上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Course Paper

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

Name	：	Wei Chen
Student ID	：	121280910033
Course	：	Advanced Heat Transfer
Supervisor	：	Prof. Tao Ren
Department	：	China-UK Low Carbon College

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1 Introduction

Machine learning has been already applied to solve scientific problems in a wide range of disciplines, including protein structure prediction[1], solving fluid dynamic problem[2], scalar field retrieval[3-5]. These applications have already achieved outstanding results and are able to solve the problem, that is difficult or even impossible to be solved by conventional methods. However, the machine learning methods mentioned above require huge amount of available and reliable training data, which may be not always available[6].

In fact, for the machine learning model especially in physics and biological systems, the traditional methods have partly or totally ignored a lot of prior knowledge. Most of the so-called prior knowledge can be explained into a set of differential equations with boundary conditions and initial conditions. These factors can be utilized to impose a constrain on the machine learning model and therefore to be trained within little or even no training label. The new way to derive the model without huge amount of training data is physics-informed neural networks (PINNs). This method use governing equations, boundary conditions and initial conditions as the model's loss function with the help of the auto . Only with spatio-temporal coordinates (and equation parameters, if required), the model can be trained.[7]

Solving differential equations alone doesn't make PINNs become an outstanding and attractive theory. A couple of mature numerical methods, such as finite volume method (FVM), finite element method (FEM) and finite difference method (FDM), can solve the problems with high accuracy, consistency and stability. In fact, to the best of authors knowledge, the computational cost (including the training time) and the error of the results computed by PINNs are normally much higher than the traditional methods. Nevertheless, the PINNs have the following non-negligible features[8]:

1. Solving the inverse problems: An inverse problem is the process finding the casual factors from the observations[9]. For complicated system, a great effort is required to find the results and no common methods or guide line exists for different problems in a traditional way. However, PINNs can solve the inverse problems by adding a dimension of output and slightly modify the training process.
2. Solving the high dimensional problems: The mesh size or the computational cost would increase with the problem's dimension in powers by conventional methods. For example, if the mesh is uniformly divided into 100 parts in each dimensions, the mesh size would be only 10^4 when the dimension is two by 10^8 when the dimension is four. The computational cost would be unimaginable if the dimension is high. However, due to the neural networks' own feature, the increase of dimension will not lead to dramatic rise of training and prediction time as well.

In this report, the following problems are solved with PINNs and traditional methods (either analytical or numerical):

1. Forward problem of 2D transient heat conduction with von Neumann boundary condition and three Dirichlet boundary condition (PINNs + numerical solution),
2. Forward problem of 2D transient heat convection in a channel (PINNs + numerical solution)
3. Forward problem of 1D transient heat conduction with two Dirichlet boundary condition and thermal diffusivity as a variable (PINNs + analytical solution),
4. Inverse problem of 1D transient heat conduction with two Dirichlet boundary condition (PINNs + analytical solution).

The results by the traditional methods are served as a benchmark of the accuracy. The theory of the mentioned methods in each problems are provided in Chapter 2, and the results are illustrated with figures and some discussion in Chapter 3.

Chapter 2 Methodology

2.1 Abstract Framework

Nonlinear partial differential equations can be described in a general form as

$$u_t + \mathcal{N}[u;\lambda] = 0, \mathbf{x} \in \Omega, t \in [0,T], \tag{2–1}$$

where $u(\mathbf{x},t)$ is the equations' solution and will be approximated by a neural network, $\mathcal{N}[\cdot;\lambda]$ is a nonlinear differential operator parameterized by λ and $\boldsymbol{\Omega}$ is a subset of $\mathbb{R}^D$. This PDEs' description can cover most of problems in heat transfer and fluid dynamics, including heat conduction equation, Navier-Stokes equation, etc.

The loss function of the neural network from the governing equations can be obtained from the left-hand-side of Equation 2–1 as

$$f := u_t + \mathcal{N}[u;\lambda]. \tag{2–2}$$

Similarly, the loss function of the neural network from the boundary conditions (BCs) or initial condition (IC) can be written as (by taking Dirichlet boundary condition as an example)

$$g := u(\mathbf{x},t) - u, \mathbf{x} \in \partial\Omega \tag{2–3}$$

The differential term can be obtained by the automatic differentiation feature of the neural network. Therefore, the network's parameters can be optimized by the mean squared error loss of f and g

$$\mathrm{MSE} = \mathrm{MSE}_f + \mathrm{MSE}_g, \tag{2–4}$$

where

$$\mathrm{MSE}_f = \frac{1}{N_f}\sum_{i=1}^{N_f} |f(\mathrm{x}_f^i, t_f^i)|^2$$

and

$$\mathrm{MSE}_g = \frac{1}{N_g}\sum_{i=1}^{N_g} |g(\mathrm{x}_g^i, t_g^i)|^2$$

One of the PINNs' significance can be shown by the training process of the network, where no label data are required to optimize the PINNs' parameters.

2.2 Forward Problem with PINNs

2.2.1 2D Transient Heat Conduction

The governing equation of 2D transient heat conduction is

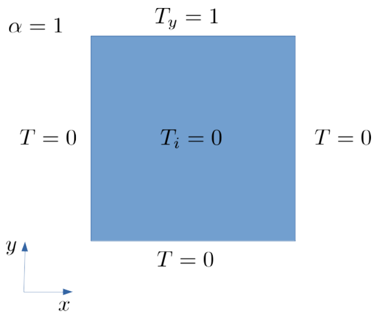


Figure 2–1 BCs and IC of 2D transient heat conduction

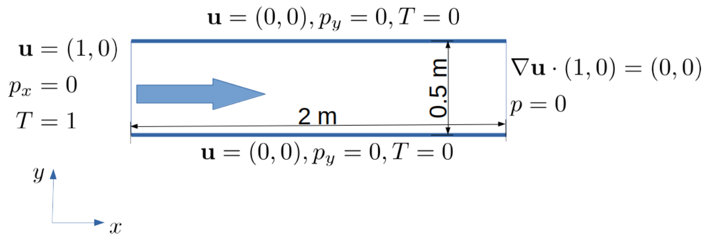


Figure 2–2 BCs and IC of 2D transient heat convection

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T. \tag{2–5}$$

In this report, a two dimensional square plate's heat conduction problem is solved. Its BCs and IC are shown in Figure 2–1.

The governing equation's loss function of this problem is

$$f(\mathbf{x}, t) = T_t(\mathbf{x}, t) - \alpha(T_{xx}(\mathbf{x}, t) + T_{yy}(\mathbf{x}, t)), \mathbf{x} \in \Omega, t \in [0, t_1], \tag{2–6}$$

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

The BCs' and IC's loss function of this problem is

$$g(\mathbf{x}, t) = \begin{cases} T & t = 0 \text{ or } x = 0 \text{ or } x = 1 \text{ or } y = 0 \\ T_y - 1 & y = 1 \end{cases} \tag{2–7}$$

The mean square error can be obtained from Equation 2–4.

The network architecture is shown in Figure 2–3[①] with layers of $3 + 10 + 20 + 40 \times 3 + 20 + 10 + 5 + 1$.

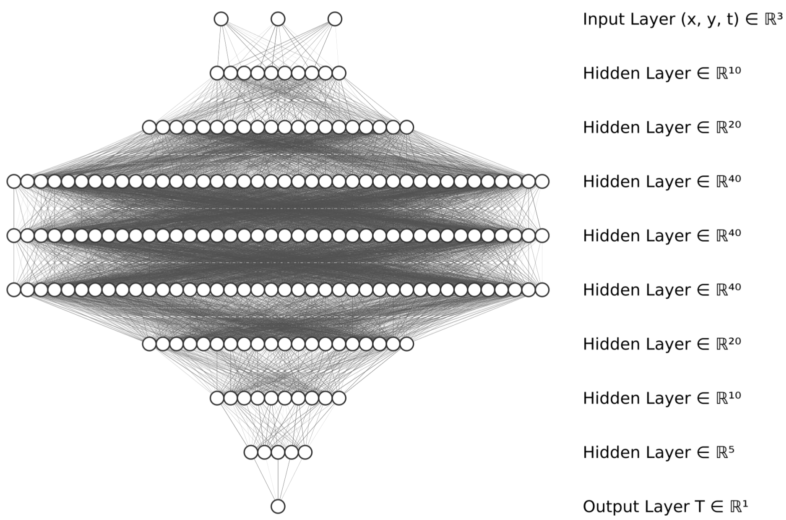


Figure 2–3 Network architecture of 2D transient heat conduction

① Plot by the online tool NN SVG: :http://alexlenail.me/NN-SVG/index.html

2.2.2 2D Transient Heat Convection

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

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{2–8}$$

momentum equation

$$\begin{aligned} \frac{\partial u}{\partial t} + \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) &= -p_x + v\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\ \frac{\partial v}{\partial t} + \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) &= -p_y + v\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)' \end{aligned} \tag{2–9}$$

and energy equation

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right). \tag{2–10}$$

By introduction stream function ψ as

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

the continuity equation is satisfied automatically.

This report solves a 2-dimensional channel flow with the boundary conditions as illustrated in Figure 2–2. Since the velocity equation is not coupled with the temperature term. Two terms can be solved individually by two neural networks. The velocity field is obtained first and it is utilized to train the temperature neural networks. The problem's loss function is obtained in a similar way as mentioned in Subsection 2.2.1, which will not be repeated here.

The network architecture for velocity is $3 + 20 \times 8 + 2$, which takes spatial coordinate x, y temporal term and t as input neurons and pressure p and stream functionψ as output neurons. Temperature networks has a structure of 3+20×8+2, that takes temperature T as output.

2.2.3 1D Transient Heat Conduction with thermal diffusivity

A 1D transient heat conduction with uniform but variable thermal diffusivity is solved. Equation 2–5 can also serve as the governing equation of this problem. A rod with constant temperature $T_b = 0$ at both sides as BCs and $T_i = 1$ as IC is considered. Obviously, thermal diffusivity α is no longer a constant but an input neurons of the networks. So the governing equation's loss function is written as

$$f(x, t, \alpha) = u_t(x, t) - \alpha u_{xx}(x, t). \tag{2–11}$$

The network has layers as $3 + 20 \times 8 + 1$, which takes x, t, and α as input neurons and T as output neuron.

The mean square error can be obtained from Equation 2–4.

2.3 Inverse Problem with PINNs

A similar physical setup as mentioned in Subsection 2.2.3 is taken as the BCs and IC of this problem. Nevertheless, in order to solve the inverse problem, thermal conductivity α is registered as an additional parameter of the neural network and is trained together with the rest of them. Therefore, the governing equation's loss function is written as

$$f(x,t) = u_t(x,t) - \alpha u_{xx}(x,t). \tag{2–12}$$

He et al.[10] solve the 1D heat conduction inverse problem by building two neural networks. One network output temperature and another yield thermal diffusivity. Two networks are coupled with each other and optimized simultaneously. The α in their problem is also uniform distributed in spatio-temporal domain. However, this attempt is in fact unnecessary because the output of the second neural network is theoretically a constant. A comparison between two methods is shown in Section 3.2

Table 2–1 Neural network's layers number

Problem	Layers
2D conduction	$3+10+20+40\times3+20+10+5+1$
2D convection, velocity	$3+20\times8+2$
2D convection, energy	$3+20\times8+1$
1D conduction, α	$3+20\times8+1$
1D conduction, inverse	$3+10+20+40\times3+20+10+5+1$

2.4 Traditional Solution

2.4.1 Analytical: 1D Transient Heat Conduction

Originally, we can have the governing equation in the form of:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)$$

We also have the boundary conditions:

$$k\frac{\partial T}{\partial x}|_{x=0} = 0 T|_{x=L} = T_1$$

initial conditions:

$$T(x,0) = T_i$$

If we define several non-dimensional variable with:

$$\begin{aligned} \text{length:} \quad & \eta = x/L \\ \text{time:} \quad & \tau = \alpha t/L^2 \\ \text{temperature:} \quad & \theta = \frac{T-T_1}{T_i-T_1} \end{aligned}$$

Where $\alpha = \frac{k}{\rho c_p}$ is the thermal diffusivity. So we can get the equation and the boundary as well as the initial condition in the form of:

$$\frac{\partial \theta}{\partial \tau} = \frac{\partial^2 \theta}{\partial \eta^2} \begin{cases} \text{Boundary conditions:} & \frac{\partial \theta}{\partial \eta}|_{\eta=0} = 0 \\ & \theta|_{\eta=L} = 0 \\ \text{Initial conditions:} & \theta|_{\tau=0} = 1 \end{cases}$$

Since we have two homogenous boundary conditions and the equation is also homogenous and linear, we can apply the seperation of variable (SOV) to solve this problem.

Let

$$\theta(\eta, \tau) = X(\eta)F(\tau)$$

We can have:

$$XF' = FX'' \rightarrow F'/F = X''/X$$

Let $F'/F = X''/X = -\lambda^2$, we can have the following equations:

$$X'' + \lambda^2 X = 0F' + \lambda^2 F = 0$$

For the first equation, we have the general solution:

$$X(\eta) = C_1 \sin(\lambda\eta) + C_2 \cos(\lambda\eta)$$

For the second equation, we have:

$$F(\tau) = C_3 \mathrm{e}^{-\lambda^2 \tau}$$

So that we can have the general solution of the non-dimensional equation as:

$$\theta(\eta, \tau) = [D_1 \sin(\lambda\eta) + D_2 \cos(\lambda\eta)]\, \mathrm{e}^{-\lambda^2 \tau}$$

And its derivative:

$$\frac{\partial \theta}{\partial \eta} = [D_1 \cos(\lambda\eta) - D_2 \sin(\lambda\eta)]\, \mathrm{e}^{-\lambda^2 \tau}$$

Apply the boundary condition we have into the general solution and its derivative. In $\eta = 0$, we have:

$$\frac{\partial \theta}{\partial \eta}|_{\eta=0} = D_1 \mathrm{e}^{-\lambda^2 \tau} \rightarrow D_1 = 0$$

In $\eta = 1$, we have:

$$\theta(\eta, \tau) = D_2 \cos\lambda \mathrm{e}^{-\lambda^2 \tau} = 0 \rightarrow \lambda_n = \pi/2 + n\pi \qquad (n = 0, 1, 2 \cdots)$$

So that we can rewrite the general solution in the form:

$$\theta(\eta, \tau) = \sum_{n=1}^{\infty} C_n \cos(\lambda_n \eta) \mathrm{e}^{-\lambda_n^2 \tau}$$

We could apply the initial condition to the general solution with the help of orthogonality:

$$\theta(\eta,0)=\sum_{n=1}^{\infty} C_n\cos(\lambda_n\eta)=1$$

$$C_n\int_0^1\cos^2(\lambda_n\eta)\mathrm{d}\eta=\int_0^1\cos(\lambda_n\eta)\mathrm{d}\eta$$

$$C_n=\frac{4\sin\lambda_n}{2\lambda_n+\sin(2\lambda_n)}$$

We can find out, that the form of the solution is almost same with the the large plate with surface convection. The only difference is the value of the eigenvalue λ_n.

2.4.2 Numerical: 2D Transient Heat Conduction

By the discretization of the governing equation, for the internal point, we have:

$$k\Delta y\frac{T_{i,j-1}^{t+1}-T_{i,j}^{t+1}}{\Delta x}+k\Delta y\frac{T_{i,j+1}^{t+1}-T_{i,j}^{t+1}}{\Delta x}+k\Delta x\frac{T_{i-1,j}^{t+1}-T_{i,j}^{t+1}}{\Delta y}+k\Delta x\frac{T_{i+1,j}^{t+1}-T_{i,j}^{t+1}}{\Delta y}=\frac{T_{i,j}^{t+1}-T_{i,j}^{t}}{\Delta t}\rho c_p\Delta x\Delta y.$$

We can rewrite it as:

$$\left(\frac{1}{\alpha\Delta t}+\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}\right)T_{i,j}^{t+1}-\frac{1}{\Delta x^2}T_{i,j-1}^{t+1}-\frac{1}{\Delta x^2}T_{i,j+1}^{t+1}-\frac{1}{\Delta y^2}T_{i-1,j}^{t+1}-\frac{1}{\Delta y^2}T_{i+1,j}^{t+1}=\frac{1}{\alpha\Delta t}T_{i,j}^{t}..$$

For the boundary points $(0,j)$,

$$k\frac{\Delta y}{2}\frac{T_{0,j-1}^{t+1}-T_{0,j}^{t+1}}{\Delta x}+k\frac{\Delta y}{2}\frac{T_{0,j+1}^{t+1}-T_{0,j}^{t+1}}{\Delta x}+k\Delta x\frac{T_{1,j}^{t+1}-T_{0,j}^{t+1}}{\Delta y}+q_s''\Delta x=\frac{T_{0,j}^{t+1}-T_{0,j}^{t}}{\Delta t}\rho c_p\Delta x\Delta y/2$$

$$\frac{T_{0,j+1}^{t+1}+T_{0,j-1}^{t+1}-2T_{0,j}^{t+1}}{2\Delta x^2}+\frac{T_{1,j}^{t+1}-T_{0,j}^{t+1}}{\Delta y^2}+\frac{q_s''}{k\Delta y}=\frac{1}{\alpha}\frac{T_{0,j}^{t+1}-T_{0,j}^{t}}{2\Delta t}.$$

We can rewrite it as:

$$\left(\frac{1}{2\alpha\Delta t}+\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)T_{0,j}^{t+1}-\frac{1}{2\Delta x^2}T_{0,j-1}^{t+1}-\frac{1}{2\Delta x^2}T_{0,j+1}^{t+1}-\frac{1}{\Delta y^2}T_{1,j}^{t+1}=\frac{1}{2\alpha\Delta t}T_{i,j}^{t}+\frac{q_s''}{k\Delta y}.$$

For the rest of BCs we have:

$$T_{i,j}^{t}=T_1.$$

We could also rewrite the equations as:
Internal equations:

$$(1+2Fo_x+2Fo_y)T_{i,j}^{t+1}-Fo_x(T_{i,j-1}^{t+1}+T_{i,j+1}^{t+1})-Fo_y(T_{i-1,j}^{t+1}+T_{i+1,j}^{t+1})=T_{i,j}^{t}.$$

Top boundary equations:

$$(1+2Fo_x+2Fo_y)T_{0,j}^{t+1}-Fo_x(T_{0,j-1}^{t+1}+T_{0,j+1}^{t+1})-2Fo_yT_{1,j}^{t+1}=T_{0,j}^{t}+\frac{2q_s''\alpha\Delta t}{k\Delta y}.$$

2.4.3 Numerical: 2D Transient Heat Convection

This problem is solved numerically in OpenFOAM by three different ways.

1. Solving the transient pressure and velocity fields by the icoFoam first and then getting the stable temperature field by the scalarTransportFoam
2. Solving three fields together by the buoyantBoussinesqSimpleFoam
3. Solving three fields together by the buoyantBoussinesqPimpleFoam

No special treatment with the differential schemes, linear system solver or pressure velocity coupling algorithm is taken. All of them are recommended options shown in OpenFOAM's tutorial cases. The boundary condition used by these cases are shown in Table 2–2

Table 2–2 Boundary condition for numerical solution of 2D transient heat convection

	Velocity	Pressure
Position	Type	Type
Left	fixedValue	fixedFluxPressure
Top	noSlip	fixedFluxPressure
Right	zeroGradient	fixedValue ($internalField)
Bottom	noSlip	fixedFluxPressure

Chapter 3 Results and Discussion

Most ot values in the figures of this chapter do not have unit, because they are regarded as the normalized values which have no significant physical meaning.

3.1 Forward Problem

3.1.1 2D Transient Heat Conduction

The comparison between the prediction and exact value of this problem is shown as heatmap in Figure 3–1. Furthermore, the temperature fields of both results are plot over line as illustrated in Figure 3–2. The error at the beginning period is low, but at time = 1.0 s, the PINNs overestimates the temperature. Overall, the prediction accuracy is acceptable. The maximum mean absolute error occurs at time = 1 s with 1.14×10^{-2} and the minimum mean absolute error occurs at time = 0 s with 2.23×10^{-3}.

3.1.2 2D Transient Heat Convection

The prediction value of the PINNs at several time steps are shown from Figure 3–3 to Figure 3–7. Temperature fields and velocity fields are sampled along the white lines shown in these figures and plotted in Figure 3–9 to compare with exact (numerical) results. Due to the complexity of the Navier-Stokes equation, the predicted accuracy of velocity field is significantly worse in comparison to the results from Subsection 3.1.1. That also leads to the temperature field error's increase. Nevertheless, the neural networks succeed in capturing the tendency of fields' change. No matter for the entrance region or the fully developed region, the temperature and velocity magnitude distribution obey the physical law.

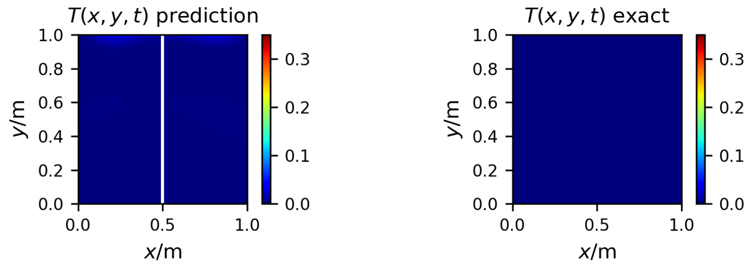
3.1.3 1D Transient Heat Conduction with thermal diffusivity

Figure 3–10 compared the prediction value with the exact value of 1D transient heat conduction problem with different thermal diffusivity. Temperature fields are also sampled along the white lines in these figures and plotted over lines as shown in Figure 3–11. The PINNs' error in this problem is almost neglectable. Only some slight differences can be found near the boundary at the beginning.

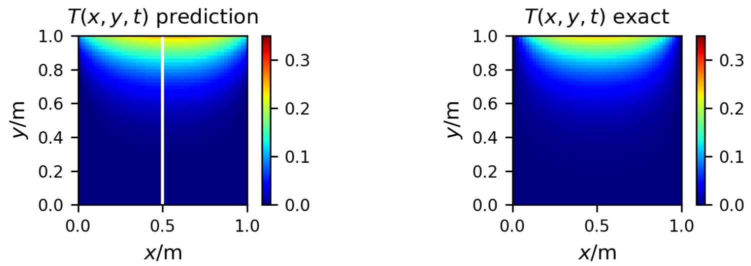
3.2 Inverse Problem: 1D Transient Heat Conduction

The inverse problem is trained as illustrated in Algorithm 3–1. The so-called known point in the algorithm means the points whose temperature value will also be passed to the neural network, and the governing point means the points that will only be used to optimize the networks by computing the loss of governing equation.

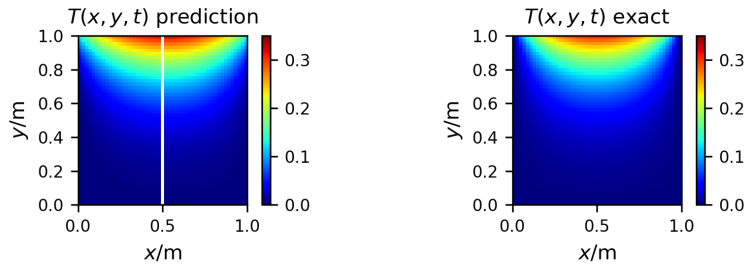
With different combinations of N and α, a couple of loss values, training time and α's errors can be obtained. They are plotted as heatmap as shown



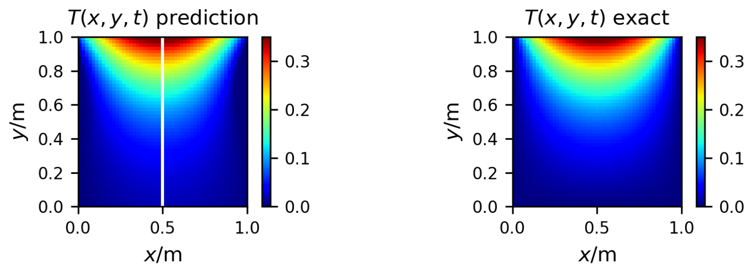
a) time = 0 s



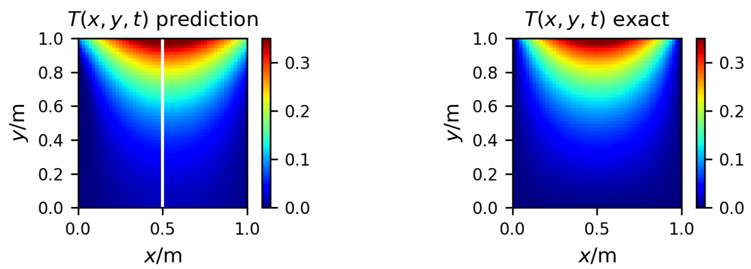
b) time = 0.05 s



c) time = 0.1 s



d) time = 0.5 s



e) time = 1.0 s

Figure 3–1 Comparison between the prediction and exact value of 2D transient heat conduction by heatmap

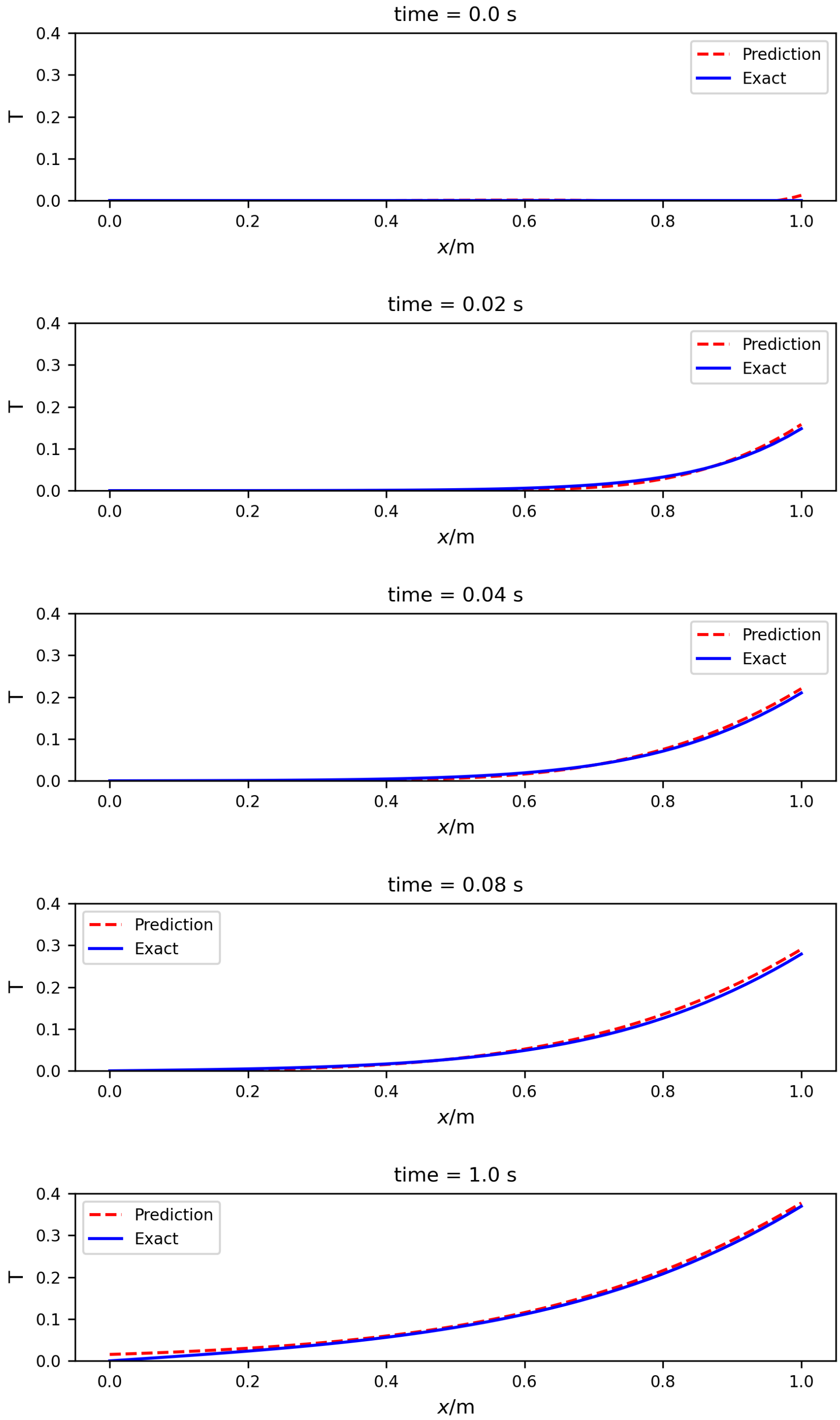


Figure 3–2 Comparison between prediction and exact value of 2D transient heat conduction by sampling at the white line of Figure 3–1

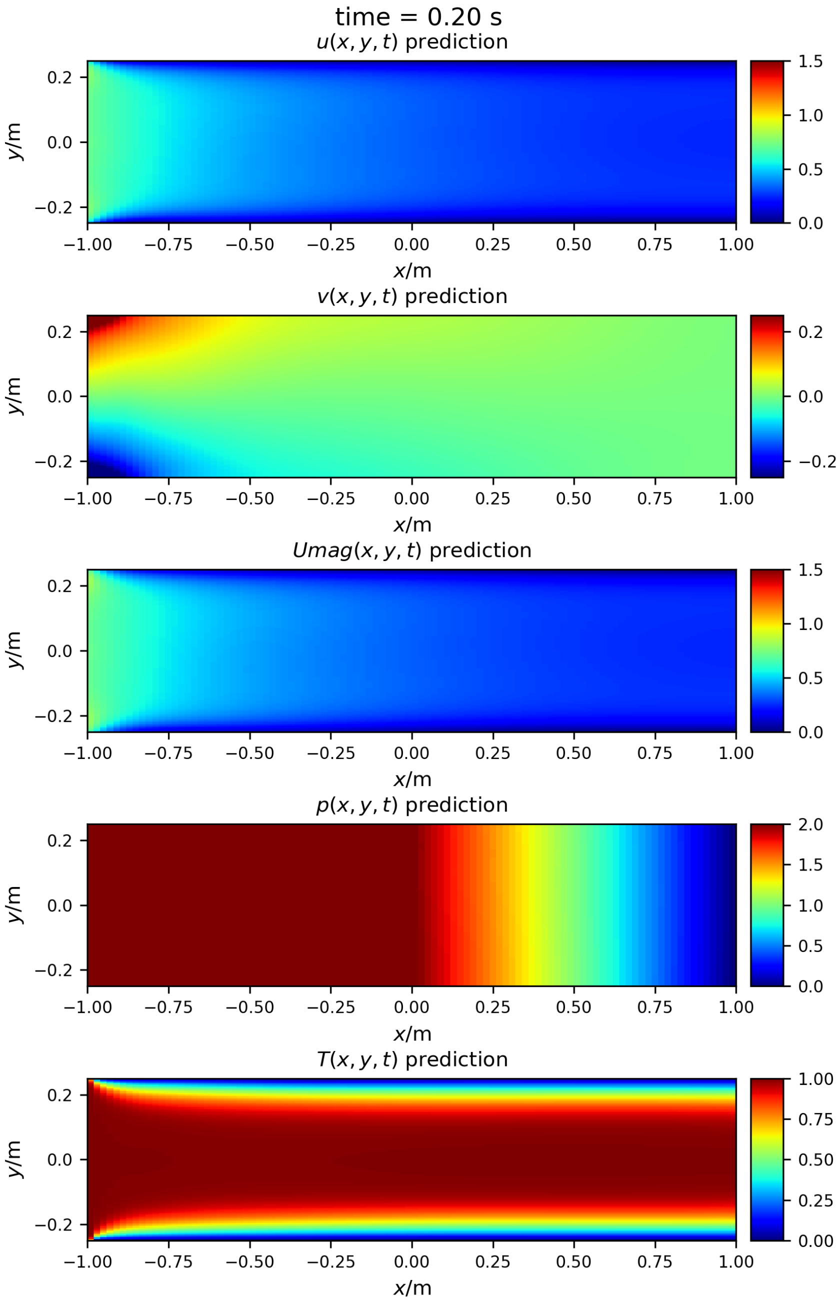


Figure 3–3 Prediction value of 2D transient heat conduction at 0.2 s

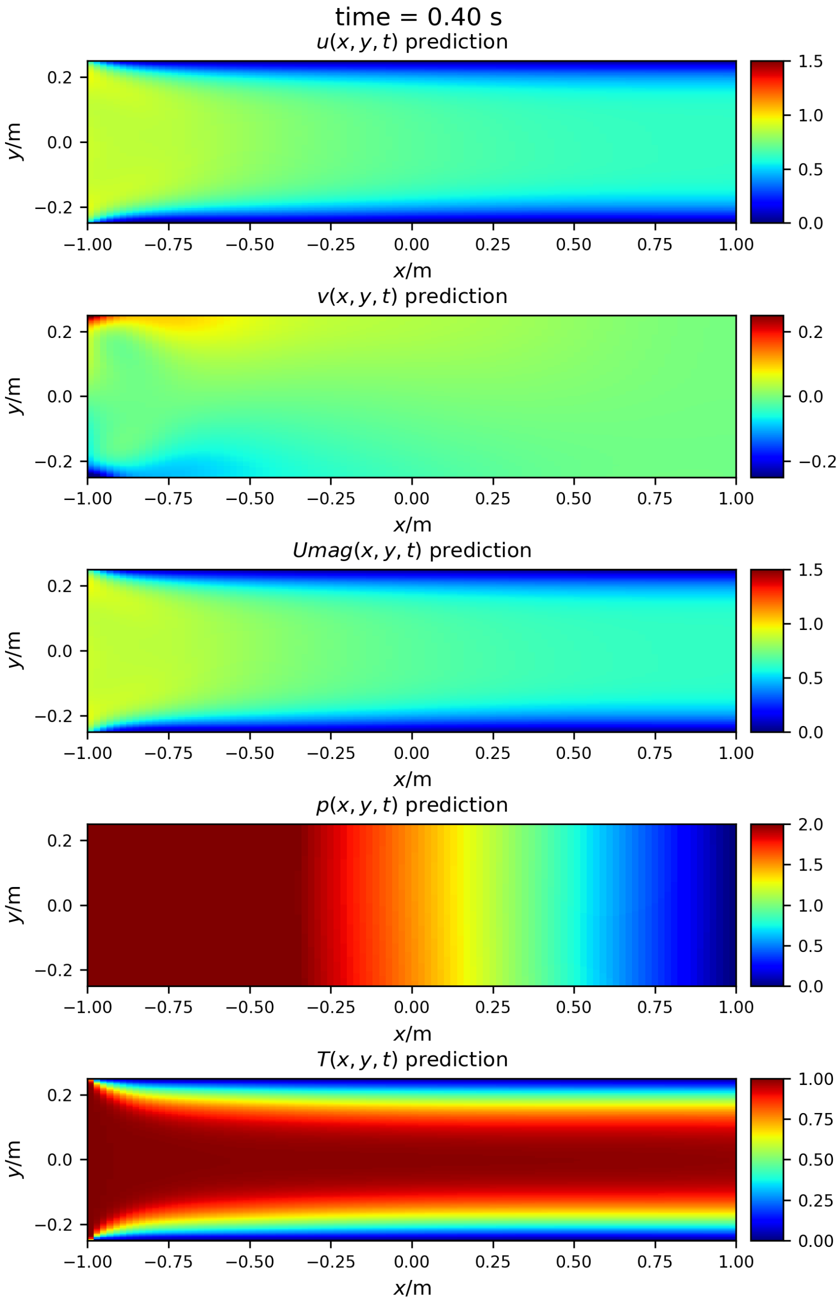


Figure 3–4 Prediction value of 2D transient heat conduction at 0.4 s

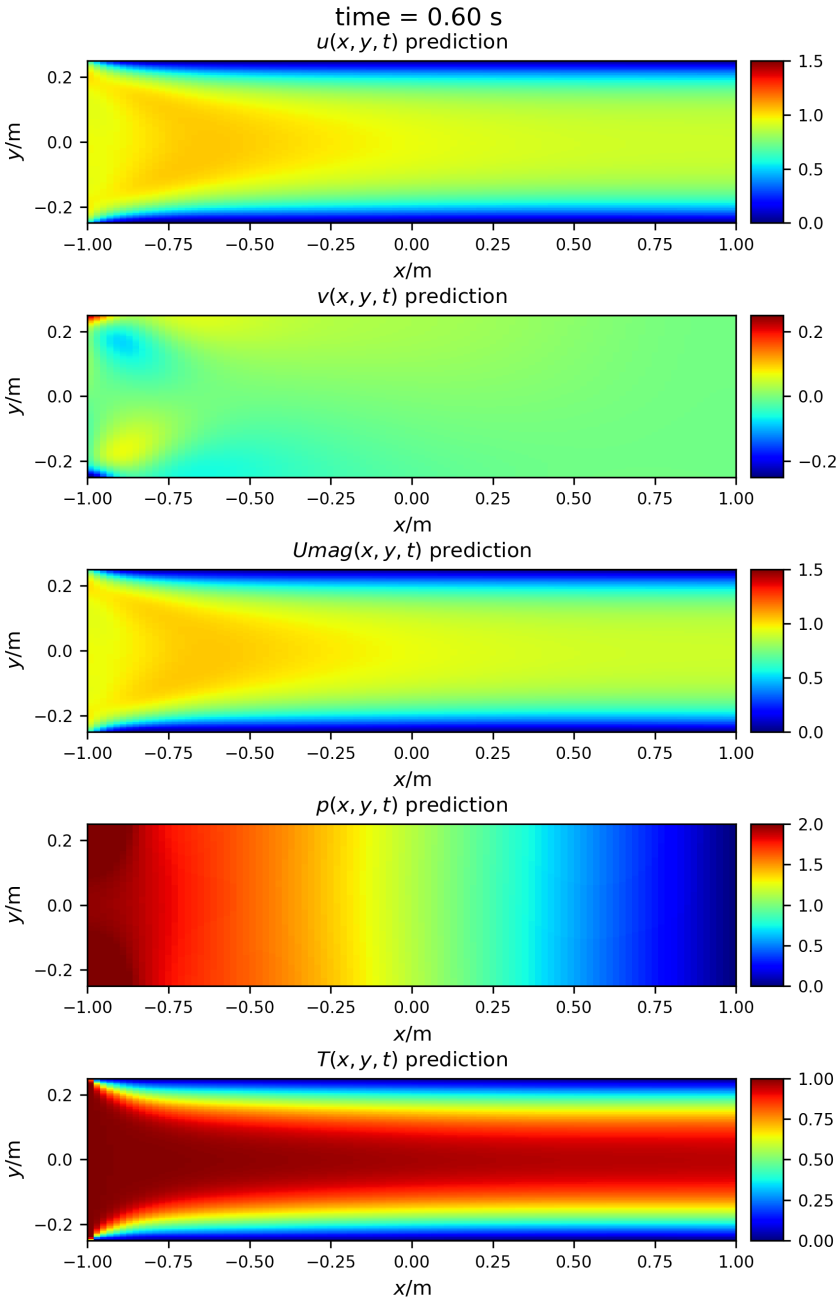


Figure 3–5 Prediction value of 2D transient heat conduction at 0.6 s

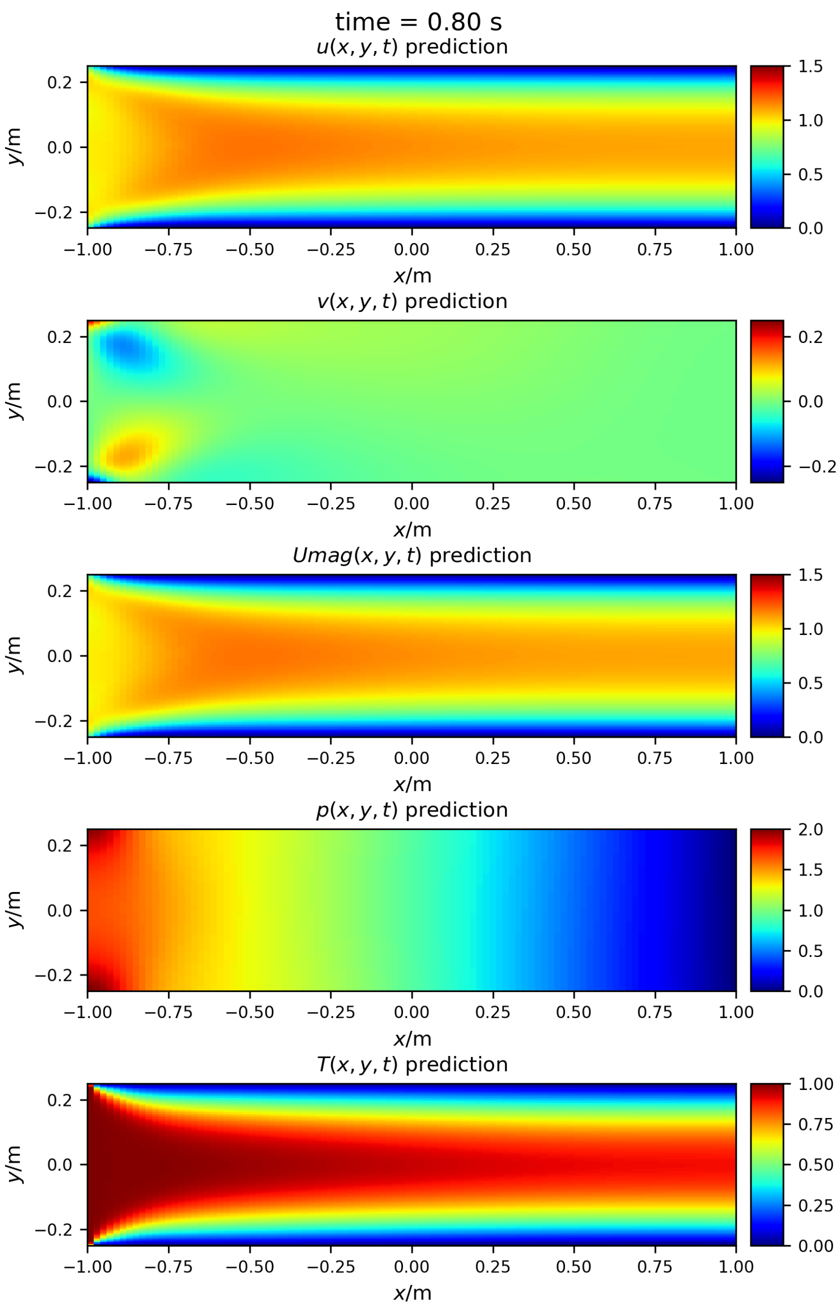


Figure 3–6 Prediction value of 2D transient heat conduction at 0.8 s

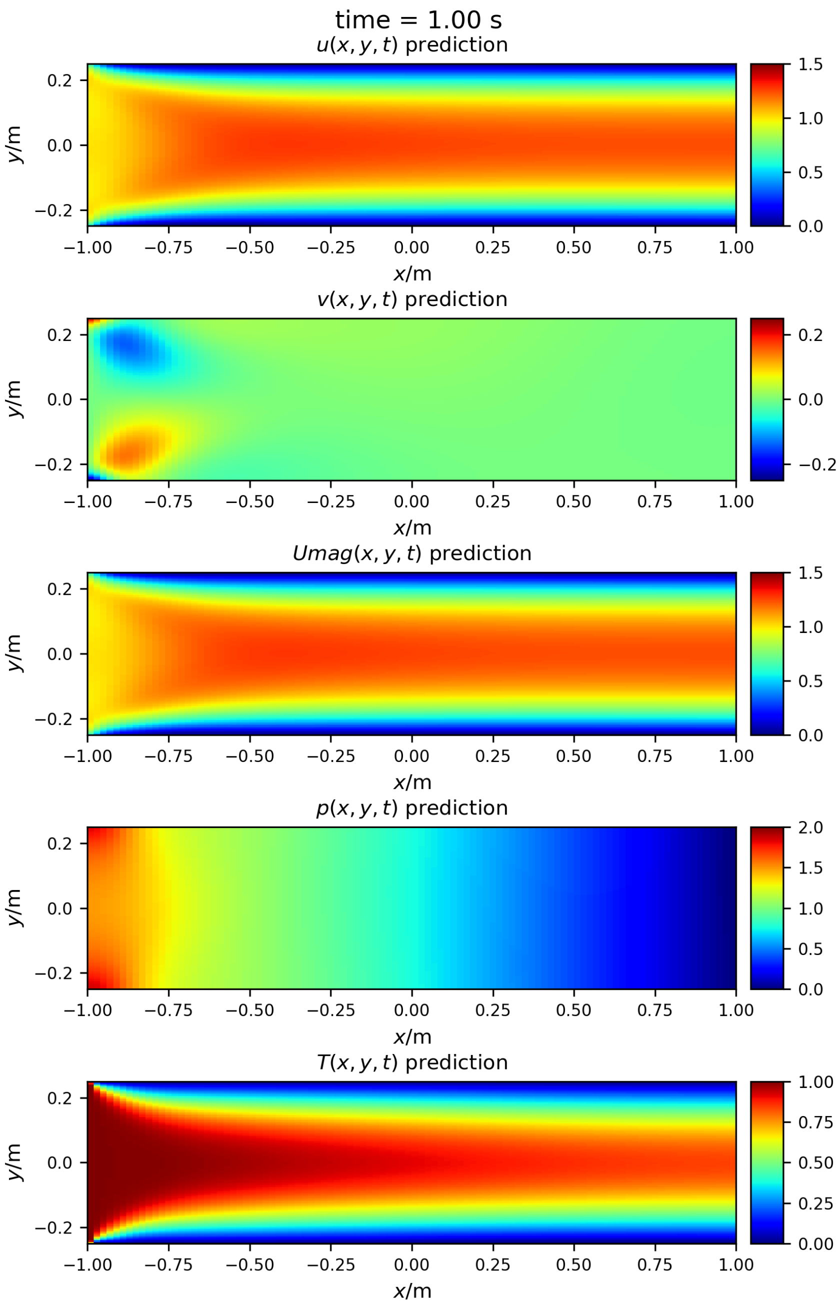


Figure 3–7 Prediction value of 2D transient heat conduction at 1.0 s

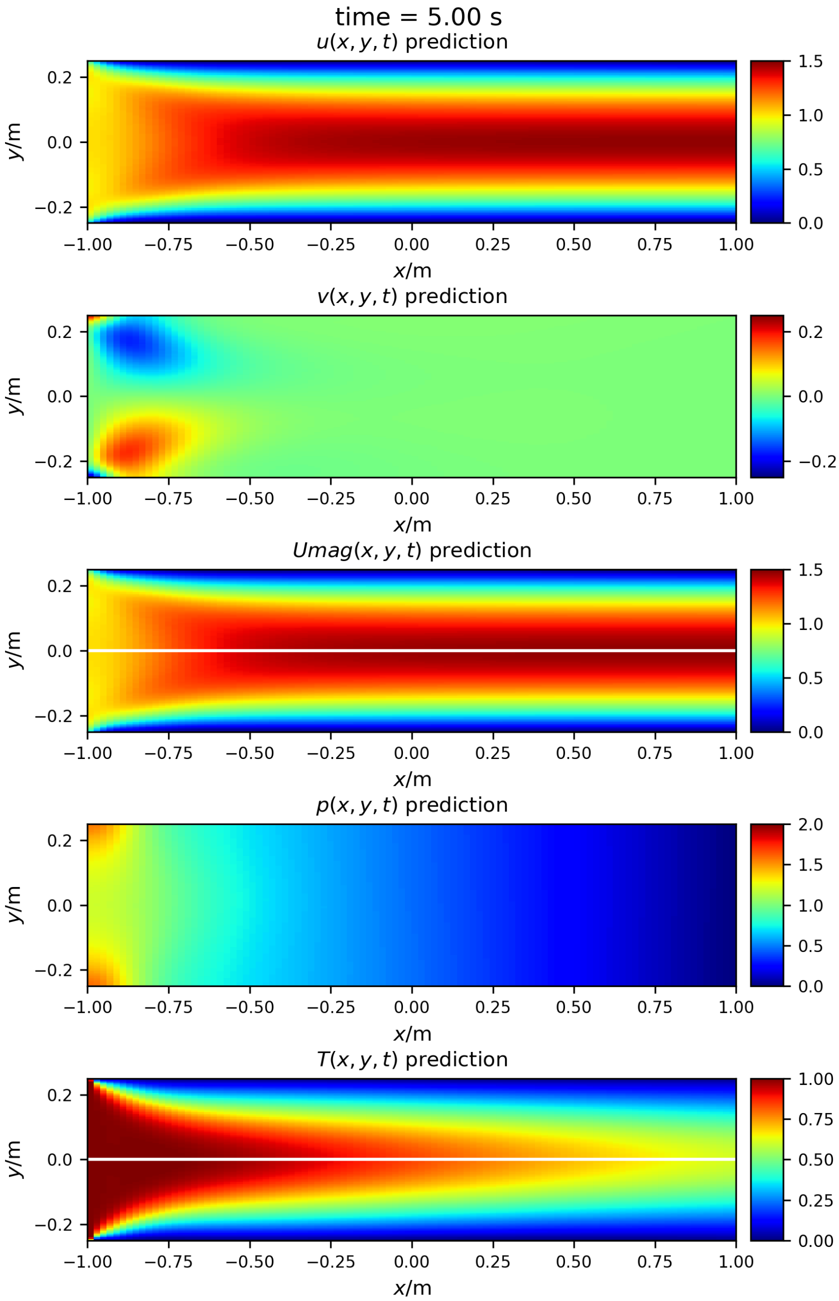


Figure 3–8 Prediction value of 2D transient heat conduction at 5.0 s

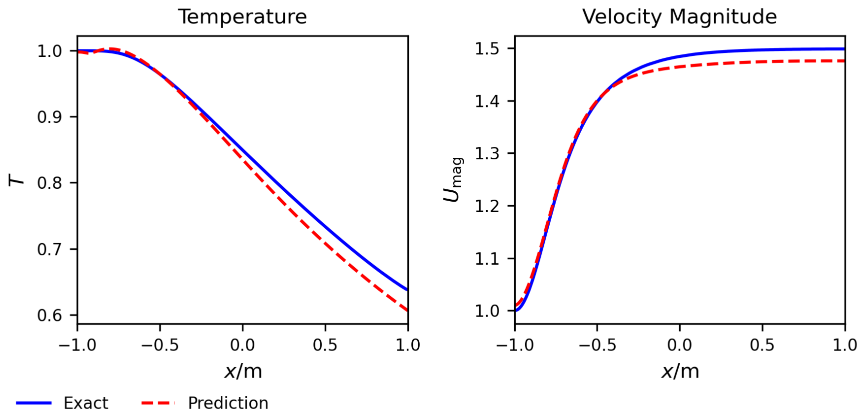


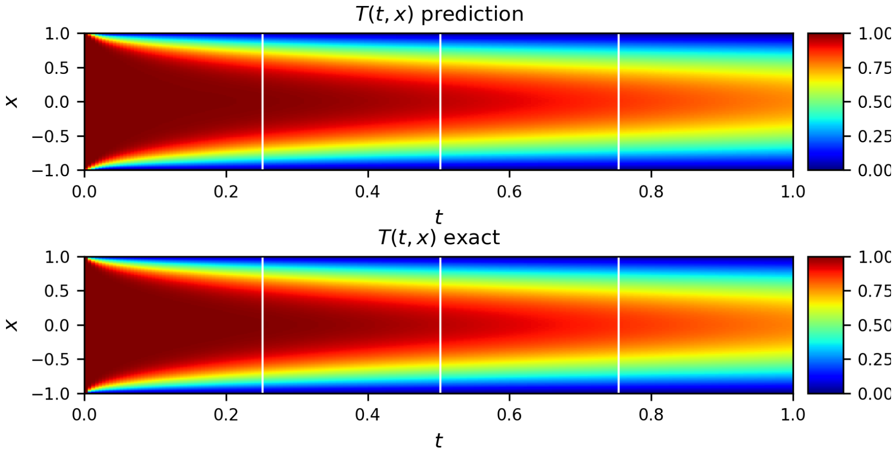
Figure 3–9 Comparison between prediction and exact value of 2D transient heat convection by sampling at the white line of Figure 3–8

Algorithm 3–1 Training process of the 1D transient heat conduction inverse problem

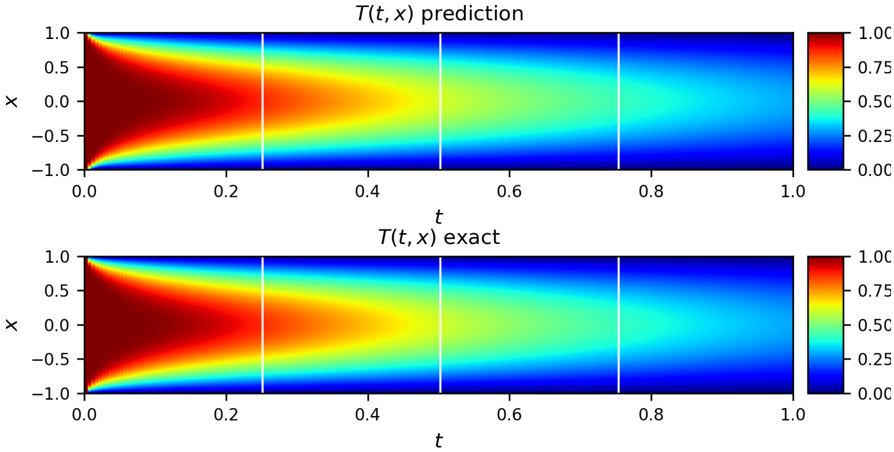
Data: Known points number N, governing points number N_f, preset thermal diffusivity α

Result: Optimized PINNs model

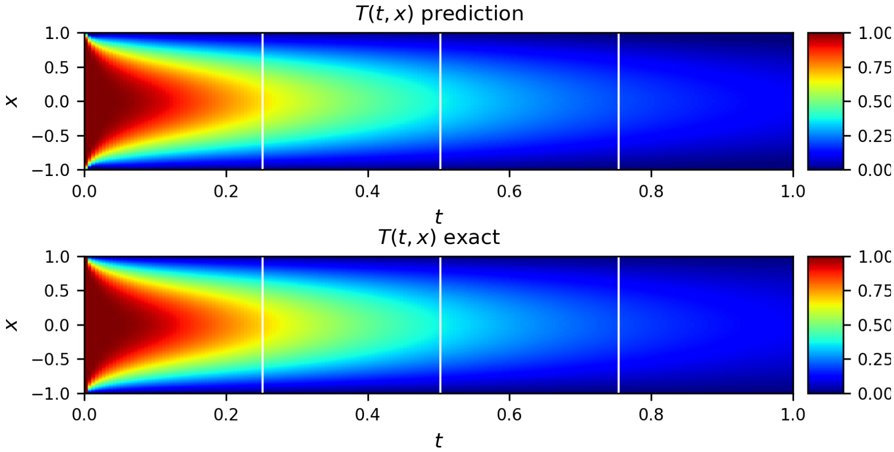
Randomly select N known points;
Compute the exact value of these points by using α;
Randomly select N_f governing points (Update part of the dataset);
estimate initial temperature;
for *Iterate for several times* **do**
    do
        Train the model;
    while *check if the loss becomes stable*;
    Randomly select N_f new known points;
end
Calculate the prediction error of α from the model's parameter.



a) $\alpha = 0.2$

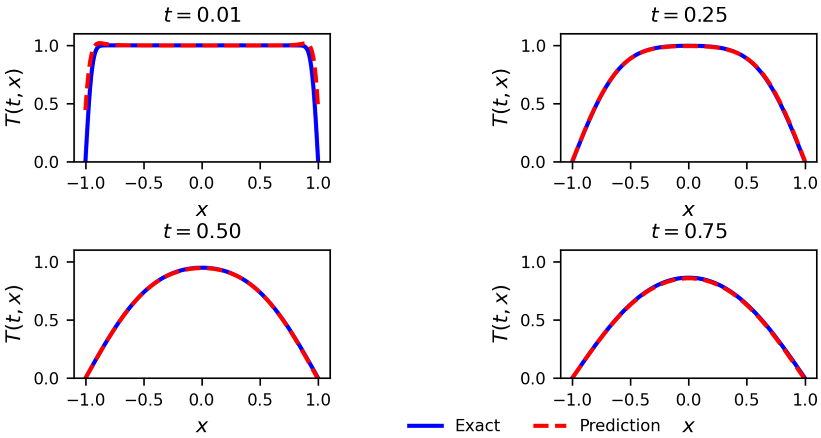


b) $\alpha = 0.6$

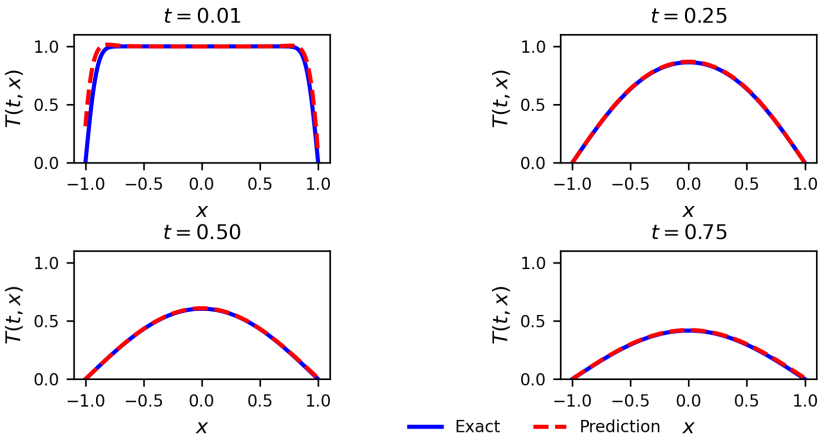


c) $\alpha = 1.0$

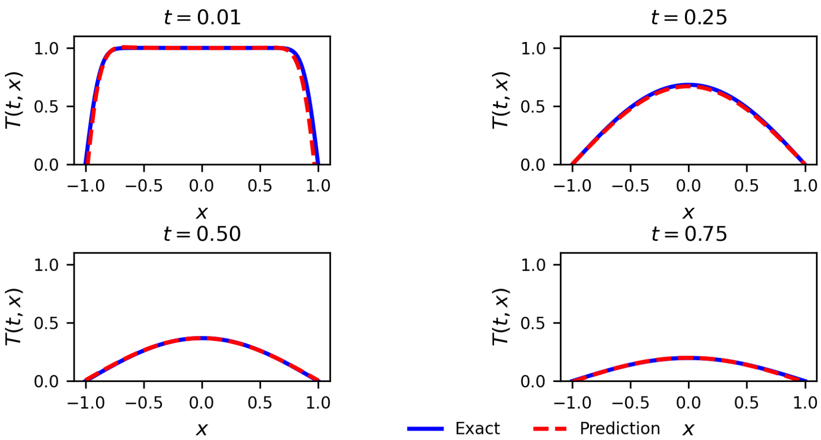
Figure 3–10 Comparison between prediction and exact value of 1d transient heat conduction with thermal diffusivity



a) $\alpha = 0.2$



b) $\alpha = 0.6$



c) $\alpha = 1.0$

Figure 3–11 Comparison between prediction and exact value of 1d transient heat conduction with thermal diffusivity by sampling at the white line of Figure 3–10

in Figure 3–12. Since during each combination, the so-called known points are given randomly and will not be changed during the whole training process, the randomness of these three values is high. However, there are still some regulated factor,

1. If the loss is high, then the error of the predicted α will also be high.
2. The optimizing with high loss normally has a short training time.
3. A low loss does not necessary mean a low predicted α error.
4. With more known points, the predicted α error will decrease. However, the decrease is expected to reach a plateau when sufficient number of points are given.
5. α near 0.8 tends to have a higher probability with lower prediction error.

The case with noise imposed on the given temperature has also been done. In this case, the known points number N is set to be a fixed value as 50. The training process is illustrated in Algorithm 3–2

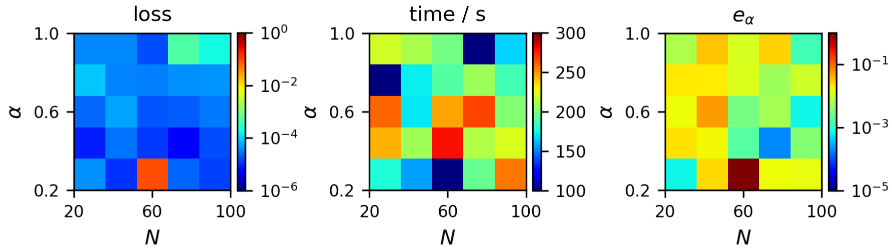
Algorithm 3–2 Training process of the 1D transient heat conduction inverse problem with noisy

Data: Noisy, governing points number N_f, preset thermal diffusivity α
Result: Optimized PINNs model

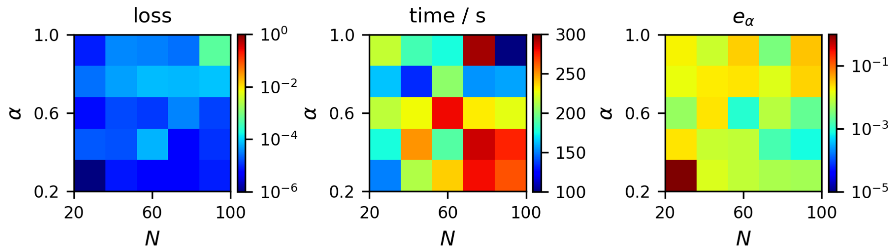
Randomly select N known points;
Compute the exact value of these points by using α;
Impose a noise on the value;
Randomly select N_f governing points (Update part of the dataset);
estimate initial temperature;
for *Iterate for several times* **do**
    do
        Train the model;
    while *check if the loss becomes stable*;
    Randomly select N_f new known points;
end
Calculate the prediction error of α from the model's parameter.

In Figure 3–13, the relationship between different factor are shown. We can find out that

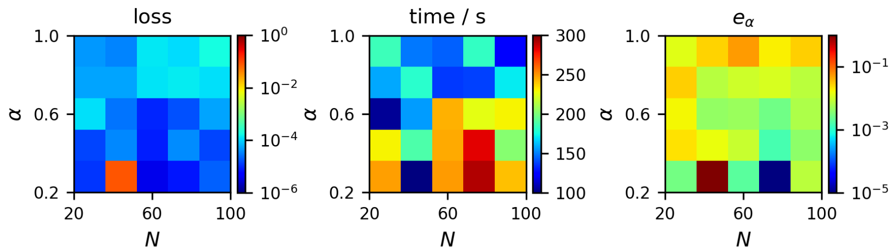
1. Even the noise is as large as 20%, the error of predicted thermal diffusivity can be still under 1%.
2. the computational cost for inverse problem is considerably lower than the forward problem, with only a few minutes.
3. The loss of the networks is proportional to the noise level.
4. Usually, a longer training time leads to a lower error of thermal diffusivity.
5. Most of the error shown in Figure 3–13 is below 1%, which is far more better than He et al.[10]'s results, which is normally beyond 1%, with 1.0575%, 1.3809%, 1.6014% for noise with 5%, 10% and 15% respectively.



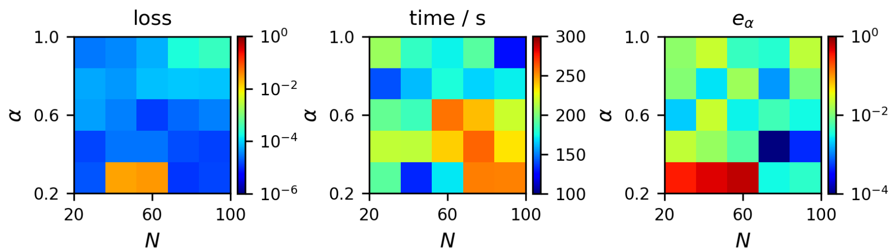
a) Trial 1



b) Trial 2



c) Trial 3



d) Mean value of the three trials

Figure 3–12 Final loss value, training time and error of the predicted α vs. different preset α and given data points N

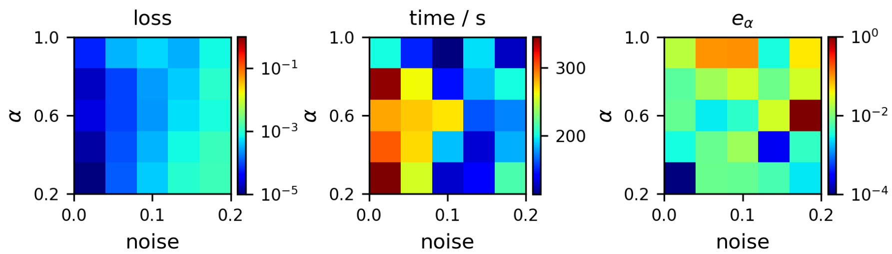


Figure 3–13 Final loss value, training time and error of the predicted α vs. different preset α and noise value

Chapter 4 Conclusion

In this report, 2D transient heat conduction, 2D transient heat convection, 1D transient heat conduction with thermal diffusivity as variable of the network, and the inverse problem of 1D transient heat conduction are solved. Prediction value and exact value are schematized with heatmap or line chart and are discussed in details. The error for either forward problem or inverse problem is low. Furthermore, the error of the inverse problem can be even lower than the external noise level, which shows the PINNs' high potential in solving these kinds of problems.

Bibliography

[1] TUNYASUVUNAKOOL K, ADLER J, WU Z, et al. Highly accurate protein structure prediction for the human proteome[J/OL]. Nature, 2021, 596(7873): 590-596 [2022-05-30]. https://www.nature.com/articles/s41586-021-03828-1. DOI: 10.1038/s41586-021-03828-1.

[2] RIBEIRO M D, REHMAN A, AHMED S, et al. DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks[R/OL]. arXiv. 2021 [2022-05-30]. http://arxiv.org/abs/2004.08826.

[3] LI H, REN T, LIU X, et al. U-Net applied to retrieve two-dimensional temperature and CO 2 concentration fields of laminar diffusion flames[J]. Fuel, 2022, 324: 124447. DOI: 10.1016/j.fuel.2022.124447.

[4] ZHAO Z, FENGXIN X, REN T, et al. Atmospheric CO2 retrieval from satellite spectral measurements by a two-step machine learning approach[J]. Journal of Quantitative Spectroscopy and Radiative Transfer, 2021: 108006. DOI: 10.1016/j.jqsrt.2021.108006.

[5] REN T, LI H, MODEST M, et al. Machine learning applied to the retrieval of three-dimensional scalar fields of laminar flames from hyperspectral measurements[J]. Journal of Quantitative Spectroscopy and Radiative Transfer, 2021, 279: 108047. DOI: 10.1016/j.jqsrt.2021.108047.

[6] REN T. Lecture 10: The Machine Learning Method[Z]. Numerical Methods in Low Carbon Energy Processing. Shanghai Jiao Tong University, 20220.

[7] RAISSI M, PERDIKARIS P, KARNIADAKIS G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J/OL]. Journal of Computational Physics, 2019, 378: 686-707 [2022-05-12]. https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125. DOI: 10.1016/j.jcp.2018.10.045.

[8] Jousef Murad. Physics-Informed Neural Networks (PINNs) - Chris Rackauckas | Podcast #42[EB/OL]. 2021 [2022-05-30]. https://www.youtube.com/watch?v=OmySUTFwh2g.

[9] Inverse problem[EB/OL]. 2022 [2022-05-30]. https://en.wikipedia.org/w/index.php?title=Inverse_problem&oldid=1088266604.

[10] HE Z, NI F, WANG W, et al. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials[J/OL]. Materials Today Communications, 2021, 28: 102719 [2022-05-19]. https://linkinghub.elsevier.com/retrieve/pii/S235249282100711X. DOI: 10.1016/j.mtcomm.2021.102719.