Finding Models of Heat Conduction via Machine Learning

Jin Zhao[a], Weifeng Zhao[b], Zhiting Ma[a], Wen-An Yong[c,*], Bin Dong[d,*]

[a]*School of Mathematical Sciences, Peking University, Beijing, 100871, China*
[b]*Department of Applied mathematics, University of Science and Technology Beijing, Beijing 100083, China*
[c]*Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China*
[d]*Beijing International Center for Mathematical Research & Institute for Artificial Intelligence, Peking University, Beijing, 100871, China*

Abstract

In this paper, we develop a method for finding models of heat conduction via machine learning. Integrating machine learning and the conservation-dissipation formulism (CDF) of irreversible thermodynamics, we obtain a system of PDEs for the heat conduction. The learned PDEs satisfy the conservation-dissipation principle and thereby are hyperbolic balance laws, which can be solved by conventional numerical methods. In the training process, we use a "warm-up" technique and train a neural network by another trained neural network. Numerical tests show that the learned models can achieve very high accuracy, perform well in a long time for a wide range of Knudsen numbers, and have an excellent generalization ability. The code is available at https://github.com/JinSZhao/Finding-Models-of-Heat-Conduction-via-Machine-Learning.

Keywords: Heat conduction, Machine learning, CDF, Warm-up, Discontinuous initial data, Hyperbolic balance laws.

1. Introduction

Heat conduction, one of the basic modes of thermal energy transport, is widely involved in the design and implementation of industrial heat exchangers. Traditionally, it is described by Fourier's law (called Fourier heat conduction), which states that the heat flux is proportional to the negative gradient of the temperature and leads to a parabolic heat equation [1]. The law is valid only in diffusive regimes where the intrinsic phonon resistive scattering dominates [2]. However, with the decreasing of the geometrical size of the electronic products and the gradual increase of packing integration, the classical law is inadequate to give an accurate description and understanding of heat transfer due to the sub-continuum effects and is no longer applicable at micro/nano scales [3, 4, 5, 6, 7, 8, 9, 10].

Up to now, various models and theories for non-Fourier heat conduction have been developed, such as the Maxwell-Cattaneo theory [11, 12] and the Guyer-Krumhansl theory [13] (see [14] for more models and theories). Deriving these models traditionally relies on the thermodynamics, a theory studying relationships among various apparently unrelated variables or parameters characterizing a thermodynamic system [15]. There have been many "schools" in this field [16], such as Classical Irreversible Thermodynamics (CIT) [17], Rational Extended Thermodynamics (RET) [18], Extended Irreversible Thermodynamics (EIT) [19], General Equation for Non-equilibrium Reversible-Irreversible Coupling (GENERIC) [20], and Conservation-dissipation Formalism (CDF) [21]. Based on the first and second laws of thermodynamics, these theories were proposed according to different physical hypotheses and mathematical simplifications. Nowadays, huge amounts of data can be easily collected and efficiently stored. Such vast quantity of data offers new opportunities for data-driven discovery of models of heat conduction [22].

Machine learning, a part of artificial intelligence, not only succeeds remarkably in computer vision, natural language processing and related areas, but also has widely been used in the field of scientific computing and modeling [23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. Similar to the conventional numerical methods, machine learning can approximate the solutions to differential equations with high accuracy. The well-known examples include the deep

*Corresponding author
Email addresses: zjin@pku.edu.cn (Jin Zhao), wfzhao@ustb.edu.cn (Weifeng Zhao), mazt@math.pku.edu.cn (Zhiting Ma), wayong@tsinghua.edu.cn (Wen-An Yong), dongbin@math.pku.edu.cn (Bin Dong)

Ritz method (DRM) [33], deep Galerkin method (DGM) [34], neural ordinary differential equations (NODE) [35], physics-informed neural networks (PINNs) [31] and so on. On the other hand, different from the traditional methods to solve inverse problems, machine learning can base datasets to learn unknown parameters or uncover the forms of the partial differential equations (PDEs), for example, PDE-Net 1.0 [36], PDE-Net 2.0 [23], and sparse identification of nonlinear dynamics (SINDy) [29]. The applications of machine learning to PDEs have been growing vigorously.

In this paper, we propose a method for finding models of heat conduction via machine learning with the CDF theory [21]. The theory is based on the first and second laws of thermodynamics, and assumes that certain conservation laws are known as a priori. Integrating machine learning and the irreversible thermodynamics, we obtain a system of interpretable and stable PDEs governing the non-equilibrium thermodynamic process, which further can be solved by traditional numerical methods. Furthermore, the learned models are hyperbolic balance laws, which naturally meet the four fundamental physical requirements proposed in [15]: the observability of physical phenomena, time-irreversibility, long-time tendency to equilibrium, and compatibility with existing classical theories.

Specifically, we consider one-dimensional heat conduction in the present work. Following CDF, we introduce one more variable to obtain a hyperbolic system of first-order PDEs with two unknown variables (conservative and dissipative variable). This process contains some freedoms which will be represented by two fully connected neural networks. On the other hand, we assume that the heat conduction obeys the mesoscopic phonon Boltzmann transport equation (BTE) in the kinetic theory, while the Bhatnagar-Gross-Krook (BGK) equation is considered in [37]. The training data are generated by solving the BTE numerically with smooth initial data. The related loss function is based on a discrete version of the CDF-based PDEs and involves two unknown functions (outputs). Unlike the loss function introduced in [37], ours does not involve the product of the two outputs, which is expected to produce more accurate results. Moreover, the "warm-up" technique [36] is used in our training process.

Some features and advantages of our learned models are verified by a number of numerical tests. Firstly, numerical results indicate that the learned model can achieve a very high accuracy with the initial data distinct from any of the training data. Secondly, the learned models can perform well in a long time, which exceeds the time of the training data. The learned models satisfy the conservation-dissipation principle automatically, and thereby are symmetrizable hyperbolic, which guarantees the long-time stability. Thirdly, the learned models can obtain satisfactory results with discontinuous initial data, although all the models are trained only with smooth initial data. Furthermore, we discuss relations between the dissipative variable and the heat flux, and show, from the numerical point of view, that the dissipative variable plays an important role in the heat conduction process.

The paper is organized as follows. The model is introduced in Section 2. The details are presented in Section 3, including the methodology of deriving the learned model and the related algorithm. We report a number of numerical results and give some discussions in Section 4. Finally, conclusions and remarks are given in Section 5.

2. Model

Heat conduction in rigid bodies obeys the conservation law of energy:

$$\partial_t u + \nabla \cdot \mathbf{q} = 0. \tag{2.1}$$

Here u is the internal energy, and $\mathbf{q}$ is the corresponding heat flux. When the heat flux obeys Fourier's law, i.e., $\mathbf{q} = -\lambda \nabla \theta$, then the equation (2.1) becomes a parabolic heat equation. However, with the rapid development of micro- and nano-electronics in recent years, the classical law fails in many fields due to the strong non-equilibrium effects [19], for example, the extremely-small heat transport [3, 38]. To overcome these defects, many methodologies have emerged, such as Classical Irreversible Thermodynamics (CIT) [17], Extended Irreversible Thermodynamics (EIT) [19], General Equation for Non-equilibrium Reversible-Irreversible Coupling (GENERIC) and Conservation-Dissipation Formalism (CDF). One can see [16] and references therein.

The CDF is a systematic methodology for mathematical modeling of irreversible phenomena [21]. It bases the first and second laws of thermodynamics to choose suitable non-equilibrium or dissipative variables. Specifically, for the heat conduction, CDF introduces a non-equilibrium variable $\mathbf{w}$ with the size of $\mathbf{q}$, which will be determined later. The whole state space is now given by the state variables $(u, \mathbf{w})$. Next, we turn to show how to generate reasonable training data for the heat conduction.

For simplicity, we only consider the one-dimensional case. Moreover, we assume that the heat conduction obeys the mesoscopic phonon Boltzmann transport equation (BTE) in kinetic theory. In the present work we are concerned about the gray model with the Debye's linear dispersion relation. The gray model employs a single phonon group speed in all directions and a single relaxation time independent of polarization and frequency. Despite the simple formulation, the model can provide some insightful predictions on the phonon transport behaviors with acceptable accuracy [39, 40], and we refer to [41] for more models. The BTE here reads as:

$$\partial_t e + v\partial_x e = -\frac{1}{\epsilon^2}(e - e^{eq}), \tag{2.2}$$

where $e = e(t, x, \mu)$ is the energy density distribution, $v = v_g\mu/\epsilon$ is the group velocity with v_g the magnitude and $\mu \in [-1, 1]$ the cosine of the angle between the phonon velocity and the positive x-axis, ϵ is the Knudsen number and

$$e^{eq} = \frac{u}{2}. \tag{2.3}$$

The total phonon energy u and heat flux q are determined by e as

$$u = \int_{-1}^{1} e(t, x, \mu)d\mu, \qquad q = \int_{-1}^{1} ve(t, x, \mu)d\mu. \tag{2.4}$$

We assume the system have a following entropy function

$$s = s(u, w; \epsilon) = s^{eq}(u) + s^{neq}(w; \epsilon). \tag{2.5}$$

Here the entropy non-equilibrium part $s^{neq}(w; \epsilon)$ is a concave function of w with the Knudsen number ϵ as a parameter. The equilibrium part has a specific entropy defined as

$$s^{eq} = s^{eq}(u) = \frac{1}{2}\int_{-1}^{1} \ln e^{eq} d\mu = \ln\frac{u}{2}. \tag{2.6}$$

Moreover, we refer to equation (2.1) and the generalized Gibbs relation

$$\mathrm{d}s = s_u \mathrm{d}u + s_w \mathrm{d}w,$$

and deduce the evolution of the entropy:

$$\begin{aligned}
\partial_t s &= s_u\partial_t u + s_w\partial_t w \\
&= -s_u\partial_x q + s_w\partial_t w \\
&= -\partial_x(s_u q) + q\partial_x s_u + s_w\partial_t w \\
&= -\partial_x(\theta^{-1}q) + q\partial_x\theta^{-1} + s_w\partial_t w
\end{aligned}$$

with $\theta^{-1} := s_u$. Here we have used

$$s_u = s_u^{eq} = \theta^{-1}.$$

CDF suggests that $\theta^{-1}q$ is the entropy flux and $q\partial_x\theta^{-1} + s_w\partial_t w$ is the entropy production.

According to the calculation above, we choose the heat flux

$$q = s_w = s_w^{neq}(w; \epsilon), \tag{2.7}$$

and the evolution equation for w as

$$\partial_t w + \partial_x\theta^{-1} = Mq \tag{2.8}$$

with $M = M(u, w; \epsilon)$ a positive function.

Therefore, we obtain a closed systems as following

$$\begin{aligned}
&\partial_t u + \partial_x q = 0, \\
&\partial_t w + \partial_x\theta^{-1} = Mq.
\end{aligned} \tag{2.9}$$

Here M is the undetermined freedom, and we will handle it in the next section. It is important to point out that this system satisfies the conservation-dissipation principle [42] and thereby is globally symmetrizable hyperbolic, and so does the multi-dimensional case.

3. Leaning Freedoms via DNNs & Prediction

3.1. Learning the Freedoms via Machine Learning

In this section, we will show how to learn the freedoms in equation (2.9) via machine learning. Since now, we can solve the equation (2.3) numerically, and thus have the data of the u and q from (2.4), however, we still have nothing about the dissipation variable, w. Therefore, it is scarcely possible to estimate M directly through the equation (2.9).

Noticing (2.7), we can multiply the first term of the second equation in (2.9) by s_{ww}/s_{ww} to obtain

$$\begin{aligned} &\partial_t u + \partial_x q = 0, \\ &g\partial_t q + \partial_x \theta^{-1} = Mq, \end{aligned} \tag{3.1}$$

where $g := 1/s_{ww}^{neq}(w;\epsilon) < 0$. For smooth solutions, the system (2.9) is equivalent to the balance laws (3.1). Recall that the non-equilibrium variable w can be globally expressed in terms of q due to the strict concavity of $s^{neq}(w;\epsilon)$ with respect to w. Then g can be viewed as a function of q. Therefore, our task becomes to set two fully connected neural networks, NN_g and NN_M, to learn the negative function $g = g(q;\epsilon)$ and the positive function $M = M(u,q;\epsilon)$, respectively.

Since the training data are known only at discrete space-time points, it is a common practice to replace or approximate the PDEs with their discrete versions [43]. Thus, the second equation in (3.1) can be written as the following discrete version:

$$q_j^{n+1} = \left[(gq)_j^n - \frac{\Delta t}{2\Delta x}\left((\theta^{-1})_{j+1}^n - (\theta^{-1})_{j-1}^n\right)\right] / \left(g_j^n - \Delta t M_j^n\right), \tag{3.2}$$

where the indices j and n together denote the space-time point $(n\Delta t, j\Delta x)$. We design two fully connected neural networks to approximate g and M, respectively. By writing the last equation in the abstract form

$$q_j^{n+1} = \mathcal{S}[g,M](V_{j-1}^n, V_j^n, V_{j+1}^n; \Delta t, \Delta x) \tag{3.3}$$

with $V_j^n = (u_j^n, q_j^n)$, we define our loss function as the mean squared error (MSE):

$$\mathcal{L}_1 = \sum_{\text{training data}} |q_j^{n+1} - \mathcal{S}[g,M](V_{j-1}^n, V_j^n, V_{j+1}^n; \Delta t, \Delta x)|^2. \tag{3.4}$$

Noting that the above loss function only involves single-step time information, we can design a loss function consisting of multi-step time information:

$$\mathcal{L}_k = \sum_{\text{training data}} |q_j^{n+k} - \mathcal{S}[g,M](\tilde{V}_{j-1}^{n+k-1}, \tilde{V}_j^{n+k-1}, \tilde{V}_{j+1}^{n+k-1}; \Delta t, \Delta x)|^2, \quad k \geq 2,$$

where $\tilde{V}_j^{n+k-1} = (u_j^{n+k-1}, \tilde{q}_j^{n+k-1})$ with $\tilde{q}_j^{n+k-1} = \mathcal{S}[g,M](\tilde{V}_{j-1}^{n+k-2}, \tilde{V}_j^{n+k-2}, \tilde{V}_{j+1}^{n+k-2}; \Delta t, \Delta x)$ and $\tilde{V}_j^n \equiv V_j^n$. Therefore, we obtain the total loss function:

$$\mathcal{L}_K = \sum_{k=1}^{K} \mathcal{L}_k. \tag{3.5}$$

Remark 1. *At the stage of pre-training, we take $\mathcal{L}_1$ as the loss function, and after a certain number of iterations, then take $\mathcal{L}_K$.*

3.2. Prediction

In the last section, we have learned the acceptable neural networks, NN_g and NN_M, approximating g and M, respectively, and then we can obtain the internal energy u through solving the equation (2.9) or (3.1). It is trivial to solve the equation (3.1) using the traditional numerical methods. Here we present the procedure to solve the equation (2.9).

Due to the strict concavity of $s_w^{neq}(w;\epsilon)$ and (2.7), w can be solved as a function

$$w = F(q;\epsilon) \tag{3.6}$$

satisfying

$$q = s_w^{neq}(F(q;\epsilon);\epsilon).$$

From this, we have the relation

$$F_q(q;\epsilon) = \frac{1}{s_{ww}^{neq}(F(q;\epsilon);\epsilon)} = \frac{1}{s_{ww}^{neq}(w;\epsilon)} = g. \tag{3.7}$$

Thus, we set a fully connected neural network $\mathcal{F}(q;\epsilon)$ to approximate $F(q;\epsilon)$, use the automatic differentiation (AD) technique [44] to derivative it with respect to q, and define the loss function

$$\mathcal{L}_F = \sum_q \left| NN_g - \mathcal{F}_q(q;\epsilon) \right|^2, \tag{3.8}$$

where NN_g is the fully connected neural network approximating g. The procedure of learning $\mathcal{F}(q;\epsilon)$ and then solving the equation (2.9) is given in Algorithm 1.

Algorithm 1 Learn $\mathcal{F}(q;\epsilon)$ and Solve (2.9) with Learned Functions

Input: NN_g, NN_M, initial data $(u^0,\ q^0)$, target predicate N
Output: (u^N, w^N)

1: Learn $\mathcal{F}(q;\epsilon)$ with NN_g and AD technique
2: Initialize w^0 with $\mathcal{F}(q^0;\epsilon)$ due to (3.6)
3: **for** $n = 0 : N-1$ **do**
4: Update q^n with the bisection method based on $w^n = \mathcal{F}(q^n;\epsilon)$
5: Update M^n with (u^n, q^n) based on $M^n = NN_M(u^n, q^n;\epsilon)$
6: Update (u^{n+1}, w^{n+1}) by solving (2.9) numerically
7: **end for**
8: **Return** (u^N, w^N)

In the above Algorithm, we take the uniform sampling to train the neural network $\mathcal{F}(q;\epsilon)$, and in order to ensure $\mathcal{F}(0;\epsilon) = 0$, we impose $\mathcal{F}(q;\epsilon) = \mathcal{F}(q;\epsilon) - \mathcal{F}(0;\epsilon)$.

4. Numerical Results & Discussions

In this section, we report the results of a number of numerical experiments and give some discussions. We refer interested readers to our code: https://github.com/JinSZhao/Finding-Models-of-Heat-Conduction-via-Machine-Learni

4.1. Generating Training Data

In this part, we show how to generate the training data. To this end, we solve the phonon BTE (2.2) in the domain $(x,t) \in [-\pi,\pi]\times[0,1]$ with periodic boundary conditions and the initial data constructed below, and obtain the training data (u,q) from (2.4). Here we give the details of solving the phonon BTE numerically.

Setting $v = v_g\mu/\epsilon$ in (2.2) with v_g the magnitude and μ the cosine of the angle between the phonon velocity and the positive x-axis [40], we discretize $\mu \in [-1,1]$ into p discrete points, $\mu_1, \mu_2, ...\mu_p$, using the discrete-ordinates method (DOM) based on the Gauss-Legendre quadrature. Denoting $e_k = e(t,x,\mu_k)$, we have

$$u = \int_{-1}^{1} e(t,x,\mu)d\mu \approx \sum_{k=1}^{p} w_k e_k,$$
$$q = \int_{-1}^{1} ve(t,x,\mu)d\mu \approx \sum_{k=1}^{p} \frac{1}{\epsilon} w_k v_g \mu_k e_k,$$

where w_k are the weights of the Gauss-Legendre quadrature. The above integrations can be evaluated accurately if there are enough discrete points. In the computation, we take $p = 100$. Then the BTE (2.2) with discrete velocities can be expressed as

$$\partial_t e_k + \frac{1}{\epsilon} v_g \mu_k \partial_x e_k = -\frac{1}{\epsilon^2}[e_k - e^{eq}], \quad k = 1,2,...,p. \tag{4.1}$$

We further discretize (4.1) in time and space by using the third-order implicit-explicit (IMEX) Runge-Kutta method [45] and the fifth-order WENO scheme [46], respectively. Due to the diffusive scaling, an restriction of the IMEX method is that when ϵ is small, the time step Δt must be small too. However, this restriction is not too serious as we are only concerned with moderate and large ϵ. In this paper, we take $\Delta t = 0.5 \times \frac{\epsilon}{v_g}\Delta x$ and $\Delta x = 2\pi/160$. We select the training data with the time internal $1/100$, i.e., the training data are (u_j^n, q_j^n), $j = 1, \cdots, 160$, $n = 1, \cdots, 100$. Furthermore, we take the Knudsen number $\epsilon = 10^{-2},\ 10^{-1},\ 10^{0},\ 10^{1},\ 10^{2}$.

Motivated by [43, 37], we constructed the initial data as

$$e_{init} = \alpha\hat{e}(u_1) + (1-\alpha)\hat{e}(u_2). \tag{4.2}$$

Here $\hat{e}(u) = e^{eq} - \epsilon v_g \mu \partial_x e^{eq} = \frac{1}{2}u - \frac{1}{2}\epsilon v_g \mu \partial_x u$ is the first-order expansion of e, α is uniformly sampled from $[0, 1]$, and

$$u_i = u_{a,i}\sin(Ax + u_{\phi,i}) + u_{b,i}, \quad i = 1, 2, \tag{4.3}$$

where $u_{a,i}$ are sampled from $[0.1, 0.3]$, $u_{b,i}$ are sampled from $[0.4, 1.0]$, $A = 1$, and $u_{\phi,i}$ are sampled from $[0, 2\pi]$.

4.2. Training

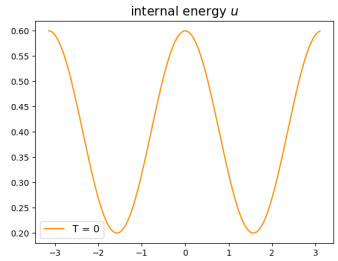
We set two fully connected neural networks, $NN_g(q)$ and $NN_M(u, q)$, to approximate the functions $g = g(q;\epsilon)$ and $M = (u, q;\epsilon)$ in (3.2) with the loss function (3.5), respectively. At the training beginning, we take $K = 1$ in the loss function (3.5) to "warm-up" the neural networks [36], and after a number of iterations, we fix $K = 4$. The both of neural networks have four hidden layers with 50 nodes in each layer. In order to ensure the positivity of M and $-g$, we take $softplus$ function as the activation function in the output player, while the *tanh* function is used in other layers. In the whole training process, we use back propagation (BP) with the stochastic gradient descent (SGD) algorithm [47] as the optimizer. The learning rate is chosen as $lr = 0.1$ at the outset and tuned up by $CosineAnnealingLR$ in the subsequent training.

We set two hidden layers with 30 nodes in each layer for the fully connected neural network, $\mathcal{F}(q;\epsilon)$, with the loss function (3.8), and choose *tanh* as the activation function in the hidden layers and the *identity* action function in the output layer. Since the AD technique is used in this training, we take $\mathcal{F}(q;\epsilon) - \mathcal{F}(0;\epsilon)$ as the approximation to $F(q;\epsilon)$. Additionally, due to the concavity of the entropy, the network $\mathcal{F}(q;\epsilon)$ is monotonous and thereby inverse, and thus, we can solve q through $w = \mathcal{F}(q;\epsilon)$ by the bisection method. The details are referred to Algorithm 1.

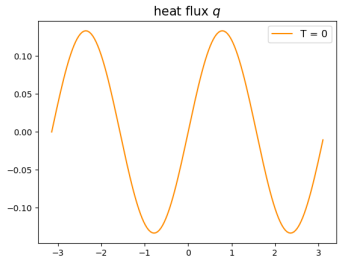
4.3. Numerical Results

To verify the performance of the learned model (2.9), we solve it by the Lax-Friedrichs scheme, and make a comparison between them and the numerical results of the phonon BTE (2.2). The initial data of these solutions are randomly sampled, which are not in the set of the training data.

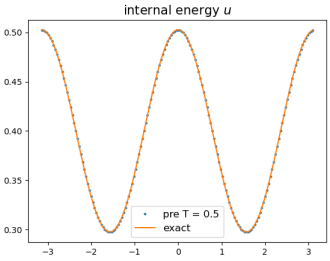
As the first experiment, we consider the cases with the period π and $\frac{2\pi}{3}$ ($A = 2$ and $A = 3$ in (4.3)), while the training data only consist of the cases with the period 2π ($A \equiv 1$ in (4.3)). We plot the related solution profiles of the internal energy, u, and the heat flux, q, obtained from the phonon BTE (exact solutions) and the learned model at $T = 0.5$ and $T = 1$ in Fig. 1 - 4 with different Knudsen numbers, $\epsilon = 10^{-2},\ 10^{-1},\ 10^{1},\ 10^{2}$. These figures indicate that the predicted results are consistent with the exact ones very well.



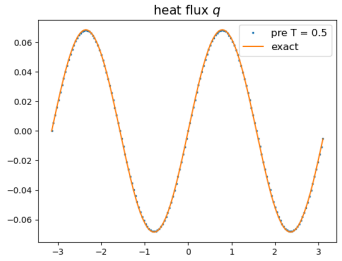
(a) initial data u



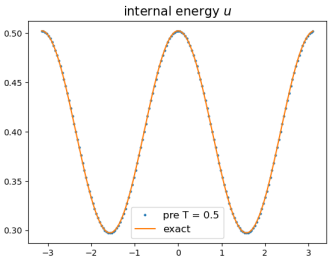
(b) initial data q



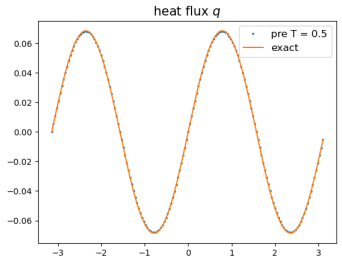
(c) $\epsilon = 10^{-2}$



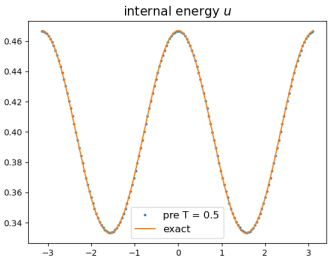
(d) $\epsilon = 10^{-2}$



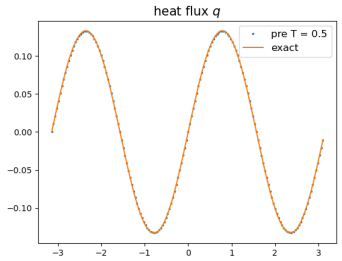
(e) $\epsilon = 10^{-1}$



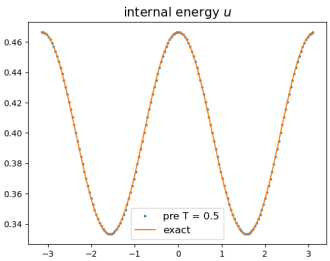
(f) $\epsilon = 10^{-1}$



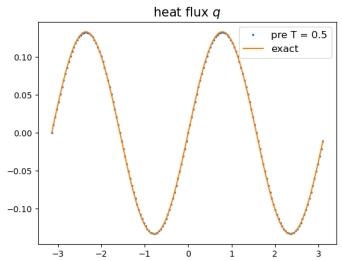
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$

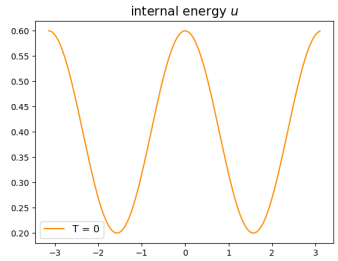


(i) $\epsilon = 10^{2}$

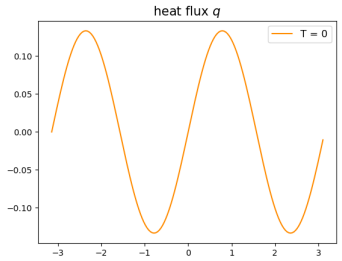


(j) $\epsilon = 10^{2}$

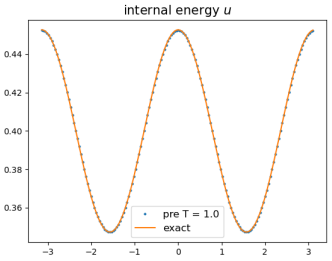
Figure 1: The results at $T = 0.5$ with the period π and different Knudsen numbers. Here "pre" represents predicting solution.



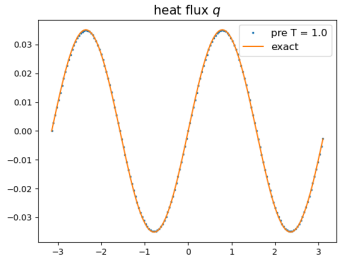
(a) initial data u



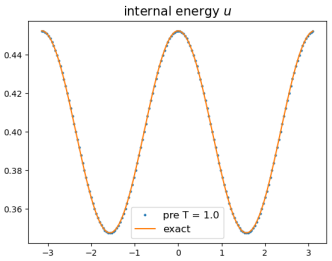
(b) initial data q



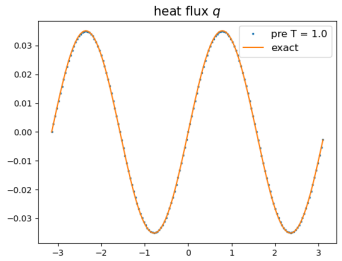
(c) $\epsilon = 10^{-2}$



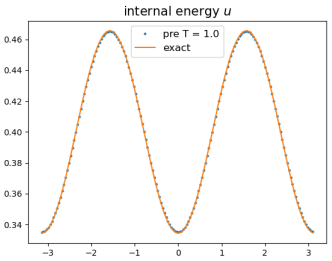
(d) $\epsilon = 10^{-2}$



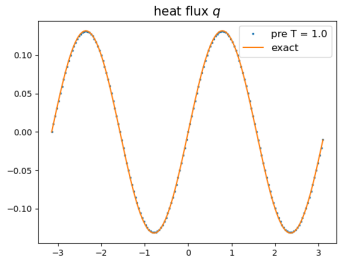
(e) $\epsilon = 10^{-1}$



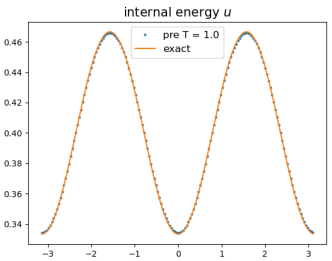
(f) $\epsilon = 10^{-1}$



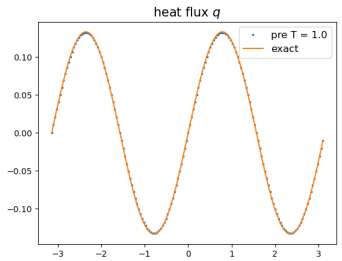
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$

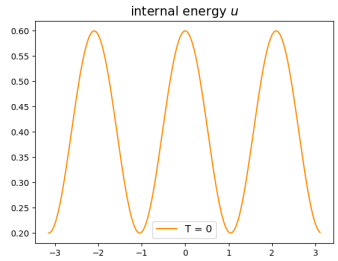


(i) $\epsilon = 10^{2}$

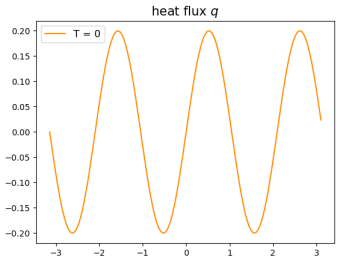


(j) $\epsilon = 10^{2}$

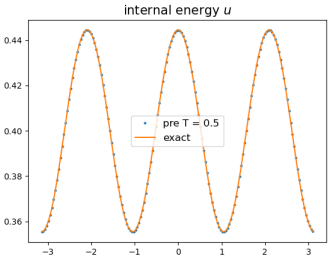
Figure 2: The results at $T = 1.0$ with the period π and different Knudsen numbers. Here "pre" represents predicting solution.



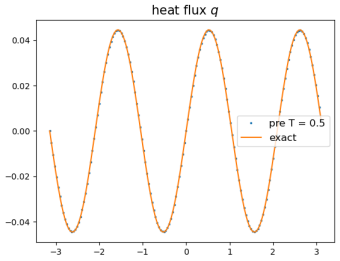
(a) initial data u



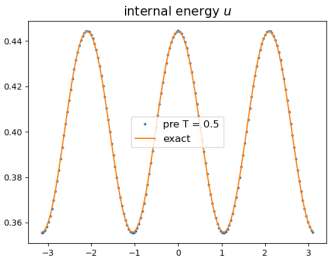
(b) initial data q



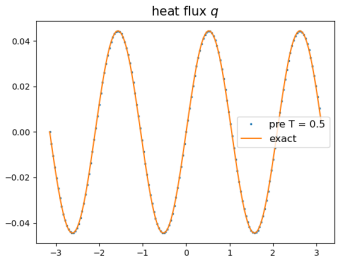
(c) $\epsilon = 10^{-2}$



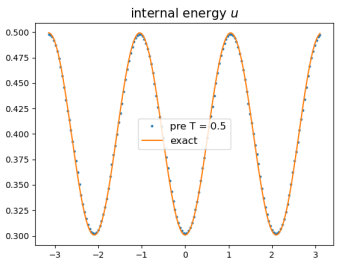
(d) $\epsilon = 10^{-2}$



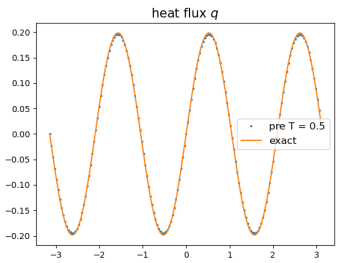
(e) $\epsilon = 10^{-1}$



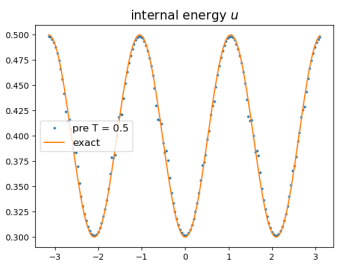
(f) $\epsilon = 10^{-1}$



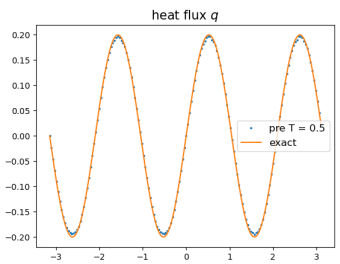
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$

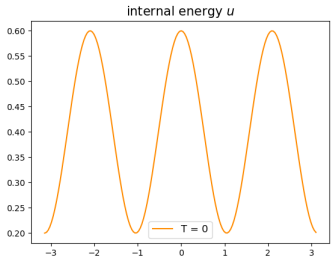


(i) $\epsilon = 10^{2}$

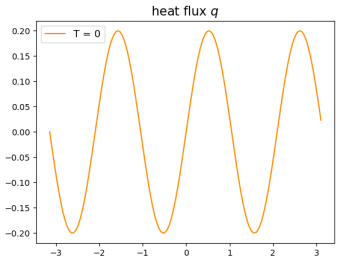


(j) $\epsilon = 10^{2}$

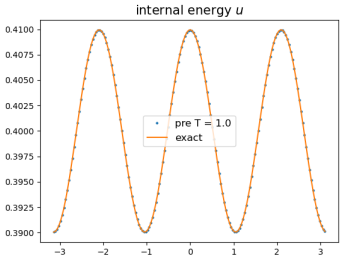
Figure 3: The results at $T = 0.5$ with the period $\frac{2\pi}{3}$ and different Knudsen numbers. Here "pre" represents predicting solution.



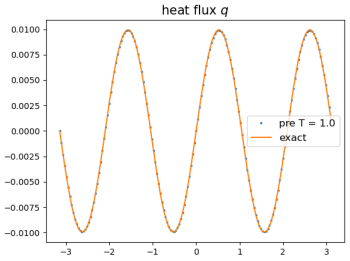
(a) initial data u



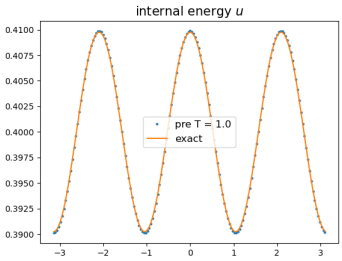
(b) initial data q



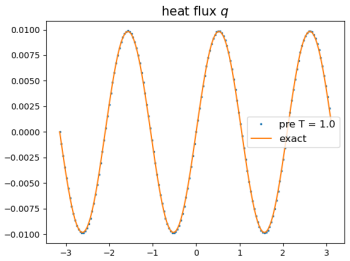
(c) $\epsilon = 10^{-2}$



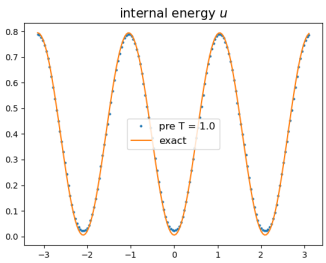
(d) $\epsilon = 10^{-2}$



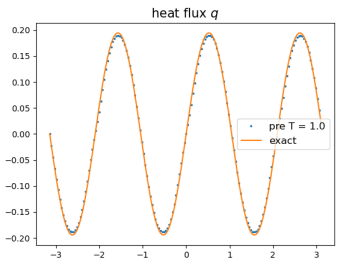
(e) $\epsilon = 10^{-1}$



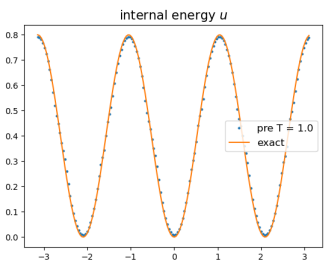
(f) $\epsilon = 10^{-1}$



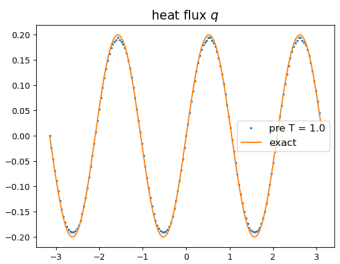
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$

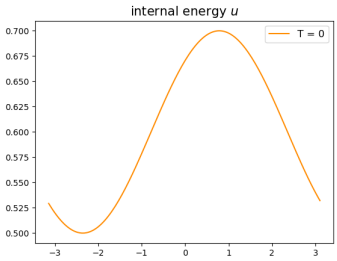


(i) $\epsilon = 10^{2}$

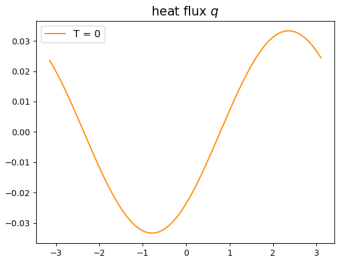


(j) $\epsilon = 10^{2}$

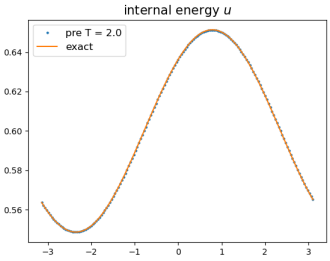
Figure 4: The results at $T = 1.0$ with the period $\frac{2\pi}{3}$ and different Knudsen numbers. Here "pre" represents predicting solution.



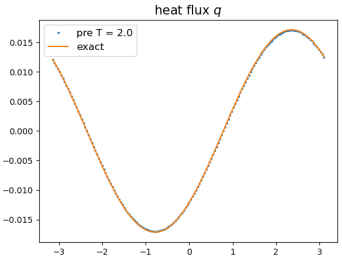
(a) initial data u



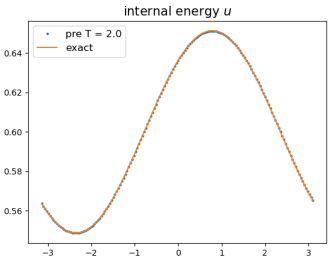
(b) initial data q



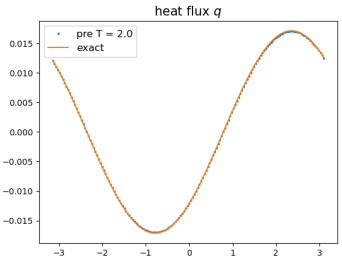
(c) $\epsilon = 10^{-2}$



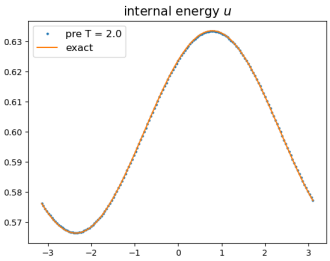
(d) $\epsilon = 10^{-2}$



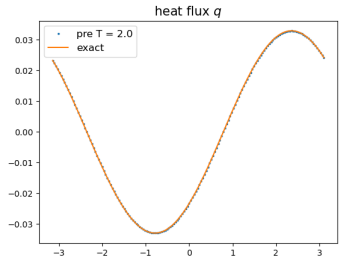
(e) $\epsilon = 10^{-1}$



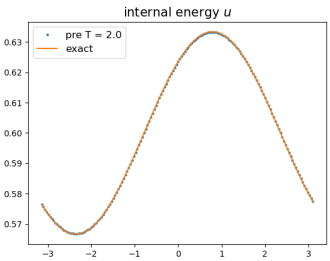
(f) $\epsilon = 10^{-1}$



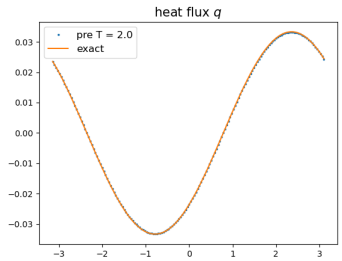
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$



(i) $\epsilon = 10^{2}$

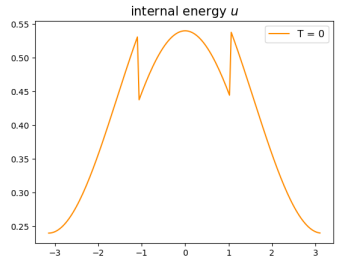


(j) $\epsilon = 10^{2}$

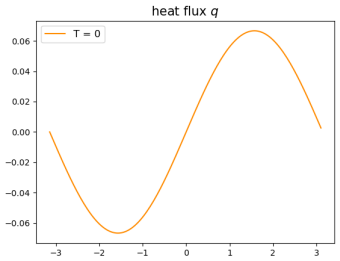
Figure 5: The results at $T = 2.0$ with different Knudsen numbers. Here "pre" represents predicting solution.

The second experiment will verify long-time behaviors of the learned models. Here we choose the example with $T = 2$ and draw the results in Fig 5. It should be pointed out that the time of the training data is limited to $T = 1$. The results indicate clearly that the learned model can predict long-time results very well. The learned model satisfies the conservation-dissipation principle automatically, and thereby is symmetrizable hyperbolic, which guarantees the long-time stability. Besides this, lots of numerical experiments with $T = 2$ (one can choose $T > 2$) are conducted, including the first and the later experiments, and they all have a good stability.

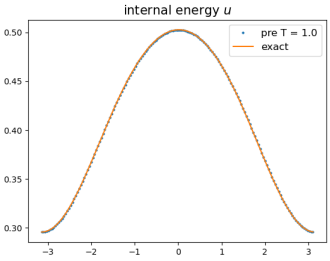
In the third experiment, we intend to explore the performance of the learned model with discontinuous initial data, although we only use the smooth training data to train the model. Fig. 6 displays the behaviors of the related numerical results with different Knudsen numbers. The results that the predicted solutions agree with the exact ones implies that the learned model is valid with discontinuous initial data.



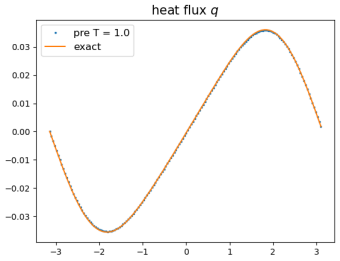
(a) initial data u



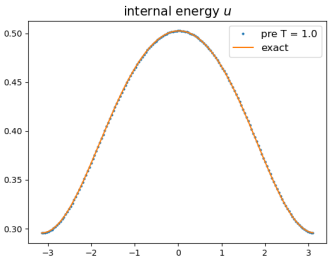
(b) initial data q



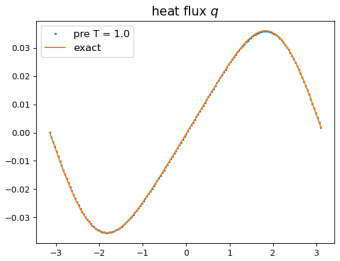
(c) $\epsilon = 10^{-2}$



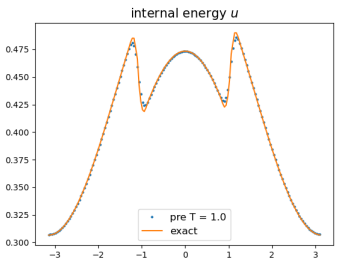
(d) $\epsilon = 10^{-2}$



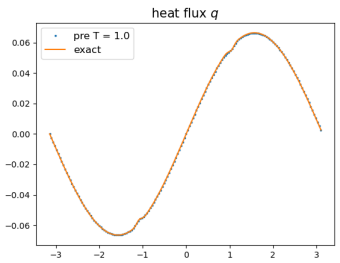
(e) $\epsilon = 10^{-1}$



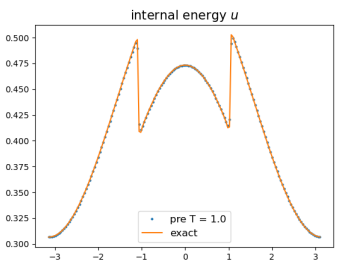
(f) $\epsilon = 10^{-1}$



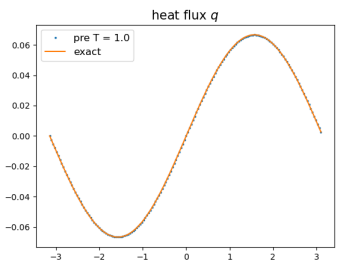
(g) $\epsilon = 10^{1}$



(h) $\epsilon = 10^{1}$



(i) $\epsilon = 10^{2}$



(j) $\epsilon = 10^{2}$

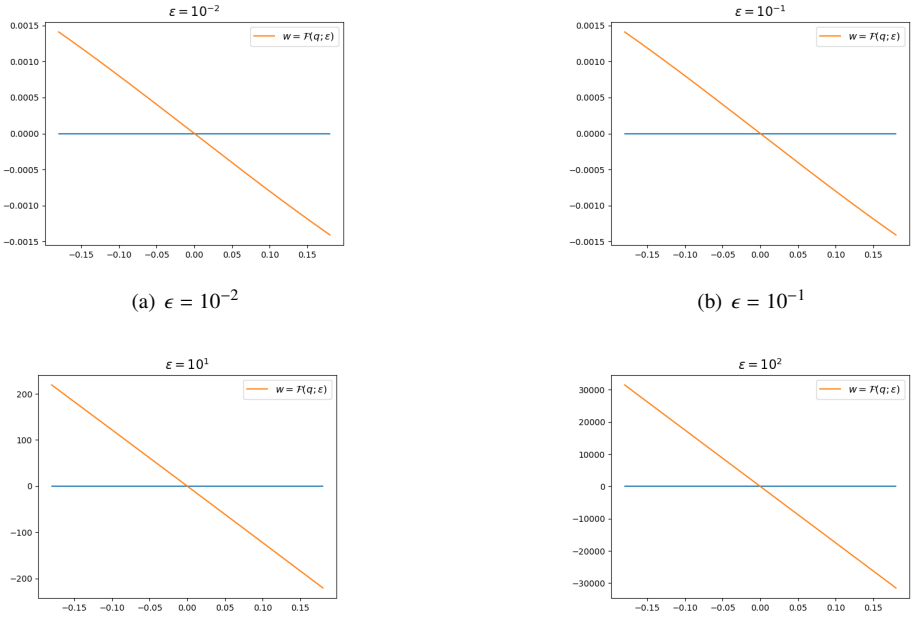
Figure 6: The results with respect to discontinuous initial data at $T = 1.0$. Here "pre" represents predicting solution.

4.4. Discussions

Besides the numerical tests above, we conduct a number of experiments to explore the features of the leaned-models. We can draw some conclusions and make useful discussions for the present work.

- $\mathcal{F}(q;\epsilon)$

We plot the profiles of the neural network $\mathcal{F}(q;\epsilon)$ with $\epsilon = 10^{-2}, 10^{-1}, 10^{1}, 10^{2}$ in Fig. 7. The four profiles suggest that w can be linear with respect to q. In this sense, we can solve the equation (3.1) with the constant g with respect to the fixed Knudsen number, ϵ. Only to obtain the conservative variable, solving (3.1) has an advantages over (2.9) in computing time. Furthermore, It is clear that g goes to zero as the Knudsen number deceases, which indicates that the second equation of (2.9) or (3.1) can be regarded as a generalization of Fourier's law.



(a) $\epsilon = 10^{-2}$ (b) $\epsilon = 10^{-1}$

(c) $\epsilon = 10^{1}$ (d) $\epsilon = 10^{2}$

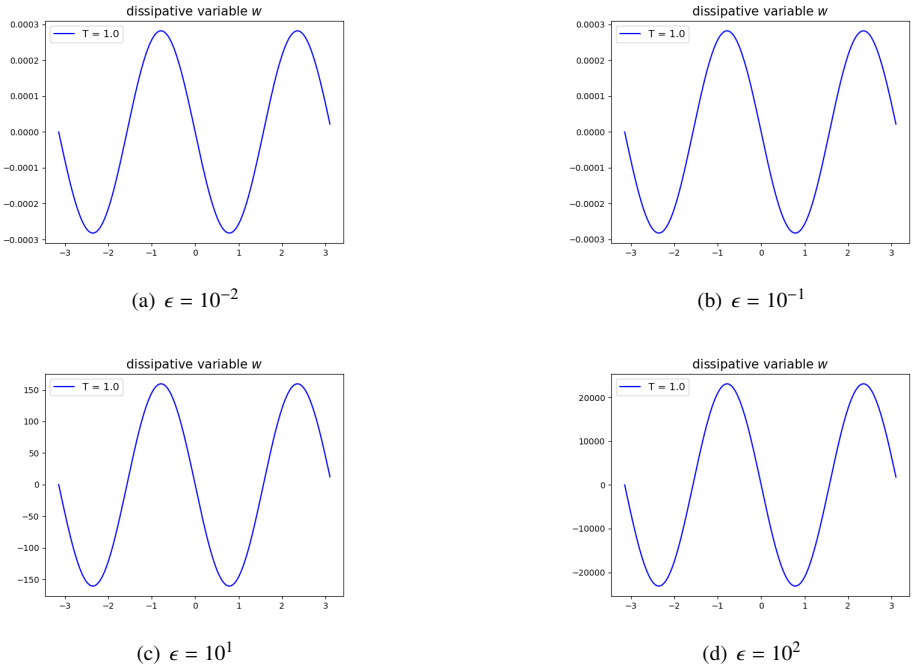
Figure 7: The profiles of $\mathcal{F}(q;\epsilon)$ with different Knudsen numbers, $\epsilon = 10^{-2}, 10^{-1}, 10^{1}, 10^{2}$.

- dissipative variable, w

The dissipative term, w, in the equation (2.8) can not be ignored to train the model with the large Knudsen number (here $\epsilon = 10^{1}, 10^{2}$). Namely, it is invalid that we apply the following model for all data:

$$\begin{aligned} \partial_t u + \partial_x q &= 0, \\ \partial_x \theta^{-1} &= Mq. \end{aligned}$$

This indicates apparently that Fourier's law is not applicable in this case. Actually, the dissipative variable plays a key role in the heat transfer process. To show this, we plot the behaviors of w corresponding to the first example (see Fig. 1) at $T = 1$ in Fig. 8. It is clear that the magnitude of w with $\epsilon = 10^{1}$ are much bigger than that with $\epsilon = 10^{-1}$, which agrees with the physical truth.



(a) $\epsilon = 10^{-2}$ (b) $\epsilon = 10^{-1}$

(c) $\epsilon = 10^{1}$ (d) $\epsilon = 10^{2}$

Figure 8: The profiles of w with $\epsilon = 10^{-2}, \epsilon = 10^{-1}, \epsilon = 10^{1}$ and $\epsilon = 10^{2}$ at $T = 1$.

In this paper, we show our idea of the whole project for modeling heat conduction with hyperbolic PDEs by introducing only one dissipative variable. The resultant model seems inadequate to describe non-Fourier heat conduction in ballistic regimes [48]. For the ballistic regime, two or more dissipative variables seem required, which is our future work. Indeed, it was pointed out in [49, 41] that two or more dissipative variables and high order derivatives should be introduced to cover both ballistic and diffusive regimes.

- boundary conditions

The present modeling is not limited by boundary conditions. To show this, we use the learned model (2.9) to simulate the quasi-1D steady cross-plane heat conduction with isothermal boundary conditions [50], where $u_L = 0.5$ and $u_R = 0.3$. In this situation, the model (2.9) reduces to the ordinary differential equation $\partial_x \theta^{-1} = Mq$. We adopt the boundary values solved by BTE [51]. The results with $Kn = 1e\{-2, -1, 1\}$ are plotted in Fig. 9, which shows that the solutions of the learned model are in a good agreement with those of the BTE.

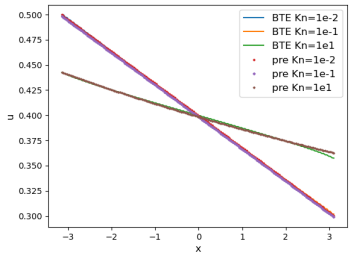


Figure 9: The profiles of internal energy u with $Kn = 1e\{-2, -1, 1\}$. Here 'pre' represents the solution of the learned model.

- $\epsilon = 10^{0}$

We here make a special efforts to investigate the performance of the examples with $\epsilon = 10^0$. To begin with, we draw the profiles of $\mathcal{F}(q;\epsilon)$ with $\epsilon = 10^0$ in Fig. 10, which shows a nonlinearity between w and q.

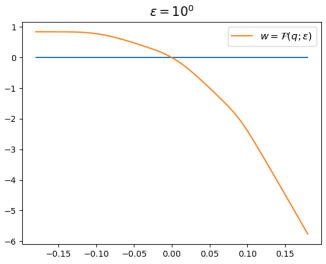
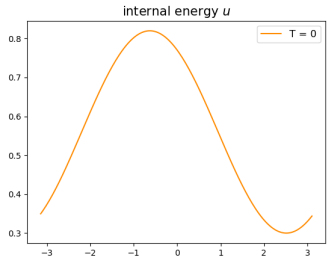
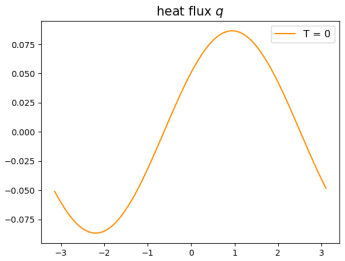


Figure 10: The profiles of $\mathcal{F}(q;10^0)$

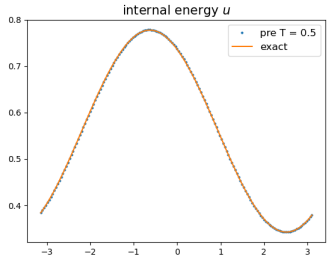
We first draw the profiles of solutions (u and q) to the learned model with the Knudsen number $\epsilon = 10^0$ at $T = 0.5,\ 1.0,\ 2.0$ in Fig. 11-12. In these figures, it is easy to see that the errors increase over time, which not occurs in the cases with $\epsilon = 10^{-2},\ 10^{-1},\ 10^1,\ 10^2$.



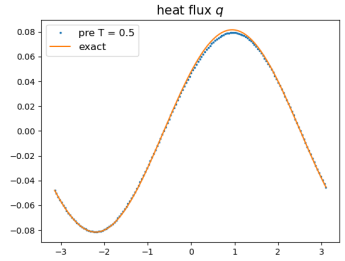
(a) initial data u



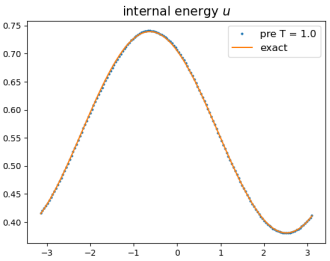
(b) initial data q



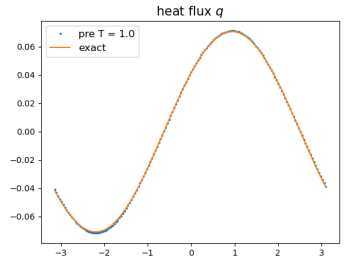
(c) $T = 0.5$



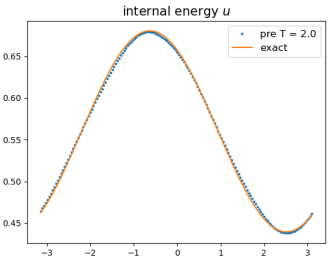
(d) $T = 0.5$



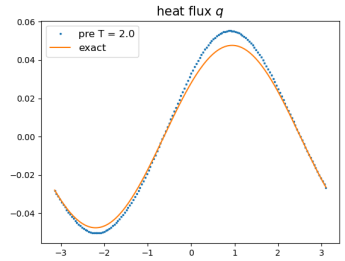
(e) $T = 1.0$



(f) $T = 1.0$

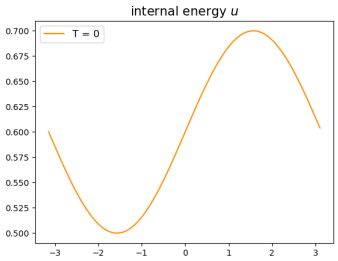


(g) $T = 2.0$

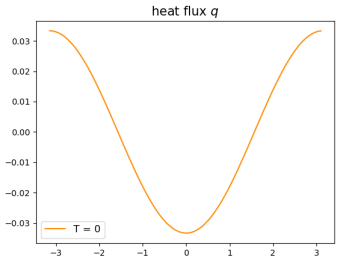


(h) $T = 2.0$

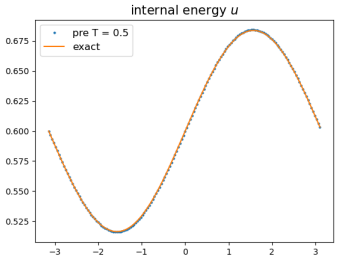
Figure 11: The profiles of u and q with $\epsilon = 10^0$ at $T = 0.5,\ 1.0,\ 2.0$. Here "pre" represents predicting solution.



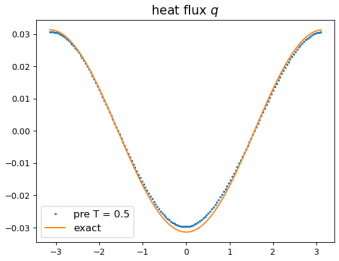
(a) initial data u



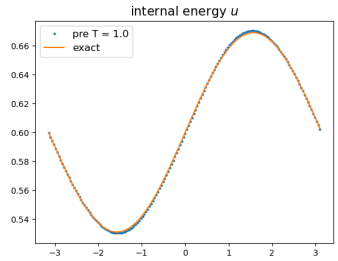
(b) initial data q



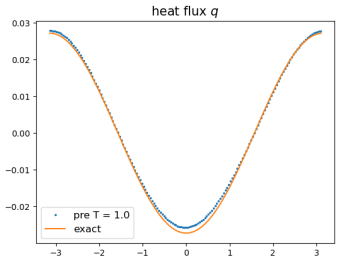
(c) $T = 0.5$



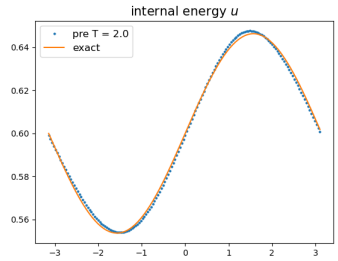
(d) $T = 0.5$



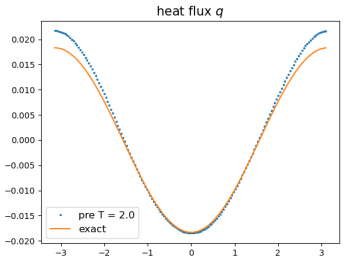
(e) $T = 1.0$



(f) $T = 1.0$



(g) $T = 2.0$



(h) $T = 2.0$

Figure 12: The profiles of u and q with $\epsilon = 10^0$ at $T = 0.5,\ 1.0,\ 2.0$. Here "pre" represents predicting solution.

On the other hand, to give a quantitative description, we define the relative L^1 and L^2 errors as

$$e_1 = \frac{\sum_j |u_j - \hat{u}_j|}{\sum_j |u_j|}$$

and

$$e_2 = \sqrt{\frac{\sum_j (u_j - \hat{u}_j)^2}{\sum_j (u_j)^2}}.$$

Here u is the solution generated by the BTE, $\hat{u}$ is solved from the learned model, and u_j and $\hat{u}_j$ are the values of u and $\hat{u}$ at j-th spatial grid point x_j and time $T = 1$ and $T = 2$. Table 1 and 2 show the relative errors and the standard derivation (std) of the internal energy u and heat flux q, respectively, corresponding to the initial data taken from 5 randomly different runs. The errors with respect to $\epsilon = 10^0$ are about 10 to 100 times bigger than others. Indeed, we can't improve the accuracy, no matter how we tune the hyperparameters of the neural networks. Maybe it is not sufficient to select one dissipative variable at the beginning of modeling in CDF theory. This motivates us to try other models through selecting more freedoms that are determined by machine learning in the forthcoming work, which is more difficult and challenging, and left for the future work.

Knudsen number ϵ	Relative L^1 error e_1 (std)		Relative L^2 error e_2 (std)	
	$T = 1$	$T = 2$	$T = 1$	$T = 2$
10^{-2}	$1.08e-4(3.31e-5)$	$1.45e-4(5.16e-5)$	$1.19e-4(3.34e-5)$	$1.60e-4(5.42e-5)$
10^{-1}	$9.48e-5(3.77e-5)$	$1.22e-4(5.81e-5)$	$1.09e-4(3.95e-5)$	$1.36e-4(6.22e-5)$
10^{0}	$1.00e-3(1.00e-4)$	$1.90e-3(2.16e-4)$	$1.21e-3(1.21e-4)$	$2.25e-3(2.15e-4)$
10^{1}	$1.50e-4(2.74e-5)$	$2.07e-4(4.59e-5)$	$1.68e-4(3.04e-5)$	$2.33e-4(5.17e-5)$
10^{2}	$1.69e-4(3.00e-5)$	$1.58e-4(2.79e-5)$	$1.87e-4(3.29e-5)$	$1.75e-4(3.12e-5)$

Table 1: The errors and std of u corresponding to the initial data taken from 5 randomly different runs.

Knudsen number ϵ	Relative L^1 error e_1 (std)		Relative L^2 error e_2 (std)	
	$T = 1$	$T = 2$	$T = 1$	$T = 2$
10^{-2}	$3.65e-3(5.28e-4)$	$4.53e-3(5.59e-3)$	$3.70e-3(4.54e-4)$	$4.56e-3(5.09e-4)$
10^{-1}	$4.11e-3(6.70e-4)$	$4.82e-4(6.04e-4)$	$4.15e-3(6.01e-4)$	$4.84e-3(5.61e-4)$
10^{0}	$2.99e-2(9.12e-3)$	$1.04e-1(8.83e-3)$	$3.29e-2(1.15e-2)$	$1.31e-1(4.63e-3)$
10^{1}	$2.06e-3(6.43e-5)$	$4.83e-4(8.46e-5)$	$2.08e-3(5.88e-5)$	$5.25e-4(9.38e-5)$
10^{2}	$1.95e-3(4.18e-7)$	$3.87e-3(1.65e-6)$	$1.97e-3(5.02e-7)$	$3.87e-3(1.62e-6)$

Table 2: The errors and std of q corresponding to the initial data taken from 5 randomly different runs.

5. Conclusions & Remarks

This paper is devoted to finding models of heat conduction via machine learning. Integrating the machine learning and the conservation-dissipation formulism (CDF) of irreversible thermodynamics, we obtain a system of PDEs for the heat conduction. The learned models satisfy the conservation-dissipation principle, and thereby are hyperbolic balance laws, which can be solved by conventional numerical methods. Lots of numerical tests for a wide range of Knudsen numbers ($\epsilon = 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}$) show that the learned models perform well in accuracy and long-time stability. Remarkably, they are still valid with discontinuous initial data, although the training data only consist of smooth initial data.

Through investigating the behaviors of the dissipative variable with respect to different Knudsen numbers, we find that the dissipative variable plays a key role in the heat transfer process, and by further comparison, it is verified that the Fourier's law is invalid for the case with large Knudsen number. In addition, the numerical results with $\epsilon = 10^0$ suggest that introducing more variables may benefit in the modeling, which is our forthcoming work.

As is known, the computable modeling process inevitably introduces errors due to the limitations of mathematical assumptions in model parameter calibration, while the modeling only based on data lacks robustness and interpretability. Integrating machine learning, physical hypotheses and mathematical simplifications exhibits lots of advantages

in the scientific computing and mathematical modeling. This paper can be regarded as the first step of the series of finding models for heat conduction via machine learning with the CDF theory. We only consider one-dimensional model in this paper, but the method is also applicable for the multi-dimensional case. In this situation, obtaining the conservative variable through solving (3.1) is much easier than (2.9). How to derive the conservative-schemes of the learned models is the key point in the multi-dimensional case, which is left for the future work.

Acknowledgements

This work is supported by the China Postdoctoral Science Foundation (No. 2020M680001).

References

[1] Z. Jiang, J. Zhao, and H. Xie. Scaling laws (chapter 3). In Zhengyi Jiang, Jingwei Zhao, and Haibo Xie, editors, *Microforming Technology*, pages 53–71. Academic Press, 2017.

[2] J. M. Ziman. *Electrons and Phonons: The Theory of Transport Phenomena in Solids*. Oxford Classic Texts in the Physical Sciences. Oxford University Press, USA, 1960.

[3] G. Chen. *Nanoscale Energy Transport and Conversion : A Parallel Treatment of Electrons, Molecules, Phonons, and Photons*. Oxford University Press, USA, 2005.

[4] S. Liu, P. Hänggi, N. Li, J. Ren, and B. Li. Anomalous heat diffusion. *Physical Review Letters*, 112(4):040601, 2014.

[5] Y. Zhang, A. Fan, M. An, W. Ma, and X. Zhang. Thermal transport characteristics of supported carbon nanotube: Molecular dynamics simulation and theoretical analysis. *International Journal of Heat and Mass Transfer*, 159:120111, 2020.

[6] D. G Cahill, W. K Ford, K. E Goodson, G. D Mahan, A. Majumdar, H. J Maris, R. Merlin, and S. R Phillpot. Nanoscale thermal transport. *Journal of Applied Physics*, 93(2):793–818, 2003.

[7] D. G Cahill, P. V Braun, G. Chen, D. R Clarke, S. Fan, K. E Goodson, P. Keblinski, W. P King, G. D Mahan, and A. Majumdar. Nanoscale thermal transport II. 2003–2012. *Applied Physics Reviews*, 1(1):011305, 2014.

[8] A. Beardo, M. Calvo-Schwarzwälder, J. Camacho, T.G. Myers, P. Torres, L. Sendra, F.X. Alvarez, and J. Bafaluy. Transport in compact and holey silicon thin films. *Physical Review Applied*, 11:034003, Mar 2019.

[9] P. Torres, A. Ziabari, A. Torelló, J. Bafaluy, J. Camacho, X. Cartoixà, A. Shakouri, and F. X. Alvarez. Emergence of hydrodynamic heat transport in semiconductors at the nanoscale. *Physical Review Materials*, 2:076001, Jul 2018.

[10] P. T. Alvarez. *Thermal Transport in Semiconductors: First Principles and Phonon Hydrodynamics*. Springer Theses, 2018.

[11] M. F. McCarthy. *Singular Surfaces and Waves*. In: A. C. Eringen, Ed., Continuum Physics, Vol. 2, Academic Press, 1972.

[12] P. J. Chen. *Growth and Decay of Waves in Solids*. Mechanics of Solids, 1973.

[13] R. A. Guyer and J. A. Krumhansl. Dispersion relation for second sound in solids. *Physical Review*, 133:A1411–A1417, Mar 1964.

[14] D. D. Joseph and L. Preziosi. Heat waves. *Reviews of Modern Physics*, 61(41-73), 1989.

[15] W.-A. Yong. Intrinsic properties of conservation-dissipation formalism of irreversible thermodynamics. *Philosophical Transactions of the Royal Society A*, 378(2170):20190177, 2020.

[16] W. Muschik. Why so many 'schools' of thermodynamics? *Forschung im Ingenieurwesen*, 71(3-4):149–161, 2007.

[17] S. R. De Groot and P. Mazur. *Non-equilibrium Thermodynamics*. Courier Corporation, 2013.

[18] I. Müller and T. Ruggeri. Rational Extended Thermodynamics. 1998.

[19] D. Jou, J. Casas-Vázquez, and G. Lebon. *Extended Irreversible Thermodynamics*. Springer, 1996.

[20] M. Pavelka, V. Klika, and M. Grmela. *Multiscale Thermo-dynamics: Introduction to GENERIC*. Walter de Gruyter GmbH & Co KG, 2018.

[21] Y. Zhu, L. Hong, Z. Yang, and W.-A. Yong. Conservation-dissipation formalism of irreversible thermodynamics. *Journal of Non-Equilibrium Thermodynamics*, 40(2):67–74, 2015.

[22] S. L Brunton and J N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[23] Z. Long, Y. Lu, and B. Dong. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[24] H. Yu, X. Tian, W. E, and Q. Li. OnsagerNet: Learning stable and interpretable dynamics using a generalized Onsager principle. *arXiv preprint arXiv:2009.02327*, 2020.

[25] J Z. Kolter and G. Manek. Learning stable deep dynamics models. In *Advances in Neural Information Processing Systems*, pages 11128–11136, 2019.

[26] Y. D. Zhong, B. Dey, and A. Chakraborty. Symplectic ODE-net: Learning Hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.

[27] P. Jin, A. Zhu, G. E. Karniadakis, and Y. Tang. Symplectic networks: Intrinsic structure-preserving networks for identifying Hamiltonian systems. *arXiv preprint arXiv:2001.03750*, 2020.

[28] S. Rudy, A. Alla, S. L Brunton, and J N. Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019.

[29] S. L Brunton, J. L Proctor, and J N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[30] J. Han, L. Zhang, and W. E. Integrating machine learning with physics-based modeling. *arXiv preprint arXiv:2006.02619*, 2020.

[31] 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, 2019.

[32] W. E. The dawning of a new era in applied mathematics. *Notice of the American Mathematical Society*, April, 2021.
[33] W. E and B. Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6:1–12, 2018.
[34] Justin S. and Konstantinos S. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.
[35] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
[36] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-Net: Learning PDEs from data. In *International Conference on Machine Learning*, pages 3208–3216, 2018.
[37] J. Huang, Z. Ma, Y. Zhou, and W.-A. Yong. Learning thermodynamically stable and Galilean invariant partial differential equations for non-equilibrium flows. *Journal of Non-Equilibrium Thermodynamics*, page 000010151520210008, 2021.
[38] D. Y. Tzou. Macro- to microscale heat transfer: The lagging behavior, 2nd edition. 2014.
[39] Y. Zhang and W. Ye. Modified ballisticdiffusive equations for transient non-continuum heat conduction. *International Journal of Heat and Mass Transfer*, 83:51–63, 2015.
[40] Z. Guo and K. Xu. Discrete unified gas kinetic scheme for multiscale heat transfer based on the phonon Boltzmann transport equation. *International Journal of Heat & Mass Transfer*, 102:944–958, 2016.
[41] Y. Guo and M. Wang. Phonon hydrodynamics and its applications in nanoscale heat transport. *Physics Reports*, 595:1–44, 2015. Phonon hydrodynamics and its applications in nanoscale heat transport.
[42] W.-A. Yong. An interesting class of partial differential equations. *Journal of Mathematical Physics*, 49(3):033503, 2008.
[43] J. Han, C. Ma, Z. Ma, and W. E. Uniformly accurate machine learning-based hydrodynamic models for kinetic equations. *Proceedings of the National Academy of Sciences*, 116(44):21983–21991, 2019.
[44] A. G. Baydin, B. A Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.
[45] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.
[46] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.
[47] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
[48] K. C. Collins, A. A. Maznev, Z. Tian, K. Esfarjani, K. A. Nelson, and G. Chen. Non-diffusive relaxation of a transient thermal grating analyzed with the Boltzmann transport equation. *Journal of Applied Physics*, 114(10):104302, 2013.
[49] G. Lebon, M. Grmela, and C. Dubois. From ballistic to diffusive regimes in heat transport at nano-scales. *Comptes Rendus Mécanique*, 339(5):324–328, 2011. Thermodiffusion and coupled phenomena.
[50] A. Majumdar. Microscale heat conduction in dielectric thin films. *Journal of Heat Transfer*, 115(1):7–16, 02 1993.
[51] Y. Guo and M. Wang. Lattice Boltzmann modeling of phonon transport. *Journal of Computational Physics*, 315:1–15, 2016.