ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

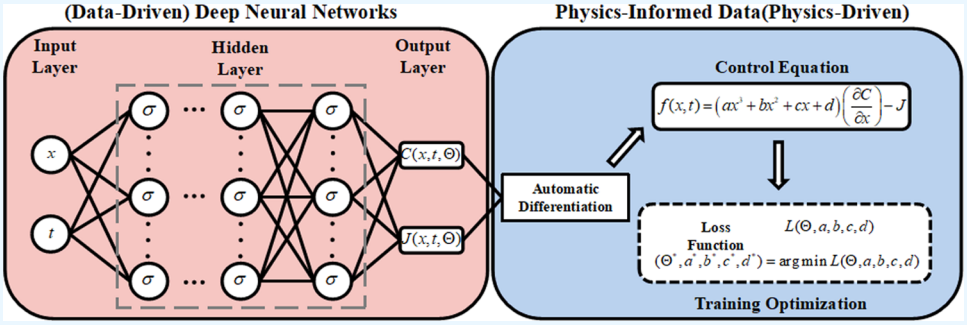
PINN Model of Diffusion Coefficient Identification Problem in Fick's Laws

Dongchen Li, Bin Yan, Tianya Gao, Guowei Li, and You Wang*

Cite This: *ACS Omega* 2024, 9, 3846−3857

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: This study tackles the complex task of determining diffusion coefficients in inverse problems, addressing the challenges of instability and computational demands. The primary objective is to introduce an efficient model for estimating diffusion coefficients under specific conditions. Through a unique fusion of Fick's laws and a Neural Network framework, a physics-informed neural network (PINN) is developed for the diffusion coefficient identification problem. The model accommodates scenarios where both diffusion flux and concentration gradient are known, where diffusion flux is known while the concentration gradient is unknown, and where diffusion flux is unknown while the concentration gradient is known. Results demonstrate the model's efficiency, obtaining diffusion coefficients in less than 1000, 2000, and 3000 iterations for the respective scenarios. Sensitivity analysis underscores the model's validity across conditions, highlighting the positive impact of a higher proportion of effective data on convergence and alignment with general diffusion coefficient patterns. In conclusion, the PINN model stands out as a powerful tool for accurately estimating diffusion coefficients under varying conditions.

1. INTRODUCTION

Adolf Fick introduced Fick's laws in 1855 to describe diffusion's macroscopic phenomena,[1] with the first law stating that a higher concentration gradient results in greater diffusion flux. Building on this, the second law shows that, in non-steady-state diffusion, the rate of change of component concentration at a point is proportional to the spatial second derivative of concentration. Despite Fick's laws being over 170 years old, reliable determination of component-dependent diffusion coefficients remains a hot research topic.[2,3] The diffusion coefficient, a crucial thermal property, plays a pivotal role in various computational and simulation processes related to mass transfer, absorption, and catalytic reactions. Understanding its dependence on different components is a pressing research issue. Experimental measurements often involve indirect methods,[4] where the initial state is established, and physical quantities related to the diffusion coefficient, such as diffusion flux and concentration distributions, are measured. The diffusion coefficient's calculation falls under parameter identification in inverse problems,[5] typically characterized by instability, nonlinearity, and computational demands.

There are many approaches to deal with this problem, e.g., the finite element method,[6−8] analytical model,[9−11] and machine learning model.[12,13] Although these methods are proven to be feasible, they take a lot of computational time and source. Recently, with the great advancement of numerical technology and intelligent algorithms, some scholars are trying to integrate the physical mechanism into the neural network (NN) framework to construct the physics-informed neural network (PINN) model. Shaban et al.[14] proposed a physics-informed deep neural network model to investigate the chloride diffusion mechanism and predict the distribution of chloride concentrations in concrete. Batuwatta-Gamage et al.[15] proposed a PINN-based surrogate framework to simulate the time-based moisture concentration and moisture-content-

Received: October 10, 2023
Revised: December 21, 2023
Accepted: December 26, 2023
Published: January 5, 2024

based shrinkage of a plant cell during drying. Lu and Christov[16] employed PINN approach to extend the utility of phenomenological models, which can simulate the particle migration in shear flow.

The biggest advantage of the PINN model, as we know, is that it incorporates physical laws, specifically Fick's laws in the present work, into the Neural Network's architecture. This integration ensures that the model adheres to fundamental principles, enhancing the reliability of the diffusion coefficient estimates. Moreover, it is evidenced that PINN proves effective in solving inverse problems, such as determining diffusion coefficients, which are inherently challenging due to nonlinearity and sensitivity to initial conditions. The combination of neural networks and physics-based constraints enables robust solutions in less computational time compared to traditional methods.

Therefore, the main target of the presented research is to propose a feasible and efficient PINN approach to deal with the diffusion coefficient identification problem even if some parameter data is unavailable. First, a PINN model will be built integrating the Fick's laws into the NN framework. Second, three different situations will be analyzed: (a) both diffusion flux and concentration gradient are known; (b) diffusion flux is known while concentration gradient is unknown; and (c) diffusion flux is unknown while concentration gradient is known. Third, the sensitivity of the proposed model will be analyzed by investigating the effects of the error level and amount of the testing data and the selection of the initial guess. Finally, conclusions through the research will be drawn.

2. MODEL DESCRIPTION

The diffusion coefficient identification itself is a highly nonlinear, uncertain, and computationally intensive problem. Neural network algorithms can be well applied to such problem.[17,18] However, traditional neural network algorithms when addressing these problems often rely solely on existing data for training to predict target parameters. This approach makes it difficult to reveal the inherent relationships among various parameters.[19] In contrast, diffusion coefficient calculations not only have abundant data sources but also are supported by corresponding physical laws. Therefore, coupling the physical mechanism into the deep neural framework to achieve data-physics dual-drive can lead to more accurate diffusion coefficient identification.[20,21] The deep neural network (DNN) algorithm is introduced and is coupled with Fick's laws to constrain neural network training, establishing the PINN diffusion coefficient identification model.

2.1. Equations of Fick's Laws. *2.1.1. Fick's First Law.* Fick's first law established the relationship between flux J, diffusion coefficient D, and concentration gradient C. The specific expression is as follows

$$J = -D\left(\frac{\partial C}{\partial x}\right) \tag{1}$$

This formula indicates that the diffusion flux through a unit cross-sectional area perpendicular to the diffusion direction per unit of time is directly proportional to the concentration gradient at that interface.

In the case of a three-dimensional diffusion system, since the diffusion flux J is a vector, it can be decomposed into three components along the x, y, and z coordinate axes. In this case, the first law can be expressed as

$$J = J_x\vec{e_i} + J_x\vec{e_j} + J_x\vec{e_k} = -D\left(\frac{\partial C}{\partial x}\vec{e_i} + \frac{\partial C}{\partial y}\vec{e_j} + \frac{\partial C}{\partial z}\vec{e_k}\right) \tag{2}$$

2.1.2. Fick's Second Law. The first law can only be applied for data calculations under the conditions of steady-state diffusion, while in actual situations, most cases involve non-steady-state diffusion. Therefore, based on the material's equilibrium, Fick established the second diffusion differential equation from the first law. The second law reflects the relationship between the rate of change of component concentration with time under non-steady-state diffusion conditions and the rate of change of diffusion flux with distance. The specific expression is as follows

$$\frac{\partial C}{\partial t} = \frac{\partial\left(D\frac{\partial C}{\partial x}\right)}{\partial x} \tag{3}$$

If the diffusion coefficient D varies only slightly with coordinates, then it can be considered as constant. In this case, eq 4 can be rewritten as follows

$$\frac{\partial C}{\partial t} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right) \tag{4}$$

After introducing one-dimensional and multidimensional equations for steady-state and non-steady-state processes, one can select an appropriate diffusion equation based on the measured diffusion flux J and diffusion coefficients D under different conditions.[22] This can be used to constrain training of the neural network.

2.2. PINN Framework of Diffusion Coefficient Identification. PINN is essentially a combination of DNN and physical information and for the diffusion space of a single component in any dimension. Its basic framework is shown in Figure 1.

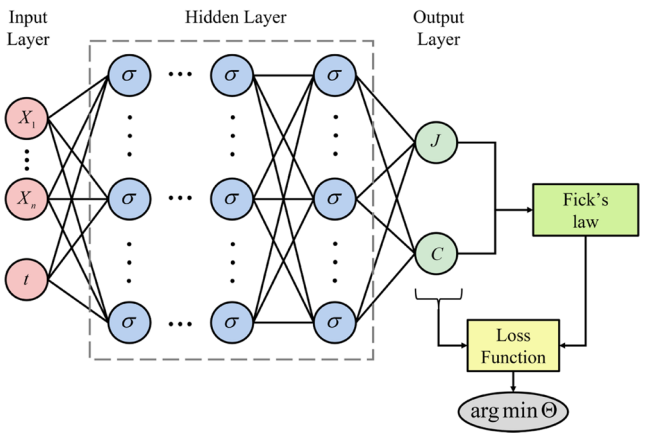


Figure 1. Basic framework of PINN.

2.2.1. Deep Neural Network Section. As shown in Figure 1, the structure of the PINN primarily consists of three parts: the input layer, hidden layers, and the output layer. Based on the available experimental data, the data used for training mainly include diffusion flux, concentration gradient, spatial coordinate, and time. Therefore, the training and testing sets for the neural network are selected from these data. In this paper, for addressing subsequent issues, 80% of the data is taken as the training set, and the remaining 20% is used as the testing set.

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

Regarding the selection of the input layer parameters, it is necessary to analyze the correlation between time and various spatial parameters X_1, $X_2 \cdots X_n$ and the diffusion flux J as well as component concentration C. For instance, calculating correlation coefficients such as Pearson, Spearman, or Kendall coefficients or plotting distribution curves can help determine their correlations. This analysis guides the selection of the input parameters. Concerning the selection of the output layer parameters, the known experimental data in this study include only diffusion flux J and component concentration C. Therefore, $J(X_1, X_2,\cdots,X_n, t)$ and $C(X_1, X_2,\cdots,X_n, t)$ are used as the output values of the neural network. Regarding the selection of the number of nodes and layers in the hidden layers, this should be determined based on the desired prediction accuracy and computational efficiency. Regarding the signal propagation process between neural network layers, from the input layer to the hidden layer and then to the output layer, it is a unidirectional propagation process, which can be expressed as follows

$$a_i^l = \sigma(z_i^l) = \sigma(w^l \cdot a_i^{l-1} + b^l) \tag{5}$$

where a_i^l represents the set of outputs of all i^{th} neural networks in the l^{th} layer; z_i^l represents the set of inputs of all i^{th} neural networks in the l^{th} layer; w^l represents the set of all weights in the l^{th} layer; b^l represents the set of all biases in the l^{th} layer; and $\sigma(z_i^l)$ represents the activation function.

Due to the gradient calculation issue with the primary design parameters, this paper selects the Tanh (hyperbolic tangent) function, which is widely applicable and has error propagation ease, as the activation function for the neural network. Its expression is as follows

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{6}$$

2.2.2. Physics Information Section. Traditional neural networks typically minimize the difference between the actual values and the network's predictions as the training objective. They commonly use the mean squared error (MSE) as the loss function for neural network training. The specific expression is as follows

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \tag{7}$$

where N represents the number of actual values; y_i represents the actual values used for neural network training; and $f(x_i)$ represents the neural network's predictions.

However, this approach requires a large number of samples for training and cannot reflect the correlations among various parameters. To further enhance the training speed and prediction accuracy of the neural network, this paper integrates Fick's laws into the loss function. It calculates the physical residual terms corresponding to the conditions of Fick's laws and combines them with the mean squared error between actual and predicted values to jointly penalize solutions that do not satisfy both predictive and physical conditions. Building upon eq 8, an improved loss function is proposed as follows

$$L(\Theta, D) = \frac{1}{N_m}\sum_{i=1}^{N_m}(|J_i - J(X_1, \cdots, X_n, t)| + |C_i - C(X_1, \cdots, X_n, t)|)^2 + \frac{1}{N_c}\sum_{j=1}^{N_c}|f(X_1, \cdots, X_n, t)|^2 \tag{8}$$

where N_m represents the number of sets of experimentally determined data; N_c represents the number of neural network nodes; Θ represents the neural network's parameters to be identified, $\Theta = [w, b]$; J_i and C_i, respectively, represent the experimentally determined values of diffusion flux and component concentration; $J(X_1,\cdots,X_n, t)$ and $C(X_1,\cdots,X_n, t)$ represent, respectively, the network's predicted values of diffusion flux and component concentration; and $f(X_1,\cdots,X_2, t)$ represents the modified loss function derived from Fick's laws under different conditions, with an optimization goal of 0. The modified equation only contains the unknown diffusion coefficient D, and the specific equation needs to be determined based on different experimental conditions.

Based on the improved loss function expression in eq 9, it is embedded into the DNN framework, ultimately forming the PINN diffusion coefficient identification model. Under the data-physical-driven conditions, the PINN network structure is continuously updated, and the parameter values of the iterative loss function are also continuously updated, ultimately yielding the global optimal values or distribution of the diffusion coefficient D and the parameters Θ to be identified. The global optimum solution can be expressed as follows

$$(\Theta^*, D^*) = \arg\min L(\Theta, D) \tag{9}$$

3. NUMERICAL SOLUTION PROCEDURE

Figure 2 presents the flowchart of the proposed PINN model. And a simplified pseudocode is presented in Appendix. Three situations are considered in this section.

3.1. Both Diffusion Flux and Concentration Gradient are Known. It is assumed that both diffusion flux J and

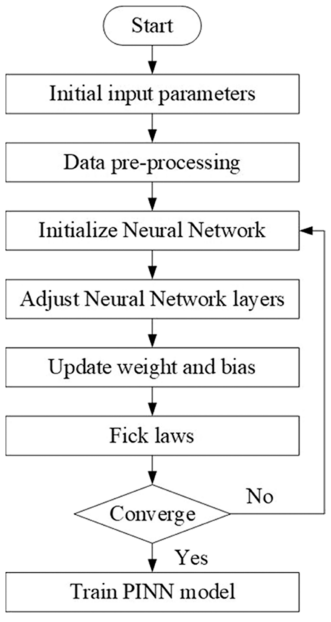


Figure 2. Flowchart of the proposed PINN model.

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

component concentration C have a strong correlation with distance and time. Therefore, the input layer parameters for the neural network are selected as distance and time (x and t), and the output parameters are diffusion flux J and component concentration C. Also, 80% of the data from the data set is used as the training data set, and the remaining 20% is used as the testing data set. The variations of MSE and iteration count with different numbers of layers are analyzed. As can be seen from Figure 3, it is found that the values of MSE decrease with

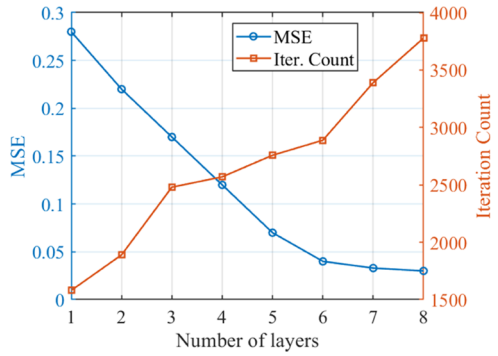


Figure 3. Variations of MSE and iteration count under different numbers of layers.

higher number of layers but vary little when the number of layers exceeds 6. In this case, more importantly, the value of the MSE is already under 0.05. That means, the error of the model satisfies the requirement and convergences under 6 hidden layers and more layers are unnecessary and will do little help. Furthermore, as the hidden layers are getting more, the iteration times are also getting longer, which means more calculation time and source will be needed. When the number of hidden layers exceeds 6, the iteration count is more than 3000. Therefore, with a target control error within 0.05, it is found that using 15 hidden nodes and 6 hidden layers achieves a relatively high accuracy and efficiency.

According to eq 10, the identification of the diffusion coefficient can be assumed as a one-dimensional problem. And, the diffusion coefficient cannot be treated as a constant, so further corrections are needed for eq 10. For the prediction error part of the diffusion flux J and component concentration C, the following equation can be derived based on eq 10

$$\mathrm{MSE} = \frac{1}{N_\mathrm{m}} \sum_{i=1}^{N_\mathrm{m}} (|J(x_i, t_i) - J_i| + |C(x_i, t_i) - C_i|)^2 \tag{10}$$

It is known that the value of diffusion coefficient D is influenced by distance x. To represent the variation of the diffusion coefficient, this paper uses a third-order polynomial to express the influence of distance on the diffusion coefficient. The specific expression is as follows

$$D(x) = ax^3 + bx^2 + cx + d \tag{11}$$

Combining eq 10 with eq 11, the physical constraint equation for diffusion motion is as follows

$$f(x, t) = (ax^3 + bx^2 + cx + d)\left(\frac{\partial C}{\partial x}\right) - J \tag{12}$$

Then, by combining eqs 10, 11, and 12, the total loss function under the conditions is obtained as follows

$$L(\theta, a, b, c, d) = \frac{1}{N_\mathrm{m}} \sum_{i=1}^{N_\mathrm{m}} (|J(x_i, t_i) - J_i| + |C(x_i, t_i) - C_i|)^2 + \frac{1}{N_\mathrm{c}} \sum_{j=1}^{N_\mathrm{c}} |f(x)|^2 \tag{13}$$

So far, the diffusion coefficient identification model based on PINN has been successfully established, and its specific structure is shown in Figure 4.

After the diffusion coefficient identification model is established, it is possible to solve for the diffusion coefficient. In this paper, the data set which includes a total of 505 sets of data for diffusion flux J and 505 sets for component concentration C. The neural network structure is given in Figure 4. After training and optimization, the convergence curve of the corresponding loss function is plotted as follows in Figure 5.

From Figure 6, it can be observed that under the condition of both physical and data-driven factors, the PINN diffusion coefficient identification model requires nearly 3000 iterations to minimize the loss function, demonstrating a relatively fast computational speed. After training the network, the globally optimal values for the diffusion coefficient are determined to be a = 0.05608, b = −0.06263, c = 0.02506, and d = 0.04971.

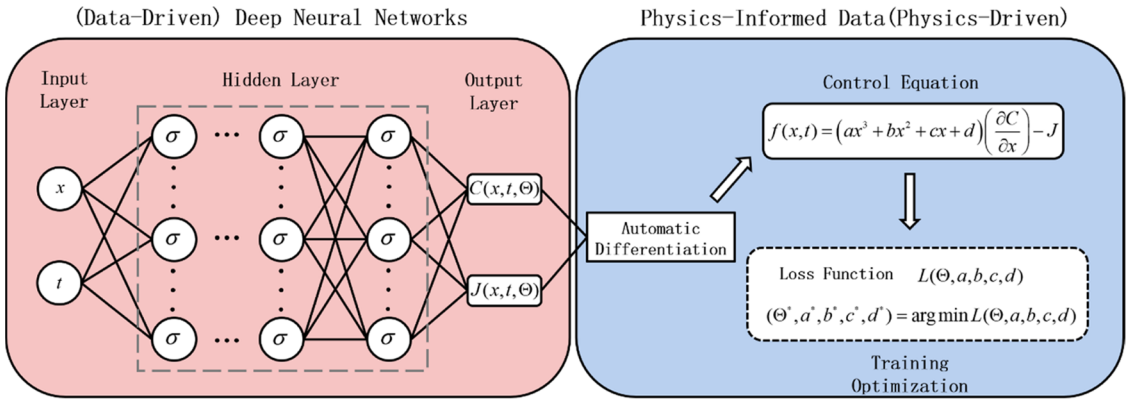


Figure 4. Diffusion coefficient identification model based on PINN.

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

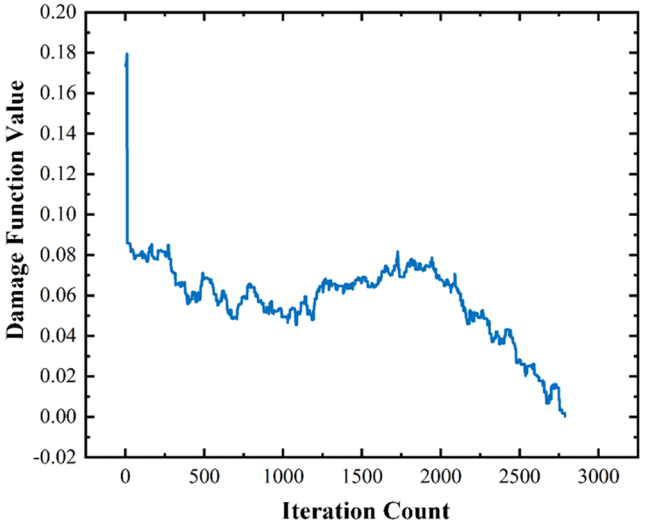


Figure 5. Convergence analysis of the PINN diffusion coefficient identification model.

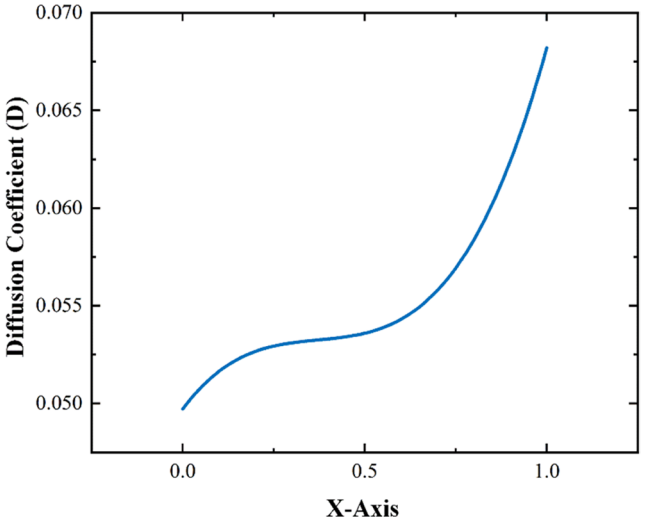


Figure 6. Diffusion coefficient variation with the component concentration.

Substituting these values into eq 14, the expression for the diffusion coefficient as a function of distance x is obtained

$$D(x) = 0.05608x^3 - 0.06263x^2 + 0.02506x + 0.04971 \tag{14}$$

Based on eq 14, the diffusion coefficient values for different distances are calculated. Combining these values with the component concentration values, the variation curve of the diffusion coefficient D with component concentration C is plotted as shown in Figure 7.

From Figure 7, it can be observed that the diffusion coefficient increases with an increase in the component concentration, forming a monotonically increasing curve. The curve's trend follows an S-shaped pattern, with a gradually slowing increase until it stabilizes, followed by a further increase. This behavior is in line with the general trend of the diffusion coefficient changing with component concentration.

3.2. Diffusion Flux is Unknown While Concentration Gradient is Known. In most cases, only one parameter would be obtained, and the other is unknown. So now, it is assumed

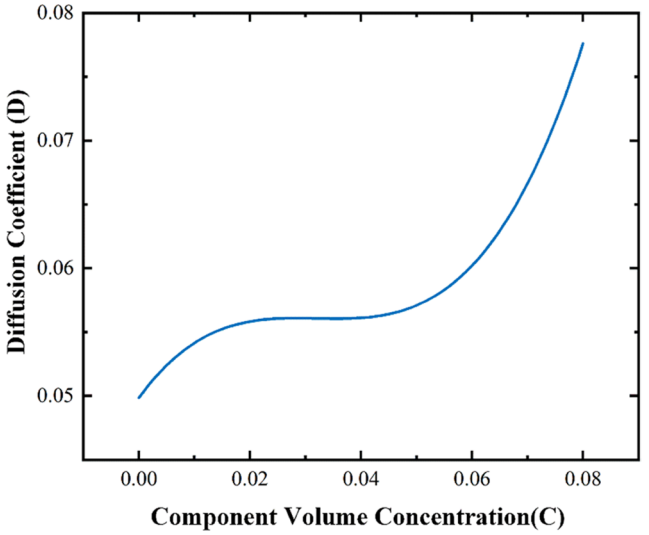


Figure 7. Diffusion coefficient behavior with the component concentration.

that the diffusion flux J is unknown while the spatial distribution of component concentration C is known. The expression of the diffusion coefficient varying with space can be assumed as

$$D(x) = ax^3 + bx^2 + cx + d \tag{15}$$

According to eqs 13 and 15, one can obtain

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) = \frac{\partial D}{\partial x}\frac{\partial C}{\partial x} + D\frac{\partial^2 C}{\partial x^2} \tag{16}$$

Therefore, the physical constraint equation under this diffusion process is

$$\begin{aligned} f(x) &= \frac{\partial C(x,t)}{\partial t} - \frac{\partial D(x)}{\partial x}\frac{\partial C(x,t)}{\partial x} - D(x)\frac{\partial^2 C(x,t)}{\partial x^2} \\ &= \frac{\partial C(x,t)}{\partial t} - (3ax^2 + 2bx + c)\frac{\partial C(x,t)}{\partial x} \\ &\quad - (ax^3 + bx^2 + cx + d)\frac{\partial^2 C(x,t)}{\partial x^2} \end{aligned} \tag{17}$$

And, the loss function can be expressed as

$$\begin{aligned} L(\Theta, a, b, c, d) &= \frac{1}{N_m}\sum_{i=1}^{N_m} |C(x_i, t_i) - C_i|^2 \\ &\quad + \frac{1}{N_c}\sum_{j=1}^{N_c} |f(x)|^2 \end{aligned} \tag{18}$$

By incorporating the loss function into the PINN model and using the data set, calculation can be performed. After less than 2000 iterations, the spatial part function of the diffusion coefficient can be obtained

$$D(x) = 0.05666x^3 - 0.05663x^2 + 0.02114x + 0.04993 \tag{19}$$

Figure 8 shows the change in the loss function value during the iterative process. As can be seen from the figure, in the early iterations, the loss function has already decreased to a low level. In the subsequent iterations, there are slight fluctuations, but they remain within an acceptable error range until

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

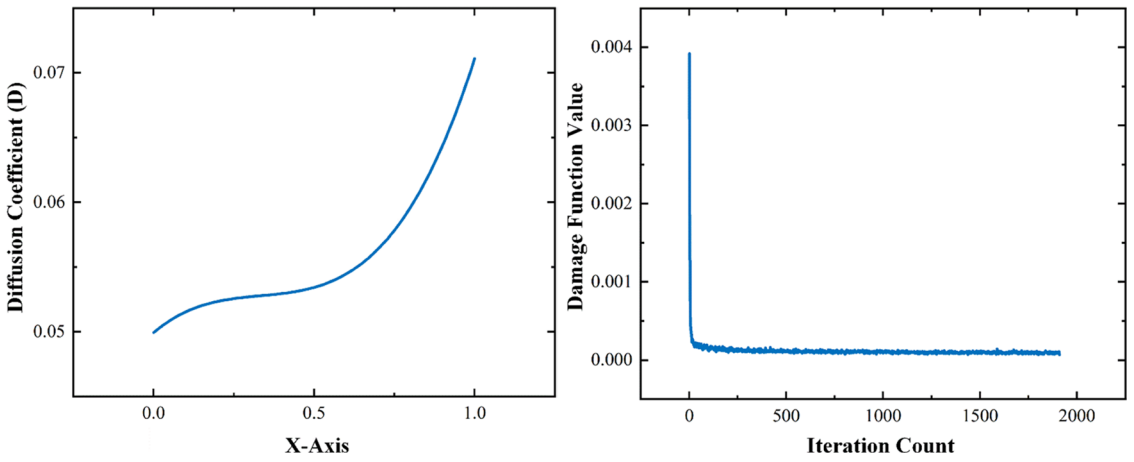


Figure 8. Convergence of the loss function during iterations.

convergence tolerance. This indicates that the solution process of the model has good convergence, achieving convergence requirements in less than 2000 iterations. Figure 9 illustrates the relationship between the component volume concentration and the diffusion coefficient in the solution. From the figure, it can be observed that the diffusion coefficient gradually increases with an increase in concentration, following an increasing cubic curve, consistent with the general trend of the diffusion coefficient changing with concentration.

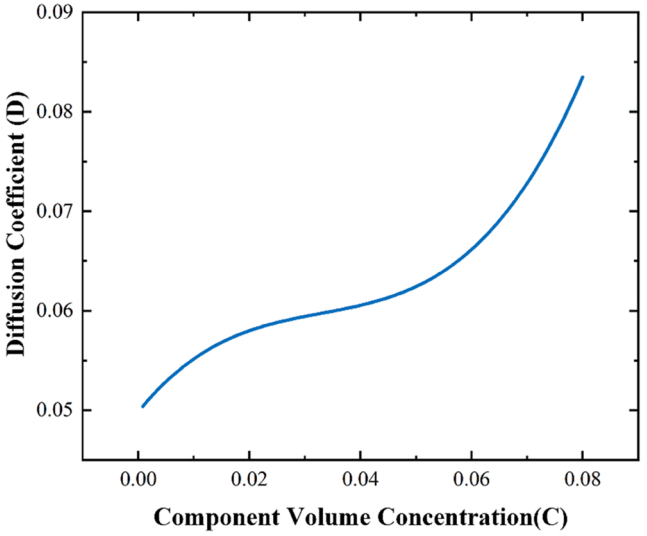


Figure 9. Diffusion coefficient vs component concentration.

3.3. Diffusion Flux is Unknown While Concentration Gradient is Known.

On the other hand, it is assumed that the diffusion flux J is known while the spatial distribution of component concentration C is unknown. The expression of the diffusion coefficient varying with space can be also assumed as

$$D(x) = ax^3 + bx^2 + cx + d \tag{20}$$

Integrating both sides of eq 20 with respect to t

$$\int \frac{\partial C}{\partial t} \mathrm{d}t = \int \frac{\partial J}{\partial x} \mathrm{d}t \tag{21}$$

Then, one has

$$C = \int \frac{\partial J}{\partial x} \mathrm{d}t + A(x) \tag{22}$$

where $A(x)$ is the integral expression with respect to x, determined by the initial condition of C. Substituting eq 21 into eq 22, one can obtain

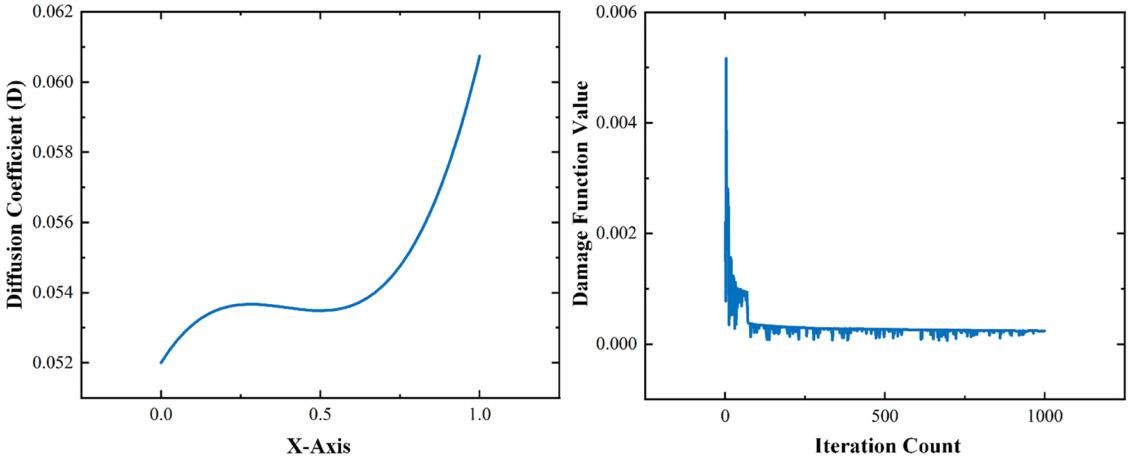


Figure 10. Loss function behavior during iterations.

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

$$J = D\frac{\partial C}{\partial x} = D\frac{\partial}{\partial x}\left(\int \frac{\partial J}{\partial x}\mathrm{d}t + A(x)\right) \tag{23}$$

Therefore, the physical constraint equation under this diffusion process can be expressed as

$$\begin{aligned} f(x) &= J - D\frac{\partial}{\partial x}\left(\int \frac{\partial J}{\partial x}\mathrm{d}t + A(x)\right) \\ &= J - (ax^3 + bx^2 + cx + d)\frac{\partial}{\partial x}\left(\int \frac{\partial J}{\partial x}\mathrm{d}t + A(x)\right) \end{aligned} \tag{24}$$

The loss function can be expressed as

$$L(\Theta, a, b, c, d) = \frac{1}{N_\mathrm{m}}\sum_{i=1}^{N_\mathrm{m}} |J(x_i, t_i) - J_i|^2 + \frac{1}{N_\mathrm{c}}\sum_{j=1}^{N_\mathrm{c}} |f(x)|^2 \tag{25}$$

By incorporating the loss function into the PINN model and using the data set, calculation can be performed. After approximately 1000 iterations, the spatial part function of the diffusion coefficient in this problem is

$$D(x) = 0.0349x^3 - 0.04082x^2 + 0.01466x + 0.052 \tag{26}$$

Figure 10 shows the change in the loss function value during the iterations. As seen in the graph, in the early iterations, the loss function exhibits significant oscillations, but the overall level still decreasing. In subsequent iterations, there are varying degrees of numerical oscillations, but they remain within an acceptable error range until convergence tolerance is reached. This is because the solution of the diffusion model involves complex numerical operations with differentiation and integration, leading to inevitable numerical oscillations and computational errors. Figure 11 shows the relationship between the component volume concentration and diffusion coefficient in the solution. From the graph, it can be observed that the diffusion coefficient tends to increase with increasing concentration, but there is a plateau in the range of component concentrations from 0.01 to 0.04, with a slow growth rate.

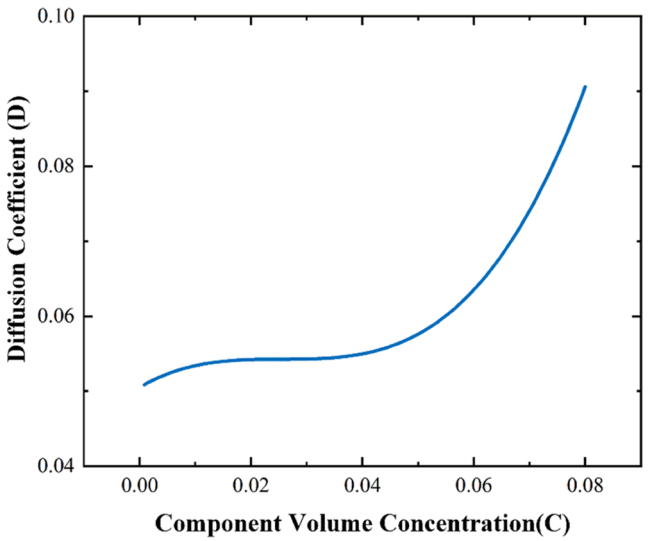


Figure 11. Diffusion coefficient vs component concentration with plateau.

4. ANALYSIS AND DISCUSSION

4.1. Effect of the Error Level of the Testing Data. Because the known experimental data consist of single-point measurements, it is impossible to objectively assess the impact of measurement data errors on the diffusion coefficient. Therefore, when considering the error level of experimental data, it is assumed that the known experimental data closely approximate the true values. One can then artificially alter the degree of deviation in the experimental data to obtain experimental data with different error levels.

For both the spatial and temporal distribution matrices of the component concentration and diffusion flux, 10% of the data points have been randomly selected for deviation. Taking the spatial distribution matrix of component concentration as an example, the specific deviation operation is expressed as

$$C'_{x_i,t_j} = [1 + (-1)^{i+j}\xi]C_{x_i,t_j} \tag{27}$$

where x_i, t_j denotes the measurement value at the i^{th} spatial location at the j^{th} time point, C'_{x_i,t_j} denotes the new component concentration values after deviation, C_{x_i,t_j} denotes the initial component concentration, and ξ denotes the degree of deviation, i.e., the artificially set error level.

The same operation is performed for the deviation of random samples of the spatiotemporal distribution matrix of the diffusion flux.

$$J'_{x_i,t_j} = [1 + (-1)^{i+j}\xi]J_{x_i,t_j} \tag{28}$$

From Figure 12, it is evident that with the same amount of bias data, when the input data are unbiased, the loss function

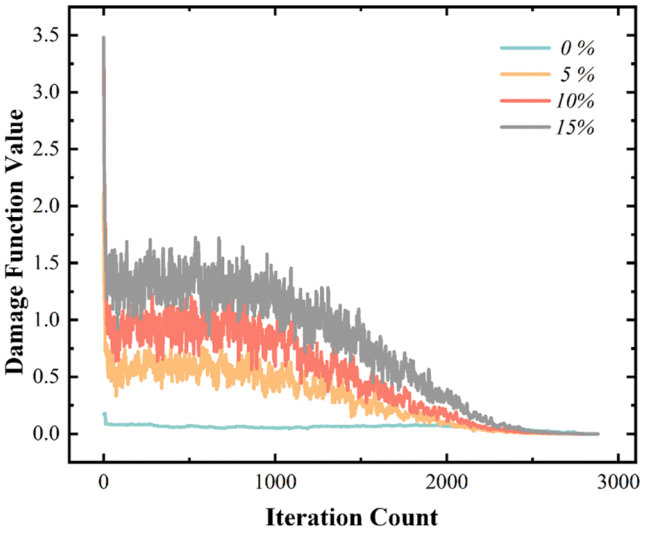


Figure 12. Understanding loss function oscillations in PINN training.

exhibits good convergence. However, when there is bias in the input data, the loss function demonstrates significant oscillations. Comparing the convergence curves of the loss function for error levels of 5, 10, and 15%, it can be further concluded that as the error level increases, the amplitude of oscillations in the loss function becomes larger, and the number of iterations required to reach convergence increases. Additionally, it is observed that when data contains errors, there is a plateau period in the first 1000 iterations, during which the loss function rapidly oscillates around a certain value

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

(plateau value). Moreover, as the error level in the input data increases, the duration of this plateau period and the plateau value both increase.

When training the PINN model, noticeable and severe oscillations in the loss function can occur. While in most cases training can continue, these oscillations are indicative of underlying issues. Through the comparative analysis in Figure 13, one can identify two primary factors that influence the

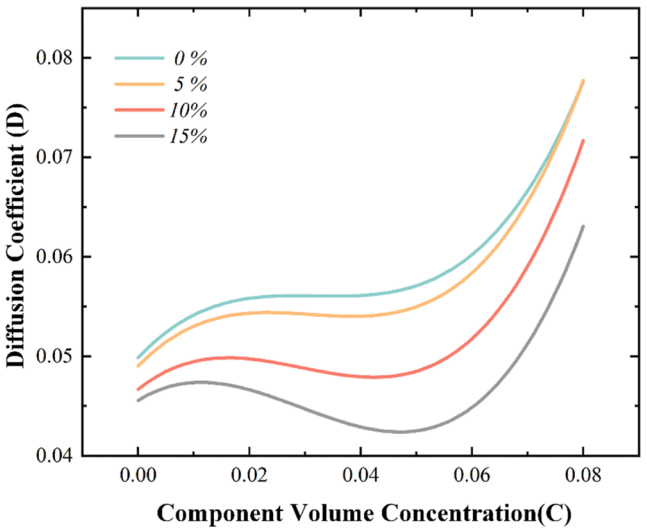


Figure 13. Exploring the impact of error levels on the relationship between component volume concentration and diffusion coefficient.

oscillation of the convergence function. First, the limited size of the training sample data set is a significant factor. Each variable has at most a numerical matrix of dimensions, provided in the attached data. The scarcity of data samples prevents the mitigation of the impact of error data, which constitutes 10% of the data set, leading to the prominent oscillations in the loss function. Second, due to the random selection of data for perturbation, we observe that, at t = 0.2s, the magnitude of data values is relatively small, with the smallest order of magnitude being on the order of 10^{-6}. In contrast, at t = 1s, the smallest order of magnitude for species concentration is on the order of 10^{-3}, and for diffusion flux, it is on the order of 10^{-4}. Changes in the higher-order magnitudes in the randomly perturbed data have a more significant impact than those in the lower-order magnitudes. This is also one of the objective factors contributing to the oscillations observed. In conclusion, these two factors, namely, the limited size of the training data set and the varying magnitudes of perturbed data, play a crucial role in causing the observed oscillations in the convergence function during the training of the PINN network.

Based on different error levels, further analysis of the relationship between component volume concentration and diffusion coefficient is presented in Figure 13. As the error level in the data set gradually increases, the relationship between the diffusion coefficient and the component volume concentration, which initially exhibited a roughly monotonically increasing trend with increasing component volume concentration, gradually changes to a pattern of initial increase followed by a decrease, reaching a local minimum before continuing to increase monotonically. When a 15% deviation occurs in the error data, which constitutes 10% of the data set, the local minimum of the diffusion coefficient with respect to the component volume concentration falls even below the initial condition's corresponding diffusion coefficient value. This implies that under conditions where dynamic equilibrium convergence has not been reached, a reverse diffusion process (aggregation of grain boundary impurities) has a reaction rate exceeding the diffusion rate. This suggests a significant reduction in the accuracy of the mathematical model for identifying the diffusion coefficient at the 15% error level. Furthermore, by comparing error levels of 5 and 10%, it can be observed that when the error level is relatively low, the trends in the relationship between the component volume concentration and diffusion coefficient remain largely consistent. This analysis indicates that the mathematical model constructed based on the known data set for identifying the diffusion coefficient is effective and can provide different feedback responses for different error levels in the data samples. Additionally, within a certain range of component volume concentrations, the diffusion coefficient consistently exhibits an approximate monotonically increasing trend as the component volume concentration increases.

4.2. Effect of the Amount of the Testing Data. In this model, the identification of diffusion coefficients in the context of substance diffusion described by Fick's laws presents significant nonlinear and uncertain characteristics. Moreover, during the experimental measurement process, the measurement methods, observation techniques, measuring instruments, and data processing procedures may all introduce certain errors, ultimately leading to bias in the identification of diffusion coefficients. When using neural networks to solve this problem, a substantial amount of data samples is often required for training to enhance the network's ability to fit and predict nonlinear functions. Insufficient data samples can lead to overfitting, where the model becomes overly influenced by the data errors during training, resulting in the amplification of sample data errors in the predictions and, consequently, inaccurate computational results. To further investigate the impact of the sample data size on the identification of diffusion coefficients in this model, calculations will be performed using randomly selected data at 100, 80, and 60% of the data. The results of these calculations are discussed to analyze how the sample data size affects the diffusion coefficient identification in the PINN model.

As shown in Figure 14, all three sets of computational results exhibit relatively good convergence and meet the convergence tolerance requirements. When 100% of the data volume is used, the loss function value does not exhibit oscillations during the iterative process and continuously decreases until it reaches convergence tolerance. This indicates an excellent convergence performance. Although a local optimal solution causes a sharp drop in the loss function value at the 300th iteration, it does not significantly impact the subsequent iterations. In comparison to the 100% data volume case, both the 80 and 60% data volume cases exhibit oscillations in the loss function value in the early stages of iteration due to the reduced data volume. The computation process is less stable than that of the 100% data volume case, with the 60% data volume case showing oscillations slightly larger than those of the 80% data volume case. Furthermore, upon comparison of the convergence curves of the three cases, the convergence curve for the 100% data volume case exhibits a slower initial descent in the loss function value. This is because the larger data volume increases the time and computational cost of

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

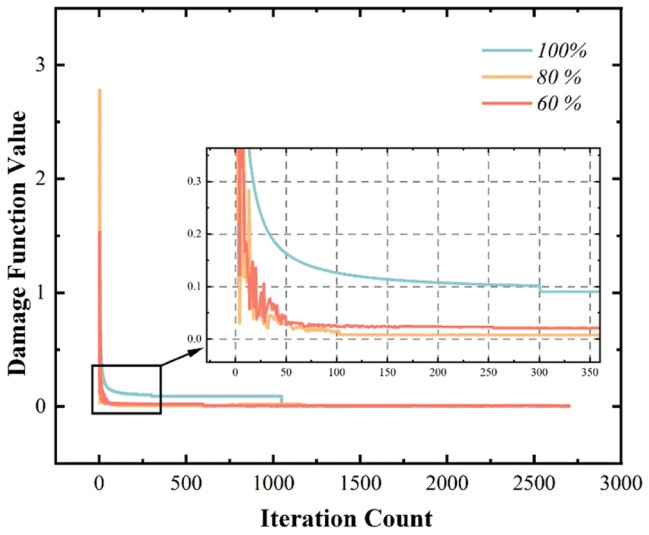


Figure 14. Diffusion coefficient variation with component volume concentration at different data volumes.

network training, making it more challenging to find the optimal solution. In contrast, the lower data volumes (such as 80 and 60%) experience faster declines in the loss function value but also exhibit convergence oscillations and instability.

As depicted in Figure 14, the computational results for all three different data volumes show a general trend of monotonically increasing diffusion coefficients with increasing component volume concentration, which aligns with the typical behavior of diffusion coefficients varying with the component volume concentration. The diffusion coefficient calculated with 80% of the data volume exhibits the fastest rate of increase. However, it starts to deviate gradually from the curve of the 100% data volume case when the species concentration exceeds approximately 0.05. This deviation is a result of the accumulated computational errors due to the reduced training data input into the neural network, leading to a decrease in prediction accuracy. Nevertheless, the error remains within an acceptable range, indicating that training the neural network with 80% of the data volume is effective. On the other hand, the curve for the 60% data volume case shows a plateau and even a decreasing trend in the diffusion coefficient within the concentration range of 0.01−0.05. This behavior occurs because the input data volume for the neural network is only 60%, resulting in insufficient data for the PINN model, leading to inaccurate predictions and an anomalous relationship between component volume concentration and diffusion coefficients. In summary, using an appropriate data volume (above 80%) for neural network training effectively ensures the accuracy of the model's prediction results, while insufficient data volumes can lead to abnormal prediction values due to overfitting.

4.3. Effect of the Selection of the Initial Guess. In a DNN model, it is essential to input initial sample data for training to obtain a neural network that meets the required accuracy. The goal is to ensure that the predicted data approach the real data within an allowable error range. The selection of initial samples plays a significant role in determining the training effectiveness and time efficiency of the neural network. An effective method for choosing initial samples can greatly enhance the training efficiency with minimal time costs and improve prediction accuracy. Therefore, this study considers four initial sample data selection methods, each involving different proportions of valid data. The goal is to investigate how the proportion of valid data in the initial samples affects the performance of the PINN model and the identification of diffusion coefficients.

It is evident that the numerical values at spatial boundaries (x = 0,1) and at the initial time (t = 0) are all zero, indicating the presence of zero boundary conditions and zero initial conditions. Additionally, the component concentration measurement data exhibit a spatial distribution change at the initial time, which can be considered as anomalous data due to the spatial discontinuity. These abrupt changes in boundary data can have an effect on the training efficiency of the neural network and are treated as invalid data. Therefore, the spatiotemporal boundary data will be considered as invalid data, while the remaining data will be considered as valid data, as depicted in Figure 15. Subsequently, initial data samples were selected using a combination of random sampling and setting the proportion of valid data, as outlined in Table 1.

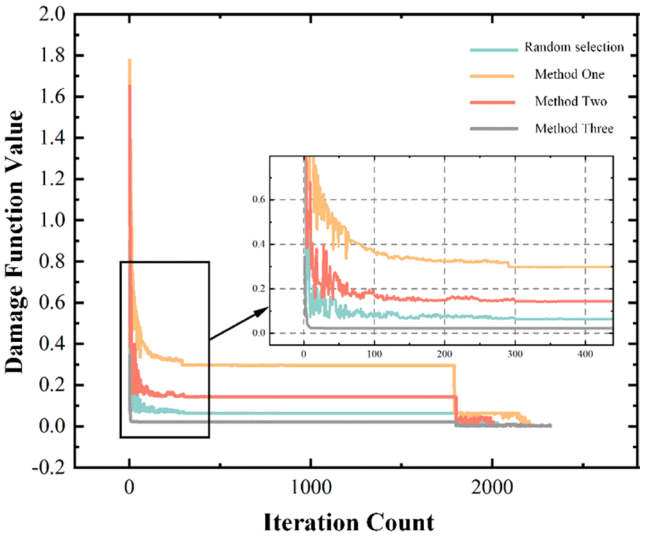


Figure 15. Comparative analysis of the component volume concentration and diffusion relationship: random selection vs Method 1.

Table 1. Initial Data Sample Selection Method

selection method	description
random selection	randomly select data
method 1	90% valid data, 10% invalid data
method 2	80% valid data, 20% invalid data
method 3	70% valid data, 30% invalid data

As shown in Figure 15, all four initial data selection methods exhibit varying degrees of oscillation in the loss function value during the early stages of iteration. This oscillation is primarily due to the significant influence of the initial values on the results in the initial computations, resulting in numerical oscillations. However, as the number of iterations increases, this oscillation gradually diminishes and the loss function value stabilizes, approaching convergence. In the later stages of iteration (approximately around 1700 iterations), there is a sharp drop in the function value. This phenomenon occurs because, during the iterative computation process, the calculation encounters a local optimal solution, leading to

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

the so-called *premature convergence* phenomenon, where the loss function value decreases. However, the model aims to find the global optimal solution; thus, this local optimal solution does not meet the requirements, and the loss function value does not satisfy the convergence tolerance. Therefore, the iterative process continues until the global optimal solution is reached.

The convergence curve for the *Random Selection* method exhibits relatively small loss function values at the beginning of the iteration with less pronounced oscillations during the early stages, indicating good convergence performance. However, this method involves no deliberate intervention and carries a high degree of uncertainty and dependence on probability. It can be used with a certain level of confidence when it is challenging to distinguish between valid and invalid data. Method 1, which includes 90% valid data, exhibits higher loss function values during the early stages of iteration but experiences smaller amplitude and shorter duration of oscillations. It demonstrates good stability and can quickly find the global optimal solution after encountering a local optimal solution in the later stages of iteration, reaching the convergence requirements. Therefore, Method 1 has the best overall computational performance. In comparison to Method 1, Method 2 contains a higher proportion of invalid data (20%), leading to more pronounced oscillations during the early stages of iteration with larger amplitudes and longer durations. It faces greater challenges in convergence throughout the entire iterative process but ultimately meets the convergence tolerance requirements. Method 3, despite not exhibiting significant oscillations during the early stages of iteration, experiences small oscillations throughout the entire iterative process and fails to meet the convergence tolerance requirements. Therefore, Method 3 is not successful in solving the problem.

As shown in Figure 16, both the *Random Selection* and Method 1 methods for component volume concentration and diffusion relationship curves exhibit similar trends. They both show an increase in the diffusion coefficient with increasing concentration, and the results are relatively close, demonstrating good consistency. This further confirms that these two methods yield better computational results. The relationship curve for Method 2 also shows an overall increasing trend, but due to its lower proportion of valid data (80%), the increase in the diffusion coefficient is less pronounced in the concentration range of 0.01 to 0.05. Additionally, it deviates significantly from the results of *Random Selection* and Method 1, indicating some bias in the computational results due to the increased amount of invalid data in Method 2. In the case of Method 3, the relationship curve shows a decrease in the diffusion coefficient within the concentration range of 0.01−0.05. This behavior is not observed in the rest of the solution process, and as previously analyzed, Method 3 does not converge during the iterative process. This further confirms that the results obtained through Method 3 are inaccurate.

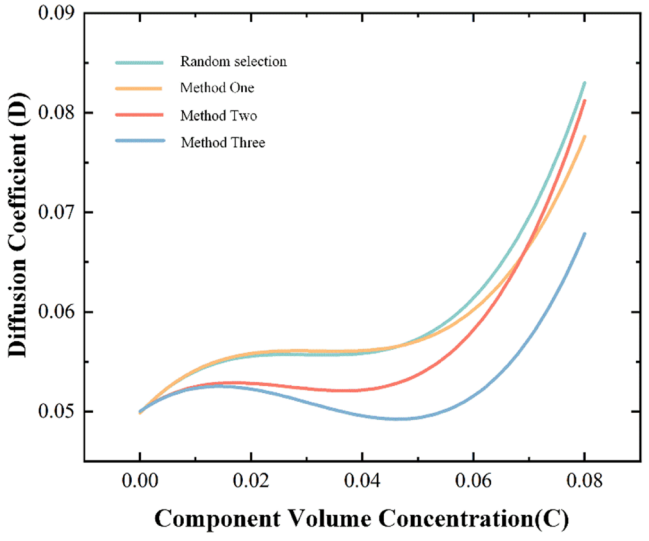


Figure 16. Effect of the valid data proportion on convergence and accuracy: random selection vs Method 1.

In conclusion, when there is a higher proportion of valid data in the initial sample data, the convergence during the solution process is better and the accuracy of the results is higher. For *Random Selection*, even though the proportion of valid data cannot be controlled, it can still meet the requirements for computational results. Therefore, in situations in which the proportion of valid data is unknown, random selection can be a viable method.

5. CONCLUSIONS

This study introduces a physics-informed neural network (PINN) model as an effective and efficient solution to the challenging task of diffusion coefficient identification in inverse problems. By seamlessly integrating Fick's laws into the neural network framework, the PINN model demonstrates its capability to provide reliable estimates under various scenarios involving known and unknown flux and concentration gradients. The main conclusions can be drawn as follows:

(1) Three specific situations are considered herein, including known diffusion flux and concentration gradient, known diffusion flux and unknown concentration gradient, and, unknown diffusion flux and known concentration gradient. Results demonstrate the model's efficiency, obtaining diffusion coefficients in less than 1000, 2000, and 3000 iterations for the respective situation.

(2) The effect of the error level of the testing data is investigated. As the error level increases, the amplitude of oscillations in the loss function becomes larger, and the number of iterations required to reach convergence increases, but all less than 3000. And changes in the higher-order magnitudes in the randomly perturbed data have a more significant impact compared to those in the lower-order magnitudes.

(3) The effect of the amount of the testing data is investigated. The lower the data volume is, the faster the loss function value declines but the more oscillation and instability exhibit. Also, insufficient data volume can lead to abnormal prediction values due to overfitting.

(4) The effect of the selection of the initial guess is investigated. When there is a higher proportion of valid data in the initial sample data, the convergence during the solution process is better, and the accuracy of the results is higher. 30% of invalid data causes a failure in solving the problem.

In the future, the proposed model would be improved to accommodate more situations, such as missing data, data incorrection, etc. In that case, the model would be introduced to deal with practical physical problems, e.g., chloride diffusion

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

in concrete. Also, extensive performance evaluators were employed to estimate the accuracy of the result.

APPENDIX

Below is a simplified pseudocode for a physics-informed neural network (PINN) model addressing the diffusion coefficient identification problem in Fick's law.

```
# Import necessary libraries
import tensorflow as tf
import numpy as np

# Define the PINN model
class PINNModel(tf.keras.Model):
    def __init__(self):
        super(PINNModel, self).__init__()

        # Define neural network layers for the solution field
        self.dense1 = tf.keras.layers.Dense(128, activation='relu')
        self.dense2 = tf.keras.layers.Dense(128, activation='relu')
        self.output_layer = tf.keras.layers.Dense(1)

        # Define neural network layers for the diffusion coefficient
        self.diff_dense1 = tf.keras.layers.Dense(64, activation='relu')
        self.diff_dense2 = tf.keras.layers.Dense(64, activation='relu')
        self.diff_output_layer = tf.keras.layers.Dense(1, activation='softplus')

    def call(self, x, t):
        # Concatenate input features (space and time)
        input_features = tf.concat([x, t], axis=1)

        # Neural network for the solution field
        solution_field = self.dense1(input_features)
        solution_field = self.dense2(solution_field)
        u = self.output_layer(solution_field)

        # Neural network for the diffusion coefficient
        diff_input = tf.concat([x, t], axis=1)
        diff_features = self.diff_dense1(diff_input)
        diff_features = self.diff_dense2(diff_features)
        D = self.diff_output_layer(diff_features)

        return u, D
```

```
# Define loss function incorporating Fick's law
def loss_function(u_pred, D_pred, x, t, u_true, D_true):
    # Define your physics-informed loss terms based on Fick's law
    # ...

    # Calculate the total loss
    total_loss = physics_loss + data_loss

    return total_loss

# Initialize PINN model
model = PINNModel()

# Compile the model
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss=loss_function)

# Training loop
for epoch in range(num_epochs):
    # Sample training data (x, t, u_true, D_true)
    # ...

    # Train the PINN model
    with tf.GradientTape() as tape:
        u_pred, D_pred = model(x, t)
        loss = loss_function(u_pred, D_pred, x, t, u_true, D_true)

    gradients = tape.gradient(loss, model.trainable_variables)
    optimizer.apply_gradients(zip(gradients, model.trainable_variables))

    # Print or log training progress
    if epoch % print_interval == 0:
        print(f"Epoch {epoch}, Loss: {loss.numpy()}")

# After training, use the trained model for predictions
# ...
```

ASSOCIATED CONTENT

Data Availability Statement

The data sets generated during and/or analyzed during the current study are not publicly available due to the confidentiality requirements of the project but are available from the corresponding author upon reasonable request.

AUTHOR INFORMATION

Corresponding Author

You Wang − *School of Civil Engineering, Central South University, Changsha 410075, People's Republic of China;* orcid.org/0000-0003-4795-8403; Email: ywang1920@csu.edu.cn

Authors

Dongchen Li − *School of Civil Engineering, Central South University, Changsha 410075, People's Republic of China*

Bin Yan − *School of Civil Engineering, Central South University, Changsha 410075, People's Republic of China*

Tianya Gao − *School of Civil Engineering, Central South University, Changsha 410075, People's Republic of China*

Guowei Li − *School of Civil Engineering, Central South University, Changsha 410075, People's Republic of China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.3c07924

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors acknowledge the financial support from the National Natural Science Foundation of China (51778633, 51308552), China Railway Construction Corporation's 2022 Annual Scientific and Technological Research and Development Plan and Funding Subjects (2022-C1), the Science and Technology Research and Development Plan of China Railway Corporation in 2020 (41, 243), and the 2022 degrees Guangzhou Metro Design and Research Institute Co. (KY-2022-014).

NOMENCLATURE

a_i^l	the set of outputs of all i neural networks in layer l
b^l	the set of all biases in layer l
C	the volume concentration of diffusing substances (components) (number of atoms/m^3 or kg/m^3)
D	diffusion coefficient, the flow rate through unit area per unit time when the concentration gradient is one unit (m^2/s)
$\tilde{D}$	the integrated diffusion coefficient (m^2/s)
$f(x_i)$	the predicted value of the neural network
J	diffusion flux is the flow of diffusing material through unit cross-sectional area perpendicular to the direction of diffusion in unit time (kg/(m^2·s))
N	the number of true values
N_A, N_B	the mole fractions of components A and B, respectively
N_c	the number of coordination points of the neural network
N_m	the number of groups of experimental measurement data
t	the measurement time of experimental data
w^l	the set of all weights in layer l
x	measurement location of experimental data (one-dimensional space)
y_i	the true values used for neural network training
z_i^l	the set of inputs of all i neural networks in layer l

GREEK LETTERS

∇	gradient operator
$\sigma(z_i^l)$	represents the activation function
Θ	the parameters to be identified of the neural network, $\Theta = [w, b]$
ξ	indicates the degree of deviation, that is, the artificially set error level
v	the translation speed of the reference plane at x (m/s)

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857

ABBREVIATIONS

DNN deep neural networks
MSE mean square error
PINN physics-informed neural network
Tanh hyperbolic tangent function

REFERENCES

(1) Wang, L. A general theory of diffusion. *Prog. Theor. Phys.* **1999**, *101* (3), 541−557.
(2) Ajithkumar, M.; Lakshminarayana, P.; Vajravelu, K. Diffusion effects on mixed convective peristaltic flow of a bi-viscous Bingham nanofluid through a porous medium with convective boundary conditions. *Phys. Fluids* **2023**, *35* (3), No. 032008, DOI: 10.1063/5.0142003.
(3) Vareda, J. P. On validity, physical meaning, mechanism insights and regression of adsorption kinetic models. *J. Mol. Liq.* **2023***376*. DOI: 10.1016/j.molliq.2023.121416.
(4) Baroghel-Bouny, V.; Kinomura, K.; Thiery, M.; Moscardelli, S. Easy assessment of durability indicators for service life prediction or quality control of concretes with high volumes of supplementary cementitious materials. *Cem. Concr. Compos.* **2011**, *33* (8), 832−847.
(5) Baroghel-Bouny, V.; Thiéry, M.; Wang, X. Performance-based assessment of durability and prediction of RC structure service life: transport properties as input data for physical models. *Mater. Struct.* **2014**, *47* (10), 1669−1691.
(6) Liberati, E. A. P.; Nogueira, C. G.; Leonel, E. D.; Chateauneuf, A. Nonlinear formulation based on FEM, Mazars damage criterion and Fick's law applied to failure assessment of reinforced concrete structures subjected to chloride ingress and reinforcements corrosion. *Eng. Failure Anal.* **2014**, *46*, 247−268.
(7) Kim, D.-W.; Cho, H.-H.; Lee, W.-B.; Cho, K. T.; Cho, Y.-G.; Kim, S.-J.; Han, H. N. A finite element simulation for carburizing heat treatment of automotive gear ring incorporating transformation plasticity. *Mater. Des.* **2016**, *99*, 243−253.
(8) Cevoli, C.; Panarese, V.; Catalogne, C.; Fabbri, A. Estimation of the effective moisture diffusivity in cake baking by the inversion of a finite element model. *J. Food Eng.* **2020**, *270*, No. 109769.
(9) Jiang, X.; Xu, M.; Qi, H. The fractional diffusion model with an absorption term and modified Fick's law for non-local transport processes. *Nonlinear Anal.: Real World Appl.* **2010**, *11* (1), 262−269.
(10) Lowney, J. R.; Larrabee, R. D. The use of Fick's law in modeling diffusion processes. *IEEE Trans. Electron Devices* **1980**, *27* (9), 1795−1798.
(11) Ghazanfari, A.; Emami, S.; Tabil, L. G.; Panigrahi, S. Thin-Layer Drying of Flax Fiber: I. Analysis of Modeling Using Fick's Second Law of Diffusion. *Drying Technol.* **2006**, *24* (12), 1631−1635.
(12) Shaban, W. M.; Elbaz, K.; Yang, J.; Shen, S.-L. A multi-objective optimization algorithm for forecasting the compressive strength of RAC with pozzolanic materials. *J. Cleaner Prod.* **2021**, *327*, No. 129355.
(13) Shaban, W. M.; Yang, J.; Elbaz, K.; Xie, J.; Li, L. Fuzzy-metaheuristic ensembles for predicting the compressive strength of brick aggregate concrete. *Resour., Conserv. Recycl.* **2021**, *169*, No. 105443.
(14) Shaban, W. M.; Elbaz, K.; Zhou, A.; Shen, S.-L. Physics-informed deep neural network for modeling the chloride diffusion in concrete. *Eng. Appl. Artif. Intell.* **2023**, *125*, No. 106691.
(15) Batuwatta-Gamage, C. P.; Rathnayaka, C. M.; Karunasena, H. C. P.; Wijerathne, W. D. C. C.; Jeong, H.; Welsh, Z. G.; Karim, M. A.; Gu, Y. T. A physics-informed neural network-based surrogate framework to predict moisture concentration and shrinkage of a plant cell during drying. *J. Food Eng.* **2022**, *332*, No. 111137.
(16) Lu, D.; Christov, I. C. Physics-informed neural networks for understanding shear migration of particles in viscous flow. *Int. J. Multiphase Flow* **2023**, *165*, No. 104476.
(17) Fang, Y.; Wu, G.-Z.; Wang, Y.-Y.; Dai, C.-Q. Data-driven femtosecond optical soliton excitations and parameters discovery of the high-order NLSE using the PINN. *Nonlinear Dyn.* **2021**, *105* (1), 603−616.
(18) Haghighat, E.; Raissi, M.; Moure, A.; Gomez, H.; Juanes, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mechanics Eng.* **2021**, *379*, No. 113741.
(19) De Florio, M.; Schiassi, E.; Calabrò, F.; Furfaro, R. Physics-Informed Neural Networks for 2nd order ODEs with sharp gradients. *J. Comput. Appl. Math.* **2024**, *436*, No. 115396.
(20) Borland, L.; Menchero, J. G.; Tsallis, C. Anomalous diffusion and nonextensive scaling in a one-dimensional quantum model with long-range interactions. *Phys. Rev. B* **2000**, *61* (3), 1650−1653.
(21) Kluth, G.; Ripoll, J.-F.; Has, S.; Fischer, A.; Mougeot, M.; Camporeale, E. Machine Learning Methods Applied to the Global Modeling of Event-Driven Pitch Angle Diffusion Coefficients During High Speed Streams. *Front. Phys.* **2022**, *10*, No. 183, DOI: 10.3389/fphy.2022.786639.
(22) Bréhier, C.-E. The averaging principle for stochastic differential equations driven by a Wiener process revisited. *C. R. Math.* **2022**, *260* (G3), 265−273, DOI: 10.5802/crmath.297.

https://doi.org/10.1021/acsomega.3c07924
ACS Omega 2024, 9, 3846−3857