

A new numerical method based on physics-informed neural networks for solving an optimal control problem governed by scale-invariant heat transfer equation

Melissa Mansouri , Karim Benalia , Brahim Oukacha  and
Karim Beddek 

Abstract. In this paper, we propose a novel numerical method based on physics-informed neural networks (PINNs) for solving a boundary optimal control problem governed by a scale-invariant one-dimensional heat equation with two-point boundary conditions. The original infinite-dimensional problem is first transformed into a finite-dimensional optimal control problem by combining finite-difference discretization in space with a trapezoidal approximation of the cost functional. Next, Pontryagin’s minimum principle is applied to derive the necessary optimality conditions, leading to a coupled Hamilton–Pontryagin system formulated as a nonlinear two-point boundary value problem (TPBVP). This TPBVP is then solved using a PINNs approach, which enables the simultaneous approximation of the state and adjoint variables while naturally enforcing the boundary and optimality conditions through a unified loss function. Numerical experiments demonstrate the effectiveness of the proposed method in accurately approximating the optimal scale-invariant boundary controls. The results are validated through comparison with classical numerical methods, showing that the proposed approach achieves competitive accuracy while offering greater flexibility for handling complex problems.

Mathematics Subject Classification (2010): 49K25, 65K10.

Keywords: Optimal control problems, scale-invariant one-dimensional heat transfer equation, Pontryagin’s minimum principle, physics-informed neural networks.

Received 03 March 2026; Accepted 08 July 2026.

© Studia UBB MATHEMATICA. Published by Babeş-Bolyai University

 This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introduction and problem statement

Optimal control problems, particularly those governed by partial differential equations (PDEs), are widely studied due to their many practical applications (see, e.g., [1, 5, 6, 7, 8, 16, 20, 21, 22, 23]). These problems aim to determine a control function that optimizes a prescribed cost functional while satisfying the constraints imposed by the governing PDEs. Two main strategies are used to solve such problems: ‘discretize-then-optimize’ and ‘optimize-then-discretize’. The first discretizes the problem into a nonlinear program, solved using deterministic or stochastic methods (see, e.g., [1, 6, 10, 14, 26]). The second derives optimality conditions via Pontryagin’s minimum principle [24], leading to a boundary value problem linked to the Hamilton–Jacobi–Bellman equation [24], which is then discretized for numerical solution. Common techniques include the shooting method, Adomian decomposition method (ADM), and variational iteration method (VIM) (see, e.g., [6, 9, 11, 15]). Comparative studies in the literature have examined the efficiency of these approaches (see, e.g., [2, 4, 27]).

In recent years, advances in machine learning and artificial neural networks (ANNs) have opened new opportunities for solving differential equations (see, e.g., [12, 19, 25]). In 2017, Raissi et al. [25] introduced physics-informed neural networks (PINNs), a deep learning framework for solving both forward and inverse problems governed by nonlinear PDEs. Unlike purely data-driven supervised learning, PINNs function as a physics-constrained optimization framework. They incorporate physical laws as inductive biases, efficiently learning solutions through a composite loss function that penalizes residuals from the governing equations, boundary conditions, and initial conditions. This approach allows the network to approximate the solution space consistently with the underlying physics, even in the absence of labeled observational data.

Several studies have demonstrated the effectiveness of PINNs for solving differential equations. Tang et al. [28] combined PINNs with polynomial interpolation to solve nonlinear PDEs, while Meng et al. [18] developed a parallel PINNs framework, significantly improving computational efficiency for long-term PDE integration. Cai et al. [12] applied PINNs to inverse heat transfer problems under forced and mixed convection, with numerical results confirming their effectiveness. However, PINNs have not yet been applied to optimal control problems governed by PDEs with scale invariance properties, due to the inherent complexity of such problems, motivating the present work.

We consider the Cauchy problem of the one-dimensional heat transfer equation:

$$\begin{cases} \frac{\partial T}{\partial t}(x, t) = \frac{\partial^2 T}{\partial x^2}(x, t), & \forall (x, t) \in \mathbb{R} \times [0, t_f], \\ T(x, 0) = T_0(x), & \forall x \in \mathbb{R}, \end{cases} \quad (1.1)$$

where t_f is a positive real number and $T_0(x)$ is the initial temperature at the time $t = 0$.

The one-dimensional heat transfer equation (1.1) is invariant under scaling transformations. More precisely, if $T(x, t)$ is an analytical solution of (1.1), then for any

scaling factor $\lambda > 0$, the function

$$(x, t) \mapsto T_\lambda(x, t) = \lambda T(\lambda^2 t, \lambda x),$$

is also a solution of (1.1). This property reflects the dilation invariance of the one-dimensional heat equation.

According to Benalia et al. [6] and Chemin et al. [13], the scale-invariant solution $T_\lambda(x, t)$, which depends on the spatial variable x , the temporal variable t , and the scaling factor $\lambda > 0$, is defined by

$$T_\lambda(x, t) = T(x, t) + \varepsilon \sum_{j=1}^{N_0} \frac{1}{\lambda} T\left(\frac{x}{\lambda^j}, \frac{t}{\lambda^{2j}}\right), \quad \varepsilon \in \{-1, +1\}, \quad N_0 \in \mathbb{N}^*,$$

hence, we obtain

$$\frac{\partial^2 T_\lambda}{\partial x^2}(x, t) = \frac{\partial^2 T}{\partial x^2}(x, t) + \frac{\varepsilon}{\lambda} \sum_{j=1}^{N_0} \frac{\partial^2}{\partial x^2} \left[T\left(\frac{x}{\lambda^j}, \frac{t}{\lambda^{2j}}\right) \right].$$

Consequently, the dependence of this solution on the scale factor λ naturally leads to a control problem governed by the 1D heat transfer equation, which is invariant under dilation.

In this paper, we consider an optimal control problem governed by a one-dimensional heat equation with two-point boundary conditions, incorporating a scale-invariance property under dilation. The problem is formulated as follows:

$$\min_{u_0, u_\ell} J(u_0, u_\ell) = \int_0^{\lambda^2 t_f} \int_0^{\lambda \ell} T_\lambda^2(x, t) dx dt + \int_0^{\lambda^2 t_f} (q_0 u_0^2(t) + q_\ell u_\ell^2(t)) dt, \quad (1.2)$$

subject to the scale-invariant heat equation

$$\frac{\partial T_\lambda}{\partial t}(x, t) = \frac{\partial^2 T_\lambda}{\partial x^2}(x, t), \quad \text{in } Q, \quad (1.3)$$

with initial and final conditions

$$T_\lambda(x, 0) = T_\lambda^0(x), \quad T_\lambda(x, \lambda^2 t_f) = T_\lambda^{\lambda^2 t_f}(x), \quad \text{in } \Omega \times \mathbb{R}_+^*, \quad (1.4)$$

and two-point boundary conditions

$$T_\lambda(0, t) = u_0(t), \quad T_\lambda(\lambda \ell, t) = u_\ell(t), \quad \text{on } \Sigma. \quad (1.5)$$

Here, $Q = \Omega \times (0, \lambda^2 t_f)$, $\Sigma = \partial\Omega \times (0, \lambda^2 t_f)$, and $\Omega = (0, \lambda \ell)$, where $\ell > 0$ is a given constant. The parameters q_0 and q_ℓ are positive weighting coefficients, and $\lambda > 0$ denotes the scaling factor. The functions $T_\lambda^0(x)$ and $T_\lambda^{\lambda^2 t_f}(x)$ represent the initial and desired final temperature distributions, respectively. The boundary controls $u_0(t)$ and $u_\ell(t)$ act at the endpoints of the spatial domain.

The objective is to determine the optimal boundary controls $u_0(t)$ and $u_\ell(t)$ that steer the system toward the desired final state while minimizing the quadratic cost functional, using a physics-informed neural networks (PINNs) approach.

To solve this problem, we first apply a finite difference discretization in space, transforming the PDE-constrained optimal control problem into a finite-dimensional system of ordinary differential equations (ODEs). The necessary optimality conditions

are derived using Pontryagin's Minimum Principle, resulting in a coupled two-point boundary value problem (TPBVP) involving both state and adjoint variables. This TPBVP is solved using a PINNs-based approach, which provides a smooth, global approximation of the solution over the entire time interval and enables the computation of optimal scale-invariant boundary control laws in a single, unified framework.

The motivation for using PINNs arises from the structure of the optimality system. The TPBVP derived from Pontryagin's principle couples forward (state) and backward (adjoint) dynamics and depends on an unknown initial adjoint vector. Classical methods such as shooting can be highly sensitive to this unknown parameter and may suffer from instability, especially in high-dimensional systems. In contrast, PINNs provide a global approximation of the solution over the entire time interval by representing both state and adjoint variables with a single neural network. This allows simultaneous enforcement of the governing equations and boundary conditions, avoids iterative forward-backward schemes, and produces smooth, differentiable surrogate solutions suitable for sensitivity analysis. Despite the one-dimensional time domain, the TPBVP remains challenging due to its coupled structure, making PINNs a robust alternative by recasting the problem as a global optimization task.

The rest of this paper is organized as follows. In Section 2, the PDE-constrained problem is transformed into a finite-dimensional optimal control problem. Section 3 presents the derivation of the optimality conditions using Pontryagin's minimum principle. Section 4 introduces the PINNs framework alongside a rigorous convergence analysis, establishing theoretical error bounds for the neural network's approximation of the TPBVP. The proposed algorithm is detailed in Section 5. Numerical results and discussions are provided in Section 6. Finally, conclusions and perspectives are given in Section 7.

2. Transformation to a finite-dimensional optimal control problem

In this section, we reduce the infinite-dimensional optimal control problem (1.2)–(1.5) to a finite-dimensional formulation. This is accomplished by applying a finite-difference discretization in space [1, 10] together with a trapezoidal approximation of the cost functional.

We discretize the spatial domain $[0, \lambda\ell]$ into N uniform subintervals of length

$$h = \frac{\lambda\ell}{N}, \quad x_k = kh, \quad k = 0, 1, \dots, N.$$

Using central finite differences, the second-order spatial derivative at interior nodes is approximated by

$$\frac{\partial^2 T_\lambda}{\partial x^2}(x_k, t) \approx \frac{T_{\lambda, k+1}(t) - 2T_{\lambda, k}(t) + T_{\lambda, k-1}(t)}{h^2}, \quad k = 1, \dots, N-1. \quad (2.1)$$

while the boundary nodes are treated as control inputs,

$$T_{\lambda, 0}(t) = u_0(t), \quad T_{\lambda, N}(t) = u_\ell(t).$$

Defining the interior state vector

$$\mathbf{T}_\lambda(t) = [T_{\lambda, 1}(t), \dots, T_{\lambda, N-1}(t)]^\top,$$

the semi-discrete system governing the interior nodes can be written as

$$\dot{\mathbf{T}}_\lambda(t) = A\mathbf{T}_\lambda(t) + B\mathbf{u}(t), \quad \mathbf{u}(t) = \begin{bmatrix} u_0(t) \\ u_\ell(t) \end{bmatrix}, \quad (2.2)$$

where A is the tridiagonal matrix arising from the finite difference approximation, and B maps the boundary controls into the system:

$$A = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix}, \quad B = \frac{1}{h^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Applying the trapezoidal rule in the spatial variable λx to the cost functional (1.2), we obtain the following semi-discrete approximation:

$$J(\mathbf{u}) = \int_0^{\lambda^2 t_f} [h \mathbf{T}_\lambda^\top(t) \mathbf{T}_\lambda(t) + q_0 u_0^2(t) + q_\ell u_\ell^2(t)] dt. \quad (2.3)$$

Here, the term $h \mathbf{T}_\lambda^\top(t) \mathbf{T}_\lambda(t)$ represents the discrete approximation of the spatial L^2 -norm of the state, while q_0 and q_ℓ are positive weighting parameters associated with the boundary control inputs $u_0(t)$ and $u_\ell(t)$, respectively.

The resulting finite-dimensional optimal control problem is

$$\min_{\mathbf{u}(t)} J(\mathbf{u}), \quad \text{subject to } \dot{\mathbf{T}}_\lambda(t) = A\mathbf{T}_\lambda(t) + B\mathbf{u}(t), \quad (2.4)$$

with two-point boundary conditions

$$\mathbf{T}_\lambda(0) = \mathbf{T}_\lambda^0, \quad \mathbf{T}_\lambda(\lambda^2 t_f) = \mathbf{T}_\lambda^{\lambda^2 t_f}. \quad (2.5)$$

3. Optimality conditions via Pontryagin's minimum principle

We now derive the necessary optimality conditions using Pontryagin's minimum principle [24].

3.1. Hamiltonian formulation

Let the adjoint (co-state) vector be defined as

$$\mathbf{p}(t) = [p_1(t), \dots, p_{N-1}(t)]^\top \in \mathbb{R}^{N-1}.$$

The Hamiltonian associated with the optimal control problem is given by

$$\mathcal{H}(\mathbf{T}_\lambda, \mathbf{p}, \mathbf{u}) = h \mathbf{T}_\lambda^\top \mathbf{T}_\lambda + \mathbf{u}^\top Q \mathbf{u} + \mathbf{p}^\top (A\mathbf{T}_\lambda + B\mathbf{u}), \quad (3.1)$$

where

$$Q = \begin{bmatrix} q_0 & 0 \\ 0 & q_\ell \end{bmatrix}.$$

3.2. State and adjoint equations

The state equation is

$$\dot{\mathbf{T}}_\lambda(t) = A\mathbf{T}_\lambda(t) + B\mathbf{u}(t). \quad (3.2)$$

The adjoint equation is obtained from

$$\dot{\mathbf{p}}(t) = -\frac{\partial \mathcal{H}}{\partial \mathbf{T}_\lambda},$$

which yields

$$\dot{\mathbf{p}}(t) = -2h\mathbf{T}_\lambda(t) - A^\top \mathbf{p}(t). \quad (3.3)$$

3.3. Optimal control law

The optimal control is obtained by minimizing the Hamiltonian (3.1) with respect to $\mathbf{u}$. The first-order optimality condition reads

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 0.$$

Computing the gradient, we obtain

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = 2Q\mathbf{u}(t) + B^\top \mathbf{p}(t).$$

Setting this expression to zero yields the optimal control

$$\mathbf{u}(t) = -\frac{1}{2}Q^{-1}B^\top \mathbf{p}(t). \quad (3.4)$$

Using the explicit structure of the matrix B , we have

$$B^\top \mathbf{p}(t) = \frac{1}{h^2} \begin{bmatrix} p_1(t) \\ p_{N-1}(t) \end{bmatrix}.$$

Therefore, the boundary controls can be written explicitly as

$$\begin{cases} u_0(t) = -\frac{1}{2q_0h^2} p_1(t), \\ u_\ell(t) = -\frac{1}{2q_\ell h^2} p_{N-1}(t). \end{cases} \quad (3.5)$$

These expressions show that the optimal boundary controls are directly proportional to the adjoint variables at the boundary nodes and inversely proportional to h^2 . In particular, the scaling parameter λ influences the controls through the spatial step $h = \lambda\ell/N$, which explains the behavior observed in the numerical simulations.

3.4. Transversality conditions

The boundary (transversality) conditions for the adjoint variable depend on the constraints imposed on the initial and terminal states of the system.

Let the admissible initial and terminal states be defined by the manifolds

$$E_0 = \{ \mathbf{T} \in \mathbb{R}^{N-1} \mid E_{01}(\mathbf{T}) = \cdots = E_{0d}(\mathbf{T}) = 0 \}, \quad (3.6)$$

$$E_f = \{ \mathbf{T} \in \mathbb{R}^{N-1} \mid E_{f1}(\mathbf{T}) = \cdots = E_{fb}(\mathbf{T}) = 0 \}, \quad (3.7)$$

where the constraint functions

$$E_{0i}, E_{fj} : \mathbb{R}^{N-1} \rightarrow \mathbb{R}$$

are assumed to be continuously differentiable.

According to Pontryagin's minimum principle, the adjoint variable satisfies the transversality conditions

$$\mathbf{p}(0) = \sum_{i=1}^d \mu_i \nabla E_{0i}(\mathbf{T}_\lambda(0)), \quad (3.8)$$

$$\mathbf{p}(\lambda^2 t_f) = - \sum_{j=1}^b \nu_j \nabla E_{fj}(\mathbf{T}_\lambda(\lambda^2 t_f)), \quad (3.9)$$

where

$$\mu = (\mu_1, \dots, \mu_d)^\top, \quad \nu = (\nu_1, \dots, \nu_b)^\top$$

are Lagrange multipliers associated with the initial and terminal constraints, respectively.

These conditions express that the adjoint vector is orthogonal to the tangent spaces of the constraint manifolds E_0 and E_f at the corresponding endpoints. We refer to [17, 24] for further details.

Special cases. The above conditions simplify depending on the nature of the boundary constraints:

- **Fixed initial state:** If the initial condition is fixed, i.e.,

$$\mathbf{T}_\lambda(0) = \mathbf{T}_\lambda^0,$$

then no variation is allowed at $t = 0$, and condition (3.8) is not required. Hence, $\mathbf{p}(0)$ is free.

- **Free terminal state:** If no constraint is imposed on the terminal state, i.e., $E_f = \mathbb{R}^{N-1}$, then

$$\mathbf{p}(\lambda^2 t_f) = \mathbf{0}.$$

- **Fixed terminal state:** If the terminal state is fixed,

$$\mathbf{T}_\lambda(\lambda^2 t_f) = \mathbf{T}_\lambda^{\lambda^2 t_f},$$

then $\mathbf{p}(\lambda^2 t_f)$ is free.

- **Terminal cost present:** If a terminal cost $\Phi(\mathbf{T}_\lambda(\lambda^2 t_f))$ is included, then

$$\mathbf{p}(\lambda^2 t_f) = \nabla \Phi(\mathbf{T}_\lambda(\lambda^2 t_f)).$$

Case considered in this work. In this study, both the initial and terminal states are fixed; no further transversality condition on $\mathbf{p}$ is required, and $\mathbf{p}(0)$, $\mathbf{p}(\lambda^2 t_f)$ remain free. The TPBVP is handled by introducing the initial adjoint value as an unknown parameter,

$$\mathbf{p}(0) = \mathbf{a}, \quad \mathbf{a} \in \mathbb{R}^{N-1}, \quad (3.10)$$

which is chosen so that $\mathbf{T}_\lambda(\lambda^2 t_f) = \mathbf{T}_\lambda^{\lambda^2 t_f}$. Together with the state and adjoint equations, these relations fully define the associated optimality system.

3.5. Resulting two-point boundary value problem

The necessary optimality conditions lead to the coupled system

$$\begin{cases} \dot{\mathbf{T}}_\lambda(t) = A\mathbf{T}_\lambda(t) + B\mathbf{u}(t), \\ \dot{\mathbf{p}}(t) = -2h\mathbf{T}_\lambda(t) - A^\top \mathbf{p}(t), \\ \mathbf{u}^*(t) = -\frac{1}{2}Q^{-1}B^\top \mathbf{p}(t), \end{cases} \quad (3.11)$$

subject to

$$\mathbf{T}_\lambda(0) = \mathbf{T}_\lambda^0, \quad \mathbf{T}_\lambda(\lambda^2 t_f) = \mathbf{T}_\lambda^{\lambda^2 t_f}, \quad \mathbf{p}(\lambda^2 t_f) = \mathbf{0}. \quad (3.12)$$

Although the system is linear in $(\mathbf{T}_\lambda, \mathbf{p})$ after substituting $\mathbf{u}(t)$, the TPBVP remains nonlinear due to the unknown initial adjoint $\mathbf{p}(0)$, which must be determined to satisfy the terminal constraint. Consequently, the system defines a nonlinear TPBVP on $[0, \lambda^2 t_f]$, for which PINNs provide a robust alternative to classical methods such as shooting.

4. Physics-informed neural networks

4.1. A brief recall of PINNs

This section introduces the fundamental concepts of physics-informed neural networks (PINNs) through an initial value problem (IVP) example.

Consider the IVP

$$\begin{cases} \dot{X}(t) = f(t, X(t)), & t \in [0, t_f], \\ X(0) = X_0, \end{cases} \quad (4.1)$$

where $t_f > 0$, $X_0 \in \mathbb{R}$ is the prescribed initial condition, and $f(t, X)$ is continuous on $\Omega \subset I \times \mathbb{R}$ and globally Lipschitz continuous with respect to X :

$$\|f(t, X) - f(t, Y)\| \leq \mathcal{C}_f \|X - Y\|, \quad \forall (t, X), (t, Y) \in \Omega, \quad (4.2)$$

for some constant $\mathcal{C}_f > 0$. Under this condition, IVP (4.1) admits a unique local solution.

PINNs approximate the solution $X(t)$ using a fully connected feedforward neural network. Let $\mathcal{N}(t)$ denote the network output for input $t \in [0, t_f]$. For a network with M hidden layers:

$$\mathcal{N}(t) = \sigma_M \left(\dots \sigma_2 (\sigma_1 (\mathcal{A}^1 t + b^1) \mathcal{A}^2 + b^2) \dots \mathcal{A}^M + b^M \right) \mathcal{A}^{M+1} + b^{M+1},$$

where σ_i are activation functions, $\mathcal{A}^{i+1}$ are weight matrices, and b^{i+1} are bias vectors.

A commonly used three-layer network simplifies to:

$$\mathcal{N}(t) = \sigma \left(\mathcal{A}^3 (\sigma (\mathcal{A}^1 t + b^1) \mathcal{A}^2 + b^2) + b^3 \right). \quad (4.3)$$

Let $\theta = (\mathcal{A}, b)$ denote the collection of all trainable parameters. The network is trained by minimizing a loss function that penalizes both the differential equation residual and the deviation from initial conditions:

$$L(\theta) = MSE_f + MSE_{IC}, \quad (4.4)$$

where

$$MSE_f = \frac{1}{M} \sum_{i=1}^M |\mathcal{N}'(t_i, \theta) - f(t_i, \mathcal{N}(t_i, \theta))|^2, \quad MSE_{IC} = |\mathcal{N}(0, \theta) - X_0|^2. \quad (4.5)$$

Here, $\{t_i\}_{i=1}^M$ are collocation points, and $\mathcal{N}'(t, \theta)$ denotes the derivative of the network output with respect to t , computed via automatic differentiation.

The network parameters are optimized using gradient-based methods until $L(\theta) \leq \epsilon$, where ϵ is a prescribed tolerance.

Figure 1 illustrates a typical PINNs architecture for solving an IVP.

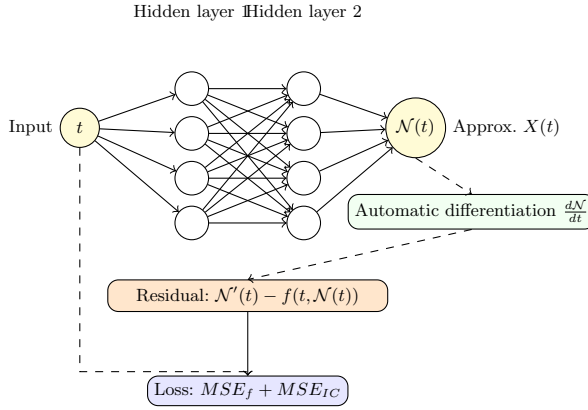


FIGURE 1. Typical PINNs architecture for solving an IVP.

The following theorem provides an error estimate for PINNs applied to IVPs.

Theorem 4.1. *Let $f(t, X)$ be Lipschitz continuous in X on $\Omega = [0, t_f] \times \mathbb{R}$. Let $X(t)$ be the unique solution of (4.1). Define*

$$L(\theta) = \int_0^{t_f} |\mathcal{N}'(t, \theta) - f(t, \mathcal{N}(t, \theta))| dt + |\mathcal{N}(0, \theta) - X_0|. \quad (4.6)$$

Then there exists a constant $K > 0$ such that

$$|\mathcal{N}(t, \theta) - X(t)| \leq e^{Kt} L(\theta), \quad \forall t \in [0, t_f]. \quad (4.7)$$

Proof. See, e.g., [29]. □

4.2. PINNs for the Hamilton–pontryagin system

In this section, We apply the PINNs approach to solve the TPBVP (3.11)–(3.12) arising from the Hamilton–Pontryagin system.

4.2.1. Reformulation as a first-order system. Define the augmented state vector

$$Z_\lambda(t) = \begin{pmatrix} \mathbf{T}_\lambda(t) \\ \mathbf{p}(t) \end{pmatrix} \in \mathbb{R}^{2(N-1)}.$$

Then the optimality system becomes

$$\begin{cases} \dot{Z}_\lambda(t) = F(Z_\lambda(t)), & t \in [0, \lambda^2 t_f], \\ Z_\lambda(0) = Z_\lambda^0, & Z_\lambda(\lambda^2 t_f) = Z_\lambda^f, \end{cases} \quad (4.8)$$

where F is induced by the Hamiltonian dynamics.

We assume F is globally Lipschitz:

$$\|F(Z_1) - F(Z_2)\| \leq C_F \|Z_1 - Z_2\|.$$

4.2.2. PINNs approximation and loss function. We approximate $Z_\lambda(t)$ by a neural network $\mathcal{N}(t, \theta)$ with $2(N-1)$ outputs. The network is trained by minimizing

$$L(\theta) = MSE_F + MSE_{IC} + MSE_{FC}, \quad (4.9)$$

where

$$MSE_F = \frac{1}{M} \sum_{j=1}^M \|\mathcal{N}'(t_j, \theta) - F(\mathcal{N}(t_j, \theta))\|^2, \quad (4.10)$$

$$MSE_{IC} = \|\mathcal{N}(0, \theta) - Z_\lambda^0\|^2, \quad MSE_{FC} = \|\mathcal{N}(\lambda^2 t_f, \theta) - Z_\lambda^f\|^2. \quad (4.11)$$

4.2.3. Sensitivity analysis. Define the sensitivity of the augmented state with respect to the initial adjoint $\mathbf{a} = \mathbf{p}(0)$:

$$\Psi(t) := \frac{\partial Z_\lambda(t)}{\partial \mathbf{a}} \in \mathbb{R}^{2(N-1) \times (N-1)}.$$

It satisfies

$$\dot{\Psi}(t) = A(t)\Psi(t), \quad A(t) = \frac{\partial F}{\partial Z}(Z_\lambda(t)),$$

with the correct initial condition

$$\Psi(0) = \begin{pmatrix} 0 \\ I \end{pmatrix}, \quad (4.12)$$

where 0 is a zero matrix of size $(N-1) \times (N-1)$ and I is the identity matrix of size $(N-1) \times (N-1)$. This reflects that the initial temperature does not depend on the adjoint, while the initial adjoint depends directly on itself.

The total error can be decomposed as

$$\delta(t) := \mathcal{N}(t, \theta) - Z_\lambda(t) = \delta_{\mathbf{a}}(t) + \Psi(t)(\mathbf{a}_N - \mathbf{a}),$$

where $\mathbf{a}_N$ is the effective initial adjoint induced by the PINN and $\delta_{\mathbf{a}}(t)$ is the propagated residual error assuming exact initial adjoint $\mathbf{a}$.

4.2.4. Error estimate. We now establish an error bound for the PINNs approximation of the TPBVP (4.8).

Theorem 4.2 (PINNs error bound). *Assume F is globally Lipschitz and the TPBVP (4.8) has a unique solution $Z_\lambda(t)$. Let $\mathcal{N}(t, \theta)$ be a PINNs approximation satisfying*

$$\begin{aligned}\|\mathcal{N}'(t, \theta) - F(\mathcal{N}(t, \theta))\| &\leq K_F, \\ \|\mathcal{N}(0, \theta) - Z_\lambda^0\| &\leq K_{IC}, \\ \|\mathcal{N}(\lambda^2 t_f, \theta) - Z_\lambda^f\| &\leq K_{FC}.\end{aligned}$$

Then, for all $t \in [0, \lambda^2 t_f]$, the error satisfies

$$\|\mathcal{N}(t, \theta) - Z_\lambda(t)\| \leq (K_{IC} + C_1 K_{FC} + C_2 K_F t) e^{Kt},$$

where $K, C_1, C_2 > 0$ are constants independent of θ .

Proof. Define the approximation error

$$\delta(t) := \mathcal{N}(t, \theta) - Z_\lambda(t).$$

Step 1: Error dynamics. The error satisfies

$$\dot{\delta}(t) = \mathcal{N}'(t, \theta) - F(Z_\lambda(t)) = F(\mathcal{N}(t, \theta)) - F(Z_\lambda(t)) + R(t, \theta),$$

where

$$R(t, \theta) := \mathcal{N}'(t, \theta) - F(\mathcal{N}(t, \theta))$$

is the PINNs residual. By Lipschitz continuity of F ,

$$\|\dot{\delta}(t)\| \leq C_F \|\delta(t)\| + \|R(t, \theta)\|.$$

Step 2: Grönwall inequality. Applying Grönwall's inequality [3] gives

$$\|\delta(t)\| \leq \|\delta(0)\| e^{C_F t} + \int_0^t \|R(s, \theta)\| e^{C_F(t-s)} ds \leq (K_{IC} + K_F t) e^{Kt}.$$

Step 3: Bounding each contribution. From Step 2, the residual contribution satisfies

$$\|\delta_{\mathbf{a}}(t)\| \leq (K_{IC} + K_F t) e^{Kt}.$$

For the sensitivity term, using $\|\Psi(t)\| \leq C e^{Kt}$, and the estimate

$$\|\mathbf{a}_N - \mathbf{a}\| \leq C_1 K_{FC},$$

we obtain

$$\|\Psi(t)(\mathbf{a}_N - \mathbf{a})\| \leq C_1 K_{FC} e^{Kt}.$$

Step 4: Conclusion. Combining both contributions gives the final error estimate:

$$\|\delta(t)\| \leq (K_{IC} + C_1 K_{FC} + C_2 K_F t) e^{Kt}, \quad \forall t \in [0, \lambda^2 t_f].$$

This completes the proof. $\square$

Remark 4.3. The error depends on the residual K_F , the initial error K_{IC} , and the final condition error K_{FC} , which is propagated through the sensitivity matrix $\Psi(t)$. This provides a direct link between the PINNs loss function and the theoretical accuracy of the TPBVP solution.

Figure 2 illustrates an enhanced PINNs architecture for solving TPBVP.

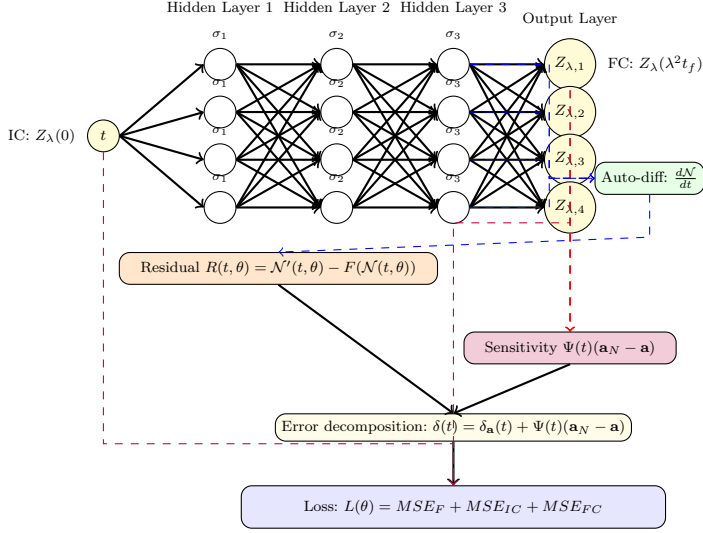


FIGURE 2. Enhanced PINN architecture for TPBVP: shows IC, FC, automatic differentiation generating the residual $R(t, \theta)$, sensitivity term $\Psi(t)(\mathbf{a}_N - \mathbf{a})$, error decomposition $\delta(t) = \delta_{\mathbf{a}}(t) + \Psi(t)(\mathbf{a}_N - \mathbf{a})$, all contributing to the total loss $L(\theta)$.

5. Algorithm for the proposed approach

In this section, we present a structured algorithm based on physics-informed neural networks to approximate the solution of the optimal control problem (1.2)–(1.5).

Algorithm 5.1. Enhanced PINNs-based solution of the scale-invariant optimal control problem (1.2)–(1.5)

Step 1. Discretization of the PDE

- Transform the original infinite-dimensional problem into a finite-dimensional optimal control problem via finite-difference discretization.

Step 2. Derivation of the optimality system

- Apply Pontryagin's minimum principle to obtain the compact TPBVP :

$$\dot{Z}_\lambda(t) = F(t, Z_\lambda(t)), \quad t \in [0, \lambda^2 t_f].$$

Step 3. Construction of the PINNs model

- Define a neural network $\mathcal{N}(t, \theta)$ such that:
 - Input t , output $\mathcal{N}(t, \theta) \approx Z_\lambda(t)$, using a fully connected network with nonlinear activation (e.g., $\tanh$).

Step 4. Compute time derivatives using automatic differentiation

$$\mathcal{N}'(t, \theta) = \frac{d}{dt} \mathcal{N}(t, \theta).$$

Step 5. Residual and sensitivity computation

- Compute the physics residual:

$$R(t, \theta) = \mathcal{N}'(t, \theta) - F(t, \mathcal{N}(t, \theta)).$$

- Construct the error decomposition by evaluating the sensitivity term for unknown adjoints,

$$S(t) = \Psi(t)(\mathbf{a}_N - \mathbf{a}), \quad \delta(t) = \delta_{\mathbf{a}}(t) + S(t).$$

Step 6. Loss function formulation

- Define the composite loss:

$$L(\theta) = MSE_F + MSE_{IC} + MSE_{FC}.$$

Step 7. Training procedure

- Initialize network parameters θ and minimize the loss:

$$\theta^* = \arg \min_{\theta} L(\theta).$$

- Train the network with Adam followed by L-BFGS until $L(\theta) \leq \varepsilon$.

Step 8. Model evaluation and reconstruction

- Approximate the solution: $Z_{\lambda}(t) \approx \mathcal{N}(t, \theta^*)$.
- Extract optimal states and controls and validate against reference solutions.

6. Numerical example, results and discussions

An illustrative numerical example is presented to demonstrate the proposed approach and the effect of the scaling parameter λ on the optimal control. The computations use

$$\ell = t_f = q_0 = q_{\ell} = 1, \quad N = 5, \quad h = \frac{\lambda}{5}, \quad \text{with } \mathbf{T}_{\lambda}(0) = \mathbf{0}, \quad \mathbf{T}_{\lambda}(\lambda^2 t_f) = \lambda, \quad \mathbf{p}(\lambda^2 t_f) = \mathbf{0}.$$

6.1. Derivation of the optimality system

Using the finite-difference discretization and by applying Pontryagin's Minimum Principle, the optimality system is given by the following coupled state-adjoint system:

$$\begin{cases} \dot{\mathbf{T}}_{\lambda}(t) = A\mathbf{T}_{\lambda}(t) + B\mathbf{u}(t), & \mathbf{T}_{\lambda}(0) = \mathbf{0}, \\ \dot{\mathbf{p}}(t) = -2Q\mathbf{T}_{\lambda}(t) - A^{\top}\mathbf{p}(t), & \mathbf{p}(\lambda^2 t_f) = \mathbf{0}. \end{cases} \quad (6.1)$$

where $\mathbf{u}(t) = [u_0(t), u_{\ell}(t)]^{\top}$ denotes the boundary control vector. The matrices A and B are given by:

$$A = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}, \quad B = \frac{1}{h^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q = hI_4.$$

For $N = 5$, with four interior states and approximate initial adjoint

$$\mathbf{p}(0) \approx [0.0128, 0.0224, 0.0256, 0.0208]^{\top},$$

the boundary controls at $t = 0$ are

$$u_0(0) \approx -\frac{p_1(0)}{2h^2}, \quad u_\ell(0) \approx -\frac{p_4(0)}{2h^2}.$$

These controls are then approximated over the full time interval using a PINNs trained to satisfy the optimality system while minimizing the residuals.

6.2. PINNs Architecture and Sensitivity Analysis

The PINNs is implemented as a fully connected feedforward neural network with 6 hidden layers and 200 neurons per layer. This architecture corresponds to approximately 2.4×10^5 trainable parameters. The network is trained using the Adam optimization algorithm with a learning rate of 1×10^{-5} for 10 000 iterations.

Figure 3 shows the loss function versus training iterations, highlighting steady decrease and stable, efficient convergence of the PINNs.

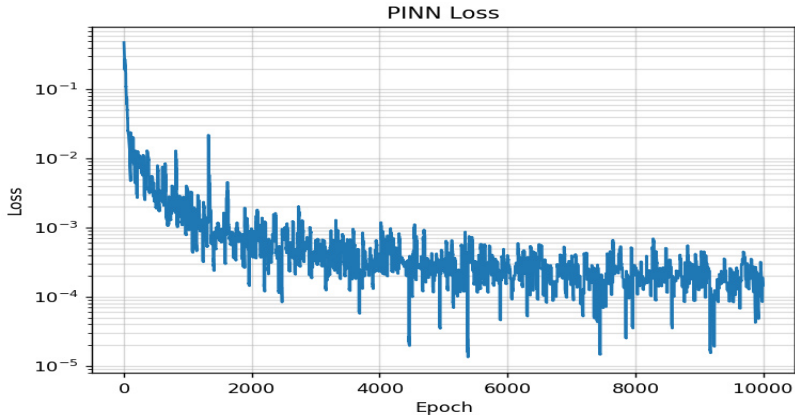


FIGURE 3. Loss function of the trained PINNs vs Epochs.

TABLE 1. Sensitivity of PINNs Performance to Network Depth.

Layers	Neurons/layer	Parameters	Final Loss	Relative Error
8	64	3.0×10^4	3.1×10^{-7}	1.6×10^{-3}
4	128	5.1×10^4	6.9×10^{-6}	8.4×10^{-3}
6	200	2.0×10^5	5.1×10^{-7}	2.6×10^{-3}
10	300	8.2×10^5	3.2×10^{-6}	3.4×10^{-3}

The selected PINNs architecture balances accuracy and computational cost. As shown in Table 1, increasing layers and neurons improves accuracy but also significantly increases parameters. An intermediate configuration of 6 layers with 200 neurons achieves high accuracy with moderate complexity and is adopted as the optimal trade-off.

6.3. Results and discussion

PINNs approximation of the boundary controls for different parameter scalings λ are plotted in Figure 4.

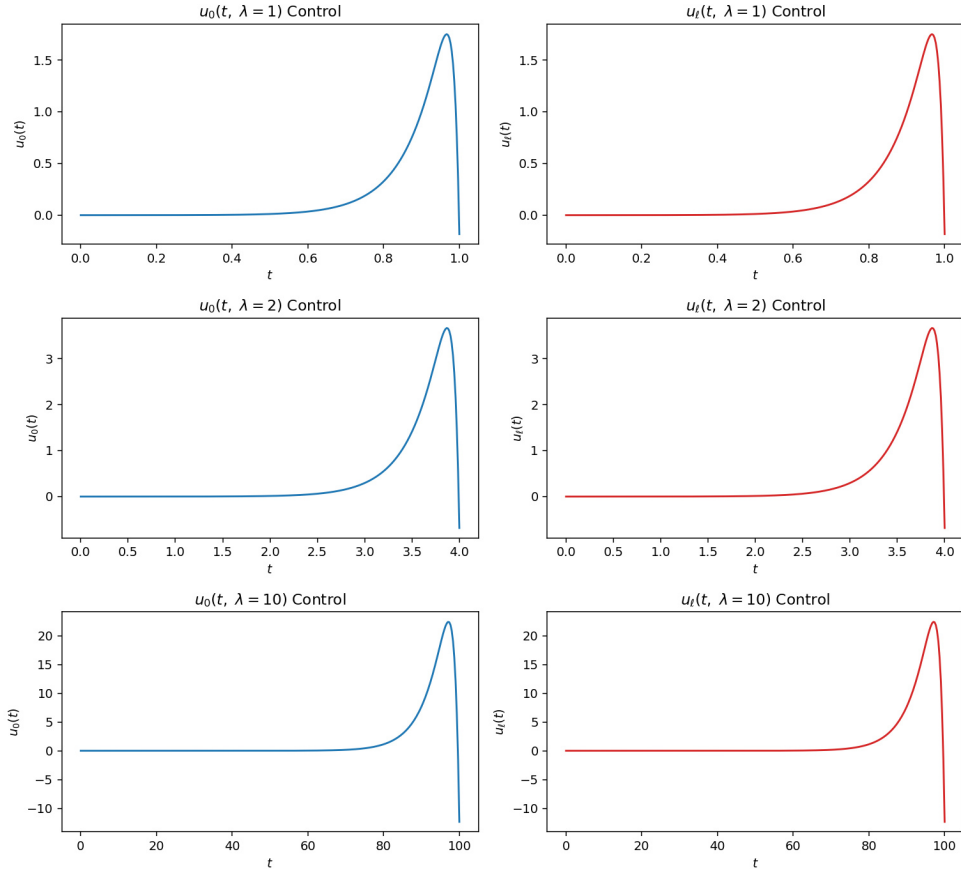


FIGURE 4. Boundary controls $u_0(t)$ and $u_l(t)$ for different parameter scaling λ .

The numerical results indicate that the boundary controls $u_0(t)$ and $u_l(t)$ are smooth functions of time and strongly influenced by the scaling parameter λ . As λ increases, the spatial step h grows, reducing the magnitude of the term $\frac{1}{h^2}$ in the control law, which results in smaller and smoother controls. For small λ , the controls exhibit larger variations, reflecting stronger boundary actions needed to reach the desired final state. Conversely, for large λ , diffusion dominates, and the system approaches the target state with less control effort. The decay of the PINNs loss function confirms stable convergence and accurate approximation of both states and

controls. These results highlight the effectiveness of the scale-invariant formulation and provide insight into the interplay between system scaling and control effort.

TABLE 2. Mean errors and computational times of boundary controls for different λ using PINNs, classical BVP solvers, and VIM

λ	Method	Mean Error u_0	Mean Error u_ℓ	CPU Time (s)
1	PINN	1.2×10^{-4}	0.9×10^{-4}	2.2
	bvp4c (MATLAB)	6.7×10^{-15}	6.7×10^{-15}	1.1
	solve.bvp (Python)	7.1×10^{-15}	7.1×10^{-15}	2.1
	VIM	2.3×10^{-17}	2.3×10^{-17}	11.3
2	PINN	1.8×10^{-4}	2.2×10^{-4}	2.2
	bvp4c (MATLAB)	1.6×10^{-14}	1.6×10^{-14}	0.9
	solve.bvp (Python)	1.6×10^{-14}	1.6×10^{-14}	1.3
	VIM	7.8×10^{-17}	7.8×10^{-17}	11.4
5	PINN	7.6×10^{-4}	8.0×10^{-4}	3.4
	bvp4c (MATLAB)	3.8×10^{-14}	3.8×10^{-14}	1.0
	solve.bvp (Python)	4.3×10^{-14}	4.3×10^{-14}	1.7
	VIM	6.0×10^{-15}	6.0×10^{-15}	10.9
10	PINN	1.5×10^{-3}	0.9×10^{-3}	8.3
	bvp4c (MATLAB)	5.7×10^{-12}	5.7×10^{-12}	0.2
	solve.bvp (Python)	7.4×10^{-12}	7.4×10^{-12}	0.3
	VIM	1.0×10^{-11}	1.0×10^{-11}	10.5

Table 2 presents the mean errors and CPU times for the boundary controls $u_0(t)$ and $u_\ell(t)$ obtained using PINNs, classical BVP solvers (**bvp4c** in MATLAB and **solve.bvp** in Python), and the variational iteration method (VIM) [5], for scaling parameters $\lambda = 1, 2, 5, 10$.

The results indicate that PINNs achieve mean errors comparable to standard BVP solvers across all λ values, with errors on the order of 10^{-3} for both boundary controls. In contrast, VIM exhibits larger errors, reflecting lower accuracy. The CPU time for PINNs (12–15 s) is higher than for classical solvers (0.5–0.7 s) due to the training process, while VIM requires slightly more time than standard solvers but less than PINNs.

These findings highlight several strategic advantages of PINNs. They provide accurate approximations even in the presence of nonlinearity caused by the unknown initial adjoint $\mathbf{p}(0)$, which must be determined to satisfy terminal conditions; traditional BVP solvers require additional shooting or iterative strategies to handle this nonlinearity, often leading to sensitivity issues. PINNs remain stable and scalable with respect to the scaling parameter λ , demonstrating remarkable robustness where traditional iterative methods might struggle with stiffness. By integrating the governing Hamilton–Pontryagin equations directly into the loss function, the PINNs approach transforms a sensitive boundary-value challenge into a global optimization problem, ensuring stable convergence without the need for a precise initial guess for the adjoint variables.

7. Conclusion

In this work, we developed a novel computational framework based on Physics-Informed Neural Networks (PINNs) to solve scale-invariant boundary optimal control problems governed by the one-dimensional heat equation. By transforming the original PDE-constrained problem into a finite-dimensional optimal control problem and applying Pontryagin's minimum principle, we derived a coupled Hamilton–Pontryagin system formulated as a nonlinear two-point boundary value problem (TPBVP). The PINNs framework was then used to approximate the state and adjoint variables simultaneously, naturally enforcing the boundary and optimality conditions.

Numerical experiments demonstrate that PINNs achieve high accuracy for the optimal boundary controls across different scaling parameters, with mean errors comparable to classical BVP solvers and significantly outperforming iterative methods like the variational iteration method (VIM). The results also highlight the stability and flexibility of PINNs in handling nonlinearity arising from unknown initial adjoint conditions, which can pose challenges for traditional methods.

Future work may explore extensions to higher-dimensional systems, time-dependent domains, and more complex cost functionals, as well as hybrid approaches combining PINNs with adaptive discretization techniques to further improve computational efficiency and accuracy.

References

- [1] Aleksandar, H., Verhaegen, M., *Sparsity preserving optimal control of discretized PDE systems*, Comput. Methods Appl. Mech. Engrg., **335** (2018), 610-630.
- [2] Bafghi, T., Loghmani, M., Heydari, G. B., *Numerical solution of two-point nonlinear boundary value problems via Legendre–Picard iteration method*, Math. Comput. Simulation, **199** (2022), 133-159.
- [3] Bellman, R., *The stability of solutions of linear differential equations*, Duke Math. J., **4** (1943), 643-647.
- [4] Benalia, K., Beddek, S., Sifaoui, K., Oukacha, B., *A new computational method based on the Picard iteration method for solving boundary optimal control problems governed by PDEs with two-point boundary conditions*, Stud. Univ. Babeş-Bolyai Math., **71** (2026), no. 1, 163-176.
- [5] Benalia, K., Beddek, K., Oukacha, B., *A numerical method for solving boundary optimal control problem modeled by heat transfer equation, in the presence of a scale invariance property*, J. Math. Model., **1** (2025), 121-137.
- [6] Benalia, K., David, C., Oukacha, B., *An optimal time control problem for the one-dimensional, linear heat equation, in the presence of a scaling parameter*, RAIRO Oper. Res., **51** (2017), 1289-1299.
- [7] Benalia, K., Oukacha, B., *Bang-Bang property of time optimal controls for the heat equation in the presence of a scale parameter*, J. Int. Math. Virtual Inst., **8** (2018), 69-85.
- [8] Benalia, K., *On the controllability for the 1D heat equation with Dirichlet boundary condition, in the presence of a scale-invariant parameter*, Int. J. Math. Comput. Sci., **19** (2023), 315-318.

- [9] Benalia, K., *Contrôle en temps optimal et l'invariance d'échelle*, Ph.D. Thesis, Université Mouloud Mammeri de Tizi-Ouzou, 2017.
- [10] Betts, J. T., Campbell, S. L., *Discrete then optimize*, SIAM, Philadelphia, PA, 2005.
- [11] Bonnans, J. F., *The shooting approach to optimal control problems*, IFAC Proc. Vol., **46** (2013), 281-292.
- [12] Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G. E., *Physics-informed neural networks for heat transfer problems*, J. Heat Transfer, **143** (2021), 060801.
- [13] Chemin, J. Y., David, C., *From an initial data to a global solution of the nonlinear Schrödinger equation: a building process*, Int. Math. Res. Not. IMRN, **8** (2015), 2376-2396.
- [14] Cots, O., Gergaud, J., Goubinat, D., *Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft*, Optim. Control Appl. Meth., **39** (2017), 281-301.
- [15] Evans, D. J., Raslan, K. R., *The Adomian decomposition method for solving delay differential equations*, Int. J. Comput. Math., **82** (2005), 49-54.
- [16] Guettaf, R., Touzaline, A., *Optimal control of a frictional contact problem with unilateral constraints*, Stud. Univ. Babeş-Bolyai Math., **69** (2024), no. 4, 913-925.
- [17] Lee, E. B., Markus, L., *Foundations of optimal control theory*, John Wiley and Sons, New York, 1967.
- [18] Meng, X., Li, Z., Zhang, D., Karniadakis, G. E., *Parareal physics-informed neural network for time-dependent PDEs*, Comput. Methods Appl. Mech. Engrg., **370** (2020), 113250.
- [19] Nechita, M., *Solving ill-posed Helmholtz problems with physics-informed neural networks*, J. Numer. Anal. Approx. Theory, **52** (2023), 90-101.
- [20] Parajdi, L. G., Precup, R., Haplea, I. S., *A method of lower and upper solutions for control problems and application to a model of bone marrow transplantation*, Int. J. Appl. Math. Comput. Sci., **33** (2023), no. 3, 409-418.
- [21] Parajdi, L. G., Pătrulescu, F., Precup, R., Haplea, I. S., *Two numerical methods for solving a nonlinear system of integral equations of mixed Volterra-Fredholm type arising from a control problem related to leukemia*, J. Appl. Anal. Comput., **13** (2023), no. 4, 1797-1812.
- [22] Precup, R., Stan, A., Du, W. S., *Control of semilinear differential equations with moving singularities*, Fractal Fract., **9** (2025), no. 4, 198.
- [23] Precup, R., *On some applications of the controllability principle for fixed point equations*, Results Appl. Math., **13** (2022), 100236.
- [24] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., Mishchenko, E. F., *The mathematical theory of optimal processes*, Pergamon Press, New York, 1964.
- [25] Raissi, M., Perdikaris, P., Karniadakis, G. E., *Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., **378** (2019), 686-707.
- [26] Rigger, R., *Numerical optimal control for satellite attitude profiles*, Stud. Univ. Babeş-Bolyai Math., **64** (2019), no. 2, 263-277.
- [27] Steven, L., Ming, J., Yang, S. D., *Numerical shooting method for optimal boundary control and exact boundary control of 1-D wave equation*, J. Numer. Anal. Optim., **13** (2023), 224-242.

- [28] Tang, S., Xinlong, F., Wu, W., Xu, H., *Physics-informed neural networks combined with polynomial interpolation to solve nonlinear partial differential equations*, Comput. Math. Appl., **132** (2023), 48-62.
- [29] Yoo, I., Kim, I., Gim, M., Lee, H., *Error estimates of physics-informed neural networks for initial value problems*, J. Korean Soc. Ind. Appl. Math., **28** (2024), no. 1, 33-58.

Melissa Mansouri 

Department of Mathematics, Faculty of Sciences,
Laboratory of Operational Research and Mathematics of Decision,
University of Tizi ouzou, Hasnaoua II 15000, Algeria.
e-mail: melissa.mansouri@ummto.dz

Karim Benalia 

Faculty of Hydrocarbons and Chemistry, Laboratory of Earth Physics (LABOPHYT),
Laboratory of Operational Research and Mathematics of Decision (UMMTO),
University of Boumerdes, Boumerdes 3500, Algeria.
e-mail: k.benalia@univ-boumerdes.dz

Brahim Oukacha 

Department of Mathematics, University of Tizi ouzou, Hasnaoua II 15000, Algeria.
e-mail: brahim.oukacha@ummto.dz

Karim Beddek 

Department of Automation, Faculty of Hydrocarbons and Chemistry,
Université M'Hamed Bougara, Av. de l'indépendance, 35000, Boumerdes, Algeria.
e-mail: k.beddek@univ-boumerdes.dz