Performance Evaluation of High-Dimensional Spatio-Temporal Evolution Simulation Using Physics-Informed Neural Networks

Kosei WADA[1)], Makoto SASAKI[1)]*, Keisuke YANO[2)]

[1)] *Nihon University, Narashino 275-8575, Japan*
[2)] *The Institute of Statistical Mathematics, Tachikawa 190-8562, Japan*

(Received 1 May 2025 / Accepted 29 July 2025)

In this study, Physics-Informed Neural Networks (PINNs), a deep learning-based framework is applied to a partial differential equation in multi-dimensional space. As a preliminary investigation, the diffusion equation is solved and we examine how computation time varies with spatial dimensionality. The computational time with that of the Finite Difference Method (FDM) with keeping the computation accuracy. The results show that the PINNs can be faster than the FDM in a higher-dimensional space due to the mesh-free characteristics.

Keywords: physics-informed machine learning, PINNs, numerical simulation, plasma simulation, calculation accuracy

DOI: 10.1585/pfr.20.1203047

Behaviors of magnetically confined plasmas are governed by a balance between losses due to turbulence driven by spatial inhomogeneities in temperature and/or density [1] and power injection by high-frequency waves and neutral beam injections [2]. Conventional transport studies have adopted the gyrokinetic approximation, which assumes the turbulence frequency is much lower than the ion cyclotron frequency, enabling the five-dimensional phase space simulations [3]. In contrast, wave heating is required to resolve the cyclotron motion so that the six-dimensional simulation is necessary. A unified treatment of heating and transport has become necessary, because the turbulence could be directly affected by the heating processes [4]. Therefore, six-dimensional simulations are required. However, the mesh-based simulations in six-dimensions are too computationally expensive even with modern computer performance. Therefore, mesh-free approaches are worth considering.

Physics-Informed Neural Networks (PINNs), deep learning-based methods, have recently gained attention. They have been applied in various fields, including fluid dynamics [5], hydrology [6], chemistry [7], materials science [8], and Earth system modeling [9]. High-accuracy results have been reported in two-dimensional fluid problems [5]. However, in such low-dimensional cases, the PINNs have generally offered no advantage over the FDM in speed or accuracy. Since the PINNs do not require spatio-temporal meshes, they may have advantages in high-dimensional problems.

As a first step toward six-dimensional plasma simulations, in this study, the six-dimensional diffusion equation is solved using both the PINNs and the FDM, and their computational performance and accuracy are evaluated. Figure 1 illustrates a schematic of the PINNs, a deep learning-based framework for solving ordinary and Partial Differential Equations (PDE) [10]. In the blue box, spatial coordinates x and time t are introduced into a network composed of linear transformations and sigmoid activations σ, producing the solution u. In the green box, automatic differentiation is used to compute partial derivatives of u, allowing reconstruction of the governing PDE. Training proceeds by minimizing this total loss until it falls below a tolerance ϵ which is explained in the following paragraph. In this way, the PINNs can incorporate physical constraints directly into the learning process, enabling solutions that remain consistent with the underlying physics.

In this study, we numerically solve the diffusion equation in a D-dimensional space using the PINNs and the FDM.

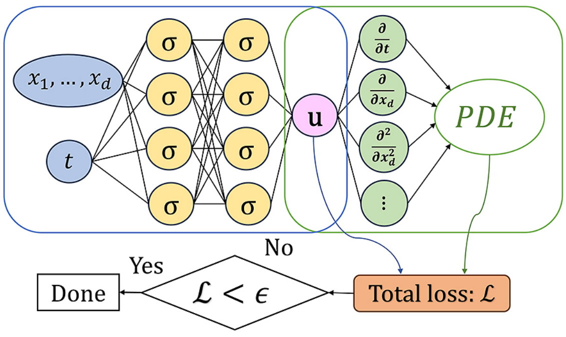

Fig. 1. Network structure of the PINNs.

*Corresponding author's e-mail: sasaki.makoto@nihon-u.ac.jp

The governing equation is given as:

$$\partial_t u = \sum_{d=1}^{D} \partial_{x_d}{}^2 u, \tag{1}$$

where x_d denotes the dth-spatial coordinate. The loss function $\mathcal{L}$ used to incorporate the physical information denoted in Fig. 1 is defined as:

$$\begin{aligned}\mathcal{L} := & \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \left| u_j - u_{IC_j} \right|^2 \\ & + \frac{1}{N_{PDE}} \sum_{j=1}^{N_{PDE}} \left| \partial_t u_j - \sum_{d=1}^{D} \partial_{x_d}{}^2 u_j \right|^2 .\end{aligned} \tag{2}$$

The first term on the right-hand side of Eq. (2) corresponds to the loss associated with the initial condition. Here, N_{IC} represents the number of training points used to enforce the initial condition. The neural network solution u, derived from these inputs, approximates the initial condition. The reference data u_{IC} represents the exact initial profile. The second term on the right-hand side of Eq. (2) corresponds to the PDE loss, where N_{PDE} denotes the number of collocation points used to enforce the governing equation. By applying automatic differentiation to the neural network output u, we obtain the residual of the multi-dimensional diffusion equation. The network is trained to minimize the total loss $\mathcal{L}$. As an initial condition, we assume a delta-function profile. The computational domain is set to $-1 \le x_d \le 1$ for each spatial direction and $1/8 \le t \le 5/8$ for the temporal range. For the PINNs, we initialize training using the analytical solution at $t = 1/8$ as the initial profile. Training is terminated once the total loss satisfies $\mathcal{L} \le 10^{-4}$ and the relative error between the PINNs solution and the analytical solution is less than 4%. Here, the analytical solution used for comparison is given by:

$$u_{exact} = \frac{1}{\left(2\sqrt{\pi t}\right)^D} e^{-\frac{\sum_d x_d{}^2}{4t}}. \tag{3}$$

In the FDM-based simulations, we apply the second-order central difference approximations. Since the numerical error scales as $\Delta x_d{}^2$, we choose a spatial step size $\Delta x_d = 0.2$ to ensure a relative error ϵ of approximately 4%, where:

$$\epsilon = \sqrt{\frac{\int |u - u_{exact}|^2 dx}{\int |u_{exact}|^2 dx}}. \tag{4}$$

The temporal step size is determined to satisfy the stability condition $\Delta t \le \Delta x_d{}^2/(2D)$. All the PINNs models are implemented using the open-source Python library PyTorch and are constructed based on the methodology proposed by M. Raissi *et al.* [11].

We evaluate computational time by varying the spatial dimensionality D from 1 to 6. Figure 2 shows the x_1-axis profile at $x_k = 0$ $(2 \le k \le 6)$ for the six-dimensional diffusion equation solved using the PINNs, along with the FDM results. The PINNs solution achieves accuracy comparable to the FDM. Figure 3 presents the computational time for both methods across dimensions. For the FDM, the computation time increases with dimensionality, following a power-law trend. The PINNs show a slower increase in contrast to the FDM. In low dimensional cases such as $D \le 4$, the PINNs is slower than the FDM. When $D \ge 5$, the PINNs become faster than the FDM, and in the case with $D = 6$, the PINNs is almost ten times faster than the FDM. The computational time is sensitive to the number of collocation points, as PINNs minimize the PDE residual on collocation points without using spatial meshes. In this analysis, 500 collocation points are used for each case. As the number of dimensions increases, the training takes longer to converge, which is the main cause for the increase of the computational time of PINNs. Thus, the PINNs could be a strong candidate for the six-dimensional simulations.

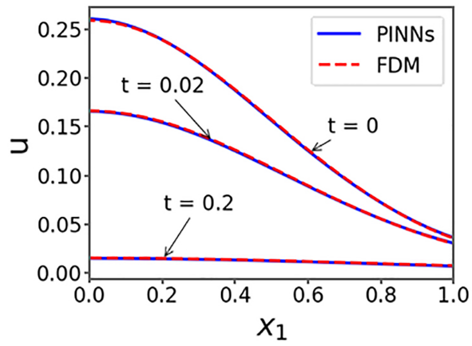

Fig. 2. Simulation of 6-dimensional diffusion equation.

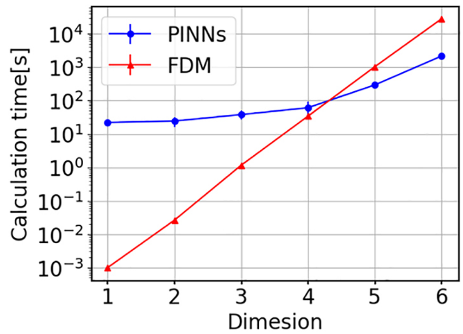

Fig. 3. Dependence of computational time with dimensions.

In conclusion, we solve the diffusion equation in multi-dimensional space using the PINNs and compare the computational time with that of the FDM. By systematically varying the number of dimensions, we demonstrate that the PINNs exhibit a clear advantage in computational efficiency over the FDM in higher-dimensional problems.

This work was partly supported by a Grant-in-Aid for Scientific Research (KAKENHI) (JP21K03509, JP24K06997, JP25K00986), the collaboration programs of NIFS (NIFS24KIPS005, NIFS24KISS041, NIFS25KISS041), that of RIAM Kyushu University (2024S2-CD-2, 2025S2-CD-3), the OML Project by the National Institutes of Natural Sciences (NINS program No. OML012511), and the "Strategic Research Projects" grant 2022-SRP-13 from ROIS (Research Organization of Information and Systems).

[1] P.H. Diamond *et al.*, Plasma Phys. Control. Fusion **47**, R35 (2005).
[2] T.H. Stix, *Waves in Plasmas*, (American Institute of Physics, New York, 1992) 566.
[3] G. Dif-Pradalier *et al.*, Commun. Phys. **5**, 229 (2022).
[4] S.I. Itoh and K. Itoh, Sci. Rep. **2**, 860 (2012).
[5] S. Cai *et al.*, Adv. Math. Sci. **37**, 1727 (2021).
[6] X. Jia *et al.*, ACM/IMS Trans. Data Sci. **2**, 1 (2021).
[7] K. Schütt *et al.*, Adv. Neural Inf. Process. Syst. **30**, 992 (2017).
[8] G.R. Schleder *et al.*, J. Phys.: Mater. **2**, 32001 (2019).
[9] M. Reichstein *et al.*, Nature **566**, 195 (2019).
[10] S.L. Brunton and J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, (Cambridge Univ. Press, 2019).
[11] M. Raissi *et al.*, J. Comput. Phys. **378**, 686 (2019).