Solving Heat Conduction and Convection Problems with Physics-informed Neural Networks

Wei Chen

China-UK Low Carbon College
Shanghai Jiao Tong University

June, 13 2022

SHANGHAI JIAO TONG UNIVERSITY

Solved problem

- Forward problem
  - 2D transient heat conduction,
  - 2D transient heat convection,
  - 1D transient heat conduction with thermal diffusivity as a variable
- Inverse problem of 1D transient heat conduction.

Benchmark

- Forward problem
  - 2D, conduction: finite difference method (FDM),
  - 2D, convection: OpenFOAM,
  - 1D, conduction: analytical solution,
- Inverse problem, 1D conduction: analytical solution.

Abstract framework[1]

For governing equation,

$$f := u_t + \mathcal{N}[u; \lambda]. \tag{1}$$

For boundary conditions or initial conditions,

$$g := u(\mathbf{x}, t) - u, \mathbf{x} \in \partial\Omega \tag{2}$$

Mean square error,

$$\mathrm{MSE} = \mathrm{MSE}_f + \mathrm{MSE}_g, \tag{3}$$

2D, conduction I

Governing equations:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T. \tag{4}$$

Loss functions for governing equation:

$$f(\mathbf{x}, t) = T_t(\mathbf{x}, t) - \alpha(T_{xx}(\mathbf{x}, t) + T_{yy}(\mathbf{x}, t)), \mathbf{x} \in \Omega, t \in [0, t_1], \tag{5}$$

where $\mathbf{x} = (x, y)$, $T_t = \partial T / \partial t$, $T_{xx} = \partial^2 T / \partial x^2$ and $T_{yy} = \partial^2 T / \partial y^2$.

2D, conduction II

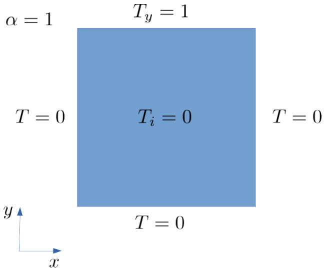


Figure 1: BCs and IC of 2D transient heat conduction

2D, conduction III

Network architecture,

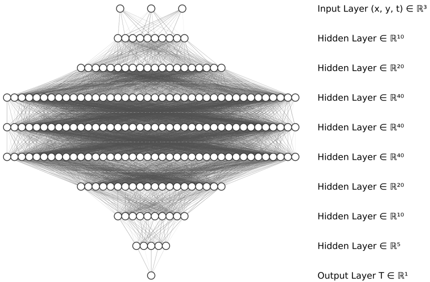


Figure 2: Network architecture of 2D transient heat conduction

2D, convection I

The governing equations of 2D transient heat convection consist of 3 parts, continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{6}$$

momentum equation

$$\begin{aligned} \frac{\partial u}{\partial t} + \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \right) = -p_x + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial v}{\partial t} + \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \right) = -p_y + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \end{aligned}, \tag{7}$$

2D, convection II

and energy equation

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right). \tag{8}$$

By introduction stream function ψ as

$$u = \frac{\partial \psi}{\partial x}, v = -\frac{\partial \psi}{\partial y},$$

the continuity equation is satisfied automatically.

2D, convection III

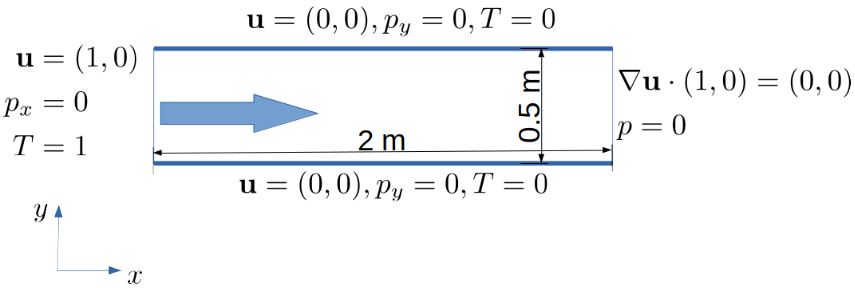


Figure 3: BCs and IC of 2D transient heat convection

1D, conduction

BCs and IC,

$$\begin{cases} T(-1,t)=0 \\ T(1,t)=0 \\ T(x,0)=1. \end{cases} \tag{9}$$

Loss function:

$$f(x,t,\alpha)=u_t(x,t)-\alpha u_{xx}(x,t). \tag{10}$$

Inverse, 1D, conduction

Loss function:

$$f(x,t) = u_t(x,t) - \alpha u_{xx}(x,t), \tag{11}$$

where α is a parameter of the neural networks[1].

[1] He et al. [2] treated α as an output neuron of a network and only solved a problem with spatial uniformly distributed α.

Network architecture

Table 1: Neural network's layers number

Problem	Layers
2D conduction	$3+10+20+40\times 3+20+10+5+1$
2D convection, velocity	$3+20\times 8+2$
2D convection, energy	$3+20\times 8+1$
1D conduction, α	$3+20\times 8+1$
1D conduction, inverse	$3+10+20+40\times 3+20+10+5+1$

2D, conduction I

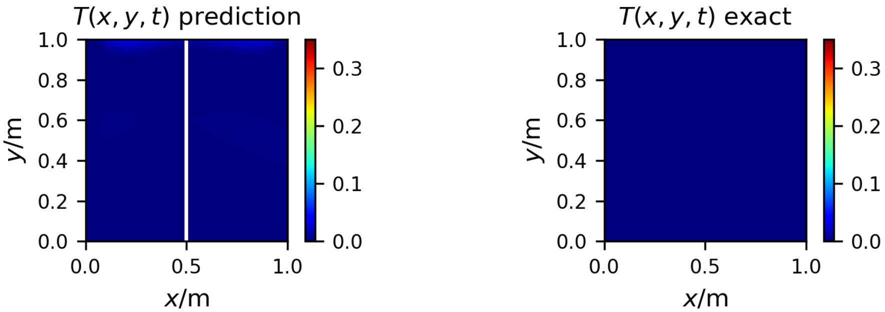


Figure 4: time $= 0$ s

2D, conduction II

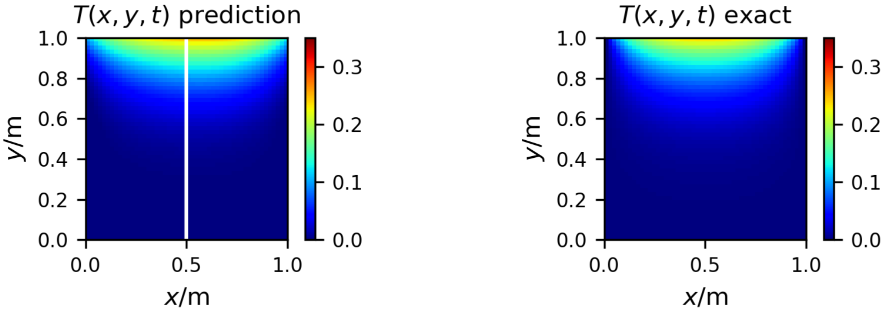


Figure 5: time $= 0.05$ s

2D, conduction III

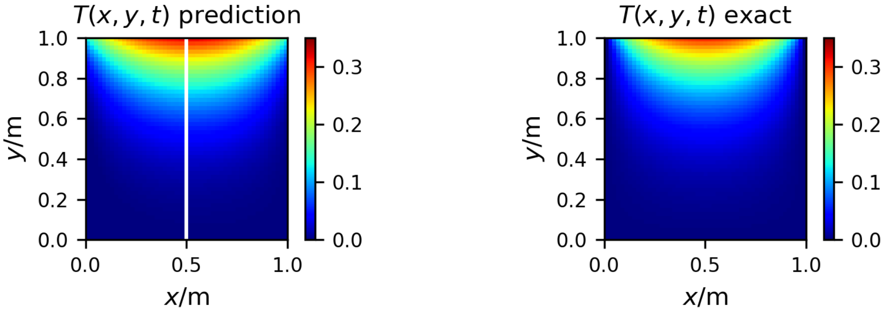


Figure 6: time $= 0.1$ s

2D, conduction IV

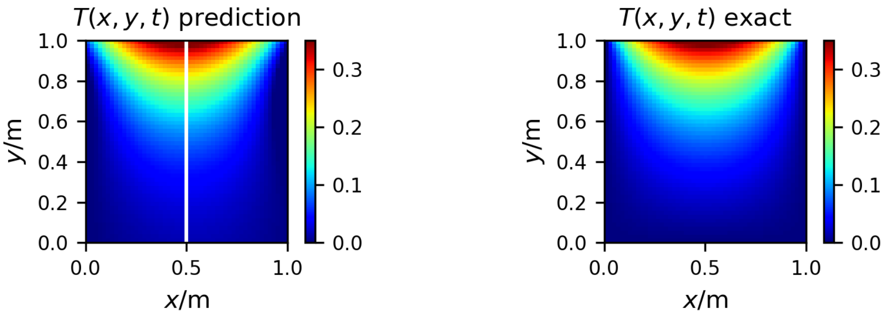


Figure 7: time $= 0.5$ s

2D, conduction V

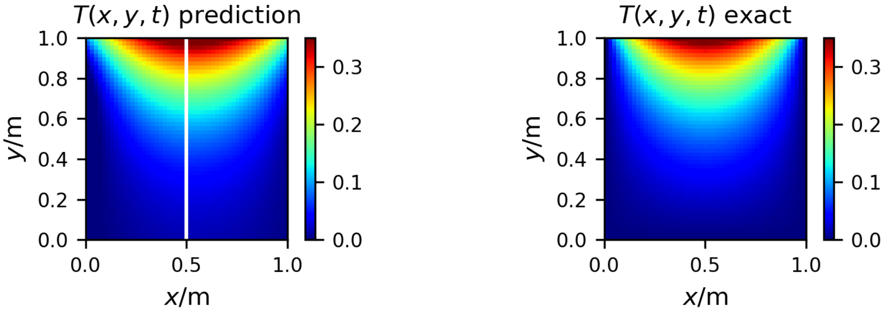


Figure 8: time $= 1.0 \mathrm{~s}$

2D, conduction VI

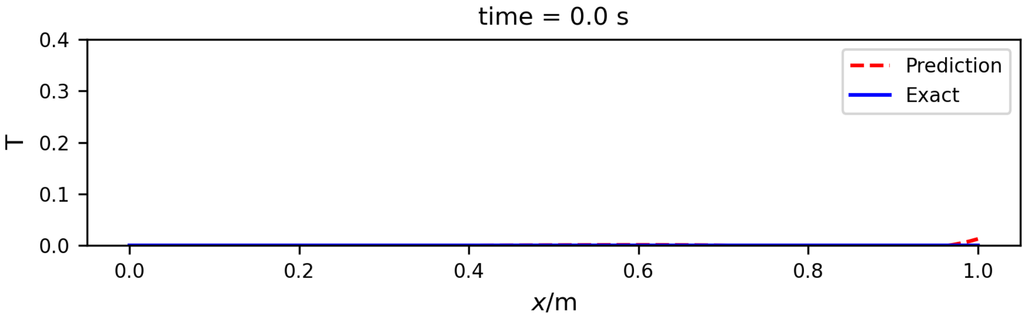


Figure 9: Line chart 1

2D, conduction VII

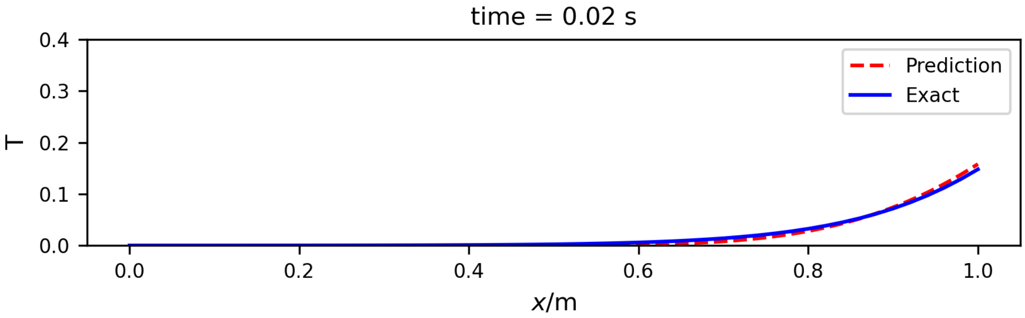


Figure 10: Line chart 2

2D, conduction VIII

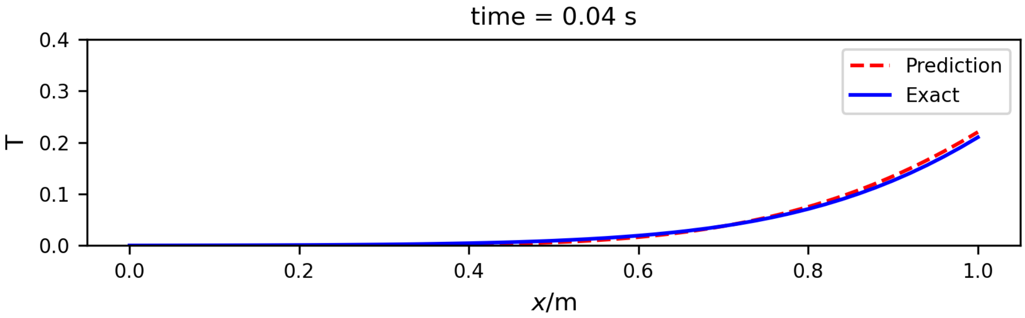


Figure 11: Line chart 3

2D, conduction IX

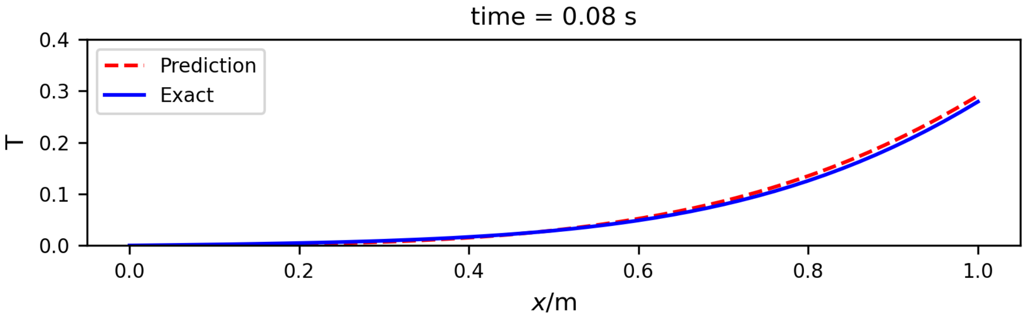


Figure 12: Line chart 4

2D, conduction X

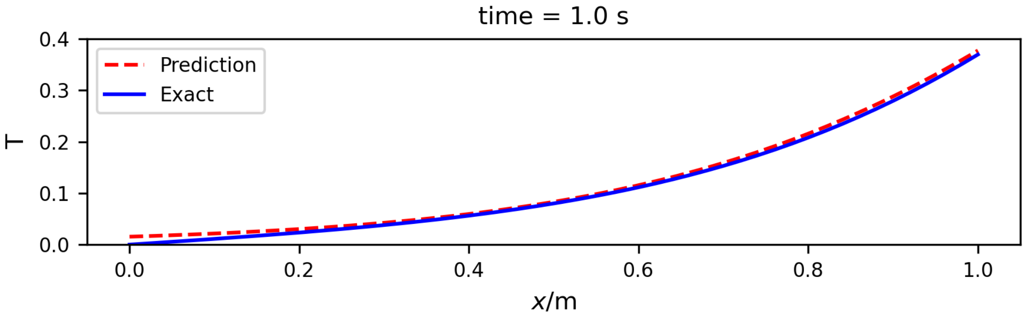
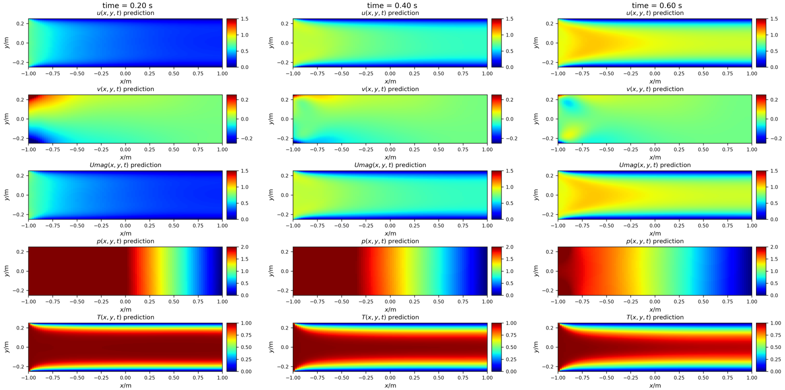


Figure 13: Line chart 5

2D, convection I



2D, convection II

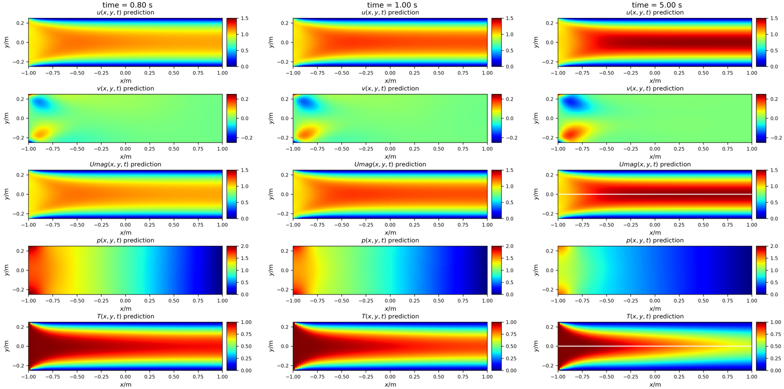


Figure 14: Prediction value of 2D transient heat conduction

2D, convection III

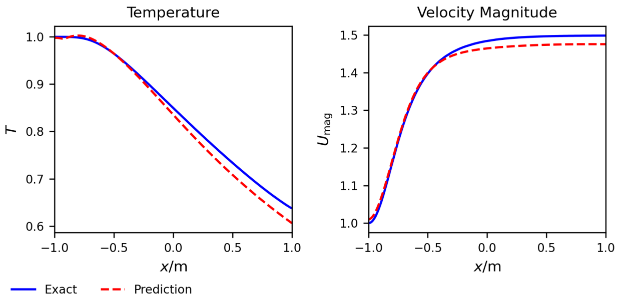


Figure 15: Comparison between prediction and exact value of 2D transient heat convection

1D, conduction I

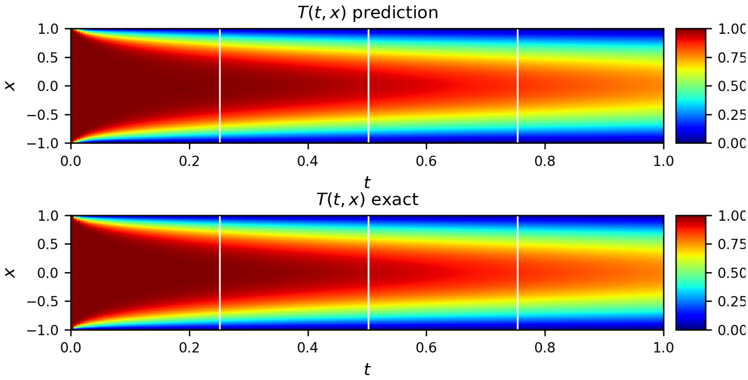


Figure 16: $\alpha = 0.2$, heatmap

1D, conduction II

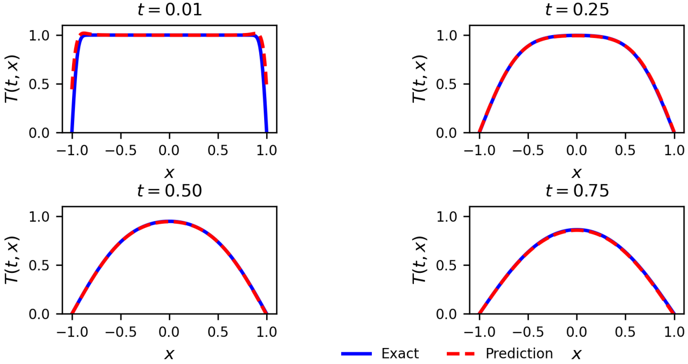


Figure 17: $\alpha = 0.2$, sampling line chart

1D, conduction III

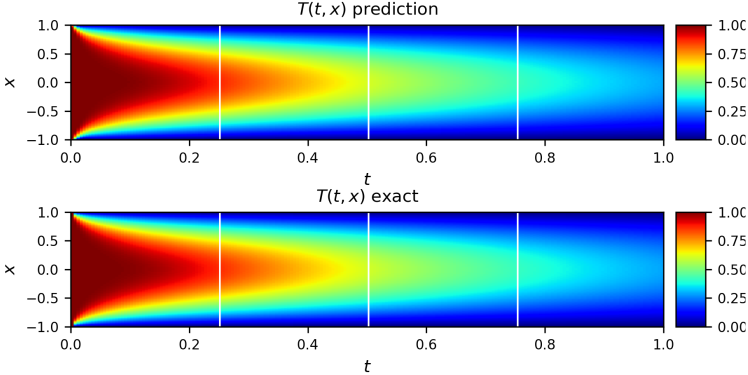


Figure 18: $\alpha = 0.6$, heatmap

1D, conduction IV

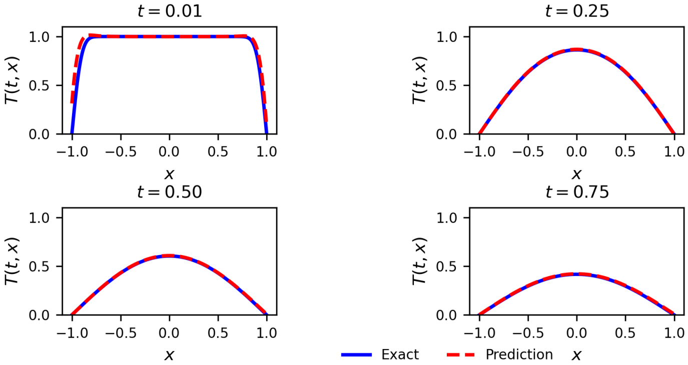


Figure 19: $\alpha = 0.6$, sampling line chart

1D, conduction V

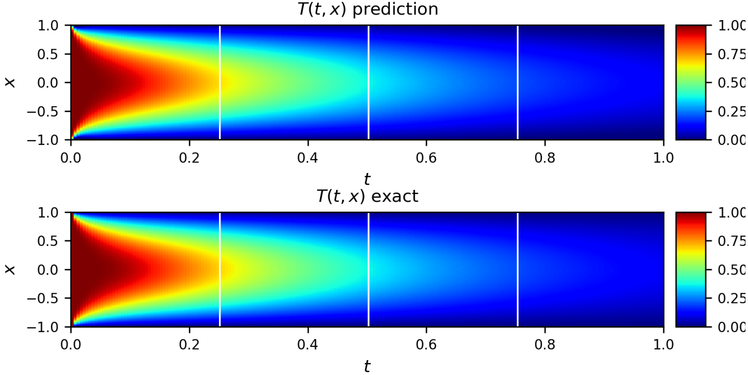


Figure 20: $\alpha = 1.0$, heatmap

1D, conduction VI

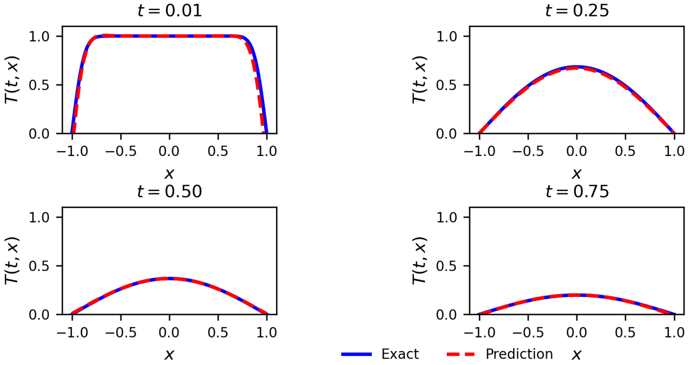


Figure 21: $\alpha = 1.0$, sampling line chart

Training time

It is not precise, just as an additional information.

1. 2D conduction: 1 hour
2. 2D convection: 2-3 hours for two networks
3. 1D conduction: less than 1 hour

They are much larger than the computational cost of traditional methods.

Inverse, 1D, conduction I

Algorithm 1 Training process for inverse problem without noise

```
Randomly select training points for governing equation
Randomly select some points with given temperature value according to the analytical solution
for epoch = 1 to 10 do
    while The loss decrease do
        Optimize the model
    end while
    Randomly select new training points for governing equation
end for
```

Inverse, 1D, conduction II

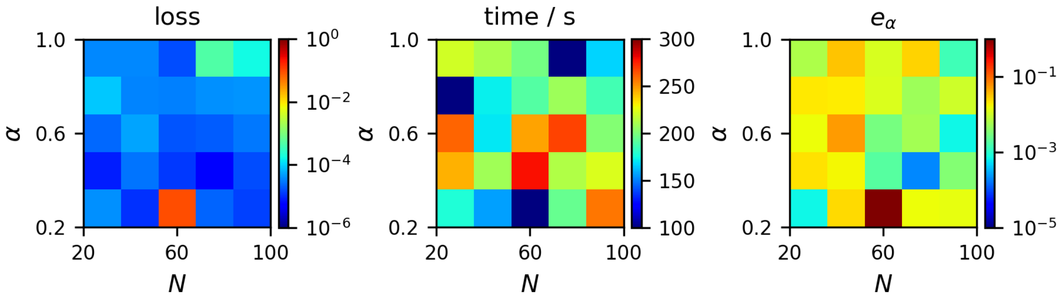


Figure 22: Trial 1 without noise

Inverse, 1D, conduction III

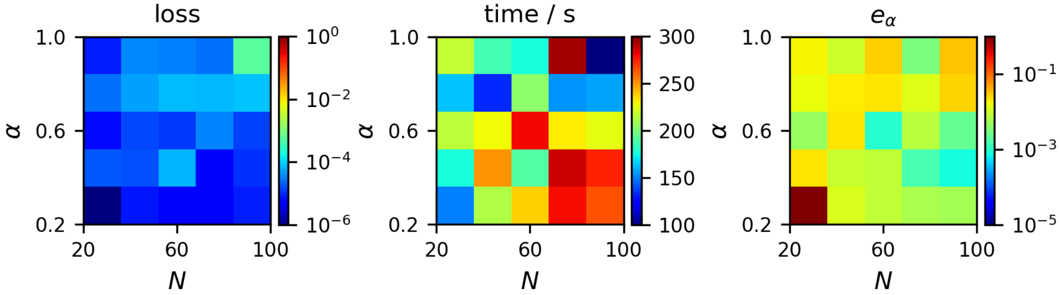


Figure 23: Trial 2 without noise

Inverse, 1D, conduction IV

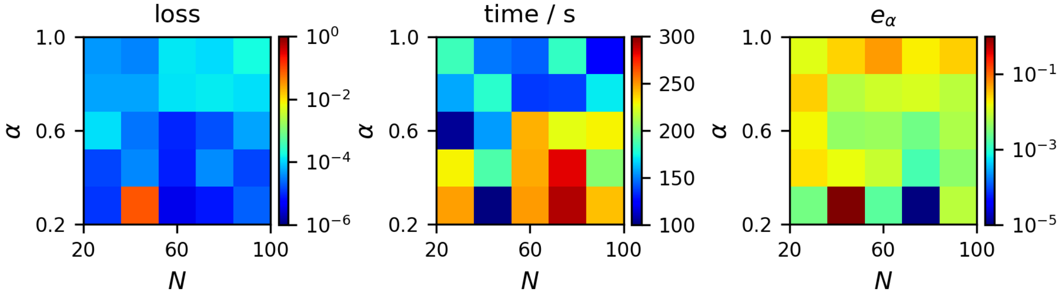


Figure 24: Trial 3 without noise

Inverse, 1D, conduction V

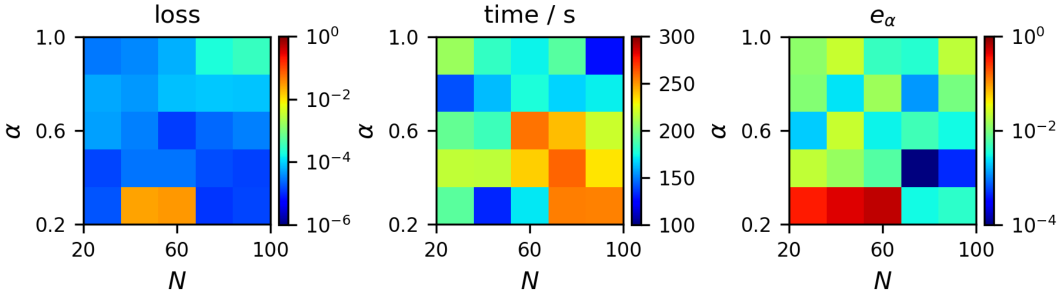


Figure 25: Mean value of the three trials without noise

Inverse, 1D, conduction VI

Algorithm 2 Training process for inverse problem with noise

Randomly select training points for governing equation
Randomly select some points with given temperature value according to the analytical solution
Impose noise on the given temperature fields.
for epoch $= 1$ to 10 **do**
    while The loss decrease **do**
        Optimize the model
    end while
    Randomly select new training points for governing equation
end for

Inverse, 1D, conduction VII

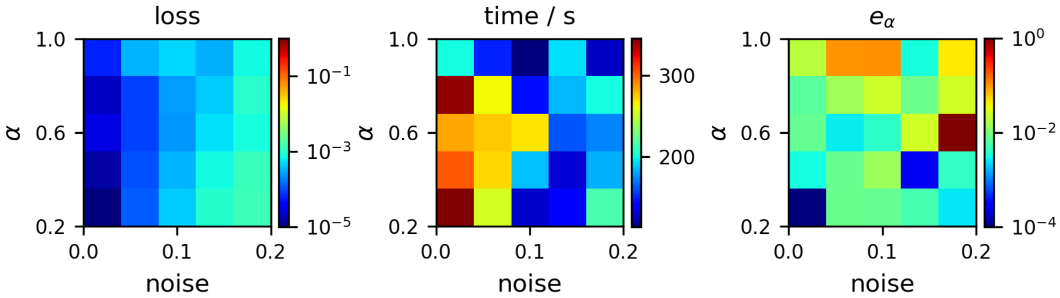


Figure 26: Results with noise

About the project

1. PINNs can solve some heat conduction and convection problem with acceptable accuracy.
2. Taking training time into consideration, the PINNs' computational cost for forward problem is significantly higher than the traditional method.
3. PINNs is able to solve inverse problem with comparatively short training time and surprisingly low error.

About my achievement

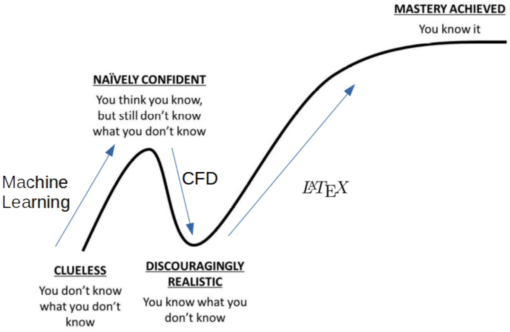


Figure 27: Learning curve

[1] 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, 378:686–707, February 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.045. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125.

[2] Zhili He, Futao Ni, Weiguo Wang, and Jian Zhang. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. Materials Today Communications, 28:102719, September 2021. ISSN 23524928. doi: 10.1016/j.mtcomm.2021.102719. URL https://linkinghub.elsevier.com/retrieve/pii/S235249282100711X.