The Eleventh International Conference on
Engineering Computational Technology
23–25 August, 2022 | Montpellier, France

Proceedings of the Eleventh International Conference on
Engineering Computational Technology
Edited by B.H.V. Topping and P. Iványi
Civil-Comp Conferences, Volume 2, Paper 6.3
Civil-Comp Press, Edinburgh, United Kingdom, 2022, doi: 10.4203/ccc.2.6.3

PINN-ADR: A neural network based simulation tool for reacting flow with multicomponent reactants

Z. Sun[1], H. Du [1], C. Miao[2], and Q. Hou,[2]

[1] College of Intelligence and Computing, Tianjin University, China
[2] School of Civil Engineering, Tianjin University, China

Abstract

The aim of this work is to analyse the use of PINN to solve forward and inverse problems of reacting flow with multicomponent reactants. For the above two problems, PINN can successfully get the correct result. In the forward problem, using the sin function as the activation function fits the discontinuous boundary problem better than using tanh, and the training speed is faster. In the inverse problem, PINN uses data to learn model parameters can not only learn the convection term and the diffusion coefficients, but also learn the parameters related to the kinetics of the reaction and mutation efficiency, and heuristically guide us to discover the physical and chemical laws in the reaction flow.

Keywords: physics-informed neural networks, inverse problem, reactive flow, advection-diffusion-reaction equation, deep learning, autocatalytic reaction.

1 Introduction

The reactive flow model plays an important role in the simulation of many physical and engineering problems, such as the transport of pollutants in water, air, and porous media [1]. Typically, a reactive flow model is an advection-diffusion-reaction (ADR) equation, which is one kind of basic partial differential equations (PDEs) with non-linear source terms with autocatalytic reactions [2]. 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 [3]. By accurately solving the reactive flow model, we not only can analyse the problems related to hydrodynamic catalytic reactions in chemical, physical, and electromagnetic fields but also can design appropriate reaction terms to optimize the process control of the entire catalytic reaction.

A variety of numerical methods can be used to solve the ADR [4,5,6], however, due to the numerical discrete calculation format, none of them is thought to be really space-time continuous. Moreover, traditional methods are usually prohibitively expensive for solving inverse problems (for inferring origins in transport processes or discovering missing physics in reactive transport, for example). Recently, the spatiotemporal continuous solution of the PDEs and the solution of inverse problems can be provided by deep learning (DL) [7, 8, 9]. Based on the governing equation, and initial and boundary conditions, continuous field information is given by the trained DL PDE solver. Using the automatic derivation ability of Neural Network (NN), physics-informed neural networks (PINN) [9] not only solves the forward problem according to the governing equation and initial boundary conditions but also solves the inverse problem according to the observed data. However, as a multiscale problem, the ADR equation includes advection, diffusion, and reaction terms, which makes the equation sensitive to changes in each term. This means that a more accurate approximation of the derivatives in the PINN is needed.

In this paper, the PINN method is applied to reacting flow problems with multicomponent reactants. To avoid the error fits due to discontinuous boundaries, the sin function instead of tanh is used as the activation function. For the inverse problem, it gives the missing physics information from the data.

2 Methods

PINN is constructed by the NN structure and a physics-informed loss function, where the solution of the PDE is approximated by NN. In this paper, we choose fully connected neural networks in the original PINN to solve the ADR equation, and it is straightforward to change the network structure into other forms. The structure of the PINN is shown in Fig. 1.

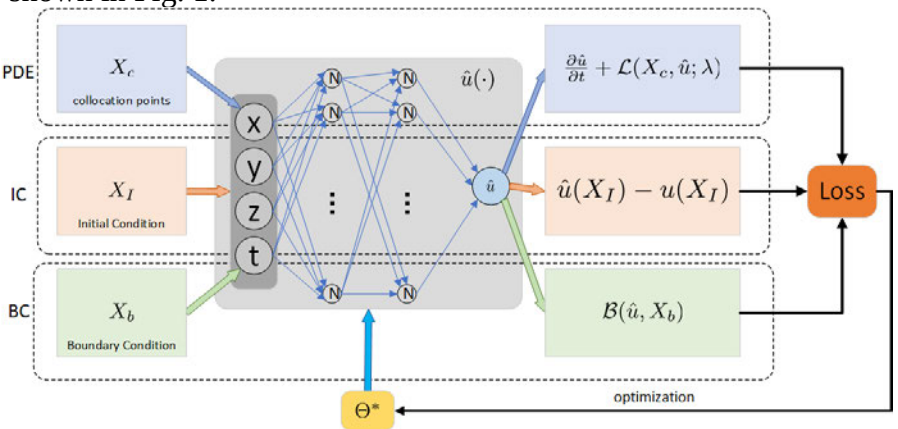

Figure 1: Structure of the PINN for solving a partial differential equation

The PINN is implemented by imposing two types of losses. One is the ℓ_s controlled by the collocation points X_c and the other is the ℓ_{v} calculated on the initial-boundary points X_i. Consequently, the PINN classifies the training points into two categories. One kind is the points in the space-time domain and the other kind is the initial-boundary points. Different from traditional numerical models, value constraint is used for PINN to train NN to fit the initial and boundary conditions, which means that there are errors in the prediction of the initial and boundary conditions. The loss function can then be defined by the L_2 norm as:

$$\ell(\Theta;\, X) = \omega_s \ell_s(\Theta;\, X_c) + \omega_v \ell_v(\Theta;\, X_{ib}) \tag{1}$$

where

$$\ell_s(\Theta; X_c) = \frac{1}{|X_c|} \sum_{X \in X_c} \left\| \frac{\partial u_\Theta}{\partial t} + v \frac{\partial u_\Theta}{\partial x} - k \frac{\partial^2 u_\Theta}{\partial x^2} - f \right\|_2^2 \tag{2}$$

$$\ell_v(\Theta; X_{ib}) = \frac{1}{|X_{ib}|} \sum_{X \in X_{ib}} \| B(u_\Theta, X_{ib}) \|_2^2 \tag{3}$$

in which $|X_c|$ and $|X_{ib}|$ represents the number of points in X_c and X_{ib}, $B(u_\Theta, X_{ib})$ is the boundary condition, and ω_s and ω_v are the weights.

PINN can also be used for inverse problems. With little modifications, the parameters of PDEs and the initial distribution can be learned. For inverse problems, observed data are required and the loss function should also be modified. Unlike solving PDEs, inverse problems no longer require initial-boundary values, but rather information about the equations in the space-time domain. The information here is provided by the distribution of concentrations $\hat{u}$ and the spatiotemporal position (x, t) in the observed data X_v.

The loss function is the same as before, but ℓ_v is no longer calculated on X_{ib}, but on X_v:

$$\ell_s(\Theta; X_c) = \frac{1}{|X_c|} \sum_{X \in X_c} \left\| \frac{\partial u_\Theta}{\partial t} + v_\theta \frac{\partial u_\Theta}{\partial x} - k_\theta \frac{\partial^2 u_\Theta}{\partial x^2} - f \right\|_2^2 \tag{4}$$

$$\ell_v(\Theta; X_v) = \frac{1}{|X_v|} \sum_{X \in X_v} \| \hat{u}(X_v) - u_\Theta(X_v) \|_2^2 \tag{5}$$

where v_θ, k_θ are the model parameters to be learned.

3 Results

The test problem is a widely used autocatalytic reaction model proposed by Alhumaizi [10], which is based on a continuous flow tubular reactor, including three reactants A, B, and C. The specific steps in the autocatalytic reaction process are as follows:

Replication of B: $A + 2\,B \xrightarrow{k_1} 3B$

Death of B: $B \xrightarrow{k_2} P_1$

Mutation of B into C: $A + 2\,B \xrightarrow{\alpha k_1} 2\,C + B$

Replication of C: $A + 2\,C \xrightarrow{\beta k_1} 3C$

Death of C: $C \xrightarrow{k_2/\beta} P_2$

where k_1, k_2, α, and β are a set of constants defining the kinetics of the reaction and mutation efficiency. If we assume dispersion to be unidirectional along the reactor axis, mass balance equations for species A, B, and C can be written in a dimensionless form as follows:

$$\frac{\partial U_1}{\partial T} + v\frac{\partial U_1}{\partial X} = D_1\frac{\partial^2 U_1}{\partial X^2} + (1-U_1)[(1+\alpha)U_2^2 + \beta U_3^2], \quad (6)$$

$$\frac{\partial U_2}{\partial T} + v\frac{\partial U_2}{\partial X} = D_2\frac{\partial^2 U_2}{\partial X^2} + (1-\alpha)(1-U_1)U_2^2 - \gamma U_2, \quad (7)$$

$$\frac{\partial U_3}{\partial T} + v\frac{\partial U_3}{\partial X} = D_3\frac{\partial^2 U_3}{\partial X^2} + (1-U_1)(\beta U_3^2 + 2\alpha U_2^2) - \frac{\gamma}{\beta}U_3, \quad (8)$$

where $U_1 = \frac{u_f - u_1}{u_f}$, $U_2 = \frac{u_2}{u_f}$, $U_3 = \frac{u_3}{u_f}$, $X = \frac{x}{L}$, $T = k_1 u_f^2 t$, $V = \frac{a}{k_1 u_f^2 L}$, $\gamma = \frac{k_2}{k_1 u_f^2}$, u_f is the feed substrate concentration, X dimensionless reactor length, T is the dimensionless time.

Initial condition:

$$U_1(X,0) = 1, \quad U_2(X,0) = 0, \quad U_3(X,0) = 0.$$

Boundary condition:

$$U_1(0,T) = 0, \quad U_2(0,T) = 1, \quad U_3(0,T) = 0,$$
$$U_1(1,T) = 1, \quad U_2(1,T) = 0, \quad U_3(1,T) = 0.$$

For this problem, two questions are considered: whether PINN can simulate dynamic behaviors of the ADR system given initial values and boundary conditions, and for the inverse problem, whether PINN can learn the correct kinetics of the reaction and mutation efficiency. The reference solution is given by the SPH method [5]. Parameter settings: α=0.065, β=2.0, γ=0.025, v=1.0, $D_1 = D_2 = D_3 = 0.05$.

The PINN results are compared with the reference solution is shown in Fig. 2. The PINN with sin as an activation function can give the correct solution. We also tested tanh as an activation function in Fig.3, but the result was wrong. Figure 4 presents the loss reduction in the training process, it is clear that PINN with sin as activation can be trained faster.

Table 1 presents all the model parameters learned by PINN from the reference solution. Table 2 presents the kinetics of the reaction and mutation efficiency learned by PINN from the reference solution. By comparing the two tables, all reaction parameters except β are correctly learned, and even the flow field and three diffusion parameters are learned correctly. Therefore, we consider whether the governing equation is β-insensitive.

Therefore, using PINN to solve the forward problem under different β conditions, the results are shown in Fig. 5. Prove that the problem is β-insensitive.

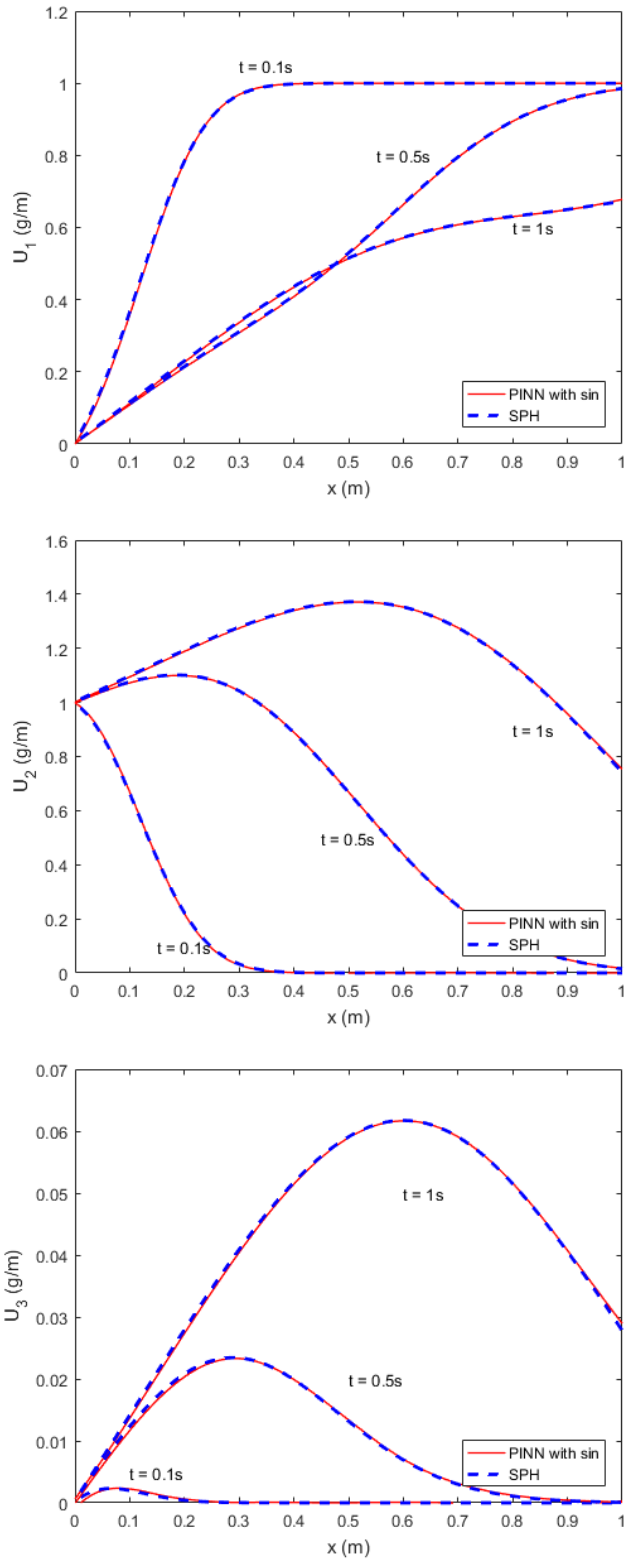

Figure 2: Comparison of the PINN results with reference solutions

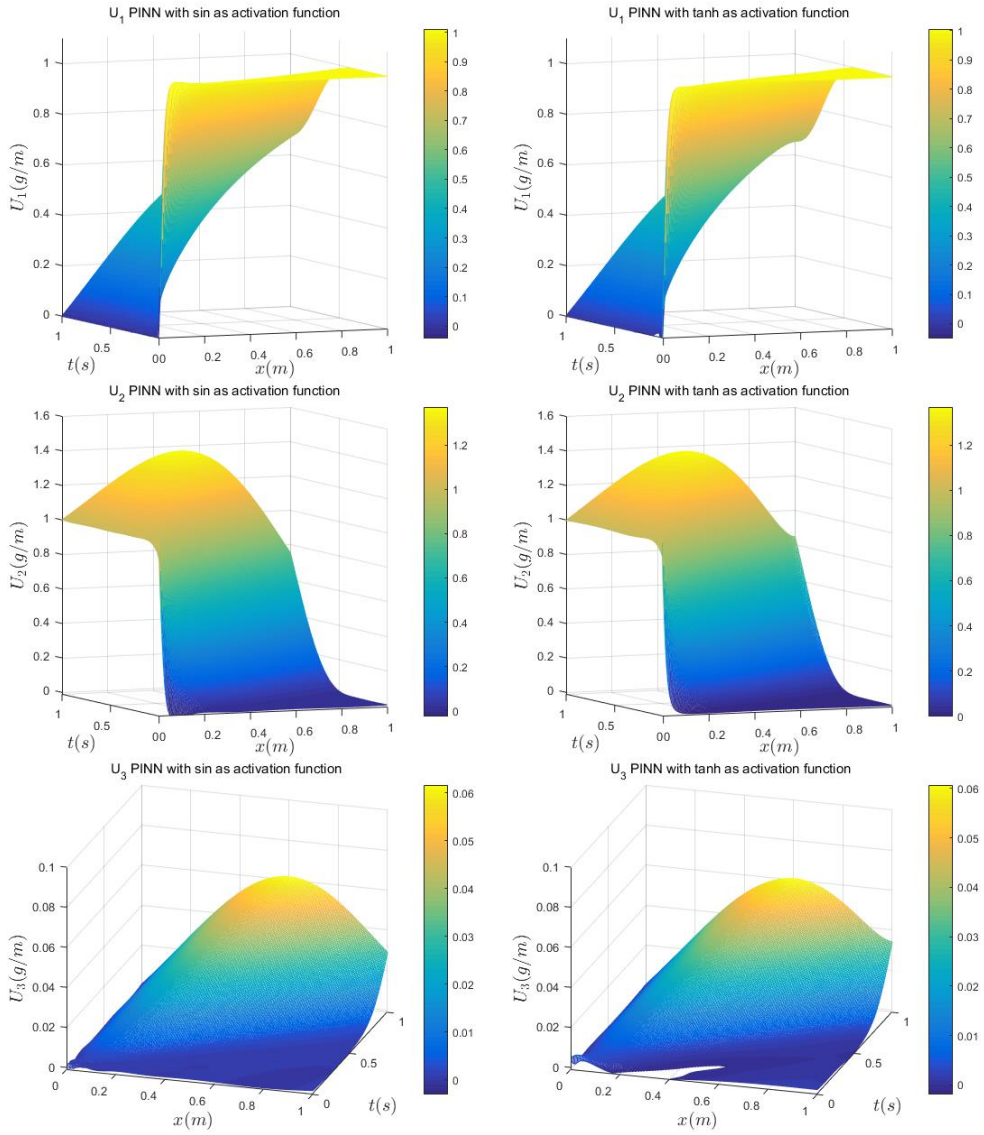

Figure 3: Comparison of the PINN results with sin and tanh as activation function

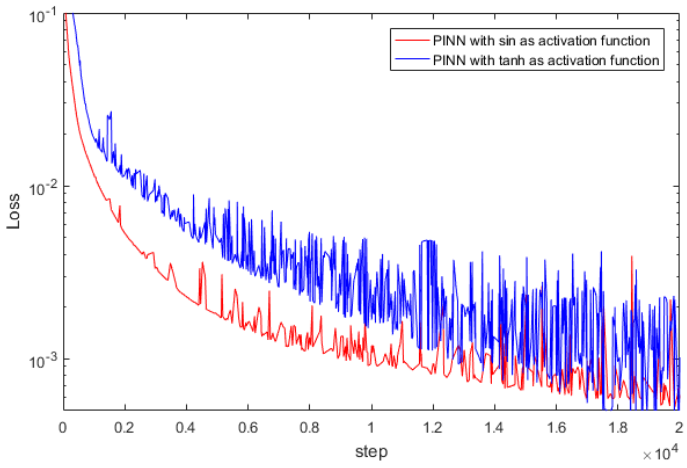

Figure 4: Comparison of the loss reduction with sin and tanh as activation function

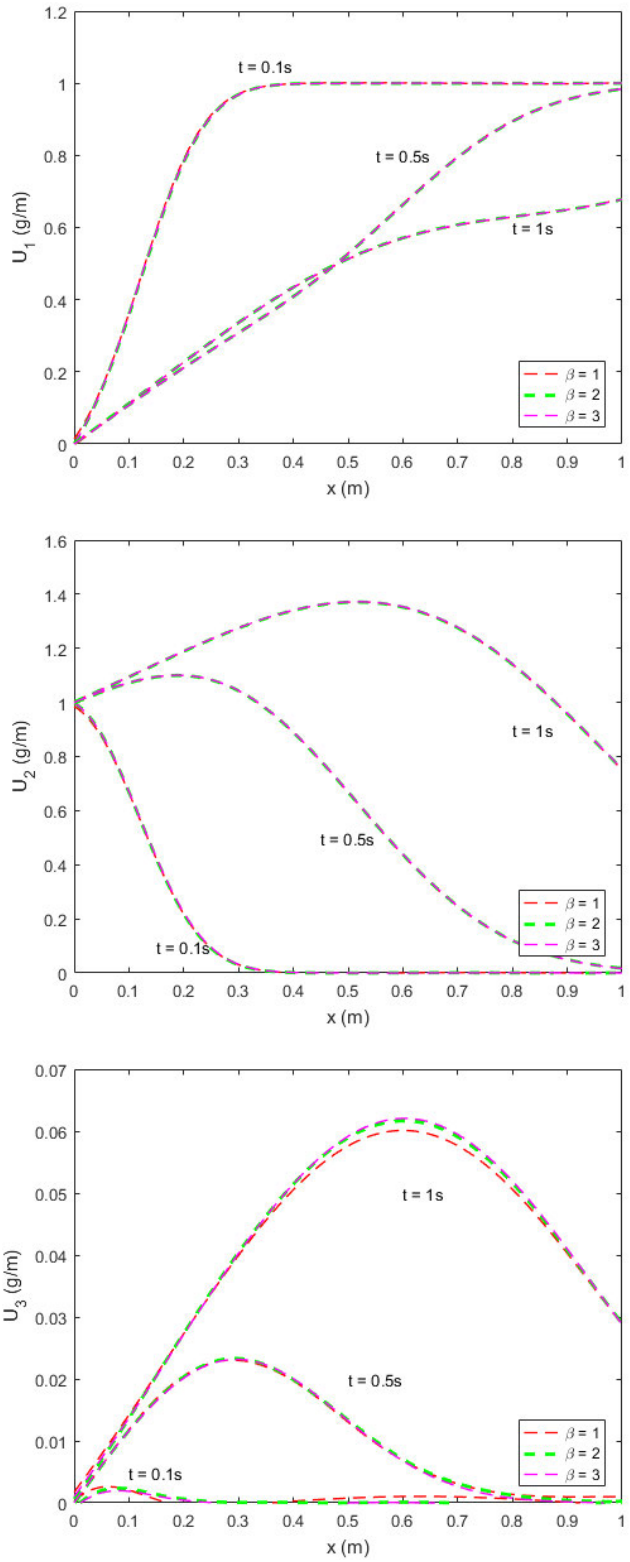

Figure 5: Comparison of the PINN results with different β

	model parameters prediction	reference	error
α	0. 06585819	0.065	1.3203 %
β	0. 8005827	2.0	59.9709%
γ	0. 025006412	0.025	0.0256%
v	0. 9987521	1	0.1248%
D_1	0. 04985645	0.05	0.2871%
D_2	0. 04979899	0.05	0.4020%
D_3	0. 052033793	0.05	4.0676%

Table 1: The error of the model parameter predicted by the PINN

	model parameters prediction	reference	error
α	0. 06547399	0.065	0.7292%
β	0. 8991418	2.0	55.043 %
γ	0. 024921743	0.025	0.313 %

Table 2: The error of the kinetics of the reaction and mutation efficiency predicted by the PINN

4 Conclusions and Contributions

In this paper, the PINN with sin as activation function is presented. The main conclusions are as follows:

For the forward problems, the strategy of using sin instead of tanh was demonstrated as an effective method for PINN to solve the ADR equation is proposed. By using sin as the activation function, the derivatives of the functions fitted by PINN are limited, and the problem of excessive derivatives caused by discontinuous boundaries in the case of using tanh will disappear. And in the process of training, the loss of the method with sin can be reduced faster than the method with tanh. The improvement has greatly improved the stability of the model and reduce the time cost by training, without increasing the need for additional data during the training process.

For the inverse problem, both strategies with three and seven parameters were tested. When testing three parameters related to the kinetics of the reaction and mutation efficiency, both α and γ were well predicted, with the exception of β, which proved to be insensitive to the equation. After adding the velocity parameter v for the convection term and the diffusion coefficients D_1, D_2, D_3 of the three reactants, PINN can still correctly predict these parameters. PINN leads us to discover and verify that this equation is insensitive to changes in β by making predictions about β errors large. PINN demonstrates a strong ability to learn missing physics on inverse problems, and it can help us better observe and explain the laws of physics and chemistry.

References

[1] Y. Zhang, C. Zhao, G. Yang, C. Zhao, K. Shao, "Solving the problem of electromagnetic convection-diffusion using a multiscale combined RBF collocation method", Journal of Huazhong University of Science & Technology, 37, 72-75, 2009.

[2] Y. Wang, K. Hutter, "Comparisons of numerical methods with respect to convectively dominated problems", International Journal for Numerical Methods in Fluids, 37, 721-745, 2001.

[3] K. Alhumaizi, A. E. Abasaeed, "On mutating autocatalytic reactions in a CSTR. I: multiplicity of steady states", Chemical Engineering Science, 55, 3919-3928, 2000.

[4] R. E. Ewing, H. Wang, "A summary of numerical methods for time-dependent advection-dominated partial differential equations", Journal of Computational and Applied Mathematics, 128, 423-445, 2001.

[5] Q. Hou, J. Liu, J.J. Lian, W.H. Lu, "A Lagrangian particle algorithm (SPH) for an autocatalytic reaction model with multicomponent reactants", Processes, 7, 1-18, 2019.

[6] D. L. Stefanovic, H. G. Stefan, "Accurate two-dimensional simulation of advective-diffusive-reactive transport", Journal of Hydraulic Engineering, 127, 728-737, 2001.

[7] L. Lyu, K. Wu, R. Du, J. Chen, "Enforcing exact boundary and initial conditions in the deep mixed residual method", arXiv:2008.01491, 2020.

[8] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, "Physics-informed machine learning", Nature Reviews Physics, 250, 422-440, 2021.

[9] M. Raissi, P. Perdikaris, G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", Journal of Computational Physics, 378, 686-707, 2019.

[10] K. Alhumaizi, "Comparison of finite difference methods for the numerical simulation of reacting flow", Computers & Chemical Engineering, 28, 1759-1769, 2004.