

Macroscopic auxiliary asymptotic preserving neural networks for the linear radiative transfer equations

Hongyan Li^a, Song Jiang^b, Wenjun Sun^b, Liwei Xu^{a,*}, Guanyu Zhou^c

^a*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu Sichuan 611731, China*

^b*Institute of Applied Physics and Computational Mathematics, Beijing 100094, China*

^c*Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu Sichuan 611731, China*

Abstract

We develop a Macroscopic Auxiliary Asymptotic-Preserving Neural Network (MA-APNN) method to solve the time-dependent linear radiative transfer equations (LRTEs), which have a multi-scale nature and high dimensionality. To achieve this, we utilize the Physics-Informed Neural Networks (PINNs) framework and design a new adaptive exponentially weighted Asymptotic-Preserving (AP) loss function, which incorporates the macroscopic auxiliary equation that is derived from the original transfer equation directly and explicitly contains the information of the diffusion limit equation. Thus, as the scale parameter tends to zero, the loss function gradually transitions from the transport state to the diffusion limit state. In addition, the initial data, boundary conditions, and conservation laws serve as the regularization terms for the loss. We present several numerical examples to demonstrate the effectiveness of MA-APNNs.

Keywords: Linear radiative transfer equation, macroscopic auxiliary equation, adaptive exponential weight, asymptotic-preserving neural network

1. Introduction

The kinetic equations describe the motion of particles in a medium, including collision and absorption. These equations are defined in phase space and can involve multiple spatial and temporal scales, as well as nonlocal operators, making numerical simulations difficult[1–4]. LRTEs, as a typical kinetic model, find wide-ranging applications in fields such as astrophysics, weapon physics, inertial/magnetic confinement fusion[5–7], and more. Accurately simulating LRTEs is nontrivial, mainly due to two challenges. The first challenge is high dimensionality. LRTEs describe the evolution of the radiation density function of a vast number of photons, which, in the most

*Corresponding author

Email addresses: lihongyan142869@163.com (Hongyan Li), jiang@iapcm.ac.cn (Song Jiang), sun_wenjun@iapcm.ac.cn (Wenjun Sun), xu1@uestc.edu.cn (Liwei Xu), zhoug@uestc.edu.cn (Guanyu Zhou)

general cases, is a seven-dimensional problem. When uncertainties are factored into the equations, the dimensionality increases further[8–10]. The second challenge is the multiscale features caused by different opacities of the background materials. LRTEs typically involve multiple spatial and/or temporal scales, characterized by the Knudsen number, which is the dimensionless mean free path. When the Knudsen number is tiny, LRTEs can be approximated by the diffusion limit equations[1]. To preserve the propagation properties of photons in different optical regions, the multiscale modeling approaches are introduced to the numerical simulation[11, 12]. When coupled radiation transport and diffusion models are solved using the same spatial grid, the mesh size should be roughly equivalent to the mean free path. This requirement can make computations incredibly expensive in optically thick mediums where the Knudsen number is very small.

Efficient computational methods that can deal with high dimensionality and multiscale characteristics are highly desirable. Classical tools such as the Implicit Monte Carlo (IMC) method [13–17] are capable of handling high dimensionality, but the convergence rate is low, and there are statistical noises. On the other hand, Asymptotic-Preserving (AP) schemes, which are uniformly stable with respect to the scaling parameter, are proposed to address the multiscale problem. As the limit of the Knudsen number approaches zero, the solution of the AP schemes converges to the diffusion limit case. The AP schemes were initially proposed to solve the neutron transport problems by Larsen and Keller[18], Larsen[19–21], and then improved by Larsen, Morel and Miller [22], Larsen and Morel[23], Jin and Levermore[24, 25]. For the unsteady cases, [26, 27] also constructed the AP schemes by decomposing the distribution into the equilibrium and disturbance parts. These methods were further developed and extended to other multiscale kinetic models[3, 28–30].

In recent years, deep learning methods and deep neural networks (DNNs) have been developed vigorously and achieved some success in solving PDEs [31–35]. Various approaches have been proposed, including the Physics-Informed Neural Networks (PINNs) [36] and the deep Galerkin method (DGM) [37] based on the least-squares formulation, the deep Ritz method (DRM) [38] utilizing the variational formulation, and others [39–41]. Compared to traditional mesh-based methods, the DNNs have an advantage as mesh-free methods in solving problems with complex geometric domains and high dimensions [42, 43]. Furthermore, DNNs use automatic differentiation, which avoids truncation errors in discretization. However, DNN methods have some drawbacks, including lack of high accuracy, long training time due to a large number of network parameters, and the use of stochastic gradient method for optimizing high-dimensional non-convex functions[44].

The PINNs, Model-Operator-Data Network (MOD-Net), and Physics-Informed DeepONets (PIDONs) have been applied to solve steady and unsteady linear radiative transfer models[32, 45–49]. For MOD-Net method, the DNN is used to parameterize the Greens function and the neural operator is obtained to approximate the solution. Combining the operator universal approximation theory and PINN’s idea, the PIDON method can solve a class of PDEs from the learned continuous operators. We find that the vanilla PINN, PIDON, or MOD-Net approach, has difficulties dealing with the multiscale characteristics. The loss function of these methods deteriorates as the scale parameter decreases to zero, which fails to capture asymptotic limits and usually

tends to learn a simplified model during the training process. To tackle this problem, the Asymptotic-Preserving Neural Networks (APNNs)[4] and Model-Data APNNs[50] are designed by using the micro-macro decomposition to solve the time-dependent linear transport equations and gray radiative transfer equations, respectively. In addition, [51] develops the APNN method based on the even-odd decomposition for the multi-scale kinetic equations. The micro-macro and even-odd decomposition techniques have also been applied to develop the Asymptotic-Preserving Convolutional Deep Operator Networks (APCONs) for the time-dependent linear radiative transport problem[52].

In this work, we devise a novel loss function incorporating the macroscopic auxiliary equation, which is capable of capturing the diffusion limit behavior as the scale parameter approaches zero. We derive the macroscopic auxiliary equation for the LRTEs by repeatedly substituting the original equation into itself and integrating it with respect to the angle direction. This was inspired by the idea of deriving the diffusion limit system for the radiative heat transport equation as described in [53]. The macroscopic auxiliary equation we derived holds true for smooth solutions and can be considered a perturbation of the diffusion limit equation when the scale parameter is small. In addition, we introduce an exponential weight that depends on the scale parameter. This weight is used to combine the original equation’s loss with that of the macroscopic auxiliary equation, to achieve the AP property. We also consider the mass conservation constraints of the radiation intensity under periodic boundaries, as well as the boundary and initial hard constraints under inflow and periodic boundaries. To evaluate the effectiveness of the proposed Macroscopic Auxiliary APNN (MA-APNN) method, we simulate low and high-dimensional linear transport equations under transport and diffusion regimes. We investigate various scenarios, such as initial layer problems, constant or variable scattering problems, uncertainty quantification problems, and more. Compared to PINNs[36], MA-APNNs have great advantages in solving diffusion dominating problems. Different from the APNNs in [4, 51] where the macro-micro or even-odd decomposition is used, it does not need the decomposition of radiation intensity in MA-APNNs, so the network’s construction is simple, and in turn cause the training time is reduced greatly with the comparable prediction accuracy as APNNs[4] under diffusion case.

The rest of this paper is organized as follows. In Section 2, we introduce the time-dependent linear radiative transfer model and its corresponding diffusion limit equation. In Section 3, we present the motivation of the MA-APNN method and define the new AP loss function. Numerous numerical experiments, including both multiscale and high dimensional uncertainty problems, are carried out in Section 4 to validate the efficiency of MA-APNNs. A conclusion remark is given in Section 5.

2. Linear radiative transfer equations

The linear radiative transfer equations describe the radiation photons transport and the energy exchange with the background materials. We denote by $f(t, r, \Omega)$ the radiation intensity of photons located at the space point $r = (x, y, z) \in D$ in time $t \in \tau$ and propagating in direction in $\Omega = (\xi, \eta, \mu) \in S^2$, where D is a bounded domain in \mathbb{R}^3 and S^2 is the unit sphere. The scaled form of the LRTEs with initial and boundary

conditions is stated as follows.

$$\varepsilon^2 \frac{\partial f}{\partial t} + \varepsilon \Omega \cdot \nabla_r f = \sigma \left(\frac{1}{4\pi} \int_{S^2} f d\Omega - f \right) - \varepsilon^2 \alpha f + \varepsilon^2 G \quad \text{in } \tau \times D \times S^2, \quad (2.1a)$$

$$Bf = f_b \quad \text{on } \tau \times \partial D \times S^2, \quad (2.1b)$$

$$f(0, r, \Omega) = f_0(r, \Omega) \quad \text{in } D \times S^2, \quad (2.1c)$$

where $\sigma(r)$ is the scattering coefficient, $\alpha(r)$ the absorption coefficient, $G(r)$ the internal source, and $\varepsilon > 0$ the Knudsen number. The average of $f(t, r, \Omega)$ on S^2 is called the incident radiation $\rho(t, r)$, saying

$$\rho := \langle f \rangle = \frac{1}{4\pi} \int_{S^2} f d\Omega.$$

Integrating (2.1a) with respect to Ω , we obtain

$$\varepsilon^2 \frac{\partial \rho}{\partial t} + \varepsilon \langle \Omega \cdot \nabla_r f \rangle = -\varepsilon^2 \alpha \rho + \varepsilon^2 G \quad \text{in } \tau \times D. \quad (2.2)$$

Furthermore, integrating (2.2) with respect to r and applying the integration by parts, we get the mass conservation equation:

$$\varepsilon^2 \frac{d}{dt} \int_D \rho dr + \varepsilon \frac{1}{4\pi} \int_{\partial D} \int_{S^2} \Omega \cdot n_r f d\Omega dr = -\varepsilon^2 \alpha \int_D \rho dr + \varepsilon^2 \int_D G dr, \quad (2.3)$$

where n_r represents the unit outer normal to ∂D at position r . In the following, we state the boundary condition (2.1b) in detail, and introduce the 1D and 2D models, the uncertainty quantification model, and the diffusion limit equation.

2.1. The boundary conditions

We separate the boundary $\Gamma = \tau \times \partial D \times S^2$ into the inflow and outflow parts as follows:

$$\Gamma_- := \{(t, r, \Omega) \in \Gamma : \Omega \cdot n_r < 0\}, \quad \Gamma_+ := \{(t, r, \Omega) \in \Gamma : \Omega \cdot n_r > 0\}.$$

In this work, we consider two types of boundary conditions for L RTEs.

(1) The inflow boundary condition

$$f(t, r, \Omega) = f_B(t, r, \Omega) \quad (t, r, \Omega) \in \Gamma_-, \quad (2.4)$$

where $f_B(t, r, \Omega)$ is a given function.

(2) If D is symmetric, we can enforce the periodic boundary condition [30]

$$f(t, r, \Omega) = f(t, \mathcal{S}(r), \Omega) \quad (t, r, \Omega) \in \Gamma, \quad (2.5)$$

where $\mathcal{S} : \partial D \rightarrow \partial D$ is a one-to-one symmetry mapping satisfying $n_r = -n_{\mathcal{S}(r)}$. At this time, we see from (2.3) that

$$\frac{d}{dt} \int_D \rho dr + \int_D \alpha \rho dr - \int_D G dr = 0. \quad (2.6)$$

2.2. The 1D and 2D linear models

In one-dimensional case, i.e., $(r, \Omega) = (x, \mu) \in (x_L, x_R) \times (-1, 1)$, the LRTes becomes

$$\varepsilon^2 \partial_t f + \varepsilon \mu \partial_x f = \sigma \left(\frac{1}{2} \int_{-1}^1 f d\mu - f \right) - \varepsilon^2 \alpha f + \varepsilon^2 G, \quad (2.7)$$

with the isotropic inflow boundary condition

$$f(t, x_L, \mu > 0) = f_L(t, \mu), \quad f(t, x_R, \mu < 0) = f_R(t, \mu), \quad (2.8)$$

or the periodic boundary condition

$$f(t, x_L, \mu) = f(t, x_R, \mu), \quad (2.9)$$

and the initial condition

$$f(0, x, \mu) = f_0(x, \mu). \quad (2.10)$$

In two-dimensional case, i.e., $(r, \Omega) = ((x, y), (\mu, \xi)) \in D \times S^1$, (2.1a) is replaced by

$$\varepsilon^2 \partial_t f + \varepsilon \Omega \cdot \nabla_r f = \sigma \left(\frac{1}{2\pi} \int_{S^1} f dv - f \right) - \varepsilon^2 \alpha f + \varepsilon^2 G. \quad (2.11)$$

where $D \subset \mathbb{R}^2$ and $S^1 = \{(\xi, \eta) : \xi^2 + \eta^2 = 1\}$.

2.3. The uncertainty quantification (UQ) model

In practice, we are interested in the LRTes that contains uncertainty in the collision cross section, source, initial or boundary data[10]. The uncertainty is characterized by the random variable $\mathbf{z} = (z_1, z_2, \dots, z_m) \in \mathbb{R}^m$ with the probability density $\omega(\mathbf{z})$. In our numerical simulation, we consider the spatial 1D LRTes with the random input \mathbf{z} , where the radiation intensity $f(t, x, \mu, \mathbf{z})$ satisfies the equation

$$\varepsilon^2 \partial_t f + \varepsilon \mu \partial_x f = \sigma(\mathbf{z}) \left(\frac{1}{2} \int_{-1}^1 f d\mu - f \right) - \varepsilon^2 \alpha(\mathbf{z}) f + \varepsilon^2 G(\mathbf{z}), \quad (2.12)$$

and the initial and inflow boundary conditions

$$f(0, x, \mu, \mathbf{z}) = f_0(x, \mu, \mathbf{z}), \quad (2.13a)$$

$$f(t, x_L, \mu, \mathbf{z}) = f_L(t, \mu, \mathbf{z}) \quad (\mu > 0), \quad (2.13b)$$

$$f(t, x_R, \mu, \mathbf{z}) = f_R(t, \mu, \mathbf{z}) \quad (\mu < 0). \quad (2.13c)$$

2.4. The diffusion limit equation

As the Knudsen number $\varepsilon \rightarrow 0$, the radiation intensity $f(t, r, \Omega)$ converges to the average density $\rho = \langle f \rangle$ and satisfies the asymptotic diffusion limit equation[26], i.e.,

$$f = \rho, \quad (2.14a)$$

$$\partial_t \rho - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \rho \right) + \alpha \rho - G = 0, \quad (2.14b)$$

where

$$\langle \Omega^2 \rangle := \frac{1}{4\pi} \int_{S^2} \mu^2 d\Omega = \frac{1}{3} \frac{1}{4\pi} \int_{S^2} \mu^2 + \xi^2 + \eta^2 d\Omega = \frac{1}{3} \frac{1}{4\pi} \int_{S^2} 1 d\Omega = \frac{1}{3}.$$

In 1D case, $\langle \Omega^2 \rangle := \frac{1}{2} \int_{-1}^1 \mu^2 d\mu = 1/3$, while in 2D case, $\langle \Omega^2 \rangle := \frac{1}{2\pi} \int_{S^1} \xi^2 d\Omega = 1/2$. An asymptotic-preserving scheme for (2.1) should be uniformly stable with respect to ε and lead to an accurate approximation to the diffusion limit equation (2.14b) when ε approaches zero.

3. Macroscopic auxiliary asymptotic preserving neural networks

We first introduce the notations for DNNs. Given an input $\mathbf{x} = (t, r, \Omega) \in \tau \times D \times S^2$, the output of the L -layer feedforward neural network f_θ is defined recursively as

$$\begin{aligned} f_\theta^{[0]}(\mathbf{x}) &= \mathbf{x}, \\ f_\theta^{[l]}(\mathbf{x}) &= \sigma^h \circ \left(W^{[l-1]} f_\theta^{[l-1]}(\mathbf{x}) + b^{[l-1]} \right), \quad l \leq L-1, \\ f_\theta(\mathbf{x}) &= f_\theta^{[L]}(\mathbf{x}) = \sigma^o \circ \left(W^{[L-1]} f_\theta^{[L-1]}(\mathbf{x}) + b^{[L-1]} \right), \end{aligned} \quad (3.1)$$

where $W^{[l]} \in \mathbb{R}^{m_{l+1} \times m_l}$ and $b^{[l]} \in \mathbb{R}^{m_{l+1}}$ represent weight matrix and bias vector, respectively, σ^h is the nonlinear activation function of the hidden layers, and σ^o is a specially designed activation function of the output layer. The common choices of σ^h are the sigmoid function, the hyperbolic tangent function, the ReLU function, and so on. The notation “ \circ ” means entry-wise composition. For simplicity, we denote the set of parameters by θ and represent the network by a list, i.e., $[m_0, m_1, \dots, m_L]$, where m_0 and m_L are the dimensions of the input and the output.

Solving PDE using DNNs involves three critical steps. Firstly, we parameterize the solution of PDE with a deep neural network. Secondly, we design the population/empirical loss function associated with the PDE, which evaluates the error between the approximate and exact solutions. Finally, we select appropriate optimization algorithms to minimize the loss function.

According to [4], for the multiscale problem, it is crucial to build a loss function that preserves the asymptotic property. In the following, we first briefly explain why the vanilla PINNs fail to resolve the multiscale LRTEs. Then, we present a formal derivation of the macroscopic auxiliary equation. After that, we build a loss function incorporating the macroscopic auxiliary equation and demonstrate the asymptotic-preserving property.

3.1. The vanilla PINNs fail to resolve the multiscale LRTEs

Let us consider the LRTEs with periodic boundary condition (2.5), which satisfies the mass conservation law (2.6). To parametrize the nonnegative radiation intensity $f(t, r, \Omega)$, we use the neural network $f_\theta^m(t, r, \Omega)$ with the nonnegative activation function $\sigma^o(\mathbf{x}) = e^{-\mathbf{x}}$ at the output layer, and adopt the least square of the residual of LRTEs as the loss function:

$$L_{\text{PINNs}}^\varepsilon = L_{\text{PINNs},g}^\varepsilon + L_{\text{PINNs},b}^\varepsilon + L_{\text{PINNs},i}^\varepsilon + L_{\text{PINNs},c}^\varepsilon,$$

where

$$\begin{aligned}
L_{\text{PINNs},g}^\varepsilon &= \lambda_g \left\| \varepsilon^2 \partial_t f_\theta^{mn} + \varepsilon \Omega \cdot \nabla_r f_\theta^{mn} - \sigma(\langle f_\theta^{mn} \rangle - f_\theta^{mn}) + \varepsilon^2 \alpha f_\theta^{mn} - \varepsilon^2 G \right\|_{L^2(\tau \times D \times S^2)}^2, \\
L_{\text{PINNs},b}^\varepsilon &= \lambda_b \|B f_\theta^{mn} - f_b\|_{L^2(\tau \times \partial D \times S^2)}^2, \\
L_{\text{PINNs},i}^\varepsilon &= \lambda_i \|f_\theta^{mn}(0) - f_0\|_{L^2(D \times S^2)}^2, \\
L_{\text{PINNs},c}^\varepsilon &= \lambda_c \left\| \varepsilon \partial_t \int_D \langle f_\theta^{mn} \rangle dr + \int_{\partial D} \langle \Omega \cdot n_r f_\theta^{mn} \rangle dr + \varepsilon \int_D \alpha \langle f_\theta^{mn} \rangle dr - \varepsilon \int_D G dr \right\|_{L^2(\tau)}^2.
\end{aligned}$$

Here, $\lambda_g, \lambda_i, \lambda_b$ and λ_c are the weight parameter to be tuned. Note that $L_{\text{PINNs},c}^\varepsilon$ is the residual of the mass conservation constraint. If the periodic boundary condition is imposed, then, by (2.6),

$$L_{\text{PINNs},c}^\varepsilon = \lambda_c \left\| \partial_t \int_D \langle f_\theta^{mn} \rangle dr + \int_D \alpha \langle f_\theta^{mn} \rangle dr - \int_D G dr \right\|_{L^2(\tau)}^2.$$

When ε is not too small, one can expect to obtain the approximate solution by minimizing the above loss function $L_{\text{PINNs}}^\varepsilon$. However, as $\varepsilon \rightarrow 0$, we find that

$$L_{\text{PINNs},g}^\varepsilon \rightarrow \lambda_g \left\| -\sigma(\langle f_\theta^{mn} \rangle - f_\theta^{mn}) \right\|_{L^2(\tau \times D \times S^2)}^2,$$

which is nothing but the residual of the following equation

$$\sigma(\langle f \rangle - f) = \sigma(\rho - f) = 0.$$

Apparently, it is not the desired diffusion limit equation (2.14b).

If we add the residual of the diffusion limit equation

$$L_{\text{PINNs},d}^\varepsilon = \lambda_d \left\| \partial_t \langle f_\theta^{mn} \rangle - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \langle f_\theta^{mn} \rangle \right) + \alpha \langle f_\theta^{mn} \rangle - G \right\|_{L^2(\tau \times D)}^2,$$

to the total loss directly, saying

$$L_{\text{PINNs}}^\varepsilon = L_{\text{PINNs},g}^\varepsilon + L_{\text{PINNs},b}^\varepsilon + L_{\text{PINNs},i}^\varepsilon + L_{\text{PINNs},c}^\varepsilon + L_{\text{PINNs},d}^\varepsilon,$$

then we will recover the diffusion limit system as $\varepsilon \rightarrow 0$. But, unfortunately, it fails to solve the problem with large or medium ε , even if we tune the weight parameters λ_g and λ_d depending on ε .

3.2. The macroscopic auxiliary asymptotic-preserving neural networks (MA-APNNs)

The above argument reveals that the vanilla PINNs are unsuitable for dealing with the multiscale feature. In this subsection, we will devise a new asymptotic-preserving loss function to tackle this problem. As a preliminary, we derive a macroscopic auxiliary equation from (2.1a). Then we demonstrate that the proposed loss function incorporating the macroscopic auxiliary equation is applicable to solve the LTREs for arbitrary ε and the solution converges to the diffusion limit as $\varepsilon \rightarrow 0$.

3.2.1. The macroscopic auxiliary equation

From (2.1a), we see that

$$f = \langle f \rangle - \frac{\varepsilon^2}{\sigma} \partial_t f - \frac{\varepsilon}{\sigma} \Omega \cdot \nabla_r f - \frac{\varepsilon^2}{\sigma} \alpha f + \frac{\varepsilon^2}{\sigma} G. \quad (3.2)$$

Replacing the f in $\frac{\varepsilon^2}{\sigma} \partial_t f$ and $\frac{\varepsilon}{\sigma} \Omega \cdot \nabla_r f$ by the right-hand side of (3.2), and arranging the result by the order of ε , we get

$$\begin{aligned} f &= \langle f \rangle - \varepsilon \frac{1}{\sigma} \Omega \cdot \nabla_r \langle f \rangle - \varepsilon^2 \left(\frac{1}{\sigma} \partial_t \langle f \rangle - \frac{1}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \Omega \cdot \nabla_r f \right) + \frac{1}{\sigma} \alpha f - \frac{1}{\sigma} G \right) \\ &+ \varepsilon^3 \left(\frac{1}{\sigma} \partial_t \left(\frac{1}{\sigma} \Omega \cdot \nabla_r f \right) + \frac{1}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \partial_t f + \frac{1}{\sigma} \alpha f - \frac{1}{\sigma} G \right) \right) \\ &+ \varepsilon^4 \left(\frac{1}{\sigma} \partial_t \left(\frac{1}{\sigma} \partial_t f + \frac{1}{\sigma} \alpha f - \frac{1}{\sigma} G \right) \right). \end{aligned}$$

Let us pay attention to the second term with order ε^2 , namely, $\frac{\varepsilon^2}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \Omega \cdot \nabla_r f \right)$. We substitute the right side of (3.2) for the f in this term, arrange the new equation by the order of ε and obtain

$$\begin{aligned} f &= \langle f \rangle - \frac{\varepsilon}{\sigma} \Omega \cdot \nabla_r \langle f \rangle - \frac{\varepsilon^2}{\sigma} \left(\partial_t \langle f \rangle - \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \Omega \cdot \nabla_r \langle f \rangle \right) + \alpha f - G \right) \\ &+ \varepsilon^3 \mathcal{A}(f, G) + \varepsilon^4 \mathcal{B}(f, G), \end{aligned} \quad (3.3)$$

where

$$\begin{aligned} \mathcal{A}(f, G) &= \frac{1}{\sigma} \partial_t \left(\frac{1}{\sigma} \Omega \cdot \nabla_r f \right) + \frac{1}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \left(\partial_t f + \alpha f - G - \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \Omega \cdot \nabla_r f \right) \right) \right), \\ \mathcal{B}(f, G) &= \frac{1}{\sigma^2} \partial_t \left(\partial_t f + \alpha f - G \right) - \frac{1}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \Omega \cdot \nabla_r \left(\frac{1}{\sigma} \left(\partial_t f + \alpha f - G \right) \right) \right). \end{aligned}$$

Integrating (3.3) with respect to Ω , in view of $\rho(t, r) = \langle f \rangle$ and $\langle \rho \rangle = \rho$, we arrive

$$\partial_t \rho - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \rho \right) + \alpha \rho - G - \varepsilon \langle \sigma \mathcal{A}(f, G) \rangle - \varepsilon^2 \langle \sigma \mathcal{B}(f, G) \rangle = 0, \quad (3.4)$$

where $\langle \Omega^2 \rangle := \langle \mu^2 \rangle$ and we have used

$$\begin{aligned} \langle \mu \rangle &= \langle \xi \rangle = \langle \eta \rangle = 0 \quad (\text{i.e., } \langle \Omega \rangle = 0), \\ \langle \mu^2 \rangle &= \langle \xi^2 \rangle = \langle \eta^2 \rangle, \quad \langle \mu \xi \rangle = \langle \mu \eta \rangle = \langle \xi \eta \rangle = 0. \end{aligned}$$

We call (3.4) the macroscopic auxiliary equation, which holds true if the solution of (2.1a) is smooth enough such that the high-order differentials in (3.4) make sense. When ε is tiny, we can regard (3.4) as the diffusion limit equation (2.14b) with the perturbation $\varepsilon \langle \sigma \mathcal{A}(f, G) \rangle + \varepsilon^2 \langle \sigma \mathcal{B}(f, G) \rangle$. When $\varepsilon \rightarrow 0$, a formal calculation indicates that (2.1a) and (3.4) approaches the expected diffusion limit system:

$$\sigma(\rho - f) = 0, \quad \partial_t \rho - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \rho \right) + \alpha \rho - G = 0.$$

A rigorous validation of the convergence is out of the scope of this work. Our motivation is to apply the macroscopic auxiliary equation to design an asymptotic-preserving loss function.

3.3. The exponentially weighted MA-APNNs loss function

We use a deep neural network $f_\theta^{mn}(t, r, \Omega)$ to parameterize the radiation intensity $f(t, r, \Omega)$. In particular, we choose the activation function $\sigma^o(\mathbf{x}) = e^{-\mathbf{x}}$ at the output layer to guarantee the nonnegativity. Now we design a new APNN loss $L_{\text{MA-APNNs}}^\varepsilon$.

$$L_{\text{MA-APNNs}}^\varepsilon = L_{\text{MA-APNNs},g}^\varepsilon + L_{\text{MA-APNNs},i}^\varepsilon + L_{\text{MA-APNNs},b}^\varepsilon + L_{\text{PINNs},c}^\varepsilon, \quad (3.5)$$

where the residual of the mass conservation law, i.e., $L_{\text{PINNs},c}^\varepsilon$, is the same to that of Section 3.1. The differences to the vanilla PINNs are shown as follows.

We add boundary and initial value constraints of the incident radiation $\langle f \rangle = \rho$ to $L_{\text{PINNs},b}^\varepsilon$ and $L_{\text{PINNs},i}^\varepsilon$, respectively.

$$\begin{aligned} L_{\text{MA-APNNs},b}^\varepsilon &= \lambda_b \left(\|Bf_\theta^{mn} - f_b\|_{L^2(\tau \times \partial D \times S^2)}^2 + \|B\langle f_\theta^{mn} \rangle - \langle f_b \rangle\|_{L^2(\tau \times \partial D)}^2 \right), \\ L_{\text{MA-APNNs},i}^\varepsilon &= \lambda_i \left(\|f_\theta^{mn}(0) - f_0\|_{L^2(D \times S^2)}^2 + \|\langle f_\theta^{mn}(0) \rangle - \langle f_0 \rangle\|_{L^2(D)}^2 \right). \end{aligned}$$

Most importantly, we replace the residual of the governing equation $L_{\text{PINNs},g}^\varepsilon$ by

$$\begin{aligned} L_{\text{MA-APNNs},g}^\varepsilon &= \left\| \lambda_{v,\beta}^{\frac{1}{2}} \left(\varepsilon^2 \partial_t f_\theta^{mn} + \varepsilon \Omega \cdot \nabla_r f_\theta^{mn} - \sigma(\langle f_\theta^{mn} \rangle - f_\theta^{mn}) + \varepsilon^2 \alpha f_\theta^{mn} - \varepsilon^2 G \right) \right\|_{L^2(\tau \times D \times S^2)}^2 \\ &\quad + \left\| (1 - \lambda_{v,\beta})^{\frac{1}{2}} \left(\partial_t \langle f_\theta^{mn} \rangle - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \langle f_\theta^{mn} \rangle \right) + \alpha \langle f_\theta^{mn} \rangle - G \right. \right. \\ &\quad \left. \left. - \varepsilon \langle \mathcal{A}(f_\theta^{mn}, G) \rangle - \varepsilon^2 \langle \mathcal{B}(f_\theta^{mn}, G) \rangle \right) \right\|_{L^2(\tau \times D)}^2, \end{aligned}$$

where $\lambda_{v,\beta}(r) := e^{-v(r)\beta_1} + \beta_2$ with $v(r) = \frac{\sigma(r)}{\varepsilon^2} + \alpha(r)$. Here, $\beta = (\beta_1, \beta_2) > 0$ is the tunable parameter satisfying $1 - \max_r \lambda_{v,\beta}(r) > 0$.

It is easy to verify that the loss $L_{\text{MA-APNNs},g}^\varepsilon$ asymptotically converges to the loss of the diffusion limit equations. In fact, since $\sigma(r) > 0$, as $\varepsilon \rightarrow 0$, we observe that $\lambda_{v,\beta}(r) \rightarrow \beta_2$ and

$$\begin{aligned} L_{\text{MA-APNNs},g}^\varepsilon &\rightarrow \beta_2 \left\| -\sigma(\langle f_\theta^{mn} \rangle - f_\theta^{mn}) \right\|_{L^2(\tau \times D \times S^2)}^2 \\ &\quad + (1 - \beta_2) \left\| \partial_t \langle f_\theta^{mn} \rangle - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \langle f_\theta^{mn} \rangle \right) + \alpha \langle f_\theta^{mn} \rangle - G \right\|_{L^2(\tau \times D)}^2, \end{aligned}$$

which is the residual of the governing equation of the desired diffusion limit system (2.14). Since the smooth solution f satisfies the macroscopic auxiliary equation (3.4), the loss (3.5) has the consistency, that is, $L_{\text{MA-APNNs}}^\varepsilon|_{f_\theta^{mn}=f} = 0$. Therefore, the loss function $L_{\text{MA-APNNs}}^\varepsilon$ possesses the AP property and is applicable to the cases under the transport, medium, and diffusion regimes. We put a schematic plot of our method in Figure 3.1.

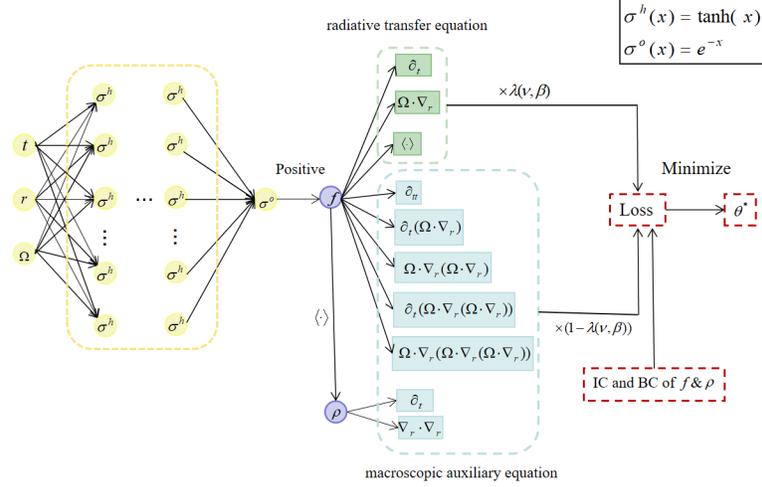


Fig. 3.1. Schematic of MA-APNNs for solving the linear radiative transfer equation with initial and boundary data.

3.4. The MA-APNN empirical loss function

In practice, we apply the quadrature rules to compute the integrals of the loss function $L_{\text{MA-APNNs}}^{\varepsilon}$.

3.4.1. Angular integration approximation

We utilize the Gauss-Legendre quadrature rule to approximate the angular integration $\langle f \rangle$:

$$\langle f \rangle = \frac{1}{4\pi} \int_{S^2} f(t, r, \Omega) d\Omega \approx \frac{1}{4\pi} \sum_{m=1}^{N_s} f(\Omega_m) \omega_m.$$

The Gauss-Legendre quadrature points $\{\Omega_m\}_{m=1}^{N_s} \subset S^2$ will be used as the training points as well, and $\{\omega_m\}_{m=1}^{N_s}$ are the associated weights.

3.4.2. Empirical loss functions, sample points, and labeled data

Here we sample the training points $\mathcal{D}_f = \{\mathbf{x}_j^f = (t_j^f, r_j^f, \Omega_j^f), j = 1, 2, \dots, N_f\}$ as a low-discrepancy Sobol sequence in the computational domain, and choose the Quasi-Monte Carlo method for numerical integration [54]. The empirical loss function of the

governing equation is stated as follows

$$\begin{aligned}
L_{\text{MA-APNNs},g}^{\varepsilon,nn} &= \frac{1}{N_{\text{int}}} \sum_{j=1}^{N_{\text{int}}} \left| \lambda_{\nu,\beta}(r_j^{\text{int}}) \left(\varepsilon^2 \partial_t f_{\theta}^{nn}(t_j^{\text{int}}, r_j^{\text{int}}, \Omega_j^{\text{int}}) + \varepsilon \Omega \cdot \nabla_r f_{\theta}^{nn}(t_j^{\text{int}}, r_j^{\text{int}}, \Omega_j^{\text{int}}) \right. \right. \\
&\quad \left. \left. - \sigma \left(\langle f_{\theta}^{nn} \rangle(t_j^{\text{int}}, r_j^{\text{int}}) - f_{\theta}^{nn}(t_j^{\text{int}}, r_j^{\text{int}}, \Omega_j^{\text{int}}) \right) + \varepsilon^2 \alpha f_{\theta}^{nn}(t_j^{\text{int}}, r_j^{\text{int}}, \Omega_j^{\text{int}}) - \varepsilon^2 G \right|^2 \\
&\quad + \frac{1}{N_{\text{int}}} \sum_{j=1}^{N_{\text{int}}} \left| \left(1 - \lambda_{\nu,\beta}(r_j^{\text{int}}) \right) \left(\partial_t \langle f_{\theta}^{nn} \rangle(t_j^{\text{int}}, r_j^{\text{int}}) - \langle \Omega^2 \rangle \nabla_r \left(\frac{1}{\sigma} \nabla_r \langle f_{\theta}^{nn} \rangle(t_j^{\text{int}}, r_j^{\text{int}}) \right) \right. \right. \\
&\quad \left. \left. + \alpha \langle f_{\theta}^{nn} \rangle(t_j^{\text{int}}, r_j^{\text{int}}) - G - \varepsilon \langle \sigma \mathcal{A}(f_{\theta}^{nn}, G) \rangle(t_j^{\text{int}}, r_j^{\text{int}}) - \varepsilon^2 \langle \sigma \mathcal{B}(f_{\theta}^{nn}, G) \rangle(t_j^{\text{int}}, r_j^{\text{int}}) \right|^2
\end{aligned}$$

where $\{(t_j^{\text{int}}, r_j^{\text{int}}, \Omega_j^{\text{int}})\}_{j=1}^{N_{\text{int}}}$ are interior Sobol sequence points. Similarly, the empirical loss functions for the boundary condition, the initial value, and conservation law are given by:

$$\begin{aligned}
L_{\text{MA-APNNs},b}^{\varepsilon,nn} &= \lambda_b \frac{1}{N_{\text{sb}}} \sum_{j=1}^{N_{\text{sb}}} \left(\left| B f_{\theta}^{nn}(t_j^{\text{sb}}, r_j^{\text{sb}}, \Omega_j^{\text{sb}}) - f_b(t_j^{\text{sb}}, r_j^{\text{sb}}, \Omega_j^{\text{sb}}) \right|^2 \right. \\
&\quad \left. + \left| B \langle f_{\theta}^{nn} \rangle(t_j^{\text{sb}}, r_j^{\text{sb}}) - \langle f_b \rangle(t_j^{\text{sb}}, r_j^{\text{sb}}) \right|^2 \right), \\
L_{\text{MA-APNNs},i}^{\varepsilon,nn} &= \lambda_i \frac{1}{N_{\text{tb}}} \sum_{j=1}^{N_{\text{tb}}} \left(\left| f_{\theta}^{nn}(0, r_j^{\text{tb}}, \Omega_j^{\text{tb}}) - f_0(r_j^{\text{tb}}, \Omega_j^{\text{tb}}) \right|^2 + \left| \langle f_{\theta}^{nn} \rangle(0, r_j^{\text{tb}}) - \langle f_0 \rangle(r_j^{\text{tb}}) \right|^2 \right), \\
L_{\text{PINNs},c}^{\varepsilon,nn} &= \lambda_c \frac{1}{N_c} \sum_{j=1}^{N_c} \left| \varepsilon \partial_t \langle f_{\theta}^{nn} \rangle(t_j^c, r_j^c) + \langle \Omega \cdot n_r f_{\theta}^{nn} \rangle(t_j^c, r_j^c) + \varepsilon \alpha \langle f_{\theta}^{nn} \rangle(t_j^c, r_j^c) - \varepsilon G \right|^2,
\end{aligned}$$

where $\{(t_j^{\text{sb}}, r_j^{\text{sb}}, \Omega_j^{\text{sb}})\}_{j=1}^{N_{\text{sb}}}$ are Sobol sequence points on boundary, $\{(0, r_j^{\text{tb}}, \Omega_j^{\text{tb}})\}_{j=1}^{N_{\text{tb}}}$ the Sobol points at $t = 0$, and $\{(t_j^c, r_j^c)\}_{j=1}^{N_c}$ the Sobol sequence points on $\tau \times D$.

We define the total empirical loss function for MA-APNNs:

$$L_{\text{MA-APNNs}}^{\varepsilon,nn} = L_{\text{MA-APNNs},g}^{\varepsilon,nn} + L_{\text{MA-APNNs},b}^{\varepsilon,nn} + L_{\text{MA-APNNs},i}^{\varepsilon,nn} + L_{\text{PINNs},c}^{\varepsilon,nn}. \quad (3.6)$$

Now, we are in the position to find the solution of the minimization problem

$$\theta^* = \arg \min_{\theta} (L_{\text{MA-APNNs}}^{\varepsilon,nn}). \quad (3.7)$$

In the simulation, we adopt appropriate optimization algorithms (e.g. Adam, LBFGS) to minimize the non-convex loss function $L_{\text{MA-APNNs}}^{\varepsilon,nn}$.

4. Numerical Experiments

We present several examples to compare the performance of PINNs, APNNs based on micro-macro decomposition[4] and MA-APNNs for solving multiscale LRTEs. The result confirms that the MA-APNNs have an advantage in solving diffusive scaling

problems. In the numerical simulations, we adopt the Adam algorithm with an initial learning rate 0.001 in the optimization process of the loss function and use $\tanh x$ as the activation function of all hidden layers. Unless stated otherwise, the reference solution f^{Ref} is obtained by the UGKS[29]. We calculate L^2 -norm relative errors of $\rho = \langle f \rangle$:

$$L_{\text{error}}^2(\rho) = \frac{\|\langle f^{\text{nm}} \rangle - \langle f^{\text{Ref}} \rangle\|_{L^2(\tau \times D)}}{\|\langle f^{\text{Ref}} \rangle\|_{L^2(\tau \times D)}}.$$

4.1. One-dimensional problems

4.1.1. Kinetic regime with isotropic boundary condition

Consider the 1D LRTes in the kinetic regime with $\varepsilon = 1$. We set $D = (0, 1)$, $\tau = (0, 4)$, $\sigma = 1$, $\alpha = G = 0$, and enforce the isotropic inflow boundary condition (2.8) and the initial condition (2.1c) with

$$f_L(t, \mu > 0) = 0, \quad f_R(t, \mu < 0) = 1, \quad f_0(x, \mu) = 0.$$

We take the network structure $f_{\theta}^{\text{nm}} = [3, 40, 40, 40, 40, 1]$ with the hyperparameters $\beta_1 = 10^{-3}$, $\beta_2 = 10^{-4}$, $\lambda_b = \lambda_i = 1$, $\lambda_c = 0$. The number of the batch training samples is $(N_{\text{int}}, N_{\text{sb}}, N_{\text{tb}}) = (2000, 400 \times 2, 400)$. According to the content introduced in [4], we set APNNs here as $g_{\theta_1}^{\text{nm}} = [3, 40, 40, 40, 40, 1]$, $\rho_{\theta_2}^{\text{nm}} = [2, 40, 40, 40, 40, 1]$. The reference solution and the prediction obtained by PINNs, APNNs and MA-APNNs are plotted in Figure 4.1. The L^2 errors and training time of PINNs, APNNs and MA-APNNs are displayed in Table 4.1. All three methods are well applicable to the LRTes in the kinetic regime.

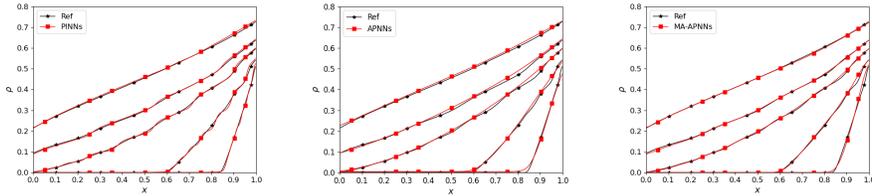


Fig. 4.1. Kinetic regime with $\varepsilon = 1$. The density ρ at times $t = 0.15, 0.4, 1.0, 1.6, 4.0$. (Left) Ref v.s. PINNs. (Middle) Ref v.s. APNNs. (Right) Ref v.s. MA-APNNs.

Table 4.1. Kinetic regime with $\varepsilon = 1$. The errors and training time of PINNs, APNNs and MA-APNNs.

| $L_{\text{error}}^2(\rho)$ | $t = 0.15$ | $t = 0.4$ | $t = 1.0$ | $t = 1.6$ | $t = 4.0$ | Training time |
|----------------------------|------------|-----------|-----------|-----------|-----------|---------------|
| PINNs | 3.48e-02 | 2.86e-02 | 1.41e-02 | 1.20e-02 | 1.01e-02 | 31min |
| APNNs | 6.19e-02 | 3.89e-02 | 2.66e-02 | 2.49e-02 | 1.47e-02 | 1h 18min |
| MA-APNNs | 6.51e-02 | 2.86e-02 | 1.22e-02 | 8.60e-03 | 5.06e-03 | 2h 15min |

4.1.2. Initial layer problem with periodic boundary condition

We adopt the same settings of the previous example, except that we set the temporal domain $\tau = (0, 1)$, replace the inflow boundary condition by the periodic boundary condition (2.9), and set the initial value

$$f_0(x, \mu) = \frac{1 + \cos(4\pi x)}{\sqrt{2\pi}} e^{-\frac{\mu^2}{2}}.$$

The solution of this 1D LRTEs is known to have an initial layer near $t = 0$. We utilize MA-APNNs with soft and hard constraints of the periodic boundary condition and APNNs with boundary hard constraints to solve this problem.

We set $(N_{\text{int}}, N_{\text{sb}}, N_{\text{tb}}) = (2000, 500 \times 2, 1000)$, and $(\beta_1, \beta_2) = (10^{-3}, 10^{-5})$. Under the soft constraints of the boundary condition, we choose $f_{\theta}^{nn} = [3, 40, 40, 40, 40, 1]$, $(\lambda_b, \lambda_c, \lambda_i) = (1, 1, 1000)$. Under the hard constraints, we add a custom layer between the input and the first hidden layer in MA-APNNs and APNNs to transform x to $(\sin 2\pi x, \cos 2\pi x)$ [55, 56] such that the periodic boundary condition is satisfied for f_{θ}^{nn} in MA-APNNs and $g_{\theta_1}^{nn}, \rho_{\theta_2}^{nn}$ in APNNs[4]. We take $f_{\theta}^{nn}/g_{\theta_1}^{nn} = [4, 40, 40, 40, 40, 1]$, $\rho_{\theta_2}^{nn} = [3, 40, 40, 40, 40, 1]$ and $(\lambda_b, \lambda_c, \lambda_i) = (0, 1, 1000)$. The numerical solutions of APNNs and MA-APNNs are plotted in Figure 4.2, together with the reference solution. The L^2 errors and training time are shown in Table 4.2. We observe that the MA-APNNs with hard constraints of the boundary condition perform slightly better than the soft constraints. APNNs and MA-APNNs with hard constraints both perform well under this case.

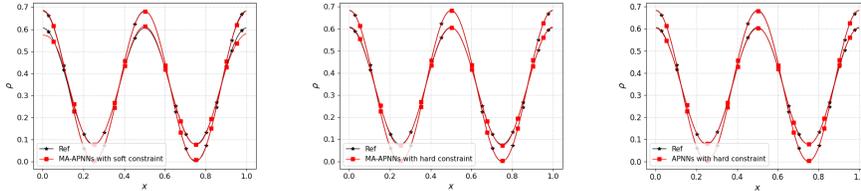


Fig. 4.2. Initial layer with $\varepsilon = 1$. The density ρ at times $t = 0.0, 0.1$. (Left) Ref v.s. MA-APNNs with boundary soft constraint. (Middle) Ref v.s. MA-APNNs with boundary hard constraint. (Right) Ref v.s. APNNs with boundary hard constraint.

Table 4.2. Initial layer with $\varepsilon = 1$. The errors and training time of MA-APNNs with boundary soft and hard constraint and APNNs with boundary hard constraint.

| $L^2_{\text{error}}(\rho)$ | $t = 0.0$ | $t = 0.1$ | Training time |
|-------------------------------|-----------|-----------|---------------|
| MA-APNNs with soft constraint | 6.38e-03 | 2.55e-02 | 3h 10min |
| MA-APNNs with hard constraint | 2.31e-03 | 7.31e-03 | 3h 13min |
| APNNs with hard constraint | 2.89e-03 | 6.28e-03 | 1h 29min |

4.1.3. Diffusion regime with a constant scattering frequency

Consider the 1D LRTes under the following settings:

$$D = (0, 1), \quad \tau = (0, 2), \quad f_L(t, \mu > 0) = 1, \quad f_R(t, \mu < 0) = 0, \\ f_0(x, \mu) = 0, \quad \sigma = 1, \quad \alpha = 0, \quad G = 0, \quad \varepsilon = 10^{-8}.$$

Since the Knudsen number ε is extremely small (in diffusion scale), the solution is close to the diffusion limit of (2.14). We choose $f_{\theta}^{nn} = [3, 40, 40, 40, 40, 1]$ in PINNs and MA-APNNs, $g_{\theta_1}^{nn} = [3, 40, 40, 40, 40, 1]$, $\rho_{\theta_2}^{nn} = [2, 40, 40, 40, 40, 1]$ in APNNs[4] and set $(N_{\text{int}}, N_{\text{sb}}, N_{\text{tb}}) = (1000, 200 \times 2, 200)$ and $(\beta_1, \beta_2, \lambda_b, \lambda_i, \lambda_c) = (10^{-5}, 10^{-16}, 10, 1, 0)$. The prediction of PINNs, APNNs and MA-APNNs are presented in Figure 4.3. The L^2 errors and training time are listed in Table 4.3. The experimental result indicates that the vanilla PINNs fail to approximate the diffusion scale problem, and APNNs do badly near initial times while the proposed MA-APNN method is well applicable to the case with very tiny ε .

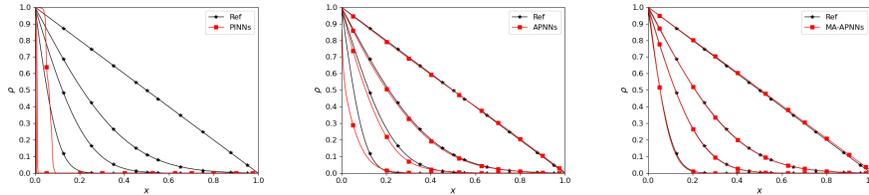


Fig. 4.3. Diffusion regime with $\varepsilon = 10^{-8}$. The density ρ at times $t = 0.01, 0.05, 0.15, 2.00$. (Left) Ref v.s. PINNs. (Middle) Ref v.s. APNNs. (Right) Ref v.s. MA-APNNs.

Table 4.3. Diffusion regime with $\varepsilon = 10^{-8}$. The errors and training time of PINNs, APNNs and MA-APNNs.

| $L^2_{\text{error}}(\rho)$ | $t = 0.01$ | $t = 0.05$ | $t = 0.15$ | $t = 2.00$ | Training time |
|----------------------------|------------|------------|------------|------------|---------------|
| PINNs | 9.75e-01 | 9.71e-01 | 9.65e-01 | 8.96e-01 | 17min 34s |
| APNNs | 3.81e-01 | 8.18e-02 | 2.37e-02 | 8.95e-03 | 2h 45min |
| MA-APNNs | 3.16e-02 | 5.79e-03 | 5.30e-03 | 1.35e-02 | 1h 24min |

4.1.4. Diffusion regime with a variable scattering frequency

Use the same settings as the previous example except that we set the scattering frequency $\sigma = 1 + (10x)^2$ and the Knudsen number $\varepsilon = 10^{-4}$. In numerical simulation, the network structures are the same as in 4.1.3 and we take $(N_{\text{int}}, N_{\text{sb}}, N_{\text{tb}}) = (1000, 200 \times 2, 400)$, $(\beta_1, \beta_2, \lambda_b, \lambda_i, \lambda_c) = (10^{-5}, 10^{-16}, 1, 1, 0)$. We see that the prediction of APNNs and MA-APNNs coincide with the reference solutions (see Figure 4.4 (middle and right)) while PINNs fail the test (see Figure 4.4 (left)). The L^2 errors and training time are shown in Table 4.4. MA-APNNs have less training time with comparable prediction accuracy to APNNs[4].

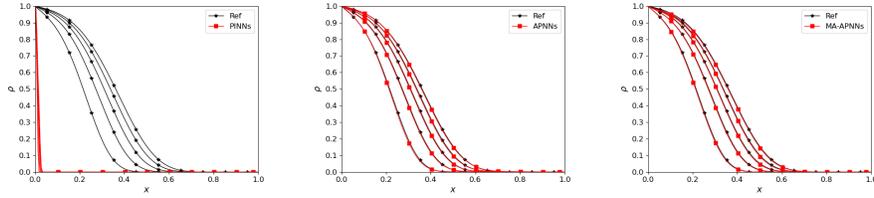


Fig. 4.4. Diffusion regime with $\varepsilon = 10^{-4}$. The density ρ at times $t = 0.2, 0.4, 0.6, 0.8, 1.0$. (Left) Ref v.s. PINNs. (Middle) Ref v.s. APNNs. (Right) Ref v.s. MA-APNNs.

Table 4.4. Diffusion regime with $\varepsilon = 10^{-4}$. The errors and training time of PINNs, APNNs and MA-APNNs.

| $L^2_{\text{error}}(\rho)$ | $t = 0.2$ | $t = 0.4$ | $t = 0.6$ | $t = 0.8$ | $t = 1.0$ | Training time |
|----------------------------|-----------|-----------|-----------|-----------|-----------|---------------|
| PINNs | 9.65e-01 | 9.66e-01 | 9.66e-01 | 9.65e-01 | 9.64e-01 | 28min 36s |
| APNNs | 1.46e-02 | 1.01e-02 | 9.18e-03 | 9.40e-03 | 8.06e-03 | 4h 26min |
| MA-APNNs | 1.28e-02 | 9.42e-03 | 8.25e-03 | 9.49e-03 | 9.76e-03 | 2h 25min |

4.1.5. Intermediate regime with a variable scattering frequency and source term

We replace the values of the source term and Knudsen number of the previous example by $G = 1$ and $\varepsilon = 10^{-2}$, respectively. We adopt the same settings of the network structure and hyperparameters except for $\beta_2 = 10^{-12}$. We apply PINNs, APNNs and MA-APNNs to solve this problem. Specially, we set the activation functions of these networks' output layers as unit functions. The numerical results are shown in Figure 4.5 and Table 4.5. Together with the above examples, we are convinced that APNNs and MA-APNNs are able to resolve the multiscale characteristics of the 1D LRTEs, while the vanilla PINNs can only treat the case $\varepsilon = O(1)$.

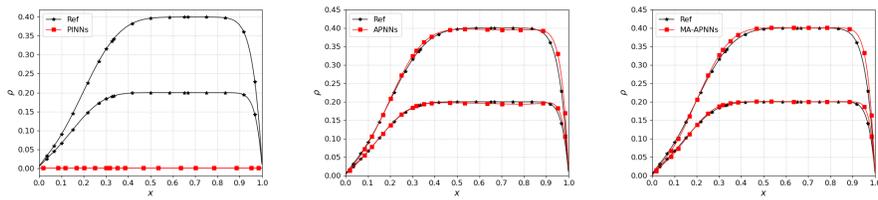


Fig. 4.5. Intermediate regime with $\varepsilon = 10^{-2}$. The density ρ at times $t = 0.2, 0.4$. (Left) Ref v.s. PINNs. (Middle) Ref v.s. APNNs. (Right) Ref v.s. MA-APNNs.

4.2. Two-dimensional problems

We apply the MA-APNNs to solve the 2D LRTEs under the following settings

$$D = (0, 1)^2, \quad \tau = (0, 1), \quad \sigma = 1, \quad \alpha = 0, \quad G = 1.$$

Table 4.5. Intermediate regime with $\varepsilon = 10^{-2}$. The errors and training time of PINNs, APNNs and MA-APNNs.

| $L^2_{\text{error}}(\rho)$ | $t = 0.2$ | $t = 0.4$ | Training time |
|----------------------------|-----------|-----------|---------------|
| PINNs | 9.98e-01 | 9.99e-01 | 30min 9s |
| APNNs | 2.87e-02 | 3.24e-02 | 4h 37min |
| MA-APNNs | 2.70e-02 | 3.44e-02 | 2h 25min |

The homogeneous initial and inflow boundary conditions are imposed, i.e., $f_0 = 0$ and $f_B|_{\Gamma_-} = 0$.

4.2.1. Kinetic regime ($\varepsilon = 1$)

We set $\tilde{f}_\theta^{nn} = [5, 40, 40, 40, 40, 1]$ and the boundary and initial conditions are enforced as hard constraints, that is,

$$f_\theta^{nn} = t(x + \text{Relu}(-\xi)^2)(1 - x + \text{Relu}(\xi)^2)(y + \text{Relu}(-\eta)^2)(1 - y + \text{Relu}(\eta)^2)\tilde{f}_\theta^{nn}.$$

And we take $N_{\text{int}} = 2000$ and $(\beta_1, \beta_2, \lambda_b, \lambda_i, \lambda_c) = (10^{-6}, 10^{-7}, 0, 0, 0)$. The reference solution and the prediction obtained by MA-APNNs are plotted in Figure 4.6. The L^2 relative errors at times $t = 0.4, 1.0$ are $2.91e - 02$ and $2.84e - 02$, respectively.

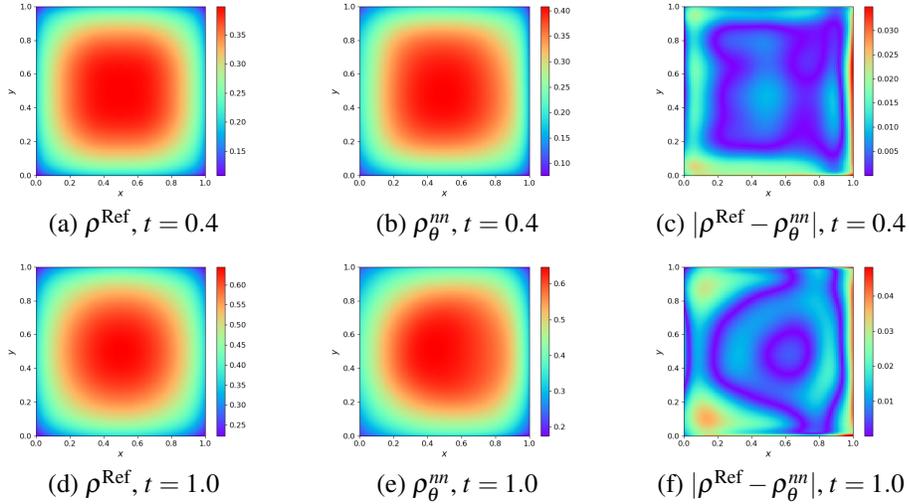


Fig. 4.6. The 2D LRTEs in the kinetic regime ($\varepsilon = 1$).

4.2.2. Diffusion regime ($\varepsilon = 10^{-8}$)

We use the same network structure and parameters as the previous example except that $\beta_1 = 10^{-5}, \beta_2 = 10^{-16}$. The reference solution is obtained by the finite difference

discretization of the diffusion limit equation. We show the reference solution and the prediction of MA-APNNs in Figure 4.7. The L^2 relative errors at times $t = 0.1, 0.8$ are $3.98e - 02$ and $4.79e - 02$, respectively.

The above two examples show the ability of MA-APNNs to solve the 2D LRTEs in both the kinetic and diffusion regimes.

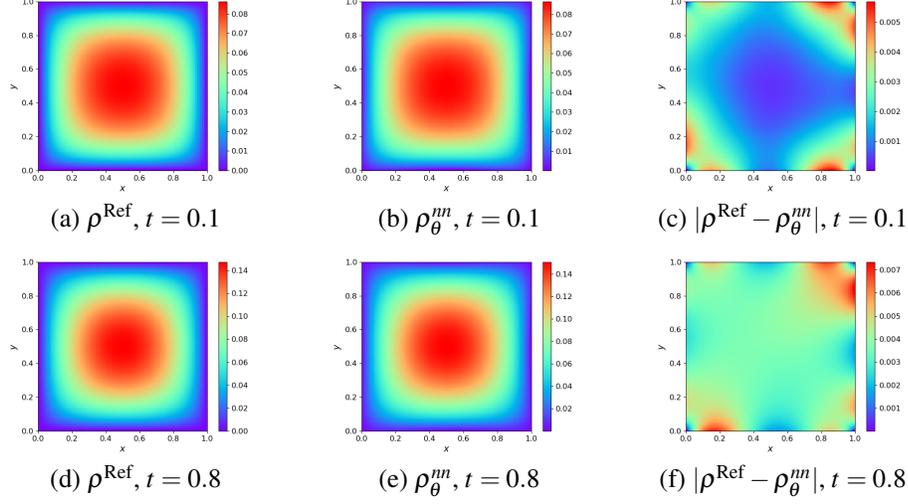


Fig. 4.7. The 2D LRTEs in the diffusion regime ($\varepsilon = 10^{-8}$).

4.3. Uncertainty quantification (UQ) problems

Let $\{z_i\}_{i=1}^d$ be the random variables with identical independent uniform distribution in $[-1, 1]$, i.e., $z_i \sim U([-1, 1])$. We apply the MA-APNNs to solve the uncertainty quantification problem (2.12) in the spatial, temporal and angular domain $D \times \tau \times S^2 = (0, 1) \times (0, 1) \times [-1, 1]$, with $\alpha = 0$, and the zero initial conditions,

$$f(0, x, \mu, z) = 0.$$

The model is a spatial 1D LRTEs with the high-dimensional random input $z \sim U([-1, 1]^d)$. The simulation is conducted for two cases with various scattering coefficients $\sigma(z)$.

Problem I. We consider the case in the kinetic regime ($\varepsilon = 1$) with the cosine scattering coefficient:

$$\sigma(z) = 1 + 0.1 \sum_{i=1}^{10} \cos(\pi z_i), \quad (d = 10).$$

The source term is set as

$$G = \frac{x(1-x)}{22} \left(\mu + 11 + \sum_{i=1}^{10} z_i \right) + \frac{\mu t(1-2x)}{22\varepsilon} \left(\mu + 11 + \sum_{i=1}^{10} z_i \right) + \frac{1}{\varepsilon^2} \sigma(z) t x (1-x) \mu.$$

We impose the boundary conditions

$$f(t, 0, \mu, \mathbf{z}) = f(t, 1, \mu, \mathbf{z}) = 0, \quad \mu \in [-1, 1].$$

The exact solution is given by

$$f(t, x, \mu, \mathbf{z}) = \frac{tx(1-x)}{22} \left(\mu + 11 + \sum_{i=1}^{10} z_i \right), \quad \rho(t, x, \mathbf{z}) = \frac{tx(1-x)}{22} \left(11 + \sum_{i=1}^{10} z_i \right).$$

Note that the expectation of ρ with respect to the random variable \mathbf{z} is

$$\mathbf{E}(\rho) = \frac{1}{2}tx(1-x).$$

We take $\tilde{f}_\theta^{nn} = [13, 40, 40, 40, 40, 1]$, and the boundaries and initial conditions are enforced by hard constraints, i.e., $f_\theta^{nn} = tx(1-x)\tilde{f}_\theta^{nn}$. And we set $N_{\text{int}} = 5000$, $\lambda_b = \lambda_i = \lambda_c = 0$, $\beta_1 = 10^{-5}$, $\beta_2 = 10^{-7}$. We carry out 10^4 times simulations for $(z_1, z_2, \dots, z_{10})$ and compute the expectation of ρ at $t = 0.2, 0.4, 0.6$. The results are plotted in Figure 4.8, which agrees well with the exact $\mathbf{E}(\rho)$. Table 4.6 shows the L^2 errors of $\mathbf{E}(\rho)$.

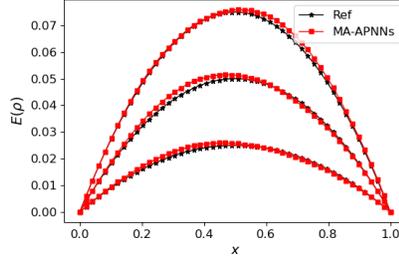


Fig. 4.8. The expectation $\mathbf{E}(\rho)$ at $t = 0.2, 0.4, 0.6$.

Table 4.6. High dimensional transport regime. The errors of MA-APNNs.

| L_{error}^2 | $t = 0.2$ | $t = 0.4$ | $t = 0.6$ |
|----------------------|-----------|-----------|-----------|
| MA-APNNs | 4.49e-02 | 3.63e-02 | 2.12e-02 |

Problem II. We consider the case in the diffusion regime ($\varepsilon = 10^{-5}$) with the sine scattering function[51]:

$$\sigma(\mathbf{z}) = 1 + 0.1 \prod_{i=1}^{20} \sin(\pi z_i), \quad (d = 20).$$

We impose the inflow boundary conditions and the zero source term,

$$f(t, 0, \mu > 0, \mathbf{z}) = 1, \quad f(t, 1, \mu < 0, \mathbf{z}) = 0, \quad G = 0.$$

We choose $f_{\theta}^{mn} = [23, 40, 40, 40, 40, 1]$, $(N_{\text{int}}, N_{\text{sb}}, N_{\text{tb}}) = (2048, 768 \times 2, 1536)$ and set $(\beta_1, \beta_2, \lambda_b, \lambda_i, \lambda_c) = (10^{-5}, 10^{-16}, 1, 1, 0)$ in our simulation. We evaluate the expectation of ρ at times $t = 0.05, 0.1$. The results are shown in Figure 4.9, which is very similar to the solutions obtained by [51]. The above two examples indicate the applicability of MA-APNNs to the 1D LRTEs with high-dimensional random input in both kinetic and diffusion regimes.

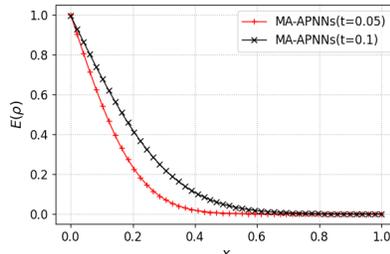


Fig. 4.9. Plot of the density ρ by taking expectation for z at $t = 0.05, 0.1$. MA-APNNs with $\varepsilon = 10^{-5}$.

5. Conclusion

In this work, we derive the macroscopic auxiliary equation of the linear radiative transfer equations (LRTEs) and propose the macroscopic auxiliary asymptotic-preserving neural networks (MA-APNNs) to tackle the multiscale characteristics and high dimensionality in the numerical computation of LRTEs. To validate the advantages of MA-APNNs compared with the vanilla PINNs and APNNs in [4], we conduct the simulation of the 1D LRTEs in kinetic, diffusive, and intermediate regimes. We also present the numerical examples of the 2D LRTEs and 1D LRTEs with high-dimensional random variables, in both kinetic and diffusive regimes, showing the applicability of MA-APNNs to deal with the multiscale nature and high dimensionality. Our future study will focus on developing MA-APNNs to solve the Vlasov-Poisson-Fokker-Planck system in the high-field regime.

Acknowledgements

This work was supported by the National Key R&D Program (2020YFA0712200) for S. Jiang, and by Beijing Natural Science Foundation (Grant No. Z230003) for W. Sun, by NSFC (No. 12071060 & No. 62231016) for L. Xu and by NSFC General Projects (No. 12171071) for G. Zhou.

References

- [1] F. Bouchut, F. Golse, M. Pulvirenti, Kinetic equations and asymptotic theory, HAL 2000.

- [2] G. Dimarco, L. Pareschi, Numerical methods for kinetic equations, *Acta Numerica* 23 (2014) 369–520.
- [3] S. Jin, Asymptotic preserving (ap) schemes for multiscale kinetic and hyperbolic equations: a review, *Lecture notes for summer school on methods and models of kinetic theory (M&MKT), Porto Ercole (Grosseto, Italy)* (2010) 177–216.
- [4] S. Jin, Z. Ma, K. Wu, Asymptotic-preserving neural networks for multiscale time-dependent linear transport equations, *Journal of Scientific Computing* 94 (3) (2023) 57.
- [5] S. Chandrasekhar, *Radiative transfer*, 2013.
- [6] E. Larsen, G. Pomraning, V. Badham, Asymptotic analysis of radiative transfer problems, *Journal of Quantitative Spectroscopy and Radiative Transfer* 29 (4) (1983) 285–310.
- [7] M. L. Adams, P. F. Nowak, Asymptotic analysis of a computational method for time-and frequency-dependent radiative transfer, *Journal of Computational Physics* 146 (1) (1998) 366–403.
- [8] S. Jin, L. Pareschi, *Uncertainty quantification for hyperbolic and kinetic equations*, Springer, 2017.
- [9] J. Hu, S. Jin, A stochastic galerkin method for the boltzmann equation with uncertainty, *Journal of Computational Physics* 315 (2016) 150–168.
- [10] S. Jin, J.-G. Liu, Z. Ma, Uniform spectral convergence of the stochastic galerkin method for the linear transport equations with random inputs in diffusive regime and a micro–macro decomposition-based asymptotic-preserving method, *Research in the Mathematical Sciences* 4 (1) (2017) 15.
- [11] W. E, *Principles of multiscale modeling*, Cambridge University Press, 2011.
- [12] P. Degond, S. Jin, A smooth transition model between kinetic and diffusion equations, *SIAM journal on numerical analysis* 42 (6) (2005) 2671–2687.
- [13] J. A. Fleck Jr, J. Cummings Jr, An implicit monte carlo scheme for calculating time and frequency dependent nonlinear radiation transport, *Journal of Computational Physics* 8 (3) (1971) 313–342.
- [14] N. Gentile, Implicit monte carlo diffusion—an acceleration method for monte carlo time-dependent radiative transfer simulations, *Journal of Computational Physics* 172 (2) (2001) 543–571.
- [15] R. G. McClarren, T. J. Urbatsch, A modified implicit monte carlo method for time-dependent radiative transfer with adaptive material coupling, *Journal of Computational Physics* 228 (16) (2009) 5669–5686.

- [16] Y. Shi, X. Han, W. Sun, P. Song, A continuous source tilting scheme for radiative transfer equations in implicit monte carlo, *Journal of Computational and Theoretical Transport* 50 (1) (2020) 1–26.
- [17] J. D. Densmore, Asymptotic analysis of the spatial discretization of radiation absorption and re-emission in implicit monte carlo, *Journal of Computational Physics* 230 (4) (2011) 1116–1133.
- [18] E. W. Larsen, J. Keller, Asymptotic solution of neutron transport problems for small mean free paths, *Journal of Mathematical Physics* 15 (1) (1974) 75–81.
- [19] E. W. Larsen, Solutions of the steady, one-speed neutron transport equation for small mean free paths, *Journal of Mathematical Physics* 15 (3) (1974) 299–305.
- [20] E. W. Larsen, Neutron transport and diffusion in inhomogeneous media. i, *Journal of Mathematical Physics* 16 (7) (1975) 1421–1427.
- [21] E. W. Larsen, Neutron transport and diffusion in inhomogeneous media. ii, *Nuclear science and engineering* 60 (4) (1976) 357–368.
- [22] E. W. Larsen, J. E. Morel, W. F. Miller Jr, Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes, *Journal of Computational Physics* 69 (2) (1987) 283–324.
- [23] J. Morel, et al., Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii, *Journal of Computational Physics* 83 (1) (1989) 212–236.
- [24] S. Jin, D. Levermore, The discrete-ordinate method in diffusive regimes, *Transport theory and statistical physics* 20 (5-6) (1991) 413–439.
- [25] S. Jin, C. D. Levermore, Fully-discrete numerical transfer in diffusive regimes, *Transport theory and statistical physics* 22 (6) (1993) 739–791.
- [26] S. Jin, L. Pareschi, G. Toscani, Uniformly accurate diffusive relaxation schemes for multiscale transport equations, *SIAM Journal on Numerical Analysis* 38 (3) (2000) 913–936.
- [27] A. Klar, An asymptotic-induced scheme for nonstationary transport equations in the diffusive limit, *SIAM Journal on Numerical Analysis* 35 (3) (1998) 1073–1094.
- [28] S. Jin, Efficient asymptotic-preserving (ap) schemes for some multiscale kinetic equations, *SIAM Journal on Scientific Computing* 21 (2) (1999) 441–454.
- [29] L. Mieussens, On the asymptotic preserving property of the unified gas kinetic scheme for the diffusion limit of linear kinetic models, *Journal of Computational Physics* 253 (2013) 138–156.

- [30] M. Lemou, L. Mieussens, A new asymptotic preserving scheme based on micro-macro formulation for linear kinetic equations in the diffusion limit, *SIAM Journal on Scientific Computing* 31 (1) (2008) 334–368.
- [31] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, *Proceedings of the National Academy of Sciences* 115 (34) (2018) 8505–8510.
- [32] L. Zhang, T. Luo, Y. Zhang, W. E, Z.-Q. J. Xu, Z. Ma, Mod-net: A machine learning approach via model-operator-data network for solving pdes, *Communications in Computational Physics* 32 (2) (2022) 299–335.
- [33] W. E, J. Han, A. Jentzen, Algorithms for solving high dimensional pdes: from nonlinear monte carlo to machine learning, *Nonlinearity* 35 (1) (2021) 278.
- [34] H. J. Hwang, J. W. Jang, H. Jo, J. Y. Lee, Trend to equilibrium for the kinetic fokker-planck equation via the neural network approach, *Journal of Computational Physics* 419 (2020) 109665.
- [35] C. Beck, M. Hutzenthaler, A. Jentzen, B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, arXiv preprint arXiv:2012.12348.
- [36] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving non-linear partial differential equations, *Journal of Computational physics* 378 (2019) 686–707.
- [37] J. Sirignano, K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, *Journal of computational physics* 375 (2018) 1339–1364.
- [38] B. Yu, et al., The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, *Communications in Mathematics and Statistics* 6 (1) (2018) 1–12.
- [39] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, *Journal of Computational Physics* 411 (2020) 109409.
- [40] Z. Cai, J. Chen, M. Liu, Least-squares relu neural network (lsnn) method for linear advection-reaction equation, *Journal of Computational Physics* 443 (2021) 110514.
- [41] L. Lyu, Z. Zhang, M. Chen, J. Chen, Mim: A deep mixed residual method for solving high-order partial differential equations, *Journal of Computational Physics* 452 (2022) 110930.
- [42] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review, *International Journal of Automation and Computing* 14 (5) (2017) 503–519.

- [43] P. Grohs, F. Hornung, A. Jentzen, P. Von Wurstemberger, A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations, arXiv preprint arXiv:1809.02362.
- [44] L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, Deepxde: A deep learning library for solving differential equations, SIAM review 63 (1) (2021) 208–228.
- [45] S. Mishra, R. Molinaro, Physics informed neural networks for simulating radiative transfer, Journal of Quantitative Spectroscopy and Radiative Transfer 270 (2021) 107705.
- [46] Y. Lu, L. Wang, W. Xu, Solving multiscale steady radiative transfer equation using neural networks with uniform stability, Research in the Mathematical Sciences 9 (3) (2022) 1–29.
- [47] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deepnets, Science advances 7 (40) (2021) eabi8605.
- [48] Z. Chen, L. Liu, L. Mu, Solving the linear transport equation by a deep neural network approach, American Institute of Mathematical Sciences 15 (4) (2022) 669–686.
- [49] L. Liu, T. Zeng, Z. Zhang, A deep neural network approach on solving the linear transport model under diffusive scaling, arXiv preprint arXiv:2102.12408.
- [50] H. Li, S. Jiang, W. Sun, L. Xu, G. Zhou, A model-data asymptotic-preserving neural network method based on micro-macro decomposition for gray radiative transfer equations, Communications in Computational Physics (2023) (accepted).
- [51] S. Jin, Z. Ma, K. Wu, Asymptotic-preserving neural networks for multiscale kinetic equations, arXiv preprint arXiv:2306.15381.
- [52] K. Wu, X.-b. Yan, S. Jin, Z. Ma, Asymptotic-preserving convolutional deepnets capture the diffusive behavior of the multiscale linear transport equations, arXiv preprint arXiv:2306.15891.
- [53] M. Ghattassi, X. Huo, N. Masmoudi, On the diffusive limits of radiative heat transfer system i: well-prepared initial and boundary conditions, SIAM Journal on Mathematical Analysis 54 (5) (2022) 5335–5387.
- [54] I. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, USSR Computational Mathematics and Mathematical Physics 7 (4) (1967) 86–112.
- [55] R. Mojjani, M. Balajewicz, P. Hassanzadeh, Kolmogorov n-width and lagrangian physics-informed neural networks: a causality-conforming manifold for convection-dominated pdes, Computer Methods in Applied Mechanics and Engineering 404 (2023) 115810.

- [56] S. Dong, N. Ni, A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks, *Journal of Computational Physics* 435 (2021) 110242.