

Research Article

PINN-CDR: A Neural Network-Based Simulation Tool for Convection-Diffusion-Reaction Systems

Qingzhi Hou ¹, Honghan Du ², Zewei Sun ², Jianping Wang ³, Xiaojing Wang ⁴, and Jianguo Wei ²

¹State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin 300350, China

²College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

³Key Laboratory of Comprehensive and Highly Efficient Utilization of Salt Lake Resources, Qinghai Institute of Salt Lakes, Chinese Academy of Sciences, Xining 810008, China

⁴School of Chemical Engineering, Tianjin University, Tianjin 300350, China

Correspondence should be addressed to Xiaojing Wang; tdxjwang@tju.edu.cn

Received 3 September 2022; Revised 24 April 2023; Accepted 27 May 2023; Published 16 August 2023

Academic Editor: Alexander Hošovský

Copyright © 2023 Qingzhi Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a discretization-free approach based on the physics-informed neural network (PINN) is proposed for solving the forward and inverse problems governed by the nonlinear convection-diffusion-reaction (CDR) systems. By embedding physical information described by the CDR system in the feedforward neural networks, PINN is trained to approximate the solution of the system without the need of labeled data. The good performance of PINN in solving the forward problem of the nonlinear CDR systems is verified by studying the problems of gas-solid adsorption and autocatalytic reacting flow. For CDR systems with different Péclet number, PINN can largely eliminate the numerical diffusion and unphysical oscillations in traditional numerical methods caused by high Péclet number. Meanwhile, the PINN framework is implemented to solve the inverse problem of nonlinear CDR systems and the results show that the unknown parameters can be effectively recognized even with high noisy data. It is concluded that the established PINN algorithm has good accuracy, convergence, and robustness for both the forward and inverse problems of CDR systems.

1. Introduction

Reacting flow models play an important role in the simulation of many physical and chemical problems, such as the pollutant transport process in water and air [1], heat conduction process in flowing fluids [2], chromatography column in reactors [3], and high-speed eddy current in electromagnetic fields [4]. A reacting flow model is often composed of a group of convection-dominated partial differential equations (PDEs) with nonlinear source terms [5–7], which usually accompanies autocatalytic reactions. A typical reacting flow model is the convection-diffusion-reaction (CDR) system [8], which is one kind of basic PDE with nonlinear source terms of autocatalytic reactions [9]. The so-called autocatalytic reaction means that through mutation, the autocatalyst will be transformed into another

form of substance, and this new substance can also undergo an autocatalytic reaction at the same time, and eventually lead to competition between the new substance and the original autocatalyst [10]. Due to the complexity of autocatalytic reactions, some of the parameters such as kinetic parameters, mutation parameters and convective diffusion coefficients are often unavailable. Therefore, the CDR problem can be further divided into the forward and inverse problems. The forward problem refers to solving the concentration of reactants at various points within a reaction for which all boundary conditions and medium parameters are known, while the inverse problem refers to the recognition of media parameters by limited known data. By accurately solving these reacting flow models, the reaction problems in chemistry, physics, electromagnetism, and fluids can be analyzed, and suitable reaction units can be designed to

optimize the process control schemes. Therefore, it is important to develop an accurate and efficient simulation tool for solving both the forward and the inverse problems of CDR systems.

For the forward problem of CDR involved in fluids, most of the traditional methods are numerical ones. Numerical schemes play a key role in the study of reacting flows and a large variety of efficient numerical methods have been developed, including finite difference [11], finite volume [12], finite element [13], and spectral methods [14]. The core of these numerical methods is to use some discrete structure to reduce the infinite dimensional operators to a finite dimensional approximation problem, that is, to divide a large space-time region into multiple, small, and simple regions that are easily processed by computers. They are used to numerically solve different types of PDEs for large variety of static and dynamic problems. However, these numerical methods are often computationally cumbersome, especially for the problems with moving steep front and complex geometries. Moreover, mesh generation usually incurs a huge burden.

With the explosive growth of computing resources over the past decade, deep learning [15, 16], especially, deep neural network (DNN) [17, 18], has undergone revolutionary development. It is increasingly used to solve fundamental PDEs in physics and chemistry problems [19–22], with the help of the general approximation theorem of neural networks and their powerful characterization capabilities, i.e., excellent nonlinear approximation of the model by the combination of multiple hidden layers and nonlinear activation functions. Nevertheless, deep learning introduces new uncertainties and other drawbacks to reacting flow problems. For example, generating an accurate alternative model of a complex physical system usually requires an extremely large sample of data, which is often prohibitively expensive or infeasible to be obtained from measurements or simulations in reality.

In recent years, a DNN framework named physics-informed neural network (PINN) [23, 24] was developed. PINN does not require manually labeled training data. No validation and testing dataset are needed. This largely differs from other deep neural networks. Making the full-use of physical information as prior knowledge, PINN is trained with few or even no labeled data as surrogate models for accurate solution of PDEs [25]. Different from traditional mesh-based discretization methods, time and space derivatives in the PINN method are evaluated using automatic differentiation [26] of the DNN that does not involve any numerical discretization. Then, the DNN coefficients are computed by minimizing the loss function that is the sum of the residuals of both the PDEs and initial and boundary conditions [24]. In addition, the PINN solution defines a function over the continuous domain, rather than a discrete solution on a grid as in traditional methods. Only initial and boundary conditions are needed to train the

PINN to accurately approximate the solution of the equations.

For the inverse problem of CDR, systematic identification and thus reconstruction of source features from sparse data are very important. However, the inverse problem is always a high challenging topic, for which many difficulties exist, such as the inherent ill-posedness, data uncertainties, and sparse and noisy observation data. To solve the inverse problem, some methods have been developed such as genetic algorithms (GAs) [27], simulated annealing (SA) [28], adaptive simulated annealing (ASA) [29], artificial neural networks (ANNs) [30], and harmony search (HS) [31]. These methods are significantly affected by the noise in the observed data. Some nonclassical optimization algorithms, namely, the population-based ones (e.g., GA), also require a great number of evaluations of the objective function, which is computationally expensive [32]. However, PINN not only solves the forward problem according to the governing equation and the initial and boundary conditions, but also solves the inverse problem according to the sparse observed data. It learns the unknown parameters of the system from a small amount of given data and has strong robustness to noise [33, 34], which can be a new way for solving the inverse problem of CDR.

Due to the excellent capability of neural networks in describing complex relationship between inputs and outputs, PINN creates a new path to solve the forward and inverse problems involving nonlinear PDEs [25]. Noisy, sparse, and multifidelity data sets are easily handled by PINN. Nowadays, many problems difficult for traditional numerical methods are solved by using the PINN-based methods [35–37]. PINN has been successfully used for solving PDEs or complex PDE-based problems in various domains, such as fluid mechanics [38, 39], medical diagnosis [40, 41] and materialogy [42]. PINN has been applied to single reactant CDR problems with good results [43]. However, there are no researches on the application of PINN to CDR systems with multiple coupled reactants. The objective of this paper is to solve the forward and inverse problems for multireactant CDR systems. The PINN is applied to the gas-solid adsorption problem and the autocatalytic reacting flow problem in a tubular reactor. In the autocatalytic reacting flow problem, the use of \sin instead of the standard activation functions such as \tanh improves the learning ability of the network for high-frequency signals. In addition, the arithmetic examples examine the computational accuracy and stability of the algorithm for the inverse problem with different amounts of training data and different levels of noise. The results show that the PINN algorithm developed in this paper is a new, simple, and effective simulation tool for solving the forward and inverse problems of CDR systems.

The rest of the paper is organized as follows: In Section 2, the gas-solid adsorption model and autocatalytic reacting flow model are presented. In Section 3, the PINN method for

solving the forward and inverse problems is introduced, and in Section 4, PINN is tested by two CDR systems, including both the forward and inverse problems. Conclusions are given in Section 5.

2. Reactor Models with Multicomponent Reactant

In this section, two kinds of multicomponent reactant models are introduced, including the gas-solid adsorption model and the autocatalytic reacting flow model.

2.1. Gas-Solid Adsorption Model. The gas-solid adsorption column without the diffusion effect is described by one PDE for flow transport, one differential equation for mass transfer and one algebraic equation for equilibrium state is [44] as follows:

$$\begin{cases} C_t^G + \frac{1-\epsilon}{\epsilon} C_t^S + v C_x^G = 0, \\ C_t^S - k(C^I - C^S) = 0, \\ C^I - K C^G = 0, \end{cases} \quad (1)$$

where gas concentration (C^G), solid concentration (C^S), gas-solid interface concentration (C^I), void fraction (ϵ), superficial gas velocity (v), speed mass transfer coefficient (k), and speed equilibrium constant (K) are denoted. Indices t and x are used for temporal and spatial derivatives, i.e., C_t and C_x , respectively.

2.2. Autocatalytic Reacting Flow Model. The chemical reactor is of the tubular type, where chemical species flow from the inlet to the outlet in one pass. The chemical reaction model includes a cubic autocatalytic reaction in which the autocatalyst is assumed to undergo a mutational process that produces another form, and it can also undergo an autocatalytic reaction and thus compete with the original autocatalyst [45]. The model captures the fundamental steps encountered in many technically important biochemical and pharmaceutical applications, such as the birth-death process of bacteria and the interaction of drugs with some other biological agents or cells. The autocatalytic reaction consists of three reagents (substrate A, autocatalyst B, and mutant C) and is carried out according to the following reaction scheme [46]:

- (i) Replication of B: $A + 2B \xrightarrow{k_1} 3B$
- (ii) Death of B: $B \xrightarrow{k_2} P_1$
- (iii) Mutation of B into C: $A + 2B \xrightarrow{\alpha k_1} 2C + B$
- (iv) Replication of C: $A + 2C \xrightarrow{\beta k_1} 3C$
- (v) Death of C: $C \xrightarrow{k_2/\beta} P_2$

Specifically, both k_1 and k_2 represent the reacting rate constants, α is the mutation constant, and β is the mutation efficiency. For simplicity, we assume that the flow rate along

the reactor is constant. Then, the transport equation describing the three reactants in dimensionless form can be expressed as follows [45]:

$$\begin{cases} \frac{\partial U_1}{\partial T} + v \frac{\partial U_1}{\partial X} = D_1 \frac{\partial^2 U_1}{\partial X^2} + (1 - U_1) [(1 + \alpha) U_2^2 + \beta U_3^2], \\ \frac{\partial U_2}{\partial T} + v \frac{\partial U_2}{\partial X} = D_2 \frac{\partial^2 U_2}{\partial X^2} + (1 - \alpha)(1 - U_1) U_2^2 - \gamma U_2, \\ \frac{\partial U_3}{\partial T} + v \frac{\partial U_3}{\partial X} = D_3 \frac{\partial^2 U_3}{\partial X^2} + (1 - U_1) (\beta U_3^2 + 2\alpha U_2^2) - \frac{\gamma}{\beta} U_3, \end{cases} \quad (2)$$

where

$$\begin{aligned} U_1 &= \frac{u_f - u_1}{u_f}, \\ U_2 &= \frac{u_2}{u_f}, \\ U_3 &= \frac{u_3}{u_f}, \\ X &= \frac{x}{L}, \\ T &= k_1 u_f^2 t, \\ v &= \frac{a}{k_1 u_f^2 L}, \\ \gamma &= \frac{k_2}{k_1 u_f^2}. \end{aligned} \quad (3)$$

In the above equations, U_i ($i = 1, 2, 3$) is the dimensionless concentration of the reactants A, B, and C, u_f is the substrate concentration, X is the dimensionless reactor length, T is the dimensionless time, L represents the length of the tubular reactor, D_i ($i = 1, 2, 3$) is the dimensionless diffusion parameter, v is the dimensionless convection velocity, and γ is a dimensionless kinetic parameter.

3. Physics-Informed Neural Network (PINN)

The PINN is a machine learning framework based on DNN. It leverages the capabilities of DNN as universal function approximators. However, different from traditional deep learning algorithms, PINN restricts the set of acceptable solutions by enforcing the validity of PDE models governing the actual physics of the problem. This is achieved within a fully connected feedforward neural network architecture leveraging automatic differentiation techniques available in the TensorFlow learning package [47]. The basic idea of the PINN algorithm is to embed the governing equations of physical prior information (such as conservation quantity, invariance, and symmetry) into the loss function

corresponding to the network training, for speeding up the network training process and improving the accuracy and interpretability of the model prediction. PINN successfully integrates the physical information with neural networks. We consider the following forward problem of a PDE with the Dirichlet boundary condition:

$$\begin{cases} u_t + D_x(u; \lambda) = 0, & x \in \Omega, t \in [t_0, t_1], \\ u(t_0, x) = u_0(x), & x \in \Omega, \\ u(t, x) = g(t, x), & x \in \delta\Omega, t \in [t_0, t_1], \end{cases} \quad (4)$$

where u denotes the solution of the equation, D_x is the differential operator respect to x , λ is the parameter in the governing equation, which is a known constant in the forward problem, $\Omega \in R$ and $\delta\Omega$ denotes the boundary, $u_0(x)$ is the initial condition at $t = t_0$, and $g(t, x)$ is the Dirichlet boundary condition.

A typical PINN framework for solving the forward problem is shown in Figure 1. The input training points (x, t) consist of three parts as follows: initial sampling point $(x_{ic}, 0)$, boundary sampling points (x_{bc}, t_{bc}) , and collocation points (x_f, t_f) in the equation domain. The predicted value is calculated using a fully connected feedforward DNN corresponding to the input point. The symbol θ is the parameter set of the DNN, including weights W , bias b , and the activation function σ . The automatic differentiation of the DNN is utilized to calculate the partial derivatives of $u_{NN}(t, x; \theta)$ with respect to x_i and t_i . The loss function is evaluated using the contributions from the initial boundary conditions and the residual from the governing equation given by the physics-informed part. Then, one seeks the optimal values of W and b to minimize the loss function below a certain specified tolerance δ or until a prescribed maximum number n of iterations.

It is implemented by imposing three types of losses. One is the loss for governing equation learning l_r controlled by the collocation points N_r , the second is the loss for initial condition learning l_{ic} calculated on the initial points N_{ic} , and the last is the loss for boundary condition learning l_{bc} calculated on the boundary points N_{bc} . To combat overfitting, the loss l_r acts as a regularization mechanism that penalizes solutions that do not satisfy the governing equation. Consequently, PINN classifies the training points into two categories. One kind is the points in the space-time domain and the other is the initial boundary points. Unlike traditional numerical methods, to fit the initial and boundary conditions, PINN uses value constraints to train the neural network, which implies that there are errors in the learning of the initial and boundary conditions. The loss function defined by the L_2 norm is then as follows:

$$\ell := \ell_r + \ell_{ic} + \ell_{bc}, \quad (5)$$

where

$$\begin{aligned} \ell_r &= \frac{1}{|N_r|} \sum_{i=1}^{N_r} \|r(t_r^i, x_r^i; \theta)\|_2^2, \\ \ell_{ic} &= \frac{1}{|N_{ic}|} \sum_{i=1}^{N_{ic}} \|u_{NN}(t_0, x_{ic}^i; \theta) - u_0(x_{ic}^i)\|_2^2, \end{aligned} \quad (6)$$

$$\ell_{bc} = \frac{1}{|N_{bc}|} \sum_{i=1}^{N_{bc}} \|u_{NN}(t_{bc}^i, x_{bc}^i; \theta) - g(t_{bc}^i, x_{bc}^i)\|_2^2,$$

$$r(t, x; \theta) := \frac{\partial u_{NN}(t, x; \theta)}{\partial t} + D_{(x)}[u_{NN}(t, x; \theta); \lambda],$$

where $D_{(x)}$ represents the learned spatial differential operator, $u_{NN}(t, x; \theta)$ is the learned solution, N_r , N_{ic} , and N_{bc} represent, respectively, the internal configuration of the sampling point data $\{(t_r^i, x_r^i), r(t_r^i, x_r^i; \theta)\}_{i=1}^{N_r}$, initial data $\{(t_0, x_{ic}^i), u_0(x_{ic}^i)\}_{i=1}^{N_{ic}}$, boundary data $\{(t_{bc}^i, x_{bc}^i), g(t_{bc}^i, x_{bc}^i)\}_{i=1}^{N_{bc}}$, and $r(t, x; \theta)$ is the residual of the PDE. The locations of the collocation points are generated by a space-filling Latin hypercube sampling (LHS) strategy [48] and the initial boundary points are selected randomly.

PINN can also be applied to the inverse problem to discover the unknown parameters λ in equation (4). Inverse problems no longer require initial boundary values but rather the observed data in the space-time domain. They are solved on the same footing as forward problems, in which cases, the loss function consists of two parts. One is the loss for governing equation learning and the other is for observed data learning. The loss function ℓ is then defined by the following:

$$\ell := \ell_m + \ell_r, \quad (7)$$

in which

$$\begin{aligned} \ell_m &= \frac{1}{|N_m|} \sum_{i=1}^{N_m} \|u_{NN}(t_m^i, x_m^i; \theta; \lambda) - u_m(t_m^i, x_m^i)\|_2^2, \\ \ell_r &= \frac{1}{|N_r|} \sum_{i=1}^{N_r} \|r(t_r^i, x_r^i; \theta; \lambda)\|_2^2, \end{aligned} \quad (8)$$

where l_m and l_r are the mean square errors of the residuals for the measured data and the governing equation, respectively, N_m is the measured data size, and $u_{NN}(t_m^i, x_m^i; \theta; \lambda)$ and $u_m(t_m^i, x_m^i)$ are the predicted and measured values at the measuring points (t_m^i, x_m^i) .

In the multicomponent CDR system, the reactor model described by a system of PDE is embedded into the loss of PINN for training. The neural network optimizes not only the loss function of the network itself during the training iterations but also the residuals of each iteration of the governing equations, so that the results obtained from the

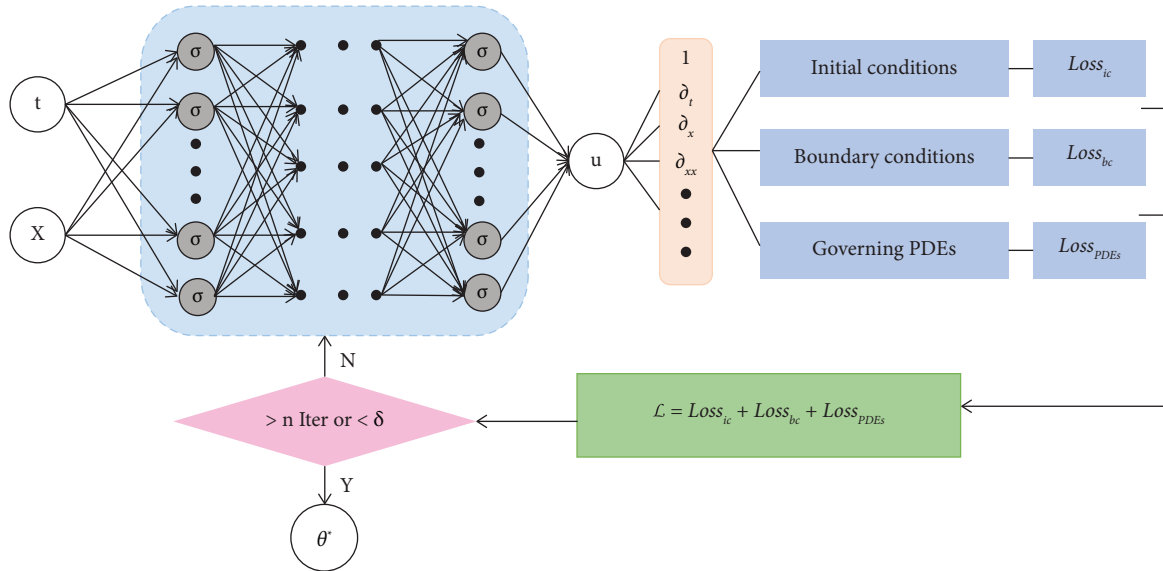


FIGURE 1: A typical PINN framework to solve the forward problem of nonlinear PDEs.

fitting better satisfy the reaction laws. In the forward problem, no manually labeled reactant concentration data are required. The PINN solves the problem by only providing the governing equations and the initial boundary conditions. In the inverse problem, the governing equation and information about the measurement points are encoded into the loss function for training. The labeled data about the unknown parameter λ are not needed either. The optimal model parameter set Θ^* is obtained by minimizing the loss function. The Adam optimization algorithm [23] is used in this paper to avoid the training process falling into the local optimum.

4. Numerical Results

In this section, numerical experiments on the nonlinear CDR systems presented in Section 2, are conducted to illustrate the capability and efficiency of the developed PINN presented in Section 3. The forward problems of the gas-solid adsorption and the autocatalytic reacting flow are addressed first in Sections 4.1 and 4.2, respectively, and the inverse problem of the autocatalytic reacting flow is then pursued in Section 4.3.

The reference solutions are given by the finite volume method [12], including the weighted essentially non-oscillation (WENO) scheme [44] and the modified total variation diminishing Lax–Friedrichs scheme with Superbee limiter (MTVDLF-Superbee) [49]. The solution of WENO is used as the reference for the gas-solid adsorption problem because it can effectively suppress the unphysical oscillations at the steep fronts. The reference solution for the autocatalytic reacting flow problem is given by the MTVDLF-Superbee scheme. It can eliminate the numerical dissipation and spurious oscillations and is considered as an optimal method for handling the CDR problems [49].

Due to the similar computational complexity of the two models, the network structure is set to be the same in all the

tests. Referring to the cases in the references [23, 43], the network structure in this study is as follows: seven hidden layers and 100 neurons in each layer. More hidden layers and more neurons have been tested, but no significant differences were observed. In the experiments, the used optimizer is Adam with a typical rate of 0.001. The used software programs are TensorFlow 1.8.0 and Python 3.6, and the experiments are conducted on a platform with NVIDIA TITAN V and Intel (R) Xeon (R) Silver 4210 CPU at 2.20 GHz.

4.1. Forward Gas-Solid Adsorption Problem. In this test, we consider whether PINN can simulate the dynamic behavior of a multireactant system with the given initial and boundary conditions. The following parameters are set: $\epsilon = 0.4$, $\nu = 0.1 \text{ m/s}$, $k = 0.0129/\text{s}$, and $K = 0.85$. The column length (L) is equal to 1.5 m.

The initial conditions are as follows:

$$\begin{aligned} C^G(x, 0) &= 0 \text{ mol/l}, \\ C^S(x, 0) &= 0 \text{ mol/l}. \end{aligned} \quad (9)$$

The Dirichlet boundary condition at $X = 0$ is as follows:

$$C^G(0, t) = 2.2 \text{ mol/l}. \quad (10)$$

The discontinuous profile given by the initial condition moves continuously along the axial direction. The reference solution, based on the result of the WENO-Roe-5 scheme on 300 fixed-grids [44], at $t = 10 \text{ s}$ is as follows:

$$\begin{cases} C^G(x, t) = 2.2 - 0.3295x, & \text{when } (x - 10\nu) \leq 0, \\ C^G(x, t) = 0.0 \text{ mol/l}, & \text{when } (x - 10\nu) > 0. \end{cases} \quad (11)$$

In this problem, the loss consists of three parts, namely, the initial concentration of the gas, the gas concentration at the boundary, and the governing equation for the gas-solid coupling reaction. Without manually labeled data, the concentrations of gas and solid reactants are learned by constraining the loss with the physical information given by the gas-solid adsorption model. During the training process, 100 initial points and 100 boundary points are randomly selected, and 2000 collocation points are generated by LHS in the space-time domain.

The PINN solution is compared with the reference solution in Figure 2, and the results show that PINN accurately captures the dynamic behavior of the gas-solid adsorption column. The running time is 0.36 h. To verify the stability of the algorithm, a set of 10 test errors has been obtained by 10 independent repetitions for this problem. The error is given by the relative root mean square error (RMSE) between the PINN and the reference solution. Then, the mean and standard deviation of the error are calculated, which is $2.16e-02 \pm 1.12e-02$ for C^G and $3.41e-02 \pm 1.52e-02$ for C^S , respectively. As shown in Figure 3, by increasing the surface gas velocity from 0.1 m/s to 0.2 m/s, the reacting reactants are accelerated to the boundary. To study the effects of void fraction ϵ on the concentration distribution, five cases with $\epsilon = 0.8, 0.4, 0.2, 0.1$, and 0.05 are studied, and the results are shown in Figure 4. It is seen that, as the void fraction decreases, the concentration no longer uniformly decreases along the range, but gradually shows a nonlinear trend, and the closer the concentration is to the inlet boundary, the faster is the decrease rate.

4.2. Forward Autocatalytic Reacting Flow Problem. In this experiment, the following parameters are taken: $\alpha = 0.065$, $\beta = 2.0$, $\gamma = 0.025$, $\nu = 1.0$, and $D_1 = D_2 = D_3 = 0.05$.

The initial conditions are as follows:

$$\begin{aligned} U_1(X, 0) &= 1, \\ U_2(X, 0) &= 0, \\ U_3(X, 0) &= 0. \end{aligned} \quad (12)$$

The Dirichlet boundary conditions at $X = 0$ are as follows:

$$\begin{aligned} U_1(0, T) &= 0, \\ U_2(0, T) &= 1, \\ U_3(0, T) &= 0. \end{aligned} \quad (13)$$

The Neumann boundary conditions at $X = 1$ are as follows:

$$\begin{aligned} \frac{\partial U_1(1, t)}{\partial x} &= 0, \\ \frac{\partial U_2(1, t)}{\partial x} &= 0, \\ \frac{\partial U_3(1, t)}{\partial x} &= 0. \end{aligned} \quad (14)$$

In this problem, the loss also consists of three parts, including the concentrations of the three reactants at the initial moment, the Dirichlet boundary condition at $X = 0$, and the Neumann boundary condition at $X = 1$, and the governing equation for the autocatalytic reacting flow. The PINN results as function of X at $T = 0.1$ and 0.5 and function of T at $X = 0.1$ and 0.5 are shown in Figure 5, compared with the reference solution given by MTVDLF-Superbee [49]. The good agreement is clearly observed. For this test, the running time is 0.46 h.

Since the sin activation function can improve the learning ability of the network for high-frequency signals [50, 51], it is used in the above tests. The tanh is also applied as the activation function and the results are shown in Figure 6, together with the PINN solution with activate function sin. Numerical oscillations at the boundary are observed. Therefore, the sin activation function is also better to fit high-frequency solutions of the CDR system studied here.

It is known that the Péclet number (Pe) defined by $Pe_i = \nu/D_i$ has an important effect on the solution of a numerical method. For convection-dominated transport (i.e., $Pe \gg 1$), the numerical solution can develop spurious oscillations (over or undershoot) or numerical dispersion [52, 53]. Its effect on the solution of PINN is also studied here. Figure 7 shows the reactant concentrations over the entire space-time domain for the cases of $Pe = 100, 1000$ and 10000 , and Figure 8 exhibits the simulated concentration distribution at the reactor center ($X = 0.5$) of the three reactants. The sharpness of the moving fronts increases with the increasing Pe number. Even for the high Pe case ($Pe = 10000$), the PINN method still converges and captures the steep gradient with no oscillations. Therefore, the PINN method has good accuracy and effectiveness in solving the autocatalytic reacting flow problems.

4.3. Inverse Autocatalytic Reacting Flow Problem. In this section, the application of PINN for the inverse autocatalytic reacting flow problems is investigated. In the forward problem, with the given initial and boundary conditions, accurate solutions of the three reactants have been obtained. In the inverse problem, the solutions of the autocatalytic reacting flow model are known at a given number of measuring points across the problem domain, while the model parameters $\lambda = [\alpha, \beta, \gamma, \nu, D_i]^T$ are unknown. The measured data are encoded as constraints into the loss function of the neural network to identify the unknown parameters and to estimate the solution in the entire space-time domain.

As no measured data is available, to illustrate the effectiveness of PINN, the dataset was generated by MTVDLF-Superbee [49] with $\alpha = 0.065$, $\beta = 2.0$, $\gamma = 0.025$, $\nu = 1.0$, and $D_1 = D_2 = D_3 = 0.05$. Different from the original PINN, with randomly obtained points in the space-time domain, an arrangement closer to the real situation is considered. The monitoring distance interval is set to be 0.025 m, and the monitoring time interval is 0.025 s, resulting in a total number of 1600 data pairs $\langle x, t, \hat{u} \rangle$.

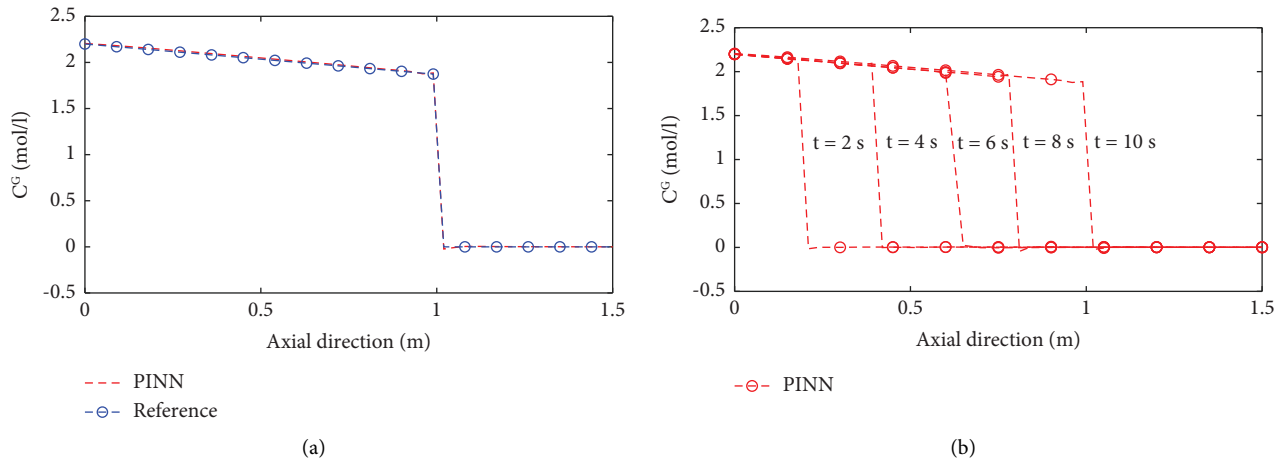


FIGURE 2: (a) Comparison of the PINN and reference solutions of the gas-solid adsorption problem at time $t = 10$ s. (b) Gas concentration profiles at different times.

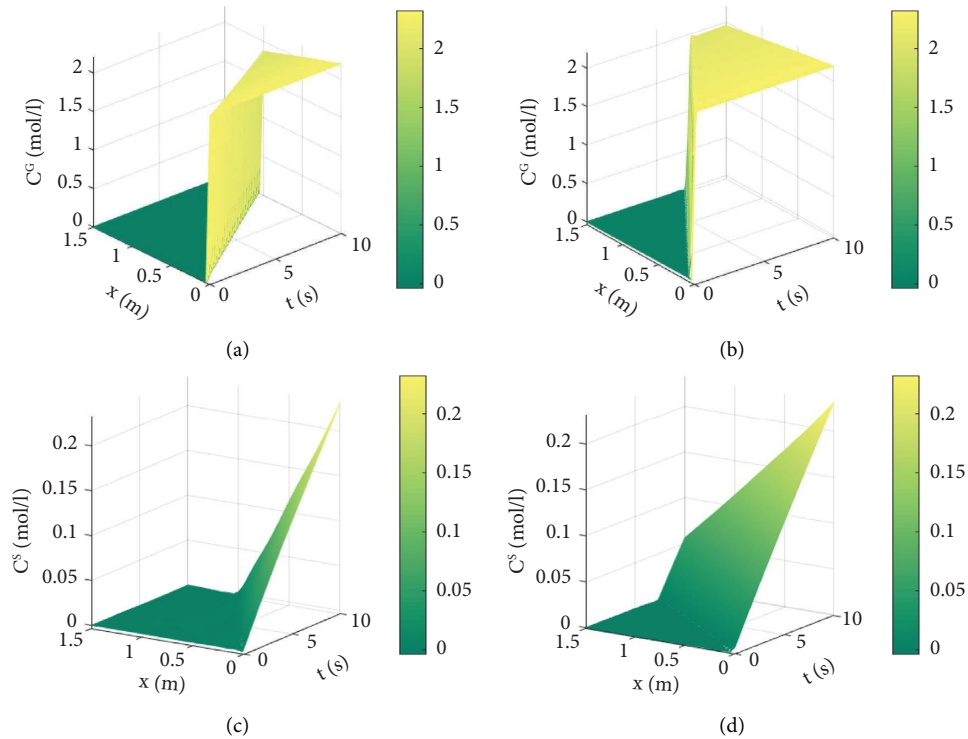


FIGURE 3: Changes of reactant concentrations with different flow velocity. $C^G(x, t)$ with (a) $v = 0.1$ m/s, and (b) $v = 0.2$ m/s, and $C^S(x, t)$ with (c) $v = 0.1$ m/s, and (d) $v = 0.2$ m/s.

Figure 9 illustrates the performance of PINN for solving the inverse problem. The concentration of the reactants learned by PINN at $T = 0.1$ and $T = 0.5$, and the time history of the concentration at $X = 0.1$ and $X = 0.5$ are shown, and the results are compared with the numerical solutions of MTVDLF-Superbee. The results reveal good agreement, indicating that PINN can accurately predict the concentration of the reactants in the entire space-time domain using limited given data. PINN simulates the inverse

problem with a running time of 0.51 h. This is a remarkable advantage over the traditional time-consuming methods [27].

In addition to providing predictions in the space-time domain, the solution to the inverse problem involves identifying the unknown parameters $\hat{\lambda} = [\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{v}, \hat{D}_i]^T$. Table 1 presents all the model parameters learned by PINN from the supposed observations. Except β , all parameters have been correctly identified with relative error less than

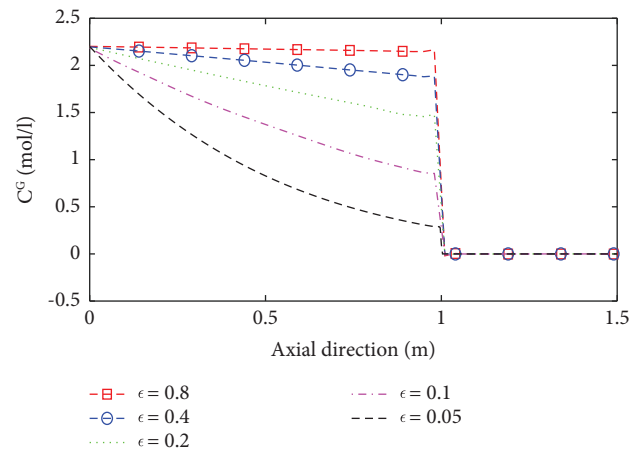


FIGURE 4: Gas concentration $C^G(x, t)$ along the tube column with different void fractions at $t = 10$ s.

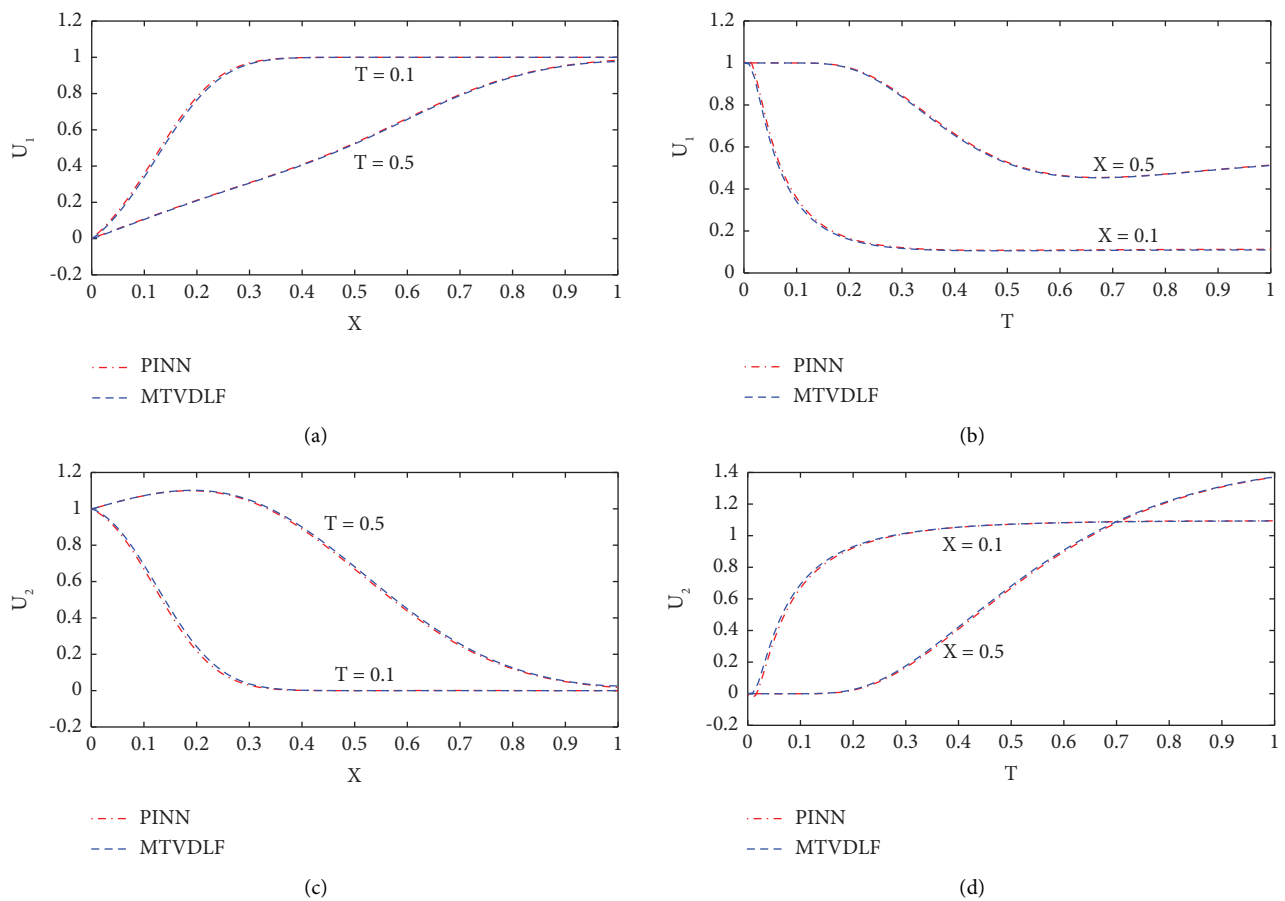


FIGURE 5: Continued.

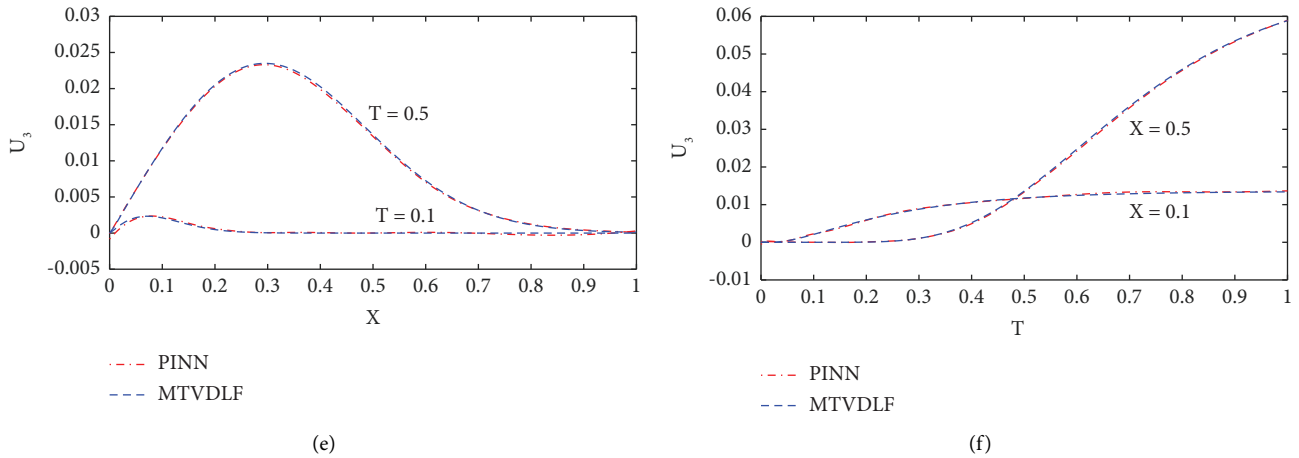


FIGURE 5: Comparison between PINN and reference solutions of (a, b) substrate A, (c, d) autocatalyst B, and (e, f) mutant C for the forward problem.

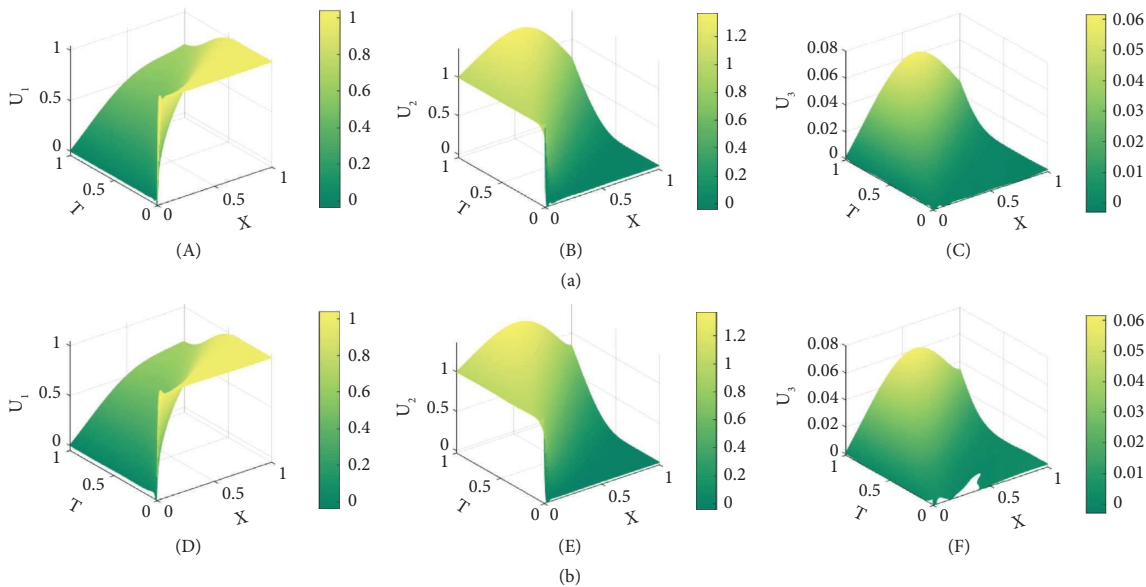


FIGURE 6: Comparison of the PINN results with different activation functions. (a) sin; (b) tanh.

5%. The parameters associated with β are α and D_3 , resulting in a relative high error of 1% for α and 4% for D_3 . Other parameters are identified with relative errors less than 0.5%. This test demonstrates the excellent capability of PINN for identifying the parameters in the CDR system. Note that the error is given by the RMSE between the predicted and the reference value defined by the following:

$$\text{RMSE} = \frac{\|\lambda_{\text{true}} - \lambda_{\text{pred}}\|_2}{\|\lambda_{\text{true}}\|_2}. \quad (15)$$

Table 2 presents the learned kinetics of the reaction and mutation efficiency, supposing the flow velocity and diffusion coefficients are known. When only the reaction-related

model parameters are predicted, the relative errors for both α and γ are less than 1%, but it remains large for β . That is, when PINN is used to identify fewer model parameters, the prediction accuracy will be improved as expected.

From the two tables, it is seen that all the reaction parameters have been correctly learned except the mutation efficiency β . The large deviance in the recognition of β is attributed to the fact that it is insensitive to the governing equation. To verify this, the forward problem with three different β was solved using PINN and the results are shown in Figure 10. The concentrations of substrate A and autocatalyst B do not change when taking different values of β (see Figures 10(a) and 10(b)). However, due to the small magnitude of the mutant C, a slight difference in the concentration of reactant C at $T = 1$ is found (Figure 10(c)).

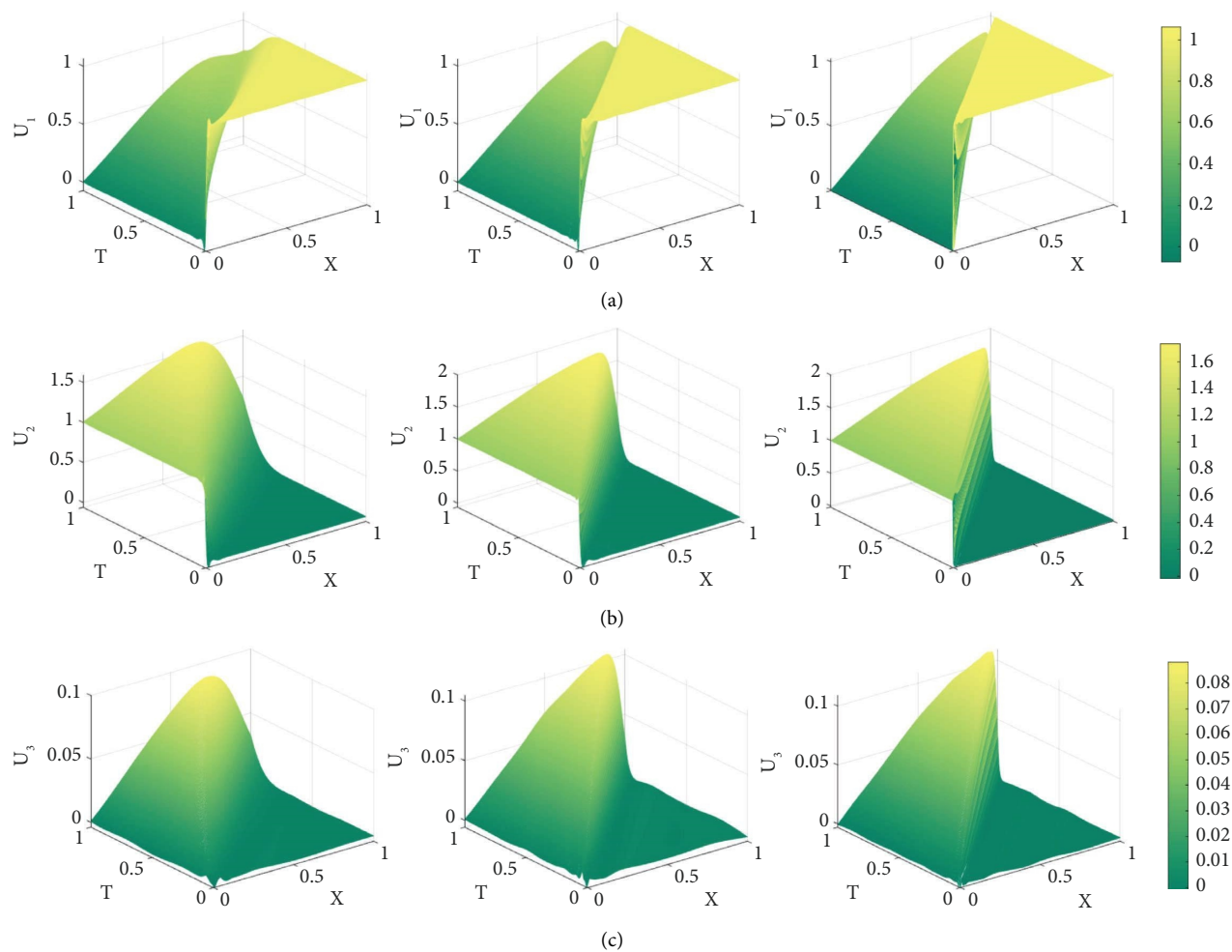


FIGURE 7: Concentration distribution of (a) substrate A, (b) autocatalyst B, and (c) mutant C with different Pe number. From left to right, $Pe = 100, 1000, 10000$.

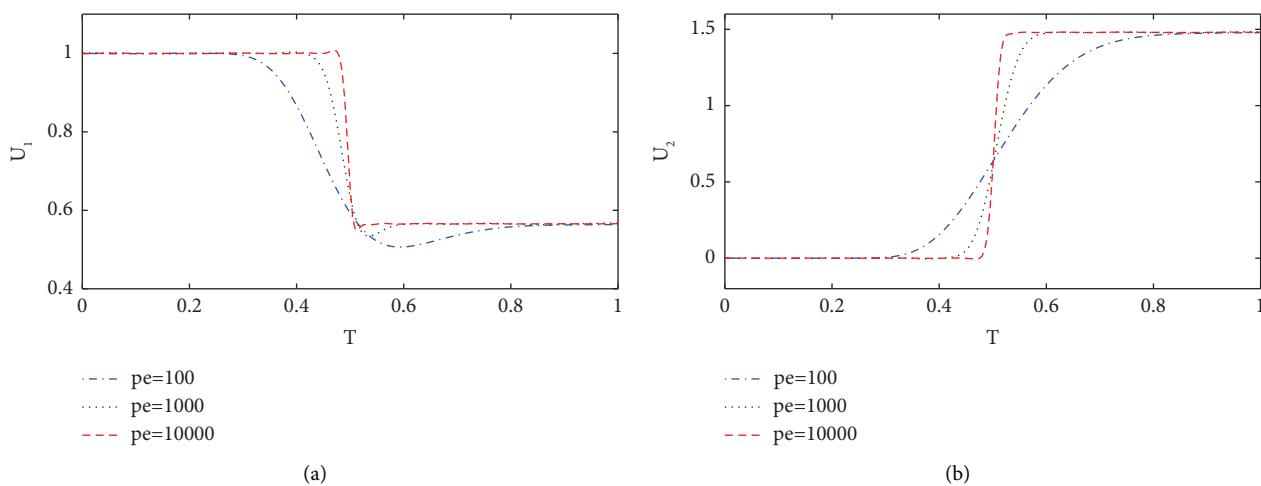


FIGURE 8: Continued.

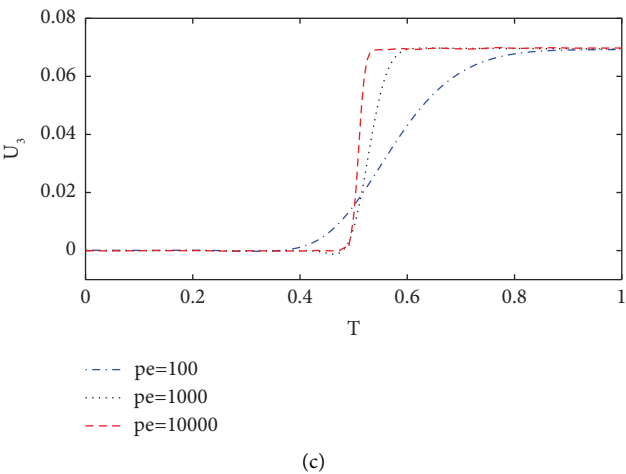


FIGURE 8: Dynamic simulation of (a) substrate A, (b) autocatalyst B, and (c) mutant C at the reactor center ($X=0.5$) with different pe number for the autocatalytic reacting flow problem.

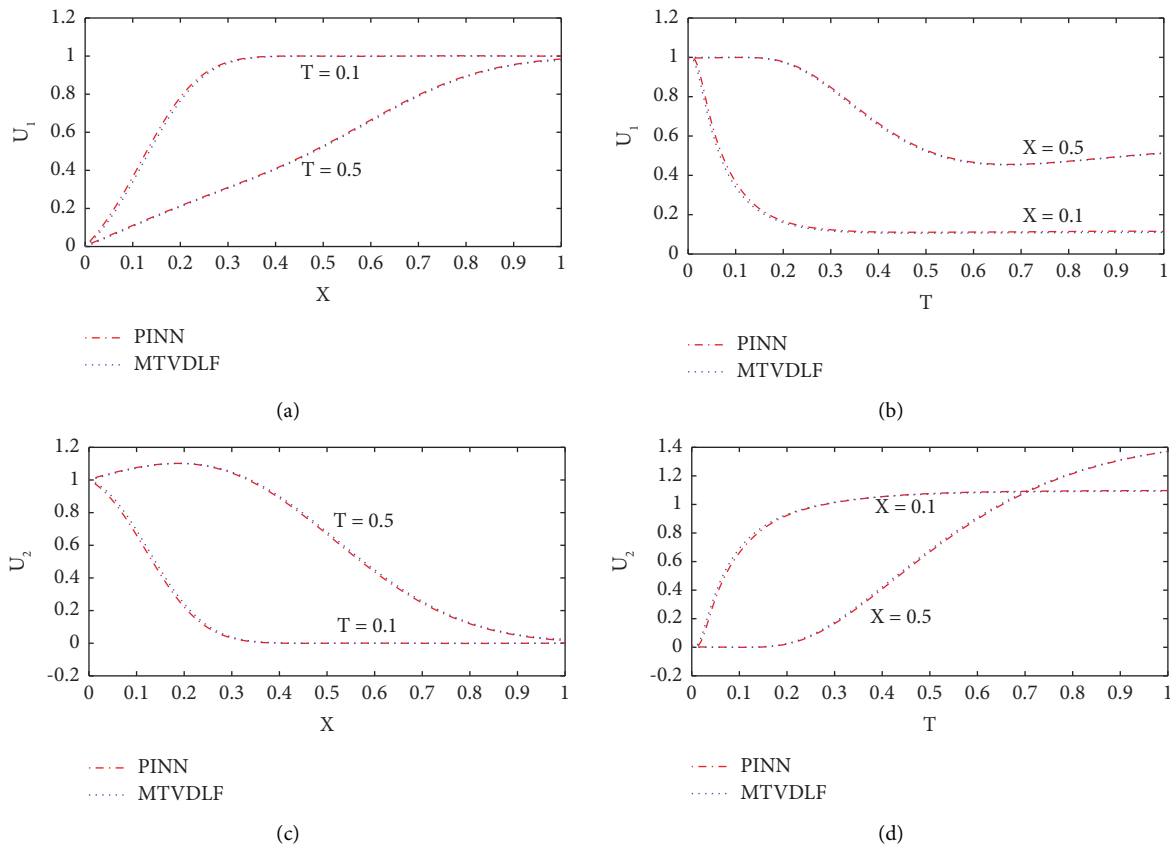


FIGURE 9: Continued.

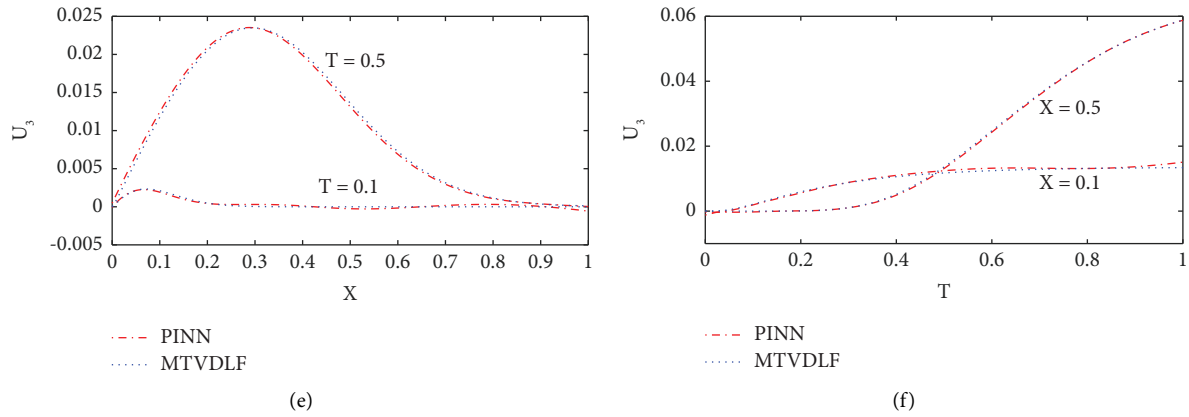


FIGURE 9: Comparison between PINN and reference solutions of (a, b) substrate A, (c, d) autocatalyst B, and (e, f) mutant C for the inverse problem.

TABLE 1: RMSE for predicted model parameters by PINN.

Model parameters	True value	Predicted value	RMSE (%)
α	0.065	0.0658	1.32
β	2.000	0.8005	59.9
γ	0.025	0.0252	0.8
ν	1.000	0.998	0.12
$D1$	0.050	0.0498	0.28
$D2$	0.050	0.0497	0.40
$D3$	0.050	0.0520	4.06

TABLE 2: RMSE of the kinetic parameters of the reaction and mutation parameters learned by PINN from the reference solution.

Model parameters	True value	Predicted value	RMSE (%)
α	0.065	0.0651	0.18
β	2.000	0.8991	55.1
γ	0.025	0.0251	0.40

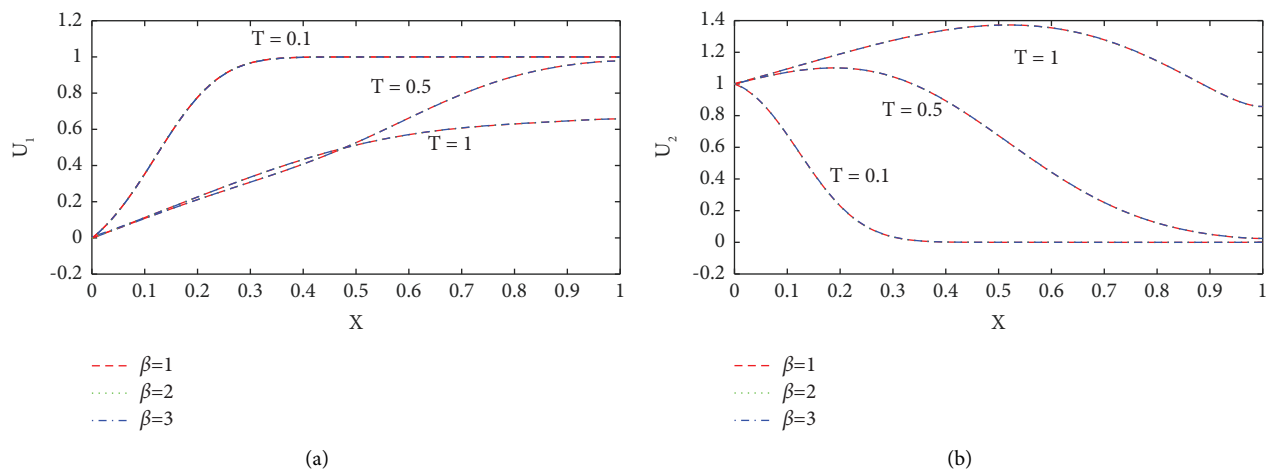


FIGURE 10: Continued.

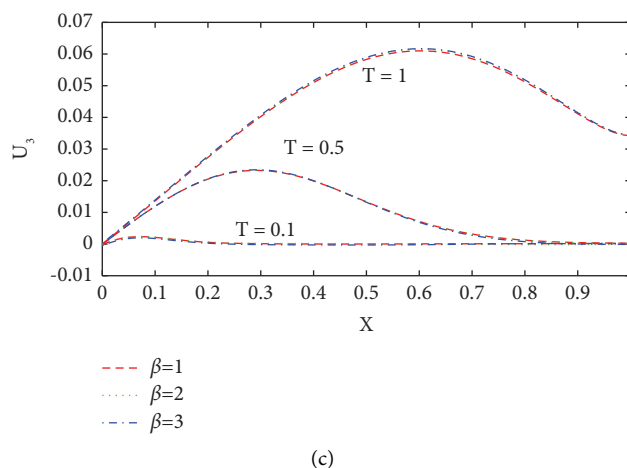


FIGURE 10: Comparison of PINN results with different β for (a) substrate A (b) autocatalyst B (c) mutant C at $T = 0.1$, $T = 0.5$ and $T = 1$.

TABLE 3: Percentage error in the identified parameters α and γ for different number of training data N corrupted by different noise levels.

Error (%)	Noise							
	α				γ			
N	0%	1%	5%	10%	0%	1%	5%	10%
1000	0.186	0.543	1.261	4.554	0.420	0.766	1.856	8.530
2000	0.178	0.212	1.257	3.523	0.369	0.658	3.481	7.223
3000	0.165	0.175	1.203	2.946	0.295	0.543	2.896	6.391

In reality, there is often noise in the measured data, which leads to various difficulties for the learning methods and highly affects the recognition accuracy of these parameters. The capacity of the PINN to solve the inverse problem with noisy data was investigated by adding Gaussian noise to the dataset. Table 3 shows the error in the PINN solution of the inverse problem at different noise levels (noise = 0%, 1%, 5%, 10%) and different sizes of the measured point ($N = 1000, 2000, 3000$), supposing the flow velocity and diffusion coefficients are known. The results indicate that the quality of the prediction over the problem domain decreases with increasing noise levels from error $\approx 10^{-3}$ for noise = 0% to error $\approx 10^{-2}$ for noise = 10%. However, the identifications of the model parameters do not show significant variations due to increasing noise levels. The proposed method appears to be robust to noise levels in the data, and a reasonable recognition accuracy is maintained even for noise corruptions up to 10%. At the same time, the error of parameter learning is hardly affected by the size of the training data. This means that for CDR systems, PINN can obtain accurate results with sparse training data.

5. Conclusion

In this paper, a PINN framework for solving the forward and inverse problems of nonlinear CDR systems is presented. In PINN, the CDR systems expressed as PDEs are incorporated into the neural network. Due to being devoid of grids or

nodes, PINN is a mesh-free method that can predict the solution at any point in the equation domain without interpolation.

For the forward gas-solid adsorption problem, the reaction process is simulated for different convection velocities and void fractions. The PINN results agree well with the reference solutions, and the moving steep fronts are correctly captured without numerical dissipation and spurious oscillations.

For the forward autocatalytic reacting flow problem, the PINN method accurately predicts the dynamic profiles of the system. Compared with the standard activation function of tanh, the sin activation function can more effectively eliminate the unphysical oscillation generated at the boundary. For high Péclet numbers, PINN still captures the sharp profiles without any unphysical oscillation or numerical dissipation.

For the inverse autocatalytic reacting flow problem, with limited data, PINN successfully identified the unknown parameters in the CDR systems, even the measured data are heavily polluted by noise. This demonstrates a strong ability of PINN to learn missing chemical information and to better observe and explain the laws in the reaction chemistry.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study has been partly funded by the National Key Research and Development Program of China (grant no. 2020YFC1807905) and the Tianjin Municipal Science and Technology Project (grant no. 19ZXZNGX00030).

References

- [1] A. Jamshidi, J. M. V. Samani, H. M. V. Samani, A. Zanini, M. G. Tanda, and M. Mazaheri, "Solving inverse problems of unknown contaminant source in groundwater-river integrated systems using a surrogate transport model based optimization," *Water*, vol. 12, no. 9, p. 2415, 2020.
- [2] A. Schaum, S. Koch, M. Reichhartinger, T. Meurer, J. A. Moreno, and M. Horn, "Nonlinear observer design for a 1D heat conduction process," in *Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1149–1154, Austin, TX, USA, December 2021.
- [3] Y. Cui and A. Ge, "A tunable high-q microwave detector for on-column capillary liquid chromatography," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 5978–5980, 2020.
- [4] Q. Xu, T. Yang, Y. He, S. Mei, F. Meng, and X. Tang, "Numerical analysis of eddy current loss of high-speed axial magnetic drive spindle," *International Journal of Wireless and Mobile Computing*, vol. 21, no. 2, pp. 119–127, 2021.
- [5] A. Pavlišić, M. Huš, A. Prašnikar, and B. Likozar, "Multiscale modelling of CO₂ reduction to methanol over industrial Cu/ZnO/Al₂O₃ heterogeneous catalyst: linking ab initio surface reaction kinetics with reactor fluid dynamics," *Journal of Cleaner Production*, vol. 275, Article ID 122958, 2020.
- [6] M. Huš, M. Grilc, A. Pavlišić, B. Likozar, and A. Hellman, "Multiscale modelling from quantum level to reactor scale: an example of ethylene epoxidation on silver catalysts," *Catalysis Today*, vol. 338, pp. 128–140, 2019.
- [7] D. Lašić Jurković, J. L. Liu, A. Pohar, and B. Likozar, "Methane dry reforming over Ni/Al₂O₃ catalyst in spark plasma reactor: linking computational fluid dynamics (CFD) with reaction kinetic modelling," *Catalysis Today*, vol. 362, pp. 11–21, 2021.
- [8] K. Alhumaizi, "Flux-limiting solution techniques for simulation of reaction–diffusion–convection system," *Communications in Nonlinear Science and Numerical Simulation*, vol. 12, no. 6, pp. 953–965, 2007.
- [9] Y. Wang and K. Hutter, "Comparisons of numerical methods with respect to convectively dominated problems," *International Journal for Numerical Methods in Fluids*, vol. 37, no. 6, pp. 721–745, 2001.
- [10] K. Alhumaizi and A. Abasaeed, "On mutating autocatalytic reactions in a CSTR. I: multiplicity of steady states," *Chemical Engineering Science*, vol. 55, no. 18, pp. 3919–3928, 2000.
- [11] E. Zuazua, "Propagation, observation, and control of waves approximated by finite difference methods," *SIAM Review*, vol. 47, no. 2, pp. 197–243, 2005.
- [12] N. Muhammad, "Finite volume method for simulation of flowing fluid via OpenFOAM," *The European Physical Journal Plus*, vol. 136, no. 10, pp. 1010–1022, 2021.
- [13] H. Egger and J. Schöberl, "A hybrid mixed discontinuous Galerkin finite-element method for convection-diffusion problems," *IMA Journal of Numerical Analysis*, vol. 30, no. 4, pp. 1206–1234, 2010.
- [14] R. M. Kirby and S. J. Sherwin, "Stabilisation of spectral/hp element methods through spectral vanishing viscosity: application to fluid mechanics modelling," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 23–24, pp. 3128–3144, 2006.
- [15] X. Yu, Q. Zhou, S. Wang, and Y. Zhang, "A systematic survey of deep learning in breast cancer," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 152–216, 2022.
- [16] Y. Huang, L. Shih, C. Tsai, and G. Shen, "Describing video scenarios using deep learning techniques," *International Journal of Intelligent Systems*, vol. 36, no. 6, pp. 2465–2490, 2021.
- [17] Z. Wu, Z. Lv, J. Kang, W. Ding, and J. Zhang, "Fingerprint bio-key generation based on a deep neural network," *International Journal of Intelligent Systems*, vol. 37, no. 7, pp. 4329–4358, 2022.
- [18] R. Zhang, J. Wei, W. Lu et al., "Cs-rep: making speaker verification networks embracing re-parameterization," in *Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7082–7086, Singapore, May 2022.
- [19] S. J. Grimberg and C. Farhat, *Hyperreduction of CFD Models of Turbulent Flows Using a Machine Learning Approach*, AIAA Scitech 2020 Forum 2020, Reston, VA, USA, 2020.
- [20] K. O. Lye, S. Mishra, and D. Ray, "Deep learning observables in computational fluid dynamics," *Journal of Computational Physics*, vol. 410, Article ID 109339, 2020.
- [21] Y. Zhu and N. Zabaras, "Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification," *Journal of Computational Physics*, vol. 366, pp. 415–447, 2018.
- [22] R. K. Tripathy and I. Bilonis, "Deep UQ: learning deep neural network surrogate models for high dimensional uncertainty quantification," *Journal of Computational Physics*, vol. 375, pp. 565–588, 2018.
- [23] 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*, vol. 378, pp. 686–707, 2019.
- [24] Q. He and A. M. Tartakovsky, "Physics-informed neural network method for forward and backward advection-dispersion equations," *Water Resources Research*, vol. 57, no. 7, Article ID e2020WR029479, 2021.
- [25] L. Yuan, Y. Ni, X. Deng, and S. Hao, "A-PINN: auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations," *Journal of Computational Physics*, vol. 462, Article ID 111260, 2022.
- [26] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 2018.
- [27] S. Momani, Z. S Abo-Hammou, and O. Mk Alsmad, "Solution of inverse kinematics problem using genetic algorithms," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, pp. 225–233, 2016.
- [28] O. H. Menin and C. T. Bauch, "Solving the patient zero inverse problem by using generalized simulated annealing," *Physica A: Statistical Mechanics and Its Applications*, vol. 490, pp. 1513–1521, 2018.
- [29] M. Jha and B. Datta, "Three-dimensional groundwater contamination source identification using adaptive simulated annealing," *Journal of Hydrologic Engineering*, vol. 18, no. 3, pp. 307–317, 2013.
- [30] Y. Li and X. Hu, "Artificial neural network approximations of Cauchy inverse problem for linear PDEs," *Applied Mathematics and Computation*, vol. 414, Article ID 126678, 2022.
- [31] E. Hetmaniok, D. Ślota, and A. Zielonka, "Solution of the inverse continuous casting problem with the aid of modified harmony search algorithm," *International Conference on Parallel Processing and Applied Mathematics*, pp. 402–411, Springer, Berlin, Germany, 2013.
- [32] B. Datta, D. Chakrabarty, and A. Dhar, "Simultaneous identification of unknown groundwater pollution sources and

- estimation of aquifer parameters,” *Journal of Hydrology*, vol. 376, no. 1-2, pp. 48–57, 2009.
- [33] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, “Physics-informed neural networks for inverse problems in nano-optics and metamaterials,” *Optics Express*, vol. 28, no. 8, Article ID 11618, 2020.
- [34] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks,” *Journal of Computational Physics*, vol. 404, Article ID 109136, 2020.
- [35] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- [36] Y. Li and F. Mei, “Deep learning-based method coupled with small sample learning for solving partial differential equations,” *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 17391–17413, Article ID 17391, 2021.
- [37] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks for heat transfer problems,” *Journal of Heat Transfer*, vol. 143, no. 6, Article ID 060801, 2021.
- [38] Q. He, D. Barajas-Solano, G. Tartakovsky, and A. M. Tartakovsky, “Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport,” *Advances in Water Resources*, vol. 141, Article ID 103610, 2020.
- [39] S. Falas, C. Konstantinou, and M. K. Michael, “Special session: physics-informed neural networks for securing water distribution systems,” in *Proceedings of the 2020 IEEE 38th International Conference on Computer Design*, pp. 37–40, Hartford, CT, USA, October 2020.
- [40] A. Arzani, J. Wang, and R. M. D’Souza, “Uncovering near-wall blood flow from sparse data with physics-informed neural networks,” *Physics of Fluids*, vol. 33, no. 7, Article ID 071905, 2021.
- [41] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, “Physics-informed neural networks for cardiac activation mapping,” *Frontiers in Physics*, vol. 8, p. 42, 2020.
- [42] Y. Mishin, “Machine-learning interatomic potentials for materials science,” *Acta Materialia*, vol. 214, Article ID 116980, 2021.
- [43] Q. Hou, Z. Sun, L. He, and A. Karemat, “Orthogonal grid physics-informed neural networks: a neural network-based simulation tool for advection–diffusion–reaction problems,” *Physics of Fluids*, vol. 34, no. 7, Article ID 077108, 2022.
- [44] Y. I. Lim, J. M. Le Lann, and X. Joulia, “Accuracy, temporal performance and stability comparisons of discretization methods for the numerical solution of partial differential equations (PDEs) in the presence of steep moving fronts,” *Computers & Chemical Engineering*, vol. 25, no. 11-12, pp. 1483–1492, 2001.
- [45] K. Alhumaizi, “Comparison of finite difference methods for the numerical simulation of reacting flow,” *Computers & Chemical Engineering*, vol. 28, no. 9, pp. 1759–1769, 2004.
- [46] K. Alhumaizi, R. Henda, and M. Soliman, “Numerical analysis of a reaction-diffusion-convection system,” *Computers & Chemical Engineering*, vol. 27, no. 4, pp. 579–594, 2003.
- [47] M. Abadi, P. Barham, J. Chen et al., *Tensorflow: a system for large-scale machine learning*, vol. 16, OSDI, Savannah, GA, USA, 2016.
- [48] M. Stein, “Large sample properties of simulations using Latin hypercube sampling,” *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
- [49] Q. Hou, J. Liu, J. Lian, and W. Lu, “A Lagrangian particle algorithm (SPH) for an autocatalytic reaction model with multicomponent reactants,” *Processes*, vol. 7, no. 7, p. 421, 2019.
- [50] M. Tancik, P. Srinivasan, B. Mildenhall et al., “Fourier features let networks learn high frequency functions in low dimensional domains,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.
- [51] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.
- [52] G. F. Pinder and W. G. Gray, *Finite Element Simulation in Surface and Subsurface Hydrology*, Elsevier, Amsterdam, Netherlands, 2013.
- [53] P. S. Huyakorn, *Computational Methods in Subsurface Flow*, Academic Press, Cambridge, MA, USA, 2012.