Physics 5300 Final Project: PINNs for Heat Equation

Peter Zhang

1 Introduction

1.1 Physics Model Setup

We study heat conduction on a one-dimensional rod of length L, governed by the diffusion equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L],\ t \in [0, T] \tag{1}$$

where $u(x, t)$ is the temperature and α is the thermal diffusivity.

Dirichlet Boundary Conditions

$$u(0, t) = 0, \quad u(L, t) = 0, \quad 0 \le t \le T \tag{2}$$

Initial Condition

$$u(x, 0) = \exp\left(-100\,(x - \tfrac{L}{2})^2\right) + \sin\left(\tfrac{x}{5}\right), \quad 0 \le x \le L \tag{3}$$

1.2 Data from Finite-Difference Solver

We discretize space into N_x points $x_i = i\,\Delta x$ and time into N_t steps $t_k = k\,\Delta t$. The explicit finite-difference update reads:

$$u_i^{k+1} = u_i^k + \frac{\alpha\,\Delta t}{\Delta x^2}\left(u_{i+1}^k - 2\,u_i^k + u_{i-1}^k\right) \tag{4}$$

where each new temperature u_i^{k+1} is computed from its two neighbors at the same time k. Stability of this scheme requires the CFL condition:

$$\Delta t \le \frac{\Delta x^2}{2\,\alpha} \tag{5}$$

Running this solver over all i, k produces the full reference solution array $\{U_{\mathrm{FD}}[i,k]\} \approx u(x_i, t_k)$, which we use both for visualization and as "ground truth" in training.

1.3 PINN Model Setup

We approximate the map $(x,t) \mapsto u(x,t)$ with a fully-connected neural network $\widehat{u}(x,t;\theta)\colon \mathbb{R}^2 \to \mathbb{R}$. The network takes $[x,t]$ as input and outputs a temperature prediction.

The training loss combines two terms:

1. **Data loss** on boundary/initial points,
2. **PDE residual loss** enforcing the heat equation via automatic differentiation:

$$\begin{aligned}\mathcal{L}(\theta) = &\frac{1}{N_{\mathrm{data}}} \sum_{i=1}^{N_{\mathrm{data}}} \left|\widehat{u}(x_u^i, t_u^i) - U_{\mathrm{FD}}(x_u^i, t_u^i)\right|^2 \\ &+ \frac{1}{N_f} \sum_{j=1}^{N_f} \left|\frac{\partial \widehat{u}}{\partial t}(x_f^j, t_f^j) - \alpha \frac{\partial^2 \widehat{u}}{\partial x^2}(x_f^j, t_f^j)\right|^2\end{aligned} \tag{6}$$

1.4 Training & Testing Sets

1. **Boundary / Initial Conditions**
   - **Initial:** $(x_i, 0, U_{\mathrm{FD}}[i,0])$ for all spatial nodes i
   - **Boundaries:** $(0, t_k, 0)$ and $(L, t_k, 0)$ for all time steps k

   These are assembled into:

$$X_{\mathrm{nu}} \in \mathbb{R}^{N_{\mathrm{nu}} \times 2} \tag{7}$$

$$Y_{\mathrm{nu}} \in \mathbb{R}^{N_{\mathrm{nu}} \times 1} \tag{8}$$

2. **Collocation Points** Sample N_f interior points (x_f^j, t_f^j) (e.g., via Latin Hypercube sampling) to enforce the PDE residual. Form:

$$X_f \in \mathbb{R}^{N_f \times 2} \tag{9}$$

3. **Train / Test Split**
   - **Training set:** All BC/IC points and collocation points

- **Testing set:** A held-out subset of finite-difference (FD) solution points used to compute test MSE:

$$\frac{1}{N_{\text{test}}} \sum |\widehat{u} - U_{\text{FD}}|^2 \tag{10}$$

This framework ensures the network learns both the prescribed data and the underlying physics of the heat equation.

2 Results and Discussion

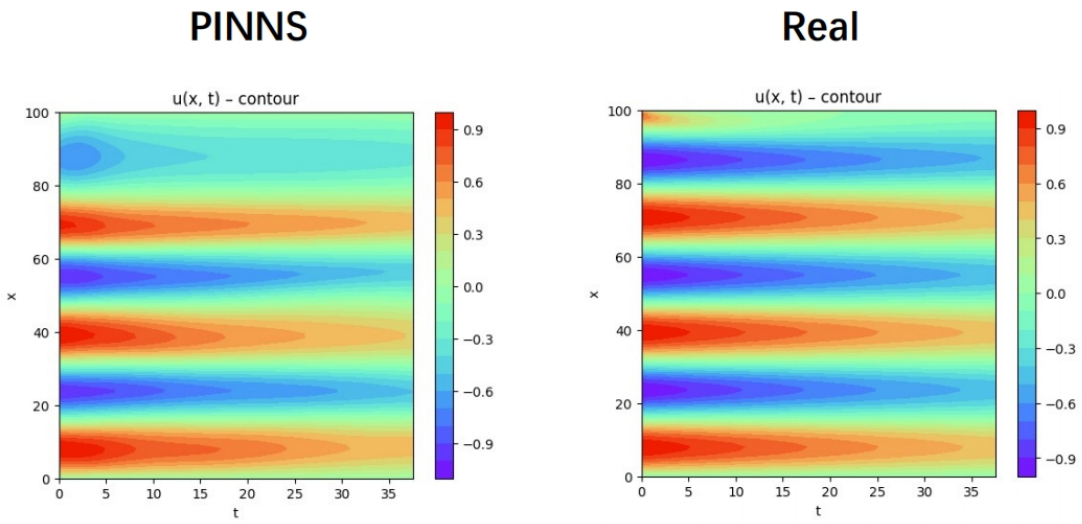

Figure 1: Comparison of PINN-predicted temperature field (left) vs. finite-difference (FD) reference solution (right).

Figure 1 compares the predicted temperature distribution $u(x, t)$ obtained via the Physics-Informed Neural Network (PINN) with the ground-truth solution generated by a finite-difference (FD) method.

Visually, both solutions capture the dominant temporal and spatial oscillations. However, the PINN solution exhibits several noticeable discrepancies, particularly in regions near the upper spatial boundary $x \approx L$. These differences can be attributed to several factors:

Boundary Gradient Effects

The Dirichlet boundary conditions impose zero temperature at both ends of the domain ($u(0, t) = u(L, t) = 0$). When the interior solution contains high-frequency components (e.g.,

from sinusoidal or Gaussian initial conditions), steep gradients can form near the boundaries. PINNs, relying on smooth activation functions (e.g., `tanh`), tend to struggle with resolving such sharp transitions accurately—especially without sufficient sampling near boundaries.

PDE Residual Sampling and Model Complexity

While the collocation points (N_f = 10,000) provide good domain coverage, the uniform or Latin Hypercube sampling strategy may undersample critical regions such as boundary layers. Additionally, the neural network used here has a shallow architecture (two hidden layers, 32 neurons each), which may limit the model's expressive capacity to approximate highly oscillatory solutions.

Training Convergence and Residual Error

Although both training and testing losses decrease significantly over epochs (e.g., training loss from $0.27 \rightarrow 0.0032$, test loss from $0.34 \rightarrow 0.0285$), a consistent offset remains. This suggests that either:

- The loss landscape contains local minima difficult to escape with standard optimizers, or
- The PDE residual loss dominates in regions of low curvature, leaving sharp features underfitted.

Suggestions for Improvement

To improve accuracy, particularly near the boundaries, we recommend:

1. Increasing the density of collocation and boundary training points,
2. Using deeper or residual-connected network architectures,
3. Incorporating adaptive sampling or residual-based weighting strategies,
4. Enforcing hard Dirichlet constraints via output transformations.

Overall, the PINN model successfully captures the qualitative structure of the heat conduction process but exhibits quantitative deviations in areas of steep gradient, particularly near the boundaries—a known limitation of current PINN formulations.