Proceedings of the Twelfth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 8, Paper 4.3
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.8.4.3

A Novel Reduced-Dimension Physics-Informed Neural Network: Application for Solving Initial Boundary Value Problems

J. Lee

Department of Architectural Engineering, Sejong University, Seoul, Republic of Korea

Abstract

This paper introduces a novel approach called reduced-dimension physics-informed neural network (rd-PINN) for solving initial boundary value problems (IBVPs). The goal of the proposed rd-PINN is to transform the partial differential equation (PDE) into a system of ordinary differential equations (ODEs). Particularly, the numerical solution is formulated in the form of a linear combination of approximation functions and coefficients, wherein the approximation functions are admissible functions and the coefficients are functions of time to be determined. Accordingly, solving the original IBVP is transferred to the task of finding coefficient functions that satisfy the obtained ODEs. To solve these ODEs, a multi-network structure is designed to parameterize coefficients. Besides, we also proposed a framework that is used to automatically impose initial conditions. The advantages of rd-PINN over the original PINN in terms of solution accuracy and training cost are demonstrated through several numerical examples with different types of PDEs, boundary conditions, and initial conditions.

Keywords: physics-informed neural network, direct method, Galerkin method, initial boundary value problems, machine learning, impose initial and boundary conditions.

1 Introduction

Recent years, physics-informed neural network (PINN) [1], a typical application of deep neural networks (DNNs), which has become popular for solving partial differential equations (PDEs) due to its simplicity and effectiveness. PINNs use automatic differentiation (AD) [2] to get over the mesh issue, which is a limitation of traditional numerical methods [3, 4]. The goal of PINN is to minimize the loss function, which is determined by using PDEs and related I/BCs to describe physical difficulties. After years of research, PINN and its variants are being applied more and more to solve forward and inverse PDEs in a variety of domains, including biomechanics [5], fluid mechanics [6], and solid mechanics [7, 8]. However, there are still some limitations in applying PINN-based models to solve PDEs, especially for IBVPs. It can be mentioned, such as PINN and its variants requires expensive cost to achive good enough solutions. Another significant concern when solving IBVPs is the precise enforcement of initial and boundary conditions (I/BCs). In static analysis, several frameworks of exact impose BCs were developed and widely used due to their merit in improving solution accuracy [9–11]. In contrast, studies on simultaneously imposing I/BCs in dynamic problems are still open.

Along this path, a novel reduced-dimension physics-informed neural network (rd-PINN) is developed to overcome above-mentioned challenges. The proposed rd-PINN operates based on direct method [12], where the PDE is transferred to a system of ordinary differential equations (ODEs). Particularly, we aim to seek a numerical solution expressed in form of a linear combination of approximation functions and coefficients. Here, the approximation functions are admissible functions and coefficients are functions of time to be determined. Subsequently, the weak form of the PDE is substituted with this direct solution, and integration over the spatial domain produces a system of ODEs. We use a multi-network structure to parameterize the unknown coefficient functions in order to solve these ODEs, with each sub-network responsible for predicting a particular field variable. The BCs are automatically satisfied via selection criteria of approximation functions. We also propose a scheme to handle ICs based on Galerkin formulation.

The rest of this paper is organized as follows. In Section 2, the establishment of rd-PINN is presented, the selection criteria of approximation functions and the technique of imposition of ICs are also mentioned. Section 3 shows the superior performance of rd-PINN over the original PINN in terms of solution accuracy and training time through several benchmark problems. Finally, we summarize this work in Section 4.

2 Reduced-dimension physics-informed neural network

2.1 Methodology of reduced-dimension physics-informed neural network

Let consider the initial boundary value problems (IBVPs), which is generally presented as

$$\begin{aligned} &\mathcal{N}\left[u\left(\boldsymbol{x},\boldsymbol{t}\right)\right]=f\left(\boldsymbol{x},\boldsymbol{t}\right), && \left(\boldsymbol{x},\boldsymbol{t}\right)\in\mathcal{D}, && \text{(1a)}\\ &\mathcal{B}\left[u\left(\boldsymbol{x},\boldsymbol{t}\right)\right]=u_{\mathcal{B}}, && \left(\boldsymbol{x},\boldsymbol{t}\right)\in\partial\Omega\times\left(0,T\right], && \text{(1b)}\\ &\mathcal{I}\left[u\left(\boldsymbol{x},0\right)\right]=u_{0}, && \boldsymbol{x}\in\Omega, && \text{(1c)}\end{aligned}$$

where $\mathcal{N}$ and f are respective differential operator and source function. $\mathcal{B}$ and I represent boundary and initial operators, where $u_{\mathcal{B}}$ and u_0 denote prescribed values on the boundary $\partial\Omega$ and at the initial instant $t=0$, respectively.

To start with the development of the rd-PINN model, we first introduce the weighted-integral statement of Eq. (1a) as

$$\int_{\Omega} w\left(\boldsymbol{x}\right)\left[\mathcal{N}\left(u\right)-f\right]d\boldsymbol{x}=0, \tag{2}$$

where w is called weight function and satisfy the essential BCs in homogeneous form. As above-mentioned, we assume that the latent solution is simulated in the form of a linear combination of approximation functions ϕ_i and coefficient functions c_i, its explicit form is given by

$$u_{rd-PINN}\left(\boldsymbol{x},\boldsymbol{t}\right)=\phi_0\left(\boldsymbol{x},\boldsymbol{t}\right)+\sum_{i=1}^{N}c_i\left(\boldsymbol{t}\right)\phi_i\left(\boldsymbol{x}\right), \tag{3}$$

The set ϕ_i are admissible functions which must satisfy several continuity and boundary conditions. Meanwhile, c_i are unknown functions of time and satisfy the system of ODEs gained by the following way.

On substituting the approximated solution (3) in Eq. (1a) and using the weight function Galerkin-based chosen, it is to identify $\phi_i, i=1,\cdots,N$ respectively. The system of one-dimensional ODEs is given by

$$\int_{\Omega}\phi_i\left(\boldsymbol{x}\right)\left[\mathcal{N}\left(u_{rd-PINN}\right)-f\right]d\boldsymbol{x}=0, \tag{4}$$

Let us assume the predicted coefficient by NN denoted as $\bar{c}_i=\mathcal{NN}\left(t,\Phi_i\right)$, the optimal results is obtained by minimize the loss function defined as follows

$$\mathcal{L}_{rd-PINN}=\sum_{i=1}^{N}\parallel R_i\left(\bar{c}_1,\cdots,\bar{c}_N\right)\parallel_{2,(0,T]}^2, \tag{5}$$

where the L^2-norm $\|\cdot\|_2$ in (5) is calculated by using Gaussian quadrature [13] and

$$R_i\left(\bar{c}_1,\cdots,\bar{c}_N\right)=\int_\Omega \phi_i\left(\boldsymbol{x}\right)\left[\mathcal{N}\left(u_{rd-PINN}\right)-f\right]d\boldsymbol{x} \tag{6}$$

It is noted that the initial and boundary conditions are automatically handled via several schemes discussed in the following sections, leading to the loss function (5) including only quantities of ODEs.

2.2 Selection criteria of approximation functions

As above-mentioned, the set (ϕ_0, ϕ_i) must satisfy several conditions following listed as:

1. ϕ_i have to be differentiability in the same order with differential operator; this requirement is put forth to avoid a trivial value $\mathcal{N}\left(u_{rd-PINN}\right)=0$.
2. The set $\{\phi_i\}$ is linear independent and complete. Then, the uniqueness of the solution to the system (4) is ensured.
3. ϕ_i requires satisfying all BCs in homogeneous forms; in other words, $\mathcal{B}\left(\phi_i\right)=0$.
4. ϕ_0 satisfies all specified BCs associated with the equations; in other words, $\mathcal{B}\left(\phi_0\right)=u_{\mathcal{B}}$.

Motivated by the growth of the Fourier series [14], we decided to employ sine, cosine, and their variants as the approximation functions in this study.

2.3 Imposition of initial conditions via Galerkin formulation

In this section, we present a framework for imposing ICs, wherein the ICs of the latent solution are replaced by the ICs of coefficients via Galerkin formulation as follows

$$\int_\Omega \phi_i\{\mathcal{I}\left(u_{rd-PINN}\right)\left(\boldsymbol{x},0\right)-u_0\}d\boldsymbol{x}=0, \qquad i=1,\cdots,N, \tag{7}$$

which gives

$$[A]\left\{c\right\}=\{b\}, \tag{8}$$

where

$$A_{ij}=\int_\Omega \phi_i\phi_j d\boldsymbol{x}, \tag{9a}$$

$$b_i=\int_\Omega \phi_i\left[-\mathcal{I}\left(\phi_0\right)\left(\boldsymbol{x},0\right)+u_0\right]d\boldsymbol{x}, \tag{9b}$$

and $c = [\mathcal{I}(c_1)(0), \cdots, \mathcal{I}(c_N)(0)]^T$ is a vector of initial values of the coefficient functions c_i, and obtained by multiplying the inverse matrix $[A]^{-1}$ into both sides of Eq. (8) as

$$\{c\} = [A]^{-1}\{b\}. \tag{10}$$

Consequently, the structure of the constrained outputs are modified as:

$$\widehat{c_i} = \Psi(\boldsymbol{t})\,\bar{c}_i + \mathcal{I}(c_i)(0), \tag{11}$$

where $\Psi(\boldsymbol{t})$ is a trial function satisfy ICs in homogeneous form. There are several previous studies in determining the trial function, such as using R-functions [10, 15], or DNNs [16].

3 Numerical examples

We use the fully-connected DNNs to parameterize predictions. The structure of each NN includes two hidden layers, each with twenty neurons; and sine function is employed as an active function. The training process is implemented in TensorFlow [17], in which sine activa- tion function was used and a hybrid optimization method between Adam [18] and L-BFGS [19] optimizers as suggested in [20].

3.1 Heat transfer problem

We first consider the transient heat transfer problem described by

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x, t < 1 \tag{12a}$$

$$u(0,t) = \frac{\partial u}{\partial x}(1,t) = 0, \quad \text{for } t > 0, \tag{12b}$$

$$u(x,0) = 1, \quad \text{for } 0 < x < 1. \tag{12c}$$

The approximation functions are chosen as

$$\phi_0 = 0, \tag{13a}$$

$$\phi_i = \sin\left[(2i-1)\frac{\pi x}{2}\right], \qquad \text{for } i > 0. \tag{13b}$$

The predicted coefficients after 10000 epochs are illustrated in Figure. 1. The exact solution of this problem is given in Figure 2, while the numerical solutions and corresponding point-wise errors of rd-PINN and PINN are showcased in Figure 3. Besides, both models respective require 0.62 and 62.43 minutes for training. It can be observed that rd-PINN provide a more accurate solution while spend less training cost over the other model.

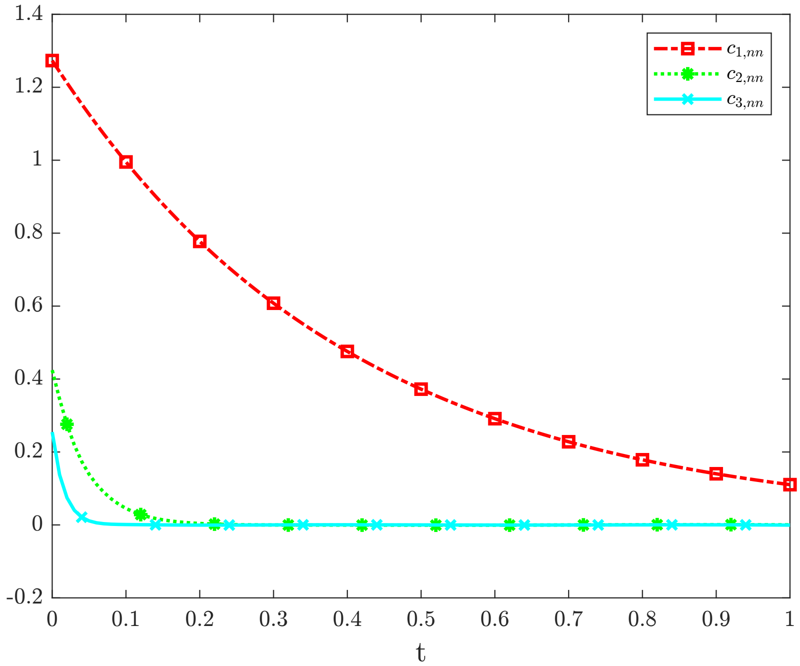

Figure 1: Predicted coefficient functions for heat transfer problem

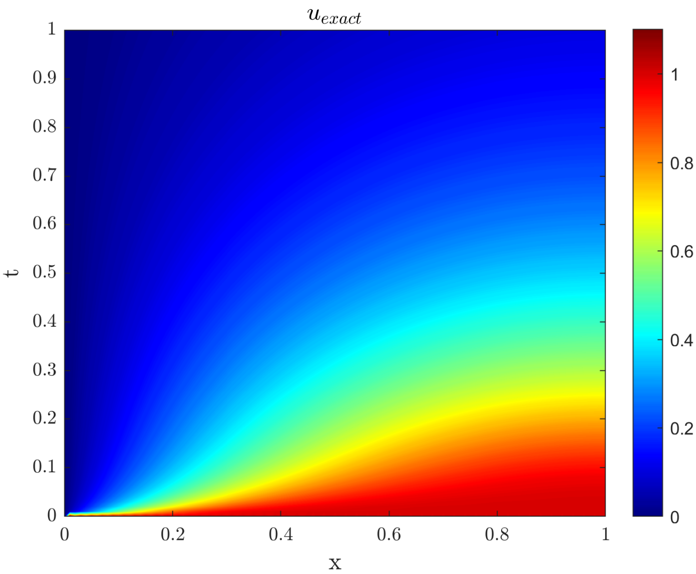

Figure 2: Exact solution for heat transfer problem

3.2 Advection-diffusion equation

Let us consider the advection-diffusion equation, where the governing equation is given by

$$\frac{\partial u}{\partial t} + \mu_1 \frac{\partial u}{\partial x} + \mu_2 \frac{\partial u}{\partial y} - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f, \quad (x, y, t) \in \Omega \times (0, 1] \tag{14}$$

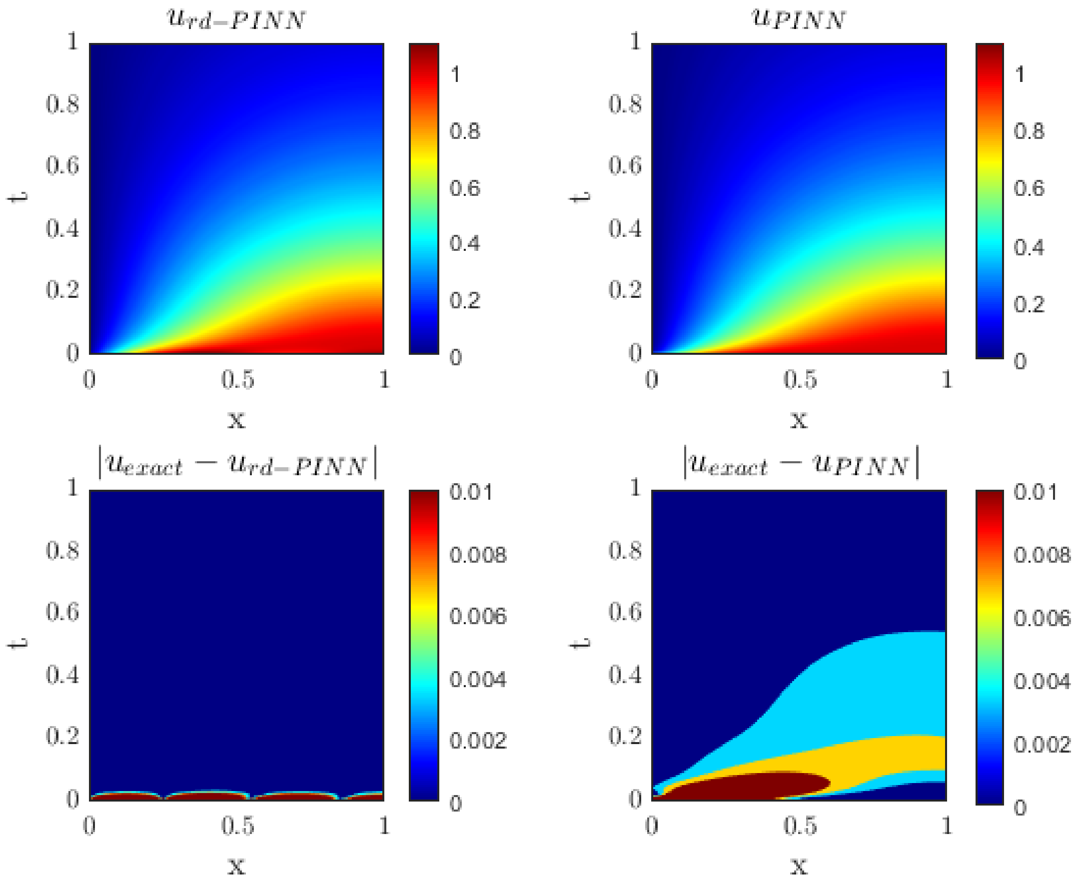

Figure 3: Approximated solutions and corresponding errors of rd-PINN and PINN for heat transfer problem

where $\mu_1 = \mu_2 = 4$ are diffusion coefficients and $\Omega = [0,4] \times [0,4]$ is a square domain. The boundary and initial conditions are expressed as

$$u(x,y,t) = 0, \qquad (x,y,t) \in \partial\Omega \times (0,1] \tag{15a}$$
$$u(x,y,0) = xy(4-x)(4-y), \qquad 0 \le x, y \le 4 \tag{15b}$$

The source term f in (14) is specified by the analytical solution , it is: $u(x,y,t) = e^{-0.25t} xy(4-x)(4-y)$. Here, the direct solution is rewritten in form:

$$u(x,y,t) = \phi_0(x,y) + \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij}(t)\, \phi_{ij}(x,y), \tag{16}$$

where the approximation functions ϕ_{ij} are chosen as:

$$\phi_0(x,y) = xy(4-x)(4-y), \tag{17a}$$
$$\phi_{ij}(x,y) = \sin\left(i\frac{\pi x}{4}\right) \sin\left(j\frac{\pi y}{4}\right), \qquad \text{for } i, j > 0 \tag{17b}$$

Figure 4 showcases the predictions of coefficients by rd-PINN. The simulations and the absolute errors of rd-PINN and original PINN are illustrated in Figures. 5 and 6, respectively. On the observation from Figure. 5, the spectrum of both rd-PINN and PINN solutions agree with the exact ones at all considered times. Nonetheless, the simulation by rd-PINN is significant better than the other when considering the point-wise absolute errors, as shown in Fig. 6. For the training time, rd-PINN and PINN respective took 0.80 and 192.69 minutes to train DNN.

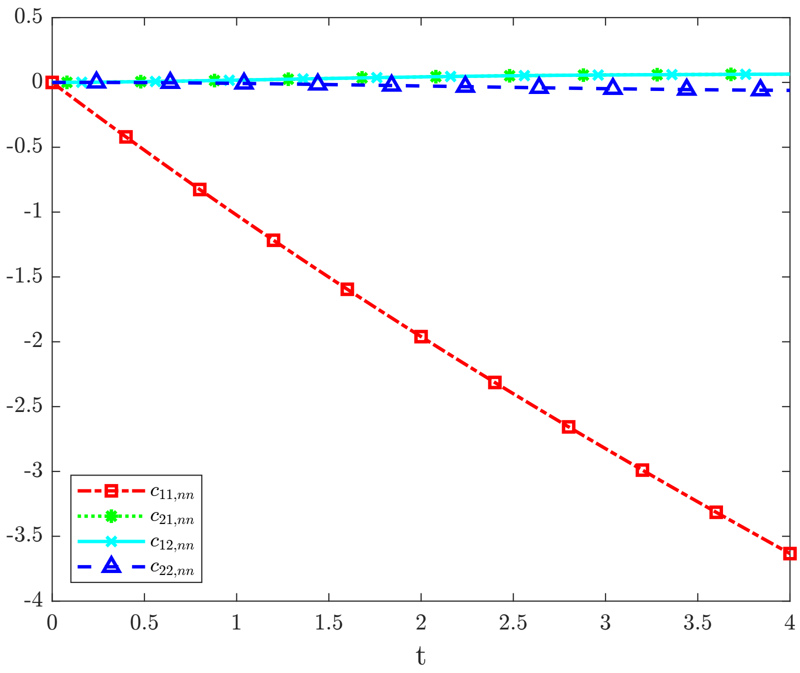

Figure 4: Predicted coefficient functions for advection-diffusion equation

3.3 Transient beam bending

For the final example, we consider the transient beam bending under initial displacement. The equilibrium equation is provided as follows:

$$\frac{\partial^2 w}{\partial t^2} + \frac{\partial^4 w}{\partial x^4} = 0, \quad (x,t) \in (0,L) \times (0,T) \tag{18}$$

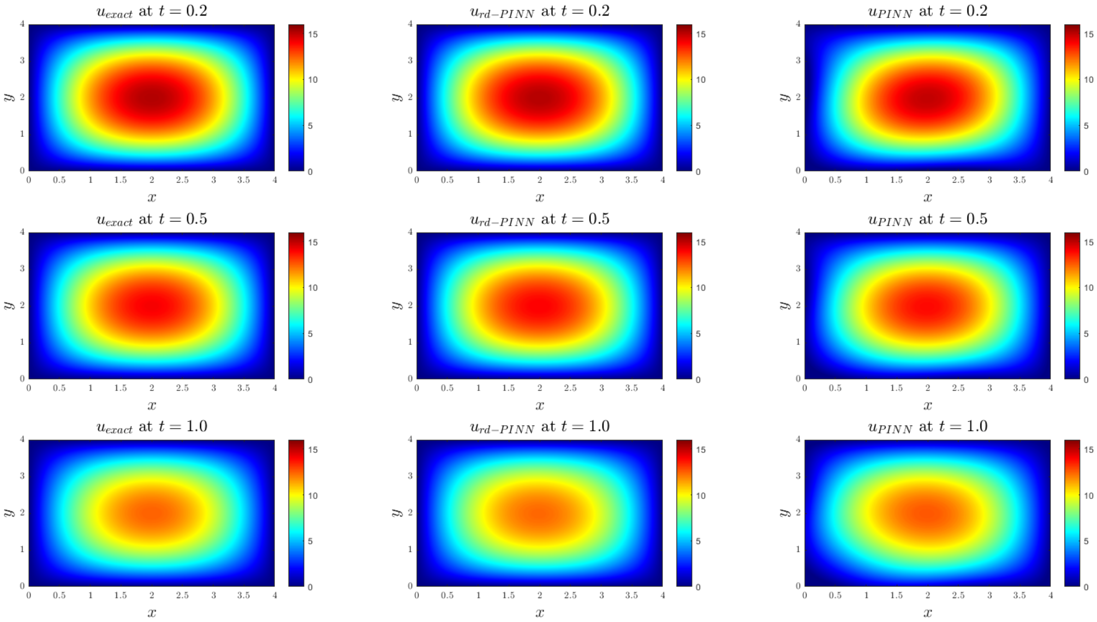

Figure 5: Exact and approximated solutions at several snapshots of rd-PINN and PINN for advection-diffusion equation

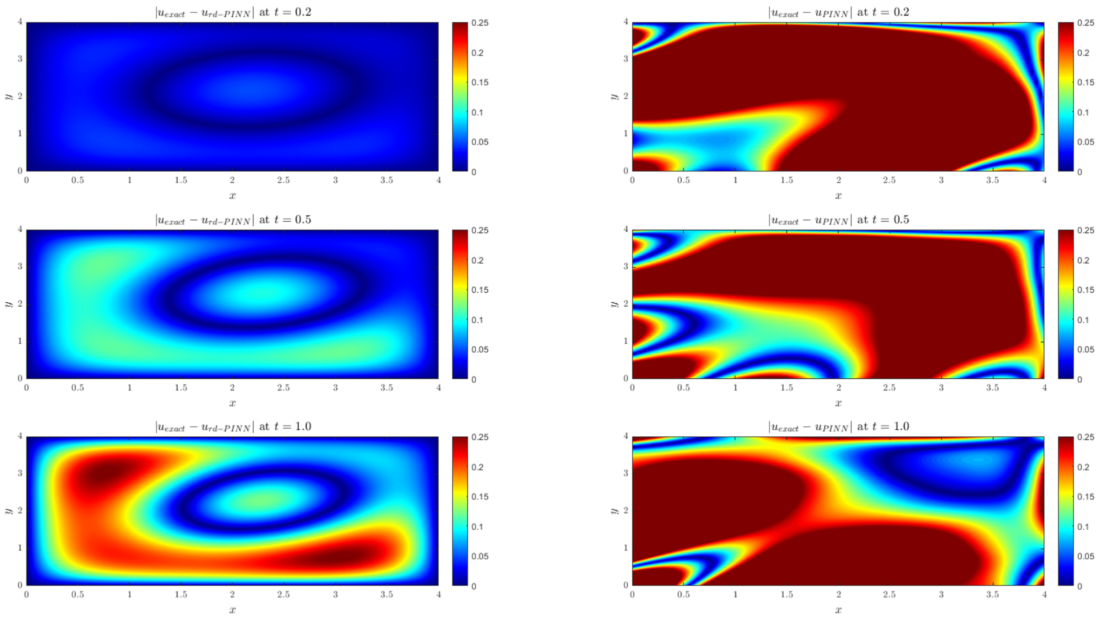

Figure 6: Absolute errors at several snapshots of rd-PINN and PINN for advection-diffusion equation

where L is the length of the beam. The beam is fixed at both ends and subjected to an initial deflection, so the I/BCs are described as

$$w(0,t) = w(L,t) = 0, \qquad 0 \leq t \leq T, \tag{19a}$$

$$\frac{\partial w}{\partial x}(0,t) = \frac{\partial w}{\partial x}(L,t) = 0, \qquad 0 \leq t \leq T, \tag{19b}$$

$$w(x,0) = \sin(\pi x) - \pi x (L - x), \qquad 0 \leq x \leq L, \tag{19c}$$

$$\frac{\partial w}{\partial t}(x,0) = 0, \qquad 0 \leq x \leq L. \tag{19d}$$

Let $L = 1$ m and $T = 1$, the latent solution w is transformed to one-parameter direct solution ($N = 1$) where the approximation functions are given by

$$\phi_0 = \sin(\pi x) - \pi x (L - x), \tag{20a}$$
$$\phi_1 = 1 - \cos(2\pi x). \tag{20b}$$

The prediction of coefficient function gained by rd-PINN after 10000 epochs is illustrated in Figure. 7. In this problem, the reference solution is generated by finite element method [21], as shown in Figure. 8. Besides, the approximated solutions and the corresponding point-wise absolute errors provided by rd-PINN and PINN are presented in Figure. 9. This figure indicates that the simulation of rd-PINN reached an agree-well in accuracy compared to the reference one. On the contrary, the PINN model failed to produce a correct solution. The causes include the PINN framework's poor performance when applied to high-order PDEs and a wide range in the convergence rates of different terms that add to the overall training error. Again, rd-PINN spent less training time than PINN. Particularly, our model required 0.7 minutes to train DNN while the other one took 190.92 minutes.

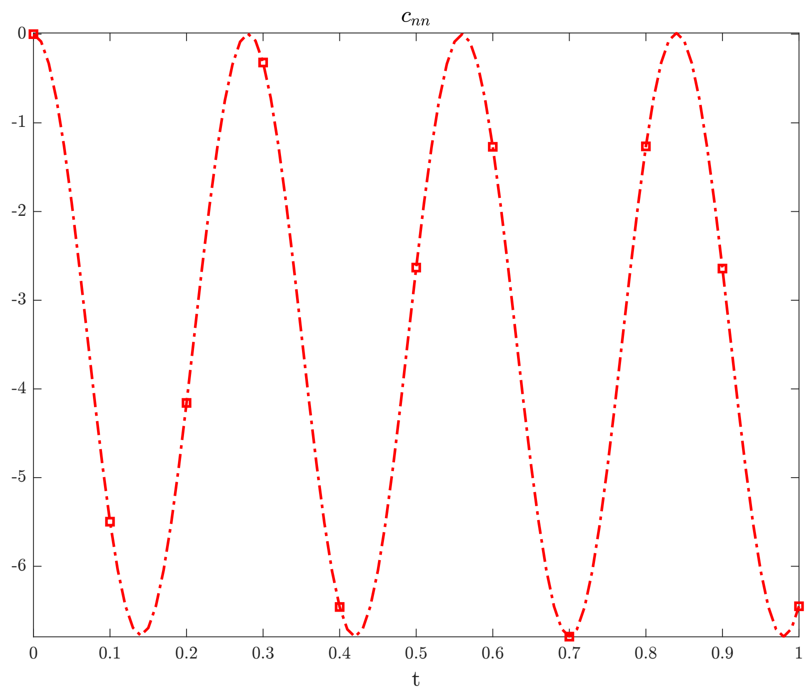

Figure 7: Predicted coefficient functions for beam bending problem

4 Concluding remarks

In this work, a novel approach called rd-PINN is developed for solving IVPs. The rd-PINN operates based on the core idea of transferring the PDE to a system of ODEs.

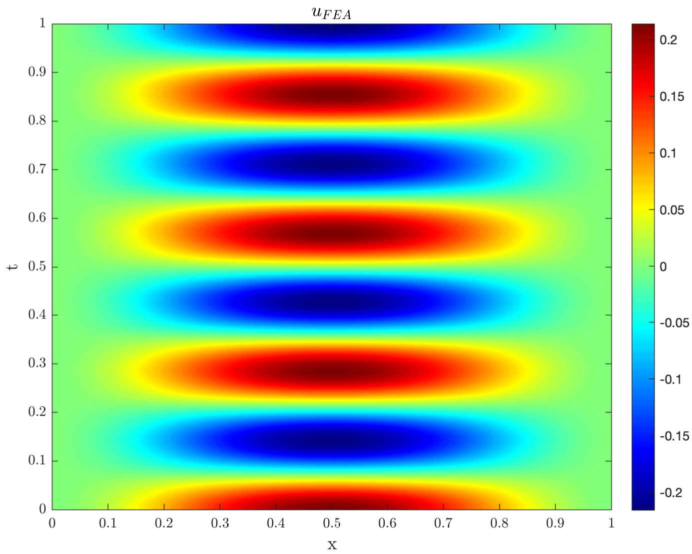

Figure 8: Exact solution for beam bending problem

In detail, the numerical solution is derived as a linear combination of approximation functions and coefficients, wherein the approximation functions are predetermined and satisfy several specified BCs, while the coefficients are functions of time to be determined. The DNN framework is integrated to solve the gained ODEs, wherein each coefficient function is parameterized by each sub-NN. Furthermore, the I/BCs are automatically enforced via the separated schemes, respectively. Particularly, the BCs are satisfied by several criteria for selecting approximation functions while the ICs are immediately imposed based on the Galerkin formulation.

Through various examples, it can be observed that the rd-PINN shows its merit in improving solution quality and also significantly reduces training time compared to the original PINN. Our method promises to be extended and applied to other problems such as inverse and seismic modeling.

Acknowledgements

This research was supported by Grant (RS-2024-00337001) from NRF (National Research Foundation of Korea) funded by MSIT (Ministry of Science and ICT) of Korean government.

References

[1] Raissi, M., Perdikaris, P. & Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving

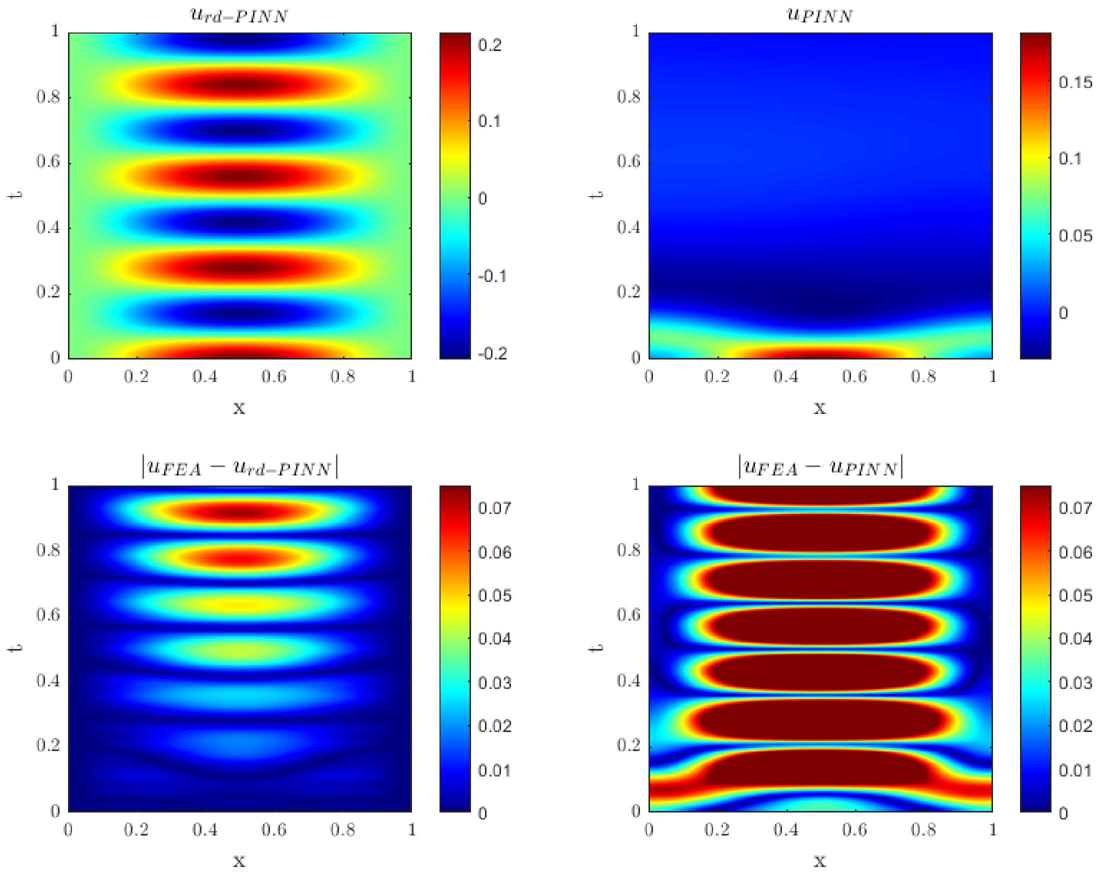

Figure 9: Approximated solutions and corresponding errors of rd-PINN and PINN for beam bending problem

nonlinear partial differential equations. *Journal Of Computational Physics*. **378** pp. 686-707 (2019).

[2] Baydin, A., Pearlmutter, B., Radul, A. & Siskind, J. Automatic differentiation in machine learning: a survey. *Journal Of Machine Learning Research*. **18**, 1-43 (2018).

[3] Koric, S., Hibbeler, L. & Thomas, B. Explicit coupled thermo-mechanical finite element model of steel solidification. *International Journal For Numerical Methods In Engineering*. **78**, 1-31 (2009).

[4] Meng, X., Li, Z., Zhang, D. & Karniadakis, G. PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Computer Methods In Applied Mechanics And Engineering*. **370** pp. 113250 (2020).

[5] Yin, M., Zheng, X., Humphrey, J. & Karniadakis, G. Non-invasive inference of thrombus material properties with physics-informed neural networks. *Computer Methods In Applied Mechanics And Engineering*. **375** pp. 113603 (2021).

[6] Cai, S., Mao, Z., Wang, Z., Yin, M. & Karniadakis, G. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*. **37**,

1727-1738 (2021).
[7] Haghighat, E., Raissi, M., Moure, A., Gomez, H. & Juanes, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods In Applied Mechanics And Engineering*. **379** pp. 113741 (2021)
[8] Le-Duc, T., Lee, S., Nguyen-Xuan, H. & Lee, J. A hierarchically normalized physics-informed neural network for solving differential equations: Application for solid mechanics problems. *Engineering Applications Of Artificial Intelligence*. **133** pp. 108400 (2024).
[9] Berg, J. & Nyström, K. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*. **317** pp. 28-41 (2018).
[10] Sukumar, N. & Srivastava, A. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods In Applied Mechanics And Engineering*. **389** pp. 114333 (2022).
[11] Luong, K., Le-Duc, T., Lee, S. & Lee, J. A novel normalized reduced-order physics-informed neural network for solving inverse problems. *Engineering With Computers*. pp. 1-20 (2024).
[12] Simpson, A. The Dynamical Behaviour of Structures. GB Warburton. Pergamon, Oxford. 1976. 354 pp. Illustrated.£6.25 (£12.50 hardback).. *The Aeronautical Journal*. **81**, 132-132 (1977).
[13] Golub, G. & Welsch, J. Calculation of Gauss quadrature rules. *Mathematics Of Computation*. **23**, 221-230 (1969).
[14] Razzaghi, M. & Razzaghi, M. Fourier series direct method for variational problems. *International Journal Of Control*. **48**, 887-895 (1988).
[15] Huang, Z. & Peng, L. An improved plate deep energy method for the bending, buckling and free vibration problems of irregular Kirchhoff plates. *Engineering Structures*. **301** pp. 117235 (2024).
[16] Luong, K., Le-Duc, T. & Lee, J. Automatically imposing boundary conditions for boundary value problems by unified physics-informed neural network. *Engineering With Computers*. pp. 1-23 (2023).
[17] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. & Others TensorFlow: a system for Large-Scale machine learning. *12th USENIX Symposium On Operating Systems Design And Implementation (OSDI 16)*. pp. 265-283 (2016).
[18] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*. (2014).
[19] Liu, D. & Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*. **45**, 503-528 (1989).
[20] Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V., Guo, H., Hamdia, K., Zhuang, X. & Rabczuk, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods In Applied Mechanics And Engineering*. **362** pp. 112790 (2020).

[21] Reddy, J. An introduction to the finite element method. *New York*. **27** pp. 14 (1993).