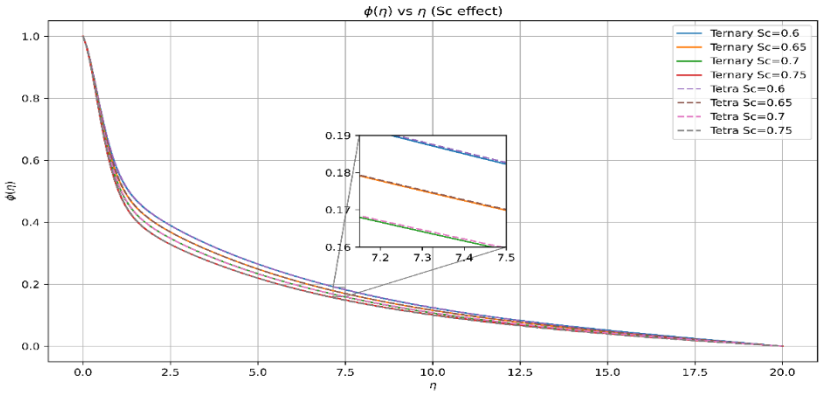


Fig. 11 ϕ Impact on SC

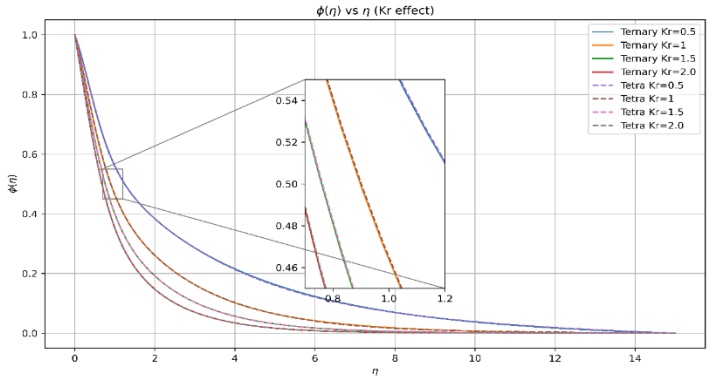


Fig. 12 Φ Impact on Kr

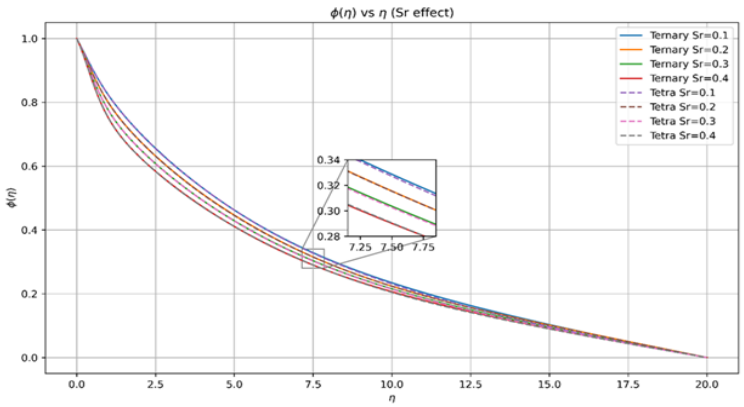


Fig. 13 ϕ Impact on SR

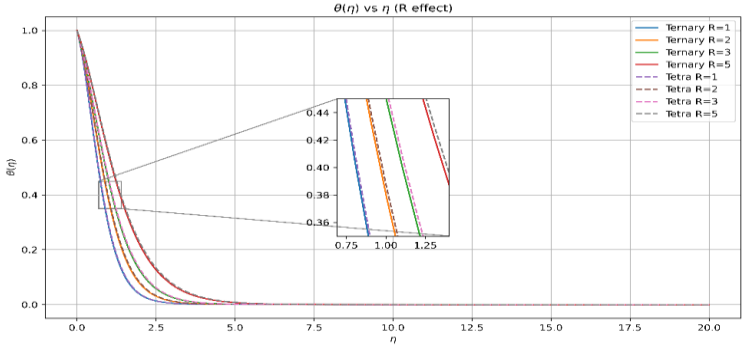


Fig. 14 ϕ Impact on R

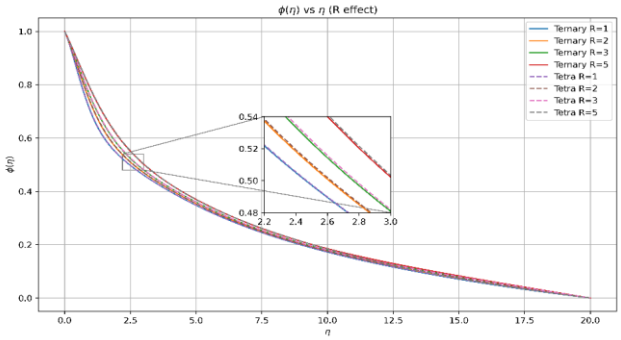


Fig. 15 ϕ Impact on R

5. Conclusion:

This study used physics-informed neural networks to mimic a Casson hybrid nanofluid under the influence of radiation, a magnetic field, a chemical reaction, and Joule heating. A ternary nanofluid (Al_2O_3, ZnO, and SiC) distributed in engine oil and a tetra nanofluid with graphene nanoplatelets added were the two distinct systems that were examined. Understanding the model's computing performance and physical behavior was the primary objective. The Tanh activation function performs the best out of all the examined scenarios from a computational perspective. It was determined that 60 hidden neurons were adequate to represent the primary solution properties, and a learning rate of 0.01 was deemed appropriate. The two-stage training approach with Adam and L-BFGS greatly enhances the PINN model's convergence behavior. The high-accuracy bvp solution was used to validate the PINN results. While concentration errors stayed below 10^{-2}, velocity and temperature L_2 errors were approximately $10^{-4} - 10^{-3}$. Skin friction and Nusselt number had very tiny relative errors (less than 0.07% and 0.6%, respectively). This suggests that the model is correct in addition to being stable. Physically, the temperature increased as the heat source and Eckert number increased because more energy was being used. In accordance with classical theory, a higher Prandtl number resulted in a thinner thermal boundary layer. The ternary system displayed somewhat steady viscosity behavior, whereas the tetra nanofluid demonstrated a significantly stronger temperature increase.

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

6. REFERENCES

[1] 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)

[2] 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)

[3] 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)

[4] 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)

[5] 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)

[6] 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)

[7] 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)

[8] 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)

[9] Zobeiry, N., Humfeld, K.D.: Physics-informed machine learning for solving heat transfer equations. *Eng. Appl. Artif. Intell.* 101, 104232 (2021)

[10] 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)

[11] 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)

[12] Tartakovsky, A.M., Perdikaris, P., Karniadakis, G.E.: Physics-informed deep neural networks for hydrological systems. *Water Resour. Res.* 56(5), e2019WR026731 (2020)

[13] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks for nonlinear PDEs. *J. Comput. Phys.* 378, 686–707 (2019)

[14] Wang, Y., Han, X., Chang, C.Y., Zha, D., Braga-Neto, U., Hu, X.: Auto-PINN: Understanding and optimizing physics-informed neural architectures. (preprint)

[15] Belbute-Peres, F.A., Sha, F., Chen, Y.-F.: HyperPINN: Learning parameterized differential equations with hypernetworks.

(preprint)
[16] Al-Hussein, A.: Physics-informed neural network framework for 2D and 3D obstacle problems. Ph.D. Thesis (2023)
[17] Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E.: Adaptive activation functions accelerate convergence in PINNs. *J. Comput. Phys.* 404, 109136 (2020)
[18] Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E.: Hyperparameter tuning of PINNs with Helmholtz applications. (preprint)
[19] Karl, A., Pielok, T.: Multi-objective hyperparameter optimization in machine learning. (preprint, 2023)
[20] Dolean, V., Heinlein, A., Mishra, S., Moseley, B.: Multilevel domain decomposition architectures for PINNs. (preprint)
[21] 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
[22] Zienkiewicz, O.C., Taylor, R.L.: *The Finite Element Method, Vol. 1: The Basis*. Butterworth-Heinemann, Oxford (2000)
[23] Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: *Spectral Methods: Fundamentals in Single Domains*. Springer, Berlin (2007)
[24] Kharazmi, E., Zhang, Z., Karniadakis, G.E.: Variational physics-informed neural networks. *arXiv preprint* arXiv:1912.00873 (2019)
[25] 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
[26] 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
[27] 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
[28] 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
[29] 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
[30] 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
[31] 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
[32] 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
[33] 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
[34] 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