

A physics-informed neural network approach for natural convection boundary-layer flow in nanofluid-saturated non-darcy porous medium

 Hamza ALI AGHA¹,  Soufiane RAHAL¹,  Faouzi DIDI²,  Maamar LAIDI¹,  Ebrahim E. Elsayed^{3,4*}

¹Biomaterials and Transport Phenomena Laboratory (LBMPPT), Faculty of technology, University Yahia Fares of Medea, Pole Urbain, Medea 26000, Algeria; aliagha.hamza@univ-medea.dz (H.A.A.); rahal.soufiane@univ-medea.dz (S.R.); laidi.maamar@univ-medea.dz (M.L.).

²Department of Common Core in Technology, Laboratory of Physics of Experimental Techniques and Their Applications, University Yahia Fares of Medea, 26.000, Algeria; didi.fauzi@univ-medea.dz (F.D.).

³Department of Scientific Research Laboratory, Azerbaijan University of Architecture and Construction, Baku AZ1073, Azerbaijan; engebrahim16@std.mans.edu.eg (E.E.E.).

⁴Department of ECE, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt.

Abstract: This study presents a robust deep learning framework based on Physics-Informed Neural Networks (PINNs) for modeling natural convection boundary-layer flow over a vertical plate embedded in a non-Darcy porous medium saturated with nanofluid. The mathematical formulation is based on the extended Cheng–Minkowycz problem, incorporating the Buongiorno nanofluid model to account for Brownian motion and thermophoresis effects, along with the Forchheimer–Darcy model to describe nonlinear porous medium resistance. Through appropriate similarity transformations, the governing nonlinear partial differential equations are reduced to a system of coupled nonlinear ordinary differential equations. These equations are then solved using PINNs, enabling accurate and mesh-free approximation of the solution while preserving the underlying physical laws. A comprehensive analysis is conducted to investigate the influence of key deep learning hyperparameters, including the learning rate and neural network architecture, on convergence and solution accuracy. The reliability of the proposed PINN model is validated through comparison with numerical solutions obtained using the implicit finite difference method implemented via the bvp4c solver. Furthermore, a detailed parametric study is performed to examine the effects of governing physical parameters on velocity, temperature, and nanoparticle concentration profiles, as well as on engineering quantities of interest such as the local Nusselt number. The results reveal that both nanoparticle effects and porous medium resistance significantly influence heat and mass transfer characteristics, highlighting the effectiveness of the proposed PINN framework for complex transport phenomena.

Keywords: Brownian motion, Buongiorno nanofluid model, Heat transfer, Natural convection, Non-Darcy porous medium, Physics-Informed neural networks, Thermophoresis.

1. Introduction

Artificial intelligence, particularly machine learning, has emerged as a transformative discipline in computational science, offering powerful tools for data-driven modeling, prediction, and system identification across numerous domains [1]. Within this framework, artificial neural networks (ANNs) represent a biologically inspired computational architecture designed to approximate complex nonlinear relationships [2]. Over the past few decades, neural networks have attracted considerable attention in disciplines such as materials science [3], chemical engineering [4–6], biomedical systems [7], robotics [8], and climate modeling [9]. Their popularity largely stems from their universal function approximation capability, which ensures that sufficiently parameterized networks can approximate any

continuous function with arbitrary accuracy. As problem complexity increases, particularly in nonlinear dynamical systems, the need for deeper architectures with multiple hidden layers and enhanced representational capacity becomes essential, giving rise to deep learning methodologies [4].

In fluid dynamics and heat transfer, governing equations are often highly nonlinear, strongly coupled, and high-dimensional, posing substantial challenges for traditional modeling approaches. Consequently, researchers have increasingly explored neural network-based models to capture flow behavior, reconstruct velocity and pressure fields, and develop reduced-order representations of turbulent structures [10]. Early implementations of shallow neural networks were used for flow visualization and particle tracking, while modern deep architectures have demonstrated improved performance in turbulence closure modeling, flow regime classification, and vortex pattern identification [11]. These advancements highlight the growing integration of neural networks into computational fluid dynamics as complementary tools for modeling complex transport phenomena.

Artificial neural networks have been investigated as alternative solvers for ordinary and partial differential equations by reformulating the governing equations into optimization problems. Early contributions by researchers such as Lagaris, et al. [12] demonstrated that neural networks could approximate differential equation solutions by embedding them within specially constructed trial functions. In this formulation, the approximate solution is expressed as a combination of two components: one analytically designed to satisfy boundary or initial conditions, and another represented by a trainable neural network responsible for minimizing the equation residuals. The network parameters, including weights and biases, were iteratively adjusted to reduce the discrepancy between the predicted and exact differential operators. Although this methodology proved that even shallow networks possess the capacity to approximate solutions accurately, the construction of suitable trial functions becomes increasingly challenging for multidimensional, nonlinear, and time-dependent systems, particularly in complex geometrical domains.

To address these limitations, a new framework known as Physics-Informed Neural Networks (PINNs) was introduced by Raissi, et al. [13]. Unlike earlier approaches, PINNs eliminate the need for explicitly defined trial functions by directly incorporating the governing equations and boundary conditions into the loss function [14]. The neural network is trained not using labeled output data as in traditional supervised learning, but by minimizing a composite loss function consisting of two principal components: the residuals of the governing differential equations evaluated at collocation points within the domain, and the mismatch between predicted and prescribed values at the boundaries or initial conditions. Through automatic differentiation, the network efficiently computes the necessary derivatives, ensuring consistency with the physical laws [15]. Since their introduction, PINNs have been successfully applied to a wide range of forward and inverse problems in computational fluid dynamics and heat transfer, demonstrating their potential as a mesh-free and physics-consistent alternative to classical numerical methods.

In this context, Bararnia and Esmailpour [16] used deep learning via PINNs to examine the dynamic and thermal behaviors of some classical boundary-layer flows, including Blasius–Pohlhausen, Falkner–Skan, and natural convection cases [16]. In this study, TensorFlow was used to build and train the models, and the predicted results were compared with those obtained by the finite-difference technique combined with Richardson extrapolation. The authors reported that the optimized PINN results were in close agreement with the numerical solutions. Sarthak and Srinivasacharya [17] used PINNs to simulate the complex dynamics of magnetohydrodynamic (MHD) natural convection along a vertical plate in a non-Darcy porous medium, accounting for thermal radiation and cross-diffusion. This study underscores the efficacy of deep-learning-based PINNs in addressing complex fluid dynamics and heat transfer challenges. By optimizing key parameters, the authors achieved enhanced solution accuracy and computational efficiency. Jawad, et al. [18] examined the time-dependent Jeffery hybrid nanofluid flow with thermal and mass convection in a rotating system in the presence of a porous medium. To analyze the dynamic, thermal, and mass behaviors, they used a combination of the PINN approach and the homotopy analysis method (HAM) to solve the resulting nondimensional partial

differential equations. Their results demonstrated that integrating HAM with PINNs provides significant advantages for solving complex fluid dynamics equations. Cai, et al. [19] demonstrated the application of PINNs to complex scenarios such as forced and mixed convection heat transfer with unknown boundary conditions, as well as to the Stefan problem involving moving interfaces and phase change, using sparse temperature measurements. Overall, their results show that PINNs can effectively address ill-posed and industry-relevant problems, thereby bridging the gap between experimental observations and numerical modeling. Li, et al. [20] proposed a physics-informed Bayesian deep learning framework that integrates Bayesian neural networks with variational inference to solve forward and inverse problems governed by the phonon Boltzmann transport equation using sparse and noisy data. This study addressed the challenges of nondiffusive phonon transport in micro- and nanoscale thermal systems, particularly issues related to inverse modeling, measurement noise, and uncertainty quantification. Numerical results demonstrated that the method accurately reconstructs temperature and heat flux fields, estimates key physical parameters such as the Knudsen number, and provides robust uncertainty quantification. Rudrabhatla and Srinivasacharya [21] investigated the effects of control parameters on Casson fluid flow and heat and mass transfer in the presence of a chemical reaction and viscous dissipation. The governing equations of the problem were solved using an optimized PINN, and the obtained results were compared with those from the finite-difference collocation method. The study concluded that the proposed PINN framework provides an accurate, efficient, and mesh-free alternative to traditional numerical methods. A. Jan, et al. [22] employed the PINN method to numerically solve a mixed convection flow of a viscous incompressible fluid over a vertical stretching sheet under viscous dissipation and Ohmic heating induced by Lorentz forces in conducting fluids. Their results demonstrated the capability and efficiency of PINNs in solving nonlinear multiphysics problems with greater flexibility and effectiveness than conventional solvers, opening opportunities for broad engineering applications.

Natural convection in porous media filled with nanofluids has attracted significant research interest due to its importance in enhancing heat transfer performance for a wide range of applications, including geothermal energy systems, thermal energy storage, electronic cooling, and solar collectors. Unlike conventional fluids, nanofluids are suspensions of nanoparticles in a base fluid that exhibit improved thermal properties such as increased thermal conductivity and modified buoyancy effects, which can substantially alter flow and heat transfer characteristics within porous structures. The presence of solid matrix effects, coupled with nanoparticle–fluid interactions, leads to complex thermal behaviors that challenge classical modelling approaches and necessitate advanced theoretical and numerical formulations. Recent studies have demonstrated that nanofluid characteristics (e.g., nanoparticle size, concentration, and type) and porous medium properties (e.g., porosity and permeability) jointly influence the onset of convection and the overall heat transfer enhancement in both Darcy and non-Darcy flow regimes [23, 24]. Over the past decades, numerous efforts have been made to develop convective transport models for nanofluids [25]. To address the limitations of earlier formulations, Buongiorno [26] proposed an alternative model in 2006 that has since gained widespread acceptance. Using scaling analysis, he demonstrated that Brownian diffusion and thermophoresis are the dominant slip mechanisms in laminar forced convection of nanofluids, with thermophoresis potentially being up to two orders of magnitude more significant than Brownian motion. The resulting framework is a non-homogeneous, two-component, four-equation model comprising partial differential equations governing particle volume fraction, thermal energy, and flow momentum. In this formulation, the liquid volume fraction is determined by that of the nanoparticles, and thermal equilibrium between the base fluid and the particles is assumed, leading to a common temperature field.

Many researchers have adopted Buongiorno's model in subsequent studies. For instance, Ramadevu, et al. [27] used a homotopy analysis method to examine the dynamic, thermal, and mass transport behaviors of viscoelastic flow in a Darcy porous medium filled with Buongiorno nanofluid over a nonlinear stretching sheet. Their analysis revealed that the Rayleigh number, nanoparticle volume fraction, permeability, and thermal conductivity ratio play crucial roles in modifying the flow velocity,

dimensionless temperature, and concentration distribution. A. Khan, et al. [28] investigated cross-diffusion through Soret and Dufour mechanisms and the effects of solar radiation on quadratic mixed convection of Maxwell-Buongiorno nanofluid over a cubic stratified surface. Alrawashdeh, et al. [29] used the Buongiorno nanofluid model to analyze unsteady natural convection flow of a non-Newtonian magnetohydrodynamic fluid via an oscillating vertical porous plate immersed in a saturated porous medium, taking into account a combination of chemical reaction, Joule heating, heat absorption/generation, and Soret and Dufour effects. Ratha, et al. [30] investigated two-dimensional unsteady flow of an incompressible Casson-Buongiorno nanofluid over a shrinking horizontal sheet under the influence of an inclined magnetic field and Joule heating. In this study, an analytical method via the Adomian decomposition method was used to solve the governing equations. In this study, the major contribution is that the Casson parameter, along with suction/injection, favors a smooth enhancement in the velocity profiles, and the fluid temperature is also enhanced by the augmentation of the Brownian and thermophoresis parameters. Vinodkumar Reddy, et al. [31] conducted a numerical analysis of the Cattaneo–Christov heat flux model for magnetohydrodynamic (MHD) flow of Ree–Eyring nanofluid through a Darcy–Forchheimer porous medium, accounting for entropy generation, thermal radiation, heat sources, activation energy, and gyrotactic microorganisms. Ahmed and Rashed [32] performed a numerical investigation of magnetohydrodynamic free convection and heat transfer of nanofluids in wavy enclosures filled with a heat-generating porous medium. The Buongiorno model was employed to describe nanoparticle transport, incorporating Brownian motion and thermophoresis, while the porous matrix was modeled using Darcy’s law under the assumption of slow flow. Chu, et al. [33] presented the bioconvection flow of nanofluid induced by two stretchable disks separated at confined distance. The Buongiorno nanofluid model was employed to examine the Brownian motion and thermophoresis aspects.

The purpose of the current paper is to bridge the gap between conventional numerical approaches and emerging deep learning methodologies by employing physics-informed neural networks (PINNs) to solve the highly nonlinear ordinary differential equations (ODEs) governing Buongiorno nanofluid natural convection flow. By embedding the underlying physical laws directly into the neural network architecture, the proposed framework improves both computational efficiency and predictive accuracy in modeling transport phenomena within non-Darcy porous media. The effects of key control parameters on the flow behavior and physical quantities, including the local Nusselt number, are presented graphically and discussed in detail. The findings of this work provide deeper insight into heat and mass transfer mechanisms in nanofluid systems and establish a robust foundation for the development of advanced, energy-efficient engineering applications.

2. Physical Problem Formulation

Let us consider the steady natural convection boundary-layer flow past a vertical semi-infinite flat plate embedded in a non-Darcy porous medium filled with an incompressible viscous nanofluid. The porous medium is described by the Darcy–Forchheimer model, which represents the inertia effect at the pore scale. The physical parameters of the porous medium, namely, the permeability and porosity, possess constant values. The Buongiorno model is adopted for the nanofluid. The physical model, direction of gravitational acceleration, and coordinate system are shown in Figure 1. The temperature T and nanoparticle volume fraction ϕ at the plate surface are T_w and ϕ_w , respectively. The corresponding ambient values, attained as y tends to infinity, are denoted by T_∞ and ϕ_∞ . The Oberbeck–Boussinesq approximation and boundary-layer assumptions for the nanofluid are assumed valid [34, 35]. Local thermal equilibrium in the homogeneous porous medium is also assumed; i.e., no slip occurs between the base fluid and the nanoparticles. It is assumed here that there are no radiation or viscous dissipation effects [36]. As mentioned before, only Brownian motion and thermophoresis effects are retained in the present configuration as the main mechanisms that explain the enhanced heat transfer by a nanofluid.

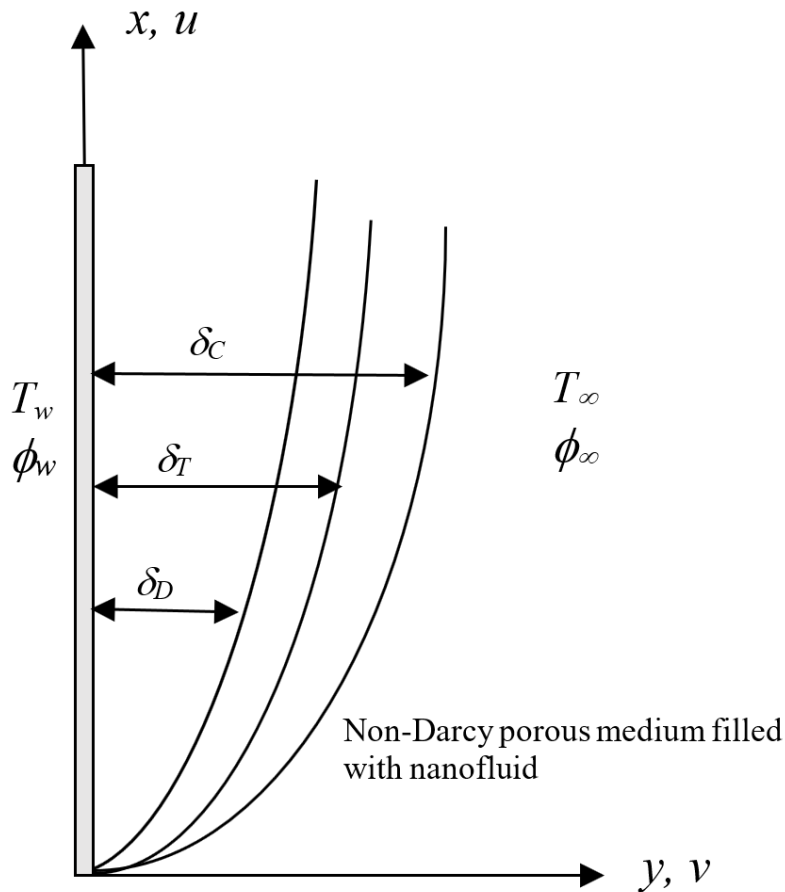


Figure 1.
Flow model and physical coordinate system.

Under the above assumptions, the boundary layer equations governing the flow, temperature, and nanoparticle concentration field can be written in dimensional form as

- Continuity equation

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

- Momentum equations

$$\frac{\partial p}{\partial x} = -\frac{\mu}{K}u - \frac{\rho_{f\infty}c_f}{\sqrt{K}}u^2 + \left[(1-\phi_{\infty})\rho_{f\infty}g\beta_T(T-T_{\infty}) - (\rho_p - \rho_{f\infty})g(\phi - \phi_{\infty}) \right] \quad (2)$$

$$\frac{\partial p}{\partial y} = 0 \quad (3)$$

- Energy equation

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \alpha_m \frac{\partial^2 T}{\partial y^2} + \tau \left[D_B \frac{\partial \phi}{\partial y} \frac{\partial T}{\partial y} + \left(\frac{D_T}{T_{\infty}} \right) \left(\frac{\partial T}{\partial y} \right)^2 \right] \quad (4)$$

- Nanoparticle volume fraction equation

$$\frac{1}{\varepsilon} \left(u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} \right) = D_B \frac{\partial^2 \phi}{\partial y^2} + \left(\frac{D_T}{T_\infty} \right) \frac{\partial^2 T}{\partial y^2} \quad (5)$$

Given the situation, p is the pressure, T is the temperature, ϕ is the nanoparticle volume fraction. K is the permeability of the porous medium; ρ , μ , and β_r are the density, the dynamic viscosity, and the volumetric thermal expansion coefficient of the base fluid, respectively. ρ_r is the density of the nanoparticles, and α_m represents the thermal diffusivity of the porous medium. The gravitational acceleration is denoted by g . The coefficients that appear in Eqs. (4) and (5) are the Brownian diffusion coefficient D_B and the thermophoretic diffusion coefficient D_T , respectively. u and v are the components of the velocity field, while x and y correspond to the vertical and horizontal coordinates, respectively. The corresponding wall and free-stream boundary conditions are defined as follows

$$\begin{aligned} y=0: \quad v=0, T=T_w, D_B \frac{\partial \phi}{\partial y} + \frac{D_T}{T_\infty} \frac{\partial T}{\partial y} &= 0 \\ y \rightarrow \infty: \quad u \rightarrow 0, T \rightarrow T_\infty, \phi &\rightarrow \phi_\infty \end{aligned} \quad (6)$$

The local Rayleigh number Ra_x is defined by $Ra_x = (1 - \phi_\infty) \rho_{f\infty} \beta_T g K x (T_w - T_\infty) / (\mu \alpha_m)$

We investigate the situation where heat transfer dominates over mass transfer and the case where the Lewis number is larger than unity. Following the scale analysis presented by Nield and Bejan [37], pertinent variables are found to be appropriate. Making use of the following transformation.

$$\eta = \frac{y}{x} Ra_x^{1/2}; \quad \psi = \alpha_m Ra_x^{1/2}; \quad \theta = \frac{T - T_\infty}{T_f - T_\infty}; \quad C = \frac{\phi - \phi_\infty}{\phi_w - \phi_\infty} \quad (7)$$

Where ψ is the classical stream function from which the velocity components are

$$u = \frac{\partial \psi}{\partial y}; \quad v = -\frac{\partial \psi}{\partial x} \quad (8)$$

Equations (1) ~ (5), with the appropriate transformations and algebraic combination described above (by eliminating the pressure p from Eqs. (2) and (3) through cross-differentiation), can be further reduced to a set of ordinary differential equations.

$$f'' + 2G f' f'' = \theta' - Nr C' \quad (9)$$

$$\theta'' + \frac{1}{2} f \theta' + Nb C' \theta' + Nt (\theta')^2 = 0 \quad (10)$$

$$C'' + \frac{1}{2} Le_p f C' + \frac{Nt}{Nb} \theta'' = 0 \quad (11)$$

The corresponding boundary conditions become

$$\eta = 0: f = 0; \theta = 1; Nb C' + Nt \theta' = 0 \quad (12)$$

$$\eta \rightarrow \infty: f' \rightarrow 0; \theta \rightarrow 0; C \rightarrow 0 \quad (13)$$

From the above equations $G = (\rho_{f\infty} \alpha_m c_f / \mu \alpha) Ra_x$ represent the no-Darcy parameter: Nr is the nanoparticle buoyancy ratio, Nb is the Brownian motion parameter $Nb = \tau D_B (\phi_w - \phi_\infty) / \alpha_m$, Nt is the

thermophoresis parameter $Nt = \tau D_T (T_w - T_\infty) / T_\infty \alpha_m$, and Le_p is the nanoparticles Lewis number $Le_p = \varepsilon \alpha_m / D_B$.

The physical quantities of interest are the local Nusselt number Nu_x , and the nanofluid Sherwood number Sh_x , which represents the dimensionless mass flux and is identically zero due to the proposed boundary conditions at the wall [34].

The local Nusselt number Nu_x is defined by the following equation $Nu_x = -\frac{x q_w}{k_m (T_w - T_\infty)}$ where

$$q_w = k_m \left(\frac{\partial T}{\partial y} \right)_{y=0}$$

Using the dimensionless variables, we obtain $Nu_x = -\theta'(0)$

3. Physics-Informed Neural Networks (PINN)

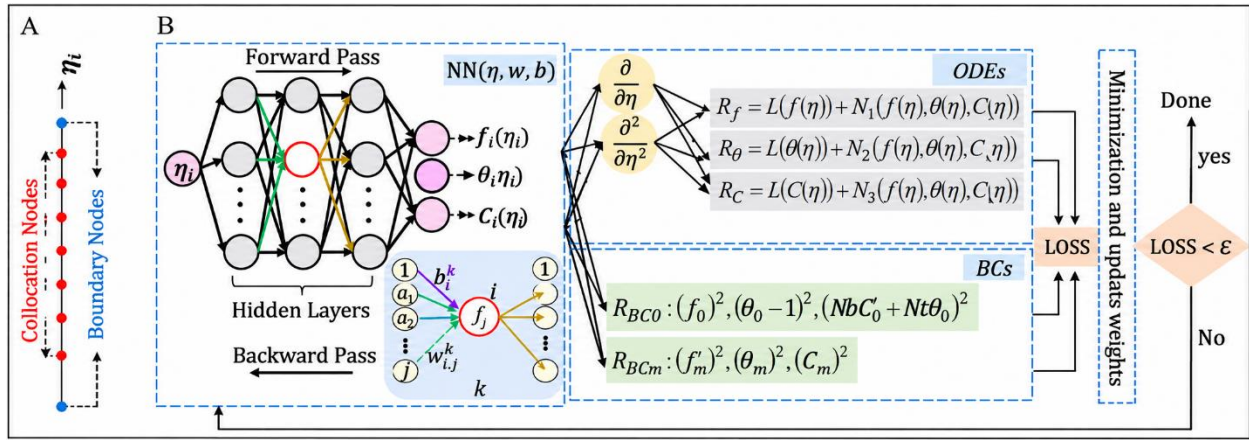


Figure 2. Schematic Diagram of a physics-informed neural network (PINN): (A) discretization of the one-dimensional domain, (B) neural network structure, and the corresponding computational algorithm.

To solve Equations (9–11) using a Physics-Informed Neural Network (PINN) framework, the one-dimensional computational domain is discretized along the similarity variable η into a finite set of nodes. This discretization enables the systematic evaluation of the governing equations and facilitates the training of the network through the incorporation of penalty terms that enforce adherence to both the differential equations and the associated boundary conditions. The selected nodes correspond to the independent variable η , which spans the interval $[0, \eta_\infty]$. As illustrated in Figure 2A, two distinct categories of nodes are employed: collocation points (denoted in red), where the residuals of the governing equations are minimized, and boundary points (denoted in blue), where the boundary conditions are explicitly imposed.

The computational domain along the (η) -axis is uniformly discretized into (m) equal subintervals, denoted by (η_i) for $(i = 0, 1, 2, \dots, m)$, such that $(\eta_i = i\Delta)$, where $(\Delta = \eta_\infty/m)$. In contrast to the conventional PINN framework, in which collocation points are commonly sampled randomly within the domain, the present study adopts a structured and uniform discretization strategy. A total of $(m+1)$ nodes are therefore defined along the (η) -axis. The first and last nodes, (η_0) and (η_m) , correspond to boundary nodes at which the prescribed boundary conditions at $(\eta_0 = 0)$ and $(\eta = \eta_\infty)$ are enforced. The

remaining intermediate nodes, (η_i) for $(i = 1, 2, \dots, m - 1)$, serve as collocation points where the residuals of the governing ordinary differential equations (ODEs) are minimized. The neural network architecture consists of an input layer with a single neuron, as illustrated in Figure 2B, which receives the independent variable (η_i) as input data. A nonlinear mapping function is then constructed to relate the input $(x_i = \eta_i)$ to the predicted solution vector $(\hat{u}_i = [f(\eta_i), \theta(\eta_i), C(\eta_i)])$, representing the approximate solutions of the ODE system. This mapping can be expressed as $(\hat{u}_i = \mathbf{F}(x_i, w, b))$ where (w) and (b) denote the trainable weights and biases of the network, respectively. These parameters are iteratively optimized during the training process to achieve an accurate functional approximation, ensuring that the predicted outputs $(\hat{u}_i)$ closely satisfy the governing equations and boundary conditions. The discrepancy between the predicted and target values is quantified through a loss function formulated in terms of the mean squared error (MSE), defined as follows

$$Loss = \left[\sum_{i=0}^{i=m} (u_i - \hat{u}_i)^2 \right] / m \quad (14)$$

The function $\mathbf{F}$ represents the sequence of algebraic and nonlinear transformations performed by all neurons constituting the neural network. To clarify its operation, Figure 2B illustrates the network architecture adopted, in which the input variable (η_i) is introduced. The first layer, referred to as the input layer, consists of a single neuron responsible for receiving and transmitting the input data. This layer is followed by three hidden layers, each containing an arbitrary number of neurons, providing sufficient representational capacity to approximate the solution. The output layer comprises three neurons, corresponding to the three unknown functions to be determined, namely, the dimensionless velocity, temperature, and nanoparticle volume fraction.

The network is structured into layers indexed by $k = 0, 1, 2, \dots, l$, where each layer contains multiple neurons indexed by $j = 1, 2, \dots, n$. The neurons are fully connected between successive layers through weighted links, which transmit information forward from one layer to the next. To describe the computations within a representative neuron, consider a neuron in layer k . The incoming signals from the previous layer $(k-1)$ are multiplied by corresponding weights $(w_{i,j}^k)$, which regulate the strength of each connection by amplifying or attenuating the transmitted information. These weights quantify the relative contribution of each neuron in the preceding layer to the current neuron's output. The neuron then computes the weighted sum of all incoming signals and adds a bias term, yielding an intermediate value (s_i) , as expressed in Eq. (15).

$$s_i = b_i + \left(\sum_{j=1}^{j=n} a_j w_{i,j}^k \right) \quad (15)$$

The bias term (b_i) can be interpreted as an additional parameter introduced to shift the activation function and improve the flexibility of the model. Practically, it may be viewed as a neuron with a constant value of one that is connected to neurons in the input and hidden layers, but not in the output layer. Its role is to adjust the predicted values and reduce systematic deviations between the estimated and desired outputs. By analogy with linear regression, where a function is expressed as $f(x) = \alpha x + \beta$, the weights and biases in a neural network play roles similar to the coefficients α and β , respectively. The computed intermediate value (s_i) is subsequently passed through an activation function, which introduces nonlinearity and constrains the neuron's output within a specific range, thereby enhancing the network's capacity to approximate complex nonlinear relationships. The function used in this study is the hyperbolic tangent activation function:

$$\text{output}(i) = \tanh(s_i) \quad (16)$$

The information within the network propagates strictly in a forward direction, implying that the connections between neurons are acyclic and do not form feedback loops. At each neuron, a set of computations is performed to generate an intermediate output value (s_i) . This process, commonly

referred to as forward propagation (or the forward pass), proceeds sequentially from one layer to the next until the output layer is reached. In the output layer, no activation function is applied; thus, the computed values (s_i) are directly interpreted as the predicted solutions ($\hat{u}_i$). Since the network parameters, namely, the weights and biases, are initially assigned arbitrary values, a discrepancy generally exists between the predicted outputs (s_i) and the exact solutions (u_i). Reducing this discrepancy requires minimizing a suitably defined loss function through systematic adjustment of the trainable parameters. Identifying the optimal set of weights and biases that drives the loss function toward zero constitutes the core of the optimization process. This is accomplished by computing the gradients of the loss function with respect to all trainable parameters using the chain rule in conjunction with the backpropagation algorithm.

The training procedure is conducted in two successive stages. First, the Adam optimizer is employed to ensure stable and efficient global convergence during the initial training phase. Subsequently, the quasi-Newton L-BFGS optimizer is applied to refine the solution and improve its accuracy. These forward and backward passes (iterations) are repeated until the loss function satisfies a predefined convergence criterion. Once the training process is completed, the calibrated model is capable of predicting solutions for new, previously unseen inputs through a single forward pass. In the present framework, the exact analytical solutions (u_i) are not available, which raises the fundamental question of how to define an appropriate loss function. Raissi, et al. [13] and Raissi, et al. [14] addressed this challenge by introducing a physics-based loss function derived directly from the governing ordinary differential equations (ODEs). The central idea is that the predicted outputs ($\hat{u}_i = [f(\eta_i), \theta(\eta_i), C(\eta_i)]$) must satisfy the underlying differential equations. Accordingly, the loss function is formulated to quantify the extent to which the predicted solutions fulfill the ODE system. Within this methodology, the neural network approximation is simultaneously responsible for enforcing the governing equations and satisfying the associated boundary conditions, thereby ensuring physical consistency of the obtained solution.

The total loss function for solving an ODE problem is decomposed into two components: (1) a loss term associated with the boundary conditions, and (2) a loss term related to the residuals of the governing ODEs. As illustrated in Figure 2B, multiple loss contributions are considered. The residual losses (R_f), (R_θ), and (R_C) correspond to the coupled flow, thermal, and concentration equations, respectively, while the boundary losses related to Dirichlet and Neumann conditions are denoted by (R_{BC0}) and (R_{BCm}). The subscripts (f), (θ), and (C) specify whether the boundary condition is associated with the flow, thermal, or concentration equation. In the following sections, these individual loss terms, as well as the overall loss function, are derived for the case studies and subsequently minimized through the optimization process.

$$R_f = \sum_{i=1}^{i=m-1} f_i'' + 2G f_i' f_i'' - (\theta_i' - NrC_i') / m - 1 \quad (17)$$

$$R_\theta = \sum_{i=1}^{i=m-1} \theta_i'' + \frac{1}{2} f_i \theta_i' + NbC_i' \theta_i' + Nt(\theta_i')^2 / m - 1 \quad (18)$$

$$R_C = \sum_{i=1}^{i=m-1} C_i'' + \frac{1}{2} Le_p f_i C_i' + \frac{Nt}{Nb} \theta_i'' / m - 1 \quad (19)$$

$$R_{BC0} : (f_0')^2, (\theta_0 - 1)^2, (NbC_0' + Nt\theta_0')^2 \quad (20)$$

$$R_{BCm} : (f_m')^2, (\theta_m)^2, (C_m)^2 \quad (21)$$

The total loss is the summation of all these losses:

$$Total\ loss = R_f + R_\theta + R_C + R_{BC0} + R_{BCm} \quad (22)$$

Thus, every training point must satisfy a particular constraint according to its position, either on the boundary or within the interior of the computational domain. It is worth noting that minimizing the overall loss function to determine the optimal network weights involves performing a forward propagation followed by a backpropagation step, which together constitute one training iteration (epoch). At each epoch, the loss value is evaluated to measure how accurately the imposed physical and boundary conditions are fulfilled. Furthermore, all derivatives appearing in the loss function are computed through automatic differentiation, as implemented in the PyTorch package. During the training process, the optimizer iteratively adjusts the network's weights and biases in order to minimize the overall loss function at each epoch. Consequently, a well-trained and reliable model is characterized by a progressive reduction in the loss value as the epochs proceed. The training phase is maintained until a convergence condition is satisfied, typically when the difference in loss between two consecutive epochs becomes smaller than a prescribed tolerance (for example, 10^{-6}).

4. Results and Discussion

To enhance the credibility of our neural network and obtain an optimized model, we investigated the influence of several structural parameters, such as the width (i.e., the number of hidden layers) and the depth (i.e., the number of neurons per layer), as well as hyperparameters like the learning rate of the optimizer used.

Subsequently, a parametric study of the physical problem was carried out by analyzing the dynamic, thermal, and mass behavior (i.e., the influence of selected control parameters). To validate the developed neural network, the obtained results were compared with those computed using an implicit finite difference method (bvp4c) implemented in MATLAB, in terms of velocity profile, temperature distribution, nanoparticle volume fraction, and the local Nusselt number.

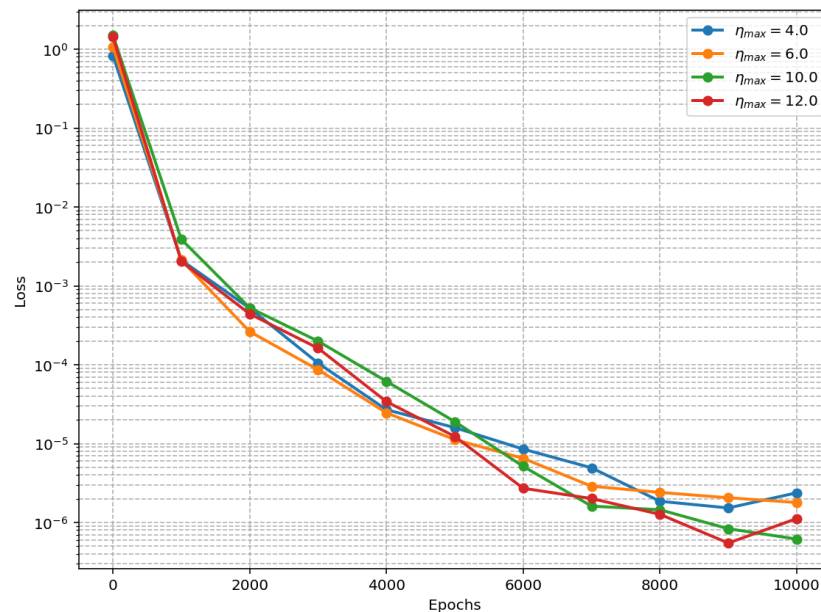


Figure 3.
Evolution of the loss function with $\eta_{\max}$ for PINN (50 neurons, 3 hidden layers).

Figure 3 shows the evolution of the loss function (on a logarithmic scale) as a function of the number of epochs for different values of $\eta_{\max}$. It is observed that, regardless of the value of $\eta_{\max}$, the loss decreases significantly during training, indicating good convergence of the PINN and an accurate solution of the governing equations. The rapid drop in the loss during the early iterations reflects

efficient learning of the global structure of the solution, while during the intermediate and final phases, the differences between the $\eta_{\max}$ values become more pronounced. The cases $\eta_{\max}=10$ and $\eta_{\max}=12$ exhibit faster convergence and reach lower loss levels (on the order of 10^{-6}) compared to smaller values, suggesting that too short a domain in η limits the solution accuracy. The slight oscillations observed at the end of training are typical of PINNs and correspond to numerical saturation. Overall, increasing $\eta_{\max}$ improves the accuracy and the satisfaction of the asymptotic conditions; however, the gains become marginal beyond $\eta_{\max} \approx 10$, making this value an optimal compromise between numerical accuracy and computational cost.

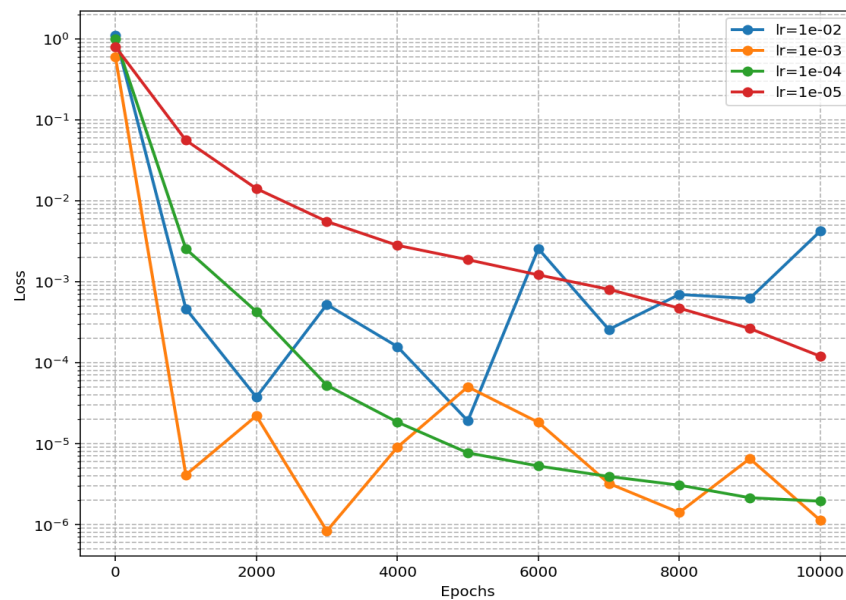


Figure 4. Influence of the learning rate on the loss function via PINN (50 neurons,3 hidden layers).

Figure 4 illustrates the evolution of the loss function as a function of the number of epochs for different values of the learning rate ($lr=10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$). Overall, all curves exhibit a decrease in the loss during training, confirming that the PINN converges in all cases, albeit with markedly different behaviors depending on the learning rate. For a high learning rate ($lr=10^{-2}$), the convergence is unstable: although the loss decreases rapidly at the beginning, it subsequently shows strong oscillations and even an increase toward the end of training, indicating that the network is having difficulty stabilizing around the minimum. The case of $lr=10^{-3}$ demonstrates fast and efficient convergence, reaching the lowest loss values (on the order of 10^{-6}), making it the best compromise between convergence speed and stability. For $lr=10^{-4}$, the convergence is more regular and stable but slower, with a gradual decrease in the loss that remains slightly higher than that obtained with $lr=10^{-3}$. Finally, a very small learning rate ($lr=10^{-5}$) leads to very slow convergence: the loss decreases monotonically but remains relatively high even after 10,000 epochs, indicating underfitting.

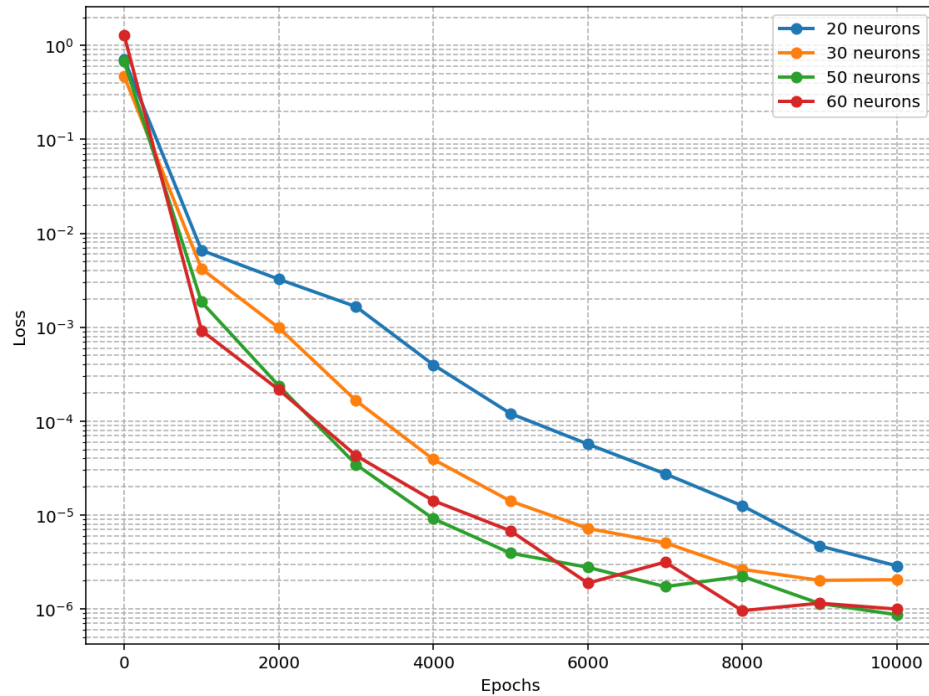


Figure 5.
Effects of the number of neurons on the loss function (3 hidden layers).

Figure 5 depicts the development of the loss as a function of the number of epochs for different network depths (20, 30, 50, and 60 neurons). First, it can be observed that for all networks, the loss decreases sharply during the initial epochs, indicating a good initial learning capability of the model. Subsequently, the decrease becomes more gradual, reflecting a phase of solution refinement. When comparing the architectures, the network with 20 neurons converges more slowly and consistently exhibits a higher loss, indicating that it is under-parameterized and therefore less capable of capturing the problem dynamics. The 30-neuron network shows a clear improvement in convergence but remains less effective than the wider architectures. The networks with 50 and 60 neurons exhibit the best convergence performance, reaching loss values on the order of 10^{-6} , with faster and more stable decay. The 60-neuron network achieves the lowest final loss; however, the improvement compared to the 50-neuron network is marginal. In summary, this figure highlights that increasing the number of neurons improves the accuracy and accelerates the convergence of the PINN up to a certain threshold. Beyond this point (50 $\rightarrow$ 60 neurons), the performance gain becomes limited, suggesting a good cost-performance trade-off around 50 neurons.

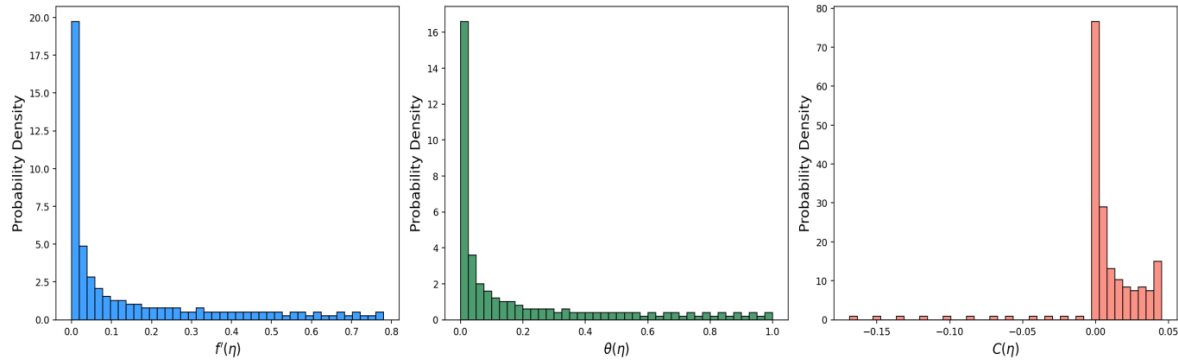


Figure 6.
Probability density functions of velocity field, temperature, and nanoparticles volume fraction.

Figure 6 explains the probability density functions (PDF) of the three dimensionless fields ($f(\eta)$), ($\theta(\eta)$), and ($C(\eta)$), providing insight into their statistical distributions over the computational domain. For ($f(\eta)$), the distribution is strongly skewed, with a pronounced peak near zero, indicating that low velocity values dominate most of the domain. The long tail toward higher values reflects the presence of larger velocity gradients localized near the wall, which is characteristic of boundary-layer-type flows. The PDF ($\theta(\eta)$) exhibits a similar trend, with a high probability density at low temperature values and a gradual decay toward larger values. This suggests that the thermal field remains close to the ambient state over most of the domain, while significant temperature variations are confined to a limited region near the wall. Regarding ($C(\eta)$), the distribution is much narrower and more centered around values close to zero, with a slight extension toward negative values. This relatively symmetric and low-dispersion behavior indicates that concentration variations are small and well-controlled, reflecting a more stable mass transfer process compared to the velocity and temperature fields. Overall, these PDFs confirm the physical consistency of the PINN solution: the velocity and temperature fields exhibit strong localized variations near the wall, while the concentration field remains more homogeneous, which is consistent with classical coupled flow, heat, and mass transfer phenomena combined with their boundary conditions.

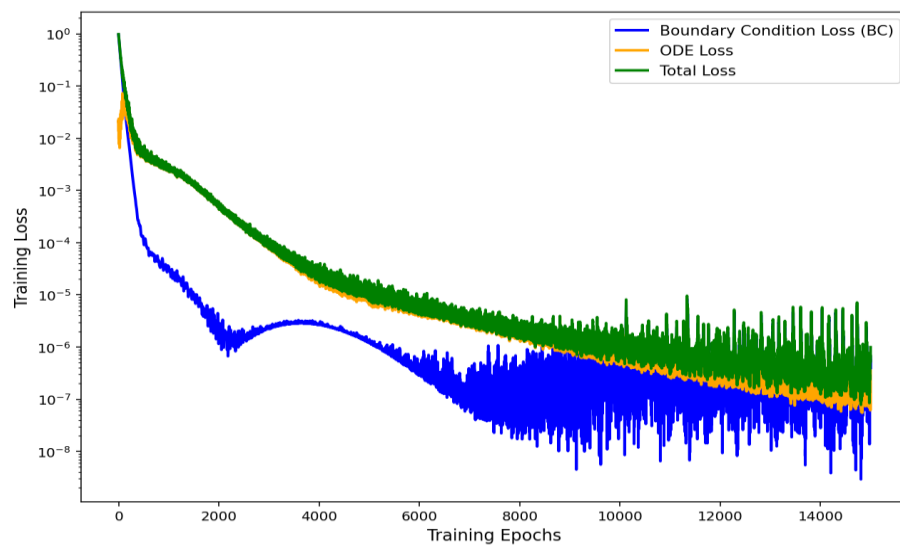


Figure 7.
Variation of loss function vs. training epochs.

Figure 7 illustrates the evolution of the training loss components as a function of the number of training epochs. The boundary condition loss (BC) decreases rapidly during the initial stages of training, indicating that the PINN quickly satisfies the imposed boundary constraints. After this sharp decay, the BC loss stabilizes at very low values, with small oscillations attributed to stochastic optimization effects. The ODE loss exhibits a more gradual, smoother decay, reflecting the progressive minimization of the residuals of the governing differential equations as training proceeds. The total loss closely follows the combined behavior of the BC and ODE losses, showing an overall monotonic decrease over several orders of magnitude, which confirms stable and effective convergence of the model. The slight fluctuations observed at later epochs, particularly in the total and BC losses, are typical in physics-informed neural networks and indicate fine-scale adjustments of the solution rather than divergence. Overall, this figure demonstrates the robustness of the training process and the ability of the PINN to simultaneously enforce boundary conditions and accurately satisfy the governing equations.

To check the accuracy of our methods (PINN and bvp4c), the corresponding results are compared with those obtained by Aziz and Khan [36] using the Nusselt number in Table 1 for the same case (Darcy flow). The error (ε) is defined as the relative difference between the estimated Nusselt number from the linear regression and the Nusselt number obtained using the methods used. It should be noted that a better estimation can be obtained with a quadratic regression, as in Aziz and Khan [36]. It is clear from this table that the present results are in good agreement with those reported in the cited work over a large range of Le_p .

Table 1.

Comparison between the maximum relative errors ε (Nusselt number).

Le	$\varepsilon(\text{Model})$ [36]	$\varepsilon(\text{bvp 4c})$ Present work	$\varepsilon(\text{PINN}); (50, 3)$ Present work
5	0.003	0.0032	0.0032
10	0.009	0.0020	0.002
20	0.003	0.0028	0.0028
50	0.004	0.0039	0.0039
100	0.005	0.00489	0.00490

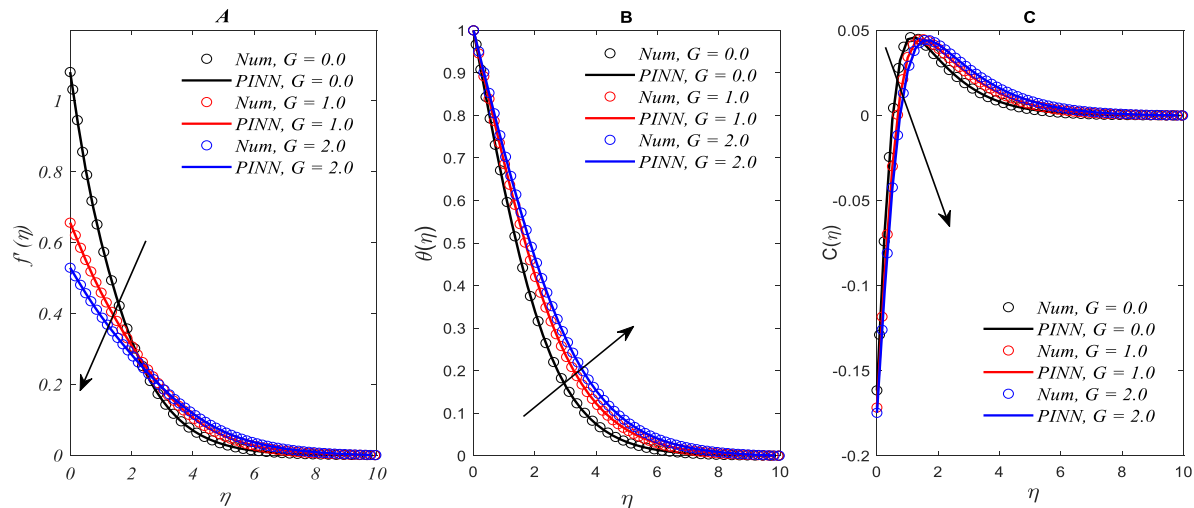


Figure 8.

Effect of the non-Darcy parameter G on : (A) Velocity field, (B) Temperature, (C) Nanoparticle volume fraction ($Nt = Nb = Nr = 0.5$, $Le_p = 10$).

Figure 8 illustrates the influence of the parameter (G) on the velocity, temperature, and volume fraction profiles, comparing PINN predictions with numerical results. An increase in (G) leads to a

noticeable reduction in fluid velocity and a thinning of the momentum boundary layer, reflecting the dissipative effect of this parameter. In contrast, higher values of (G) enhance both the temperature and the volume fraction within the boundary layer, indicating thicker thermal and concentration layers and stronger heat and mass transfer. The profiles show the expected asymptotic behavior as the similarity variable increases, and the excellent agreement between the PINN solutions and the numerical data across all cases confirms the accuracy, robustness, and reliability of the PINN approach in capturing the coupled nonlinear transport phenomena.

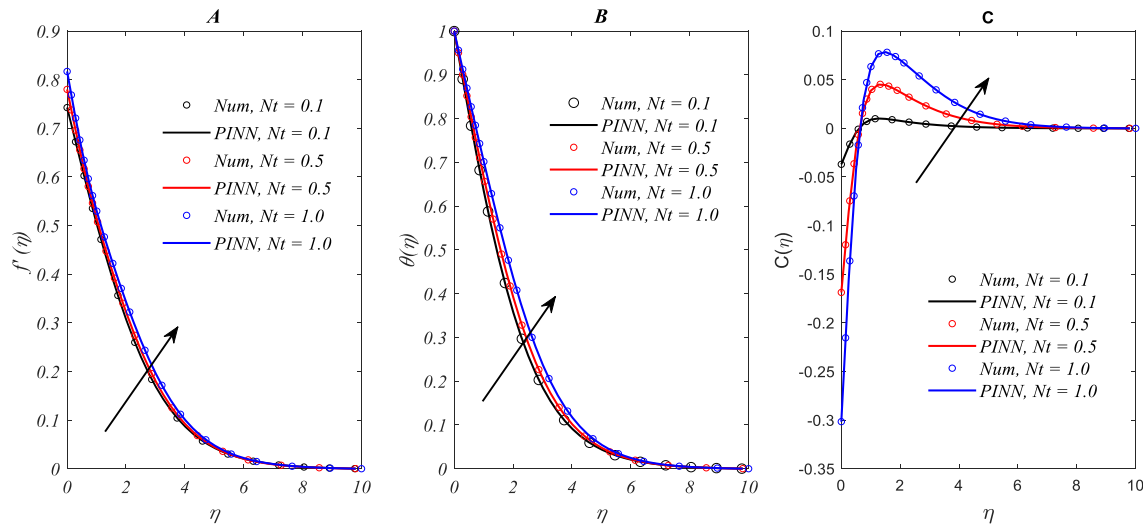


Figure 9.

Influence of the thermophoresis parameter, Nt , on (A) Velocity field, (B) Temperature, (C) Nanoparticle volume fraction; ($G = Nb = Nr = 0.5$, $Le_p = 10$).

Figure 9 illustrates the effect of the thermophoresis parameter (Nt) on the velocity, temperature, and volume fraction profiles and compares PINN predictions with numerical solutions. Increasing (Nt) has only a minor influence on the velocity profile, while it significantly enhances the temperature within the thermal boundary layer, indicating stronger thermal transport. The same results were observed by Bouaziz and Hanini [38]. In contrast, the volume fraction exhibits a pronounced non-monotonic behavior, with higher peak values as (Nt) increases, reflecting the intensified particle migration induced by temperature gradients. The profiles approach their asymptotic states smoothly, and the excellent agreement between the PINN and numerical results across all cases confirms the accuracy and reliability of the PINN approach in capturing the coupled heat and mass transfer mechanisms.

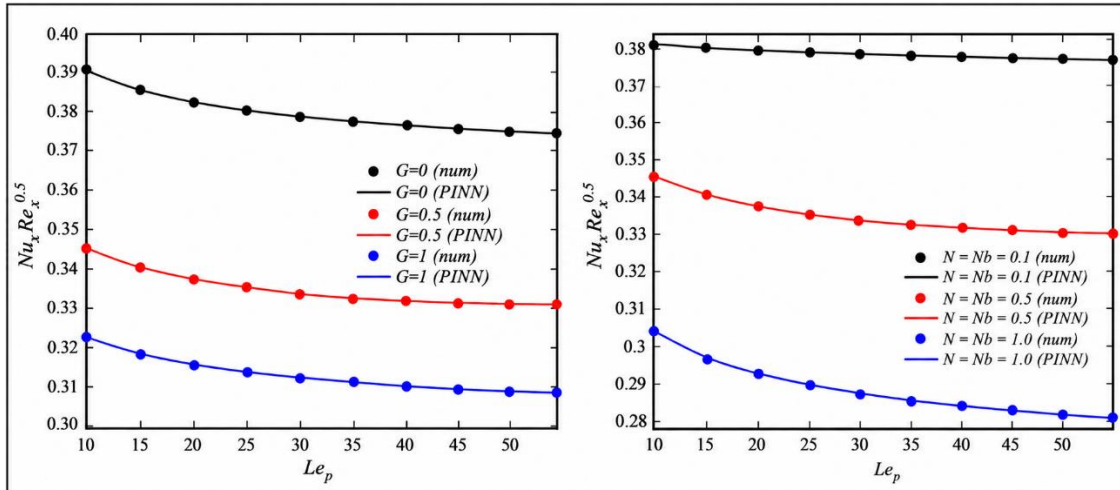


Figure 10.
Effects of the G , Nt , and Nb on heat transfer rate (Local Nusselt number).

Figure 10 illustrates the effects of the governing parameters G (non-Darcy parameter), Nt (thermophoresis), and Nb (Brownian motion) on the heat transfer rate, represented by the local Nusselt number, as a function of the nanoparticle Lewis number Le_p . The results obtained using the PINN approach are compared with numerical solutions. In both panels (a) and (b), the heat transfer rate decreases monotonically with increasing Le_p , indicating that larger Lewis numbers weaken thermal diffusion relative to mass diffusion, thereby reducing the rate of heat transfer at the surface. In Figure 10(a), increasing the parameter G leads to a noticeable reduction in the heat transfer rate, suggesting that stronger resistance effects associated with G suppress thermal transport. In Figure 10(b), higher values of the thermophoresis and Brownian motion parameters (Nt and Nb) also result in lower heat transfer rates, reflecting the enhanced coupling between the thermal and nanoparticle concentration fields, which thickens the thermal boundary layer and diminishes the surface heat flux. Across all cases, the PINN predictions are in excellent agreement with the numerical results, confirming the accuracy and robustness of the PINN methodology.

5. Concluding Remarks

The present investigation successfully demonstrated the efficacy of the Physics-Informed Neural Network (PINN) framework in resolving the complex, coupled nonlinear ordinary differential equations governing natural convection flow within a non-Darcy porous medium saturated with a Buongiorno nanofluid. By embedding the fundamental conservation laws directly into the neural network's loss function, this study bypassed the need for traditional mesh-based discretization, offering a robust alternative to classical numerical solvers.

The following key conclusions are drawn from the study:

- **Methodological Robustness:** The PINN architecture, optimized through a hybrid Adam and L-BFGS training strategy, exhibited high convergence stability. The use of automatic differentiation ensured that the physical residuals were minimized with high precision, achieving loss levels on the order of 10^{-6} .
- **Domain Optimization:** Sensitivity analysis of the similarity variable η indicated that a computational domain of $\eta_{\max} \approx 10$ provides an optimal balance between satisfying asymptotic boundary conditions and maintaining computational efficiency.
- **Physical Insights:** The model successfully captured the intricate interplay between the Brownian motion (Nb) and thermophoresis (Nt) parameters. The integration of the non-Darcy–

Forchheimer model allowed for a more realistic representation of inertia effects in the porous matrix, which significantly influences the local Nusselt number and transport phenomena.

- **Validation and Reliability:** Comparative analysis with traditional numerical methods (finite difference/bvp4c) confirmed the high fidelity of the PINN approach. The results demonstrate that deep learning-based solvers can effectively bridge the gap between theoretical modeling and complex engineering applications.

In summary, this research establishes that PINNs are not only a viable alternative for solving high-dimensional nonlinear transport problems but also offer superior flexibility in handling multiphysics couplings. Future work could extend this framework to inverse problems, where the PINN can be used to identify unknown porous media properties or nanoparticle concentrations from sparse experimental data. More complex geometries and boundary conditions may also be investigated in future studies to confirm the reliability and utility of deep learning-based PINNs in scientific and engineering applications.

Nomenclature:

C	Dimensionless nanoparticle volume fraction.
c_f	Empirical constant of the Forchheimer model.
D_B	Brownian diffusion coefficient
D_T	Thermophoretic diffusion coefficient
F	Dimensionless stream function
g	Gravitational acceleration
Gr	Non-Darcy porous medium parameter
K	Permeability of the porous medium
k_m	Effective thermal conductivity of the porous medium
Le_p	Nanofluid Lewis number
Nb	Brownian motion parameter
Nc	Regular buoyancy ratio
Nr	Buoyancy ratio parameter
Nt	Thermophoresis parameter
P	Pressure
Ra_x	Local Rayleigh number
T	Temperature
T_∞	Ambient temperature attained as y tends to infinity
u, v	Velocity components in the x and y directions
x, y	Cartesian coordinates

Greek Symbols:

α_m	Thermal diffusivity of the porous medium
β_T	Volumetric thermal expansion coefficient of the fluid
ϕ	Nanoparticles volume fraction

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

© 2026 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

References

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.
- [2] S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2009.
- [3] A. Agrawal and A. Choudhary, "Deep materials informatics: Applications of deep learning in materials science," *MRS Communications*, vol. 9, no. 3, pp. 779-792, 2019. <https://doi.org/10.1557/mrc.2019.73>
- [4] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge, England: Cambridge University Press, 2019.
- [5] S. M. Moosavi, K. M. Jablonka, and B. Smit, "The role of machine learning in the understanding and design of materials," *Journal of the American Chemical Society*, vol. 142, no. 48, pp. 20273-20287, 2020. <https://doi.org/10.1021/jacs.0c09105>
- [6] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug Discovery Today*, vol. 23, no. 6, pp. 1241-1250, 2018. <https://doi.org/10.1016/j.drudis.2018.01.039>
- [7] D. S. W. Ting *et al.*, "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *JAMA*, vol. 318, no. 22, pp. 2211-2223, 2017. <https://doi.org/10.1001/jama.2017.18152>
- [8] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: A review of recent research," *Advanced Robotics*, vol. 31, no. 16, pp. 821-835, 2017. <https://doi.org/10.1080/01691864.2017.1365009>
- [9] P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis, "Could machine learning break the convection parameterization deadlock?," *Geophysical Research Letters*, vol. 45, no. 11, pp. 5742-5751, 2018. <https://doi.org/10.1029/2018GL078202>
- [10] C. D. Ellis and H. Xia, "Vector-based neural network turbulent heat flux closures in near-wall cooling jet flows," *International Journal of Heat and Mass Transfer*, vol. 244, p. 126893, 2025. <https://doi.org/10.1016/j.ijheatmasstransfer.2025.126893>
- [11] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," *Physics of Fluids*, vol. 33, no. 5, p. 055133, 2021. <https://doi.org/10.1063/5.0048909>
- [12] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987-1000, 1998. <https://doi.org/10.1109/72.712178>
- [13] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," *arXiv:1711.10561*, 2017. <https://doi.org/10.48550/arXiv.1711.10561>
- [14] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026-1030, 2020. <https://doi.org/10.1126/science.aaw4741>
- [15] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686-707, 2019. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [16] H. Bararnia and M. Esmailpour, "On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems," *International Communications in Heat and Mass Transfer*, vol. 132, p. 105890, 2022. <https://doi.org/10.1016/j.icheatmasstransfer.2022.105890>
- [17] S. Sarthak and D. Srinivasacharya, "Physics informed neural network modeling for MHD natural convection: Thermal radiation and cross-diffusion effects on vertical plate dynamics," *European Journal of Mechanics-B/Fluids*, vol. 114, p. 204307, 2025. <https://doi.org/10.1016/j.euromechflu.2025.204307>
- [18] M. Jawad, H. Zafar, Z. Fu, S. Alshehry, M. Junaid, and I. Khan, "Physics-informed neural networks for rotating EMHD flow of Jeffrey hybrid nanofluid with Arrhenius activation energy and mass convections," *Expert Systems with Applications*, vol. 271, p. 126517, 2025. <https://doi.org/10.1016/j.eswa.2025.126517>
- [19] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *Journal of Heat Transfer*, vol. 143, no. 6, p. 060801, 2021. <https://doi.org/10.1115/1.4050542>
- [20] R. Li, J. Zhou, J.-X. Wang, and T. Luo, "Physics-informed Bayesian neural networks for solving phonon Boltzmann transport equation in forward and inverse problems with sparse and noisy data," *ASME Journal of Heat and Mass Transfer*, vol. 147, no. 3, p. 032501, 2025. <https://doi.org/10.1115/1.4067163>
- [21] S. K. Rudrabhatla and D. Srinivasacharya, "Deep learning framework for Casson fluid flow: A PINN approach to heat and mass transfer with chemical reaction and viscous dissipation," *European Journal of Mechanics-B/Fluids*, vol. 116, p. 204401, 2026. <https://doi.org/10.1016/j.euromechflu.2025.204401>

- [22] A. Jan, M. I. Khan, U. Farooq, F. Aslam, and R. Adnan, "PINN-based numerical modeling of MHD flows in porous media over linear stretching boundaries," *International Communications in Heat and Mass Transfer*, vol. 172, p. 110592, 2026. <https://doi.org/10.1016/j.icheatmasstransfer.2026.110592>
- [23] H. A. Nabwey, T. Armaghani, B. Azizimehr, A. M. Rashad, and A. J. Chamkha, "A comprehensive review of nanofluid heat transfer in porous media," *Nanomaterials*, vol. 13, no. 5, p. 937, 2023. <https://doi.org/10.3390/nano13050937>
- [24] K. Vafai, *Handbook of porous media*, 3rd ed. Boca Raton, FL: CRC Press, 2015.
- [25] E. Bänsch, S. Faghih-Naini, and P. Morin, "Convective transport in nanofluids: The stationary problem," *Journal of Mathematical Analysis and Applications*, vol. 489, no. 1, p. 124151, 2020. <https://doi.org/10.1016/j.jmaa.2020.124151>
- [26] J. Buongiorno, "Convective transport in nanofluids," *Journal of Heat Transfer*, vol. 128, no. 3, pp. 240–250, 2006. <https://doi.org/10.1115/1.2150834>
- [27] S. Ramadevu, P. V. Kumar, G. Lorenzini, S. M. Ibrahim, and K. Jyothsna, "Viscoelastic flow analysis with Buongiorno nanofluid model over a nonlinear stretching sheet: A homotopy approach," *Frontiers in Heat and Mass Transfer*, vol. 23, no. 3, pp. 857–879, 2025. <https://doi.org/10.32604/fhmt.2025.062231>
- [28] A. Khan *et al.*, "Quadratic mixed convection of Maxwell-Buongiorno nanofluid over cubic stratified surface incorporating cross diffusion effects and solar radiation," *Scientific Reports*, vol. 15, no. 1, p. 20092, 2025. <https://doi.org/10.1038/s41598-025-07140-0>
- [29] A. Alrawashdeh, M. Awais, H. Al Garalleh, and R. F. Riaz, "Numerical analysis of Williamson nanofluid flow past an oscillating vertical porous plate: Influence of Soret, Dufour, and chemical reaction effects," *Modeling Earth Systems and Environment*, vol. 12, no. 1, p. 13, 2026. <https://doi.org/10.1007/s40808-025-02639-x>
- [30] P. K. Ratha, S. Mishra, R. Tripathy, and P. K. Pattnaik, "Analytical approach on the free convection of Buongiorno model nanofluid over a shrinking surface," *Proceedings of the Institution of Mechanical Engineers, Part N: Journal of Nanomaterials, Nanoengineering and Nanosystems*, vol. 237, no. 3–4, pp. 83–95, 2023. <https://doi.org/10.1177/23977914221103982>
- [31] M. Vinodkumar Reddy, J. M. Nath, K. Ramakrishna Reddy, T. K. Das, and K. Venugopal Reddy, "Cattaneo-Christov heat flux in MHD Ree-Eyring nanofluid flow via porous medium including entropy optimization and gyrotactic microorganism," *Journal of the Korean Physical Society*, vol. 86, no. 7, pp. 615–635, 2025. <https://doi.org/10.1007/s40042-025-01316-7>
- [32] S. E. Ahmed and Z. Z. Rashed, "MHD natural convection in a heat generating porous medium-filled wavy enclosures using Buongiorno's nanofluid model," *Case Studies in Thermal Engineering*, vol. 14, p. 100430, 2019. <https://doi.org/10.1016/j.csite.2019.100430>
- [33] Y.-M. Chu *et al.*, "Study of Buongiorno's nanofluid model for flow due to stretching disks in presence of gyrotactic microorganisms," *Ain Shams Engineering Journal*, vol. 12, no. 4, pp. 3975–3985, 2021. <https://doi.org/10.1016/j.asej.2021.01.033>
- [34] D. A. Nield and A. V. Kuznetsov, "The Cheng–Minkowycz problem for natural convective boundary-layer flow in a porous medium saturated by a nanofluid," *International Journal of Heat and Mass Transfer*, vol. 52, no. 25–26, pp. 5792–5795, 2009. <https://doi.org/10.1016/j.ijheatmasstransfer.2009.07.024>
- [35] A. Aziz, W. A. Khan, and I. Pop, "Free convection boundary layer flow past a horizontal flat plate embedded in porous medium filled by nanofluid containing gyrotactic microorganisms," *International Journal of Thermal Sciences*, vol. 56, pp. 48–57, 2012. <https://doi.org/10.1016/j.ijthermalsci.2012.01.011>
- [36] A. Aziz and W. A. Khan, "Natural convective boundary layer flow of a nanofluid past a convectively heated vertical plate," *International Journal of Thermal Sciences*, vol. 52, pp. 83–90, 2012. <https://doi.org/10.1016/j.ijthermalsci.2011.10.001>
- [37] D. A. Nield and A. Bejan, *Convection in porous media*, 3rd ed. New York, NY: Springer, 2006.
- [38] A. M. Bouaziz and S. Hanini, "Double dispersion for double diffusive boundary layer in non-Darcy saturated porous medium filled by a nanofluid," *Journal of Mechanics*, vol. 32, no. 4, pp. 441–451, 2016. <https://doi.org/10.1017/jmech.2016.18>