International Journal of Heat and Fluid Flow 96 (2022) 109002

Contents lists available at ScienceDirect

International Journal of Heat and Fluid Flow

journal homepage: www.elsevier.com/locate/ijhff

Predicting high-fidelity multiphysics data from low-fidelity fluid flow and transport solvers using physics-informed neural networks

Maryam Aliakbari, Mostafa Mahmoudi, Peter Vadasz, Amirhossein Arzani *

Department of Mechanical Engineering, Northern Arizona University, Flagstaff, AZ, USA

ARTICLE INFO

Keywords:
Deep learning
Scientific machine learning
Partial differential equations
CFD
Porous media
Multi-fidelity modeling

ABSTRACT

High-fidelity models of multiphysics fluid flow processes are often computationally expensive. On the other hand, less accurate low-fidelity models could be efficiently executed to provide an approximation to the solution. Multi-fidelity approaches combine high-fidelity and low-fidelity data and/or models to obtain a desirable balance between computational efficiency and accuracy. In this manuscript, we propose a multi-fidelity approach where we combine data generated by a low-fidelity computational fluid dynamics (CFD) solution strategy (solver settings and resolution) and data-free physics-informed neural networks (PINN) to obtain improved accuracy. Specifically, transfer learning based on low-fidelity CFD data is used to initialize PINN. Subsequently, PINN with this physics-guided initialization is used to obtain the final results without any high-fidelity training data. The accuracy of the final results relies on the governing equations encoded in PINN together with the low-fidelity CFD data initialization. To investigate the accuracy of this approach, several partial differential equations were solved to predict velocity and temperature in different fluid flow, heat transfer, and porous media transport problems. Comparison with reference high-fidelity CFD data revealed that the proposed approach not only significantly improves the accuracy of low-fidelity CFD data but also improves the convergence speed and accuracy of PINN.

1. Introduction

In various engineering and scientific modeling applications, fundamental governing equations are either not known, or if known they are in an approximate form. Additionally, in many physical settings, a high-fidelity numerical solution to problems where the exact governing equations are known tends to be either challenging or computationally demanding. As a result, often numerical simulations of physical systems only provide an approximation to the true solution. In this manuscript, we focus on a common situation where one needs to provide a computationally fast approximate solution to a problem with well-defined governing equations. This could be either a multi-query problem where the solver needs to be called multiple times (e.g., optimization and uncertainty quantification) or a compromised solution strategy where high-performance computing facilities and high-fidelity solvers are not available. In these scenarios, low-resolution computational meshes, lower order solver settings, and relaxed tolerances could be employed to enable approximate solutions with low computational cost. However, these low-fidelity solver settings are known to reduce accuracy and miss important physics (Khan et al., 2015).

Machine learning–enhanced modeling and simulation is one of the forefronts of scientific machine learning research (Baker et al., 2019). Physics-informed neural networks (PINN) (Raissi et al., 2019; Karniadakis et al., 2021) is a recent paradigm in this area where governing differential equations are encoded to provide a hybrid physics-based and data-driven deep learning framework for solving forward and inverse problems. Another common approach is to use traditional physics-based models to pre-train neural networks and therefore reduce the need for large datasets (Willard et al., 2020). Overall, in challenging data-driven modeling problems, low-fidelity data or models interact with their more expensive and less accessible high-fidelity counterparts (Peherstorfer et al., 2018).

Applying PINN in a multi-fidelity data/model setting with the potential use of transfer learning offers an attractive solution to many of the aforementioned challenges. Towards this goal, various multi-fidelity PINN approaches have been proposed. Multiple neural networks were used to learn the correlation between input low fidelity and high-fidelity data (Meng and Karniadakis, 2020). A similar PINN framework was used

* Corresponding author.
E-mail address: amir.arzani@nau.edu (A. Arzani).

https://doi.org/10.1016/j.ijheatfluidflow.2022.109002
Received 3 March 2022; Received in revised form 12 May 2022; Accepted 19 May 2022
Available online 25 May 2022

in material modeling (Liu and Wang, 2019). Another interesting approach was proposed where PINN was initialized with approximated and simpler governing equations and subsequently transfer learning was used with few high-fidelity training data (Chakraborty, 2021). Finally, in a very different approach (Penwarden et al., 2021), the neural network architecture in PINN was proposed as parameters related to model fidelity.

Most of the above approaches require high-fidelity training data. Additionally, low-fidelity CFD solver settings have not been considered as the low-fidelity model. We propose to use PINN along with the data generated by low-fidelity CFD solvers to enhance CFD data accuracy and simultaneously improve PINN's accuracy and training cost. In other words, we will show that PINN can increase the fidelity of low-fidelity CFD data and at the same time the low-fidelity CFD data will improve PINN's accuracy and convergence compared to the traditional PINN approach. Similar to prior data-free PINN approaches for solving differential equations (Jin et al., 2021), our multi-fidelity approach does not require high-fidelity training data and instead relies on the data generated from the low-fidelity solver to initialize PINN for a faster and more accurate solution. Our approach could be perceived as a physics-guided initialization method (Willard et al., 2020), where it has been shown that pre-training neural networks with even incorrect parameters could improve performance (Jia et al., 2021). Finally, we focus on multiphysics fluid flow problems integrating fluid flow (Navier–Stokes equations), heat transfer (energy equation), and heterogeneous porous media transport (generalized Darcy law). While fluid flow (Cai et al., 2022), heat transfer (Cai et al., 2021), and porous media (Gasmi and Tchelepi, 2021) have all been studied with PINN, multi-fidelity modeling of multiphysics flows with PINN has received less attention.

The rest of the paper is organized as follows. The problem statement, introducing the PINN framework, and numerical experiments are represented in Section 2. In Section 3, the results showing the performance of the proposed approach are presented. Finally, the proposed approach and results are discussed in Section 4.

2. Methods

2.1. Problem statement

Let's consider a general form of PDEs describing the behavior of a physical system

$$L(\mathbf{u}) = \mathbf{f} \quad \mathbf{x} \in \Omega, \tag{1a}$$

$$B(\mathbf{u}) = \mathbf{g} \quad \mathbf{x} \in \partial\Omega, \tag{1b}$$

where L is a differential operator, B is a boundary condition operator that determines the specified boundary conditions, $\mathbf{u}$ represents the unknown variable like velocity or temperature, $\mathbf{x} \in \Omega$ is the spatial coordinates in the domain of interest, and $\partial\Omega$ denotes the boundary. In this paper, we focus on operators where L is the diffusion, advection–diffusion, and Navier–Stokes equations and consider multiphysics problems where these equations are coupled.

The main problem investigated is stated as follows: is it possible to increase the accuracy of low fidelity CFD solution strategies to multiphysics fluid flow and transport problems with PINN? Will the data produced by low fidelity solvers accelerate PINN convergence to a solution that is more accurate than the provided low-fidelity data? In our study, numerical simulations were performed in the commercial finite volume solver Ansys Fluent and the open-source finite element solver FEniCS. Low fidelity solution strategies (resolution and solver settings) as explained below were utilized together with transfer learning (TL) and PINN to obtain more accurate results, which were compared to high-fidelity simulations. The goal is to use PINN and transfer learning to map low fidelity data $\mathbf{u}_L$ to a high-fidelity level $\mathbf{u}_H$

$$\mathbf{u}_L(\mathbf{x}_L) \overset{TL\ PINN}{\mapsto} \mathbf{u}_H(\mathbf{x}_H), \tag{2}$$

where $\mathbf{x}_L \in \Omega_{h,L}$ and $\mathbf{x}_H \in \Omega_{h,H}$ are the discretized spatial domains in terms of the coarser and finer discretizations, respectively.

The proposed multi-fidelity model is composed of three steps. First, a low-fidelity CFD solution strategy is used to produce low-fidelity results that approximate the physics in the problem of interest. Next, a purely data-driven deep neural network is used to learn a nonlinear mapping from the input coordinates to the low fidelity data generated by CFD. Finally, transfer learning is utilized to initialize PINN with the learned low fidelity map and obtain a more accurate solution.

2.2. Physics-informed neural network (PINN)

In this part, we provide a brief overview of the PINN framework. Fig. 1 shows the schematic of the multi-fidelity approach using PINN. PINN is a flexible deep learning approach that encodes physics via PDE-based regularizations using automatic differentiation (Raissi et al., 2019), and as such provides a hybrid setting for integrated data-driven and physics-based solutions of forward and backward problems. In this work, we are interested in 2D, steady, multiphysics forward problems where we use PINN to approximate temperature $T(x,y)$ and velocity $\mathbf{u}(x,y)$ fields as a function of space.

First, the low fidelity simulation data is used to train a fully connected neural network that approximates temperature and velocity

$$(T, \mathbf{u}) = F_L(x, y; \Theta_L), \tag{3}$$

where Θ_L denotes the network's trainable parameters (i.e., weights and biases of each layer). To increase the expressive power of the networks, separate neural networks (F_L) were used to approximate velocity components and temperature. This step is purely data-driven where the loss function used in optimizing the Θ_L parameters is a standard mean-squared error (MSE) loss based on the low-fidelity data defined on the computational nodes of the low-fidelity CFD solver.

In the transfer learning step, F_L is used to initialize the final neural networks $F_H(x,y;\Theta_H)$ that output the high-fidelity velocity and temperature results. The parameters Θ_H are optimized using the following MSE loss function

$$\mathscr{L} = \mathscr{L}_{\mathscr{F}} + \lambda_b \mathscr{L}_{\mathscr{BC}}, \tag{4}$$

where the residual of the multiphysics PDEs described below define the physics loss function $\mathscr{L}_{\mathscr{F}}$ and boundary conditions for velocity and temperature are enforced using $\mathscr{L}_{\mathscr{BC}}$. $\lambda_b > 0$ is a hyperparameter that is applied to weight the relative contribution of the boundary condition regularization. Swish activation functions and Adam optimization were used in all cases and all test cases described below were implemented in Pytorch. We should mention that in this approach, we do not need to match the low-fidelity and high-fidelity CFD meshes. PINN can take arbitrary points as input, and therefore the low-fidelity CFD mesh nodes and the corresponding data are used in the pre-training step, whereas the high-fidelity CFD mesh nodes are used as collocation points for PINN training in the next step.

2.3. Test-case problem formulation

In this section, the different test case problems are described. For all test cases, two sets of CFD simulations were performed. The high-fidelity CFD simulation was used to verify the results and the low-fidelity CFD simulation was used for the multi-fidelity PINN initialization. Additionally, a traditional PINN simulation with random initialization (the Kaiming approach) was used to compare the convergence speed and accuracy of the two PINN approaches. In all PINN models, the same nodes as the high-fidelity CFD mesh were used for collocation points. In the low fidelity CFD simulations performed with FEniCS or Fluent (as

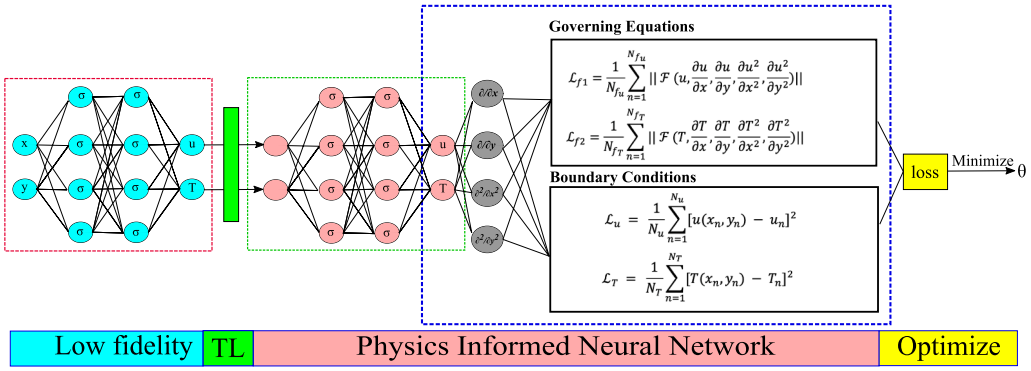

Fig. 1. A schematic overview of the proposed multi-fidelity approach using physics-informed neural networks, data coming from low-fidelity CFD solver strategies, and transfer learning (TL).

described below), a low-resolution mesh was used and the diffusion coefficient and/or kinematic viscosity was increased by 10% to represent a dissipative simulation. Additionally, for the Fluent simulations (finite volume), the least-square cell-based method was selected for gradient calculation in the high-fidelity simulations, whereas the less accurate Green-Gauss cell-based method was used for the low-fidelity

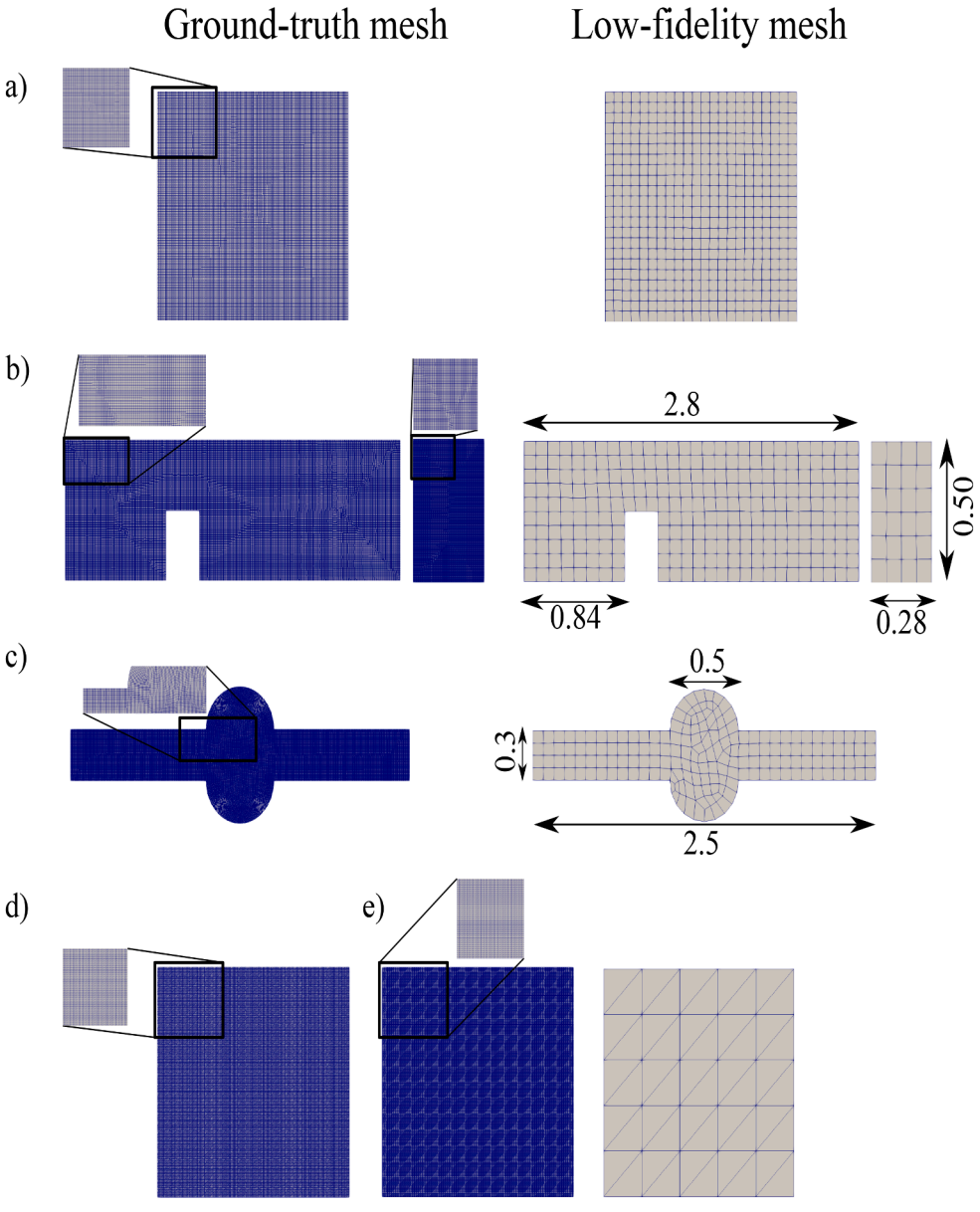

Fig. 2. The mesh and geometries used are shown. The ground-truth (high-fidelity) and low-fidelity mesh are shown in the left and right panels, respectively. All geometric lengths are dimensionless. a) Lid-driven cavity (test case 1). b) The fluid and solid domains in test cases 2 and 6 (flow over a fin). c) Aneurysm flow (test case 3). d) Rotating porous medium (test case 4). e) Rotating porous medium with heat transfer (test case 5). The same low-fidelity mesh was used for test cases 4 and 5.

cases. The first and second order upwind methods were selected for the low-fidelity and high-fidelity simulations, respectively (for both momentum and energy equations). Similarly, linear and second order discretization was selected for the pressure equation of the low-fidelity and high-fidelity models, respectively. The residual convergence criteria were set to 10^{-6} for the continuity, 10^{-8} for the momentum, and 10^{-5} for the energy equation in the high-fidelity simulations, and 10^{-4} for all equations in the low-fidelity models. In the FEniCS simulations (finite element), quadratic and linear shape functions were used in the high-fidelity and low-fidelity simulations, respectively. Overall, the options selected for the low-fidelity simulations offer a much faster simulation approach but in general, only provide a crude quantitative approximation of the physics.

2.3.1. Test case 1: 2D lid-driven cavity

Numerical solution to the lid-driven cavity flow problem has been extensively studied (Schreiber and Keller, 1983; Ghia et al., 1982) and used as a benchmark for CFD codes. For this problem, a square in the region [0,1] × [0,1] (expressed in dimensionless form) filled with a fluid with a density of $\rho = 998.2$ and kinematic viscosity of $\nu = 0.001$ is considered. The steady-state 2D Navier–Stokes equations govern the motion of the fluid

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{5a}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{5b}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \tag{5c}$$

where p is pressure, and u and v represent velocity in the x and y direction, respectively. No-slip boundary condition is applied to all the walls except for the top wall that drives the flow with u = 0.001. The high-fidelity mesh shown in Fig. 2a consisted of 40 k quadrilateral elements, while 484 elements were used for the low-fidelity simulations both carried out in Fluent. The low-fidelity data were used to initialize the neural networks with 100 epochs and a constant learning rate of 3×10^{-3}. Subsequently, the optimizer in PINN was run for 10000 epochs with a dynamic learning rate varying between 3×10^{-4} and 3×10^{-6}. Five hidden layers and 70 neurons per layer were used for each network representing u, v, and p. $\lambda_b = 20$ was used to weigh the boundary condition loss.

2.3.2. Test case 2: 2D fluid flow over a fin

As a second example, we consider the steady 2D Naiver-Stokes equations to solve the flow field over a rectangular fin ([0.84, 1.12] × [0, 0.5]) located inside a fluid domain ([0, 2.8] × [0, 1]) with a dimensionless density $\rho = 1$ and kinematic viscosity $\nu = 0.01$. A parabolic velocity profile with a peak Reynolds number of Re = 50 was applied at the inlet. No-slip boundary condition was assumed at the walls. CFD simulations were carried out in Fluent with a total number of 265 and 20 k elements for the low and high-fidelity cases, respectively, as shown in Fig. 2b. All neural network parameters were similar to the last example except that 150 neurons per layer and $\lambda_b = 30$ were used.

2.3.3. Test case 3: 2D flow in an aneurysm

We consider a 2D fluid flow problem inside a channel with a sudden focal enlargement. This test-case resembles a very idealized blood flow problem in an aneurysm (Arzani and Shadden, 2012), which is a common cardiovascular disease and has been previously modeled with PINN (Fathi et al., 2020; Arzani et al., 2021). A parabolic velocity profile with Re = 600 was defined at the inlet with no-slip at the walls. High-fidelity and low-fidelity simulations were modeled in Fluent with a total number of 59 k and 184 elements, respectively (Fig. 2c). Six hidden layers with 170 neurons per layer were selected for the neural networks. The low-fidelity neural network was trained with 200 epochs and a constant learning rate of 3×10^{-3}, and a variable learning rate (3×10^{-4} to 3×10^{-6}) with 10000 epochs was used for the PINN training with $\lambda_b = 70$.

2.3.4. Test case 4: 2D fluid flow in a rotating heterogeneous porous medium

As the next example, fluid flow through a heterogeneous porous medium in a rotating box is considered (Vadasz, 1993). In homogeneous porous media where the permeability is constant, the effect of rotation does not affect the flow, therefore, resulting in a uniform distribution of filtration velocity. However, for heterogeneous porous media, the spatial variability of permeability leads to a secondary flow due to the Coriolis effect. For this problem, the fluid is forced by an axial (in x-direction) pressure gradient into a fluid-saturated heterogeneous porous box having a square cross-section (in y-z plane) of [0,1] × [0,1] expressed in dimensionless form. The dimensionless density and permeability are $\rho = 1$ and $k = e^{-\gamma z}$, respectively, where $\gamma = -1$ was used in all computations. Consequently, the variation of the permeability is considered in the vertical z-direction only. An asymptotic expansion of the variables for large porous media Ekman numbers ($Ek = \frac{\phi\nu_*}{2\omega_* k_*}$ where ϕ is porosity, ν_* is the fluid's kinematic viscosity, ω_* is the imposed angular velocity, and k_* is a reference value of the permeability) produces a hierarchy of differential equations, by using Darcy's law extended to include the Coriolis and centrifugal effects. The steady-state solution at the leading zeroth order was obtained analytically in Vadasz (1993) producing an axial flow, $u_0 = k(z), v_0 = 0, w_0 = 0$. At the first order, the axial velocity turns out to vanish (axial velocity is part of the zeroth order solution) and secondary flow occurs in the cross-section, suggesting the introduction of a stream function ψ_1, defined by $v_1 = \frac{\partial \psi_1}{\partial z}, w_1 = -\frac{\partial \psi_1}{\partial y}$, where v_1 and w_1 are filtration velocity in the y and z direction, respectively. The latter yields the following equation governing the secondary flow (see (Vadasz, 1993) for details)

$$\frac{\partial^2 \psi_1}{\partial y^2}+\frac{\partial^2 \psi_1}{\partial z^2}-\frac{d(lnk)}{dz}\frac{\partial \psi_1}{\partial z}=-k\frac{dk}{dz}. \tag{5d}$$

Impermeability boundary condition was imposed at all walls (zero Dirichlet boundary condition for ψ_1). The simulations were conducted in FEniCS with 26 k and 100 triangular elements for high-fidelity and low-fidelity cases, respectively, as shown in Fig. 2d. The neural networks used in approximating the stream function had 5 hidden layers with 70 neurons per layer. In the initialization step, 100 epochs with a constant learning rate of 3×10^{-3} were used. In the final PINN model, the learning rate was chosen to vary between 3×10^{-4} and 3×10^{-6} and the simulation was run for 10000 iterations with $\lambda_b = 20$.

2.3.5. Test case 5: 2D convection in a rotating porous medium

Herein, we consider a steady state free convection problem in a long rotating homogeneous porous box (Vadasz, 1993) where temperature gradients and convection result from the differential heating of the horizontal walls subject to centrifugal body force (Vadasz, 2021). A main flow is generated in the axial direction accompanied by secondary flow in the cross-section, which is the focus here. The cross-section is a square [0,1] × [0,1] (dimensionless). The box is heated from the top and cooled from the bottom with insulated side wall walls. The governing equation for this problem is Darcy's law extended to include centripetal and Coriolis accelerations. The present problem focuses on the Coriolis effect on the convective flow and is solved for large values of the Ekman number. An asymptotic expansion of the variables for large porous media Ekman numbers produces a hierarchy of differential equations where the leading zeroth order solution uses the Boussinesq approximation to solve for the axial flow (Vadasz, 1993). The first order effect, solves for the secondary flow in the y-z plane. A stream function is defined similar to the previous example to derive the following set of differential equations (see (Vadasz, 1993))

$$\frac{\partial^2\psi_1}{\partial y^2}+\frac{\partial^2\psi_1}{\partial z^2}=Ra_\omega x, \tag{7}$$

$$\frac{\partial^2 T_1}{\partial y^2}+\frac{\partial^2 T_1}{\partial z^2}=-\frac{\partial\psi_1}{\partial y}, \tag{8}$$

where $x=0.5$ is the position of the cross-section. The rotation Rayleigh number in porous media was defined as $Ra_\omega=\frac{\beta_{T*}\Delta T_c\omega_c^2 L_* H_* K_* M_f}{\alpha_{e0}\nu_0}$ where β_{T*} is the thermal expansion coefficient, $\Delta T_c$is a characteristic temperature difference, ω_c is the characteristic angular velocity of the rotating box, L_* is the length of the box, H_* is the height of the box, K_* is the permeability, M_f is the ratio between the heat capacity of the fluid and the effective heat capacity of the porous domain, α_{e0} is the effective thermal diffusivity, and ν_0 is the kinematic viscosity of the fluid. $Ra_\omega=100$ was assumed in this example. These coupled equations were solved for ψ_1 (flow) and T_1 (temperature) with zero Dirichlet boundary condition on all boundaries for $\psi_1, T_1=0$ on z = 0 and z = 1, and $\frac{\partial T_1}{\partial y}=0$ on y = 0 and y = 1.

Finally, the total temperature (T) and filtration velocity (v,w) in the cross section were computed by adding the zeroth order effect with the above first order Ekman number effect using the original boundary conditions T = 0 (z = 0) and T = 1 (z = 1)

$$T=z+Ek^{-1}\times T_1, \tag{9}$$

$$w=w_1\times Ek^{-1}, \tag{10}$$

$$v=v_1\times Ek^{-1}. \tag{11}$$

It was reported by Vadasz (1993) that the Coriolis effect on free convection is controlled by the combined dimensionless group $\sigma=Ra_\omega Ek^{-1}$ and a value of $\sigma=10$ was assumed in the calculations. High-fidelity and low-fidelity simulations were conducted in FEniCS with a total number of 40 k and 100 triangular elements, respectively (Fig. 2e). The neural networks used in approximating the stream function had 4 hidden layers with 70 neurons per layer, whereas 5 hidden layers and 70 neurons per layer were selected to approximate temperature. Learning rates, number of epochs, and λ_b were set similar to the previous test case.

2.3.6. Test case 6: 2D multiphysics heat transfer in a fin

As the last example, we solved a 2D steady state problem for a rectangular fin in a fluid domain where conduction in the solid is coupled with convection in the fluid. The same geometric dimension used in test case 2 was considered. The continuity and momentum equations (Eq. 5) together with the energy equations (advection–diffusion in the fluid and diffusion in the solid domain) were solved

$$u\frac{\partial T_f}{\partial x}+v\frac{\partial T_f}{\partial y}-\alpha\left(\frac{\partial^2 T_f}{\partial x^2}+\frac{\partial^2 T_f}{\partial y^2}\right)=0, \tag{12a}$$

$$\frac{\partial^2 T_s}{\partial x^2}+\frac{\partial^2 T_s}{\partial y^2}=0, \tag{12b}$$

where u and v are velocity in x and y direction, T_f and T_s denote temperature in the fluid and solid domains, respectively, and a thermal diffusivity of $\alpha=0.02$ was considered. As boundary conditions, a parabolic velocity profile with a peak Reynolds number of $Re=50$ was prescribed at the inlet. No-slip boundary condition was applied at the walls. For thermal boundary conditions, the inlet temperature was set to zero and the base temperature of the fin was set to one (non-dimensional). At the fluid–solid interface, equal heat flux and temperature of the solid and fluid were enforced.

The simulations were carried out in Fluent. The number of grid elements to solve high-fidelity simulation was 20 k in each domain (total number 40 k) and to solve the low-fidelity problem the number of elements was 265 and 24 in the fluid and solid domains, respectively (Fig. 2b). The low-fidelity neural network was run for 200 epochs with a constant learning rate of 3×10^{-3}, whereas the final PINN simulation had 10000 epochs with a learning rate varying between 3×10^{-4} and 3×10^{-6}. The number of hidden layers to approximate velocity and pressure was 7 with 150 neurons per layer, while 6 hidden layers and 140 neurons per layer were selected for approximating temperature. $\lambda_b=30$ was used to weigh the boundary condition loss.

3. Results

In the following sections, we present the results for the different test case problems. The average error for all cases is presented in Table 1.

3.1. Test case 1: 2D lid-driven cavity

The results for test case 1 are shown in Fig. 3. The velocity contours for the ground-truth (high-fidelity) CFD simulation, low-fidelity CFD data, and PINN predictions are shown. The absolute errors with respect to the ground-truth CFD data are shown in Fig. 3b, which demonstrate the improved accuracy achieved with PINN compared to low-fidelity data. The loss vs. epoch plot in Fig. 3c shows that the given low-fidelity data can accelerate PINN convergence and improve accuracy compared to when the network is randomly initialized. In this case, the equation loss sees a more notable improvement compared to the boundary condition (BC) loss.

3.2. Test case 2: 2D fluid flow over a fin

Fig. 4 shows the test case 2 results. The multi-fidelity PINN approach provides accurate results compared to the ground-truth simulation with only a small localized region in the top left corner of the fin where the localized high error in the low-fidelity data persists. The improvement in multi-fidelity PINN convergence and accuracy (equation and BC loss) is evident compared to the original random initialization approach (Fig. 4c and Table 1). The absolute error in the low-fidelity data could be perceived as what multi-fidelity PINN needs to correct. In other words, this error is the new physics that PINN has to learn to provide the more accurate solution. Compared to case 1 (Fig. 3b), it could be seen that in case 2 this new physics (error) is qualitatively different from the ground-truth physics of the problem.

3.3. Test case 3: 2D blood flow in an aneurysm

The 2D aneurysm results are shown in Fig. 5. Similar trends could be seen where the multi-fidelity approach improves data accuracy and improves PINN convergence. However, compared to the previous two test cases, less improvement could be seen in this example. The complex pattern in the low-fidelity data error (Fig. 5b) explains this observation as PINN has to learn a more complex pattern to bridge the gap between the low-fidelity data and high-fidelity solution.

Table 1
The spatial average of the absolute error (with respect to high-resolution CFD) for different simulations is reported. V and T represent velocity and Temperature, respectively.

Test cases	Multi-fidelity PINN	Traditional PINN	Low-fidelity CFD
Test case 1 (V)	6.83e−6	1.33e−5	1.12e−4
Test case 2,6 (V)	7.6e−3	0.0106	0.0819
Test case 3(V)	0.0293	0.0415	0.2105
Test case 4(V)	7.61e−6	3.21e−5	0.0156
Test case 5 (V)	1.81e−5	6.01e−4	0.1566
Test case 5 (T)	2.89e−5	7.46e−4	7.44e−3
Test case 6 (T)	1.7e−4	1e−3	0.0160

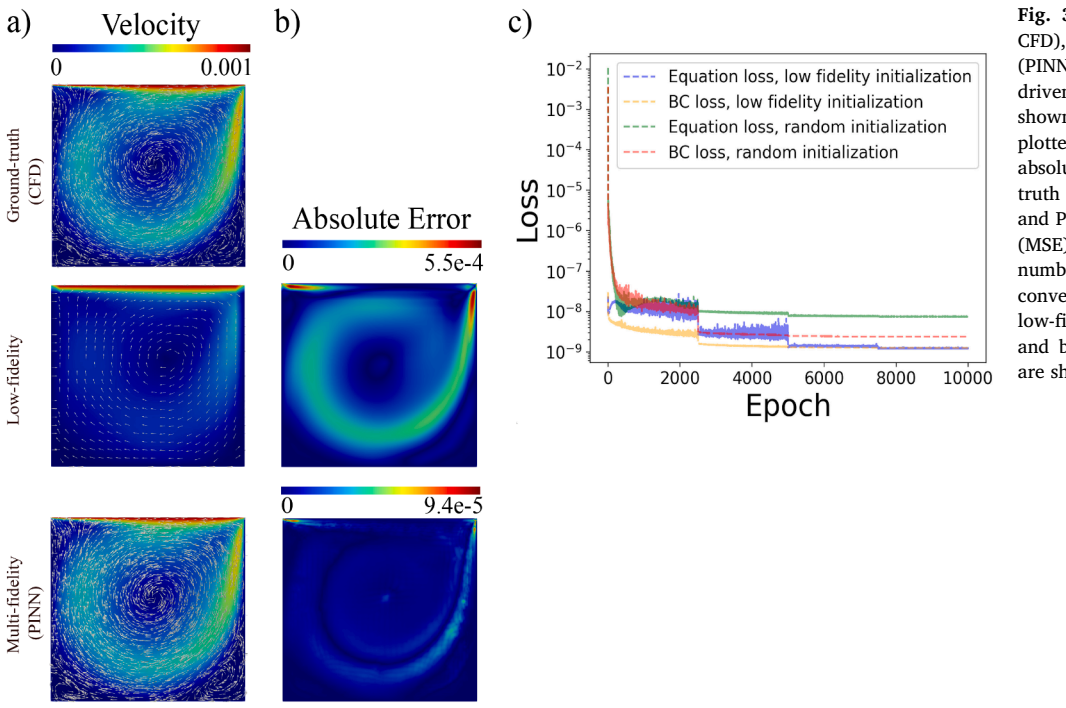

Fig. 3. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) velocity results for the 2D lid-driven cavity problem (test case 1) are shown. Normalized velocity vectors are also plotted to show the flow direction. b) The absolute errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation and boundary condition loss contributions are shown separately.

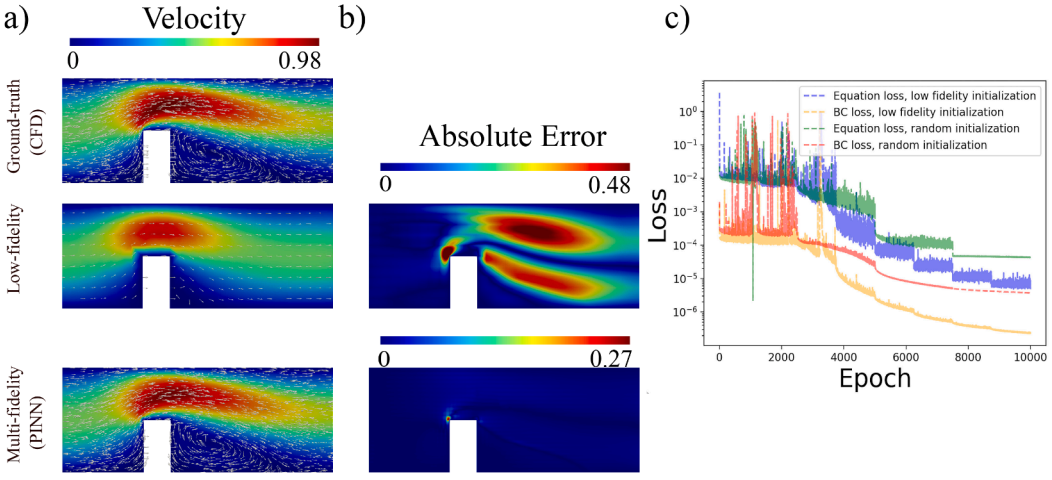

Fig. 4. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) velocity results for the flow over a fin problem (test case 2) are shown. Normalized velocity vectors are also plotted to show the flow direction. b) The absolute errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation and boundary condition loss contributions are shown separately.

3.4. Test case 4: 2D fluid flow in a rotating heterogeneous porous medium

The solution for the fluid flow in a rotating porous medium is shown in Fig. 6. The results show that the multi-fidelity procedure significantly improves the accuracy and leads to results indistinguishable from the high-fidelity results with minor errors in the bottom corners of the box. Also, the plotted graph in Fig. 6c shows that the given low fidelity data can accelerate PINN and improve the training efficiency of both equation and BC loss.

3.5. Test case 5: 2D convection in a rotating porous medium

The velocity and temperature results from test case 5 simulations are shown in Fig. 7. It could be seen that the multi-fidelity procedure remarkably improves the accuracy of low fidelity velocity and temperature data. Similar to test case 4, localized velocity errors are only observed in the corners of the box. Also, the graph in Fig. 7e and Table 1 show that low-fidelity initialization can improve PINN performance. However, compared to test case 4, this improvement is less when heat transfer is included. It should be noted that here the equation loss consists of momentum and energy equations combined.

3.6. Test case 6: 2D multiphysics heat transfer in a fin

The temperature results for test case 6 are shown in Fig. 8. The velocity results were similar to test case 2 (Fig. 4) and are not included here for brevity. We could see localized regions near the boundary between the fin and the fluid where the reduction in error is not as significant as some of the other examples. This could be due to the multiphysics nature of this problem where the energy equations (heat conduction in the solid

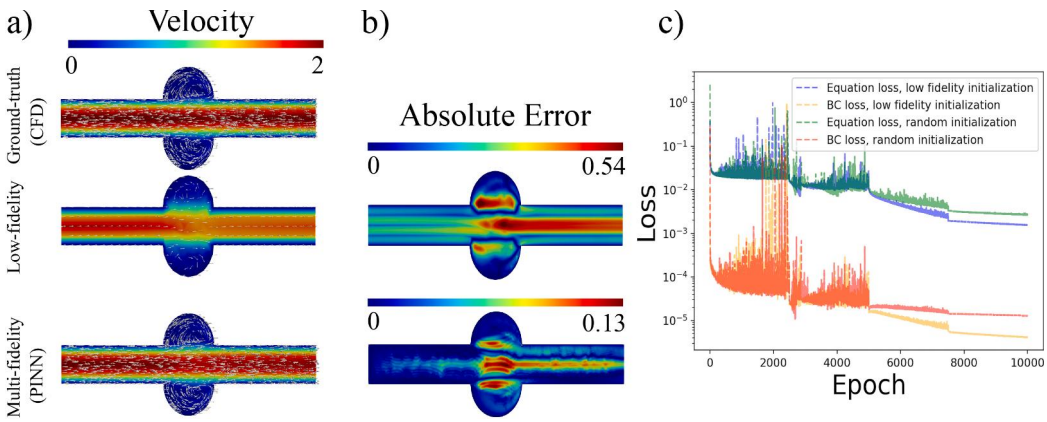

Fig. 5. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) velocity results for the idealized aneurysm problem (test case 3) are shown. Normalized velocity vectors are also plotted to show the flow direction. b) The absolute errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation and boundary condition loss contributions are shown separately.

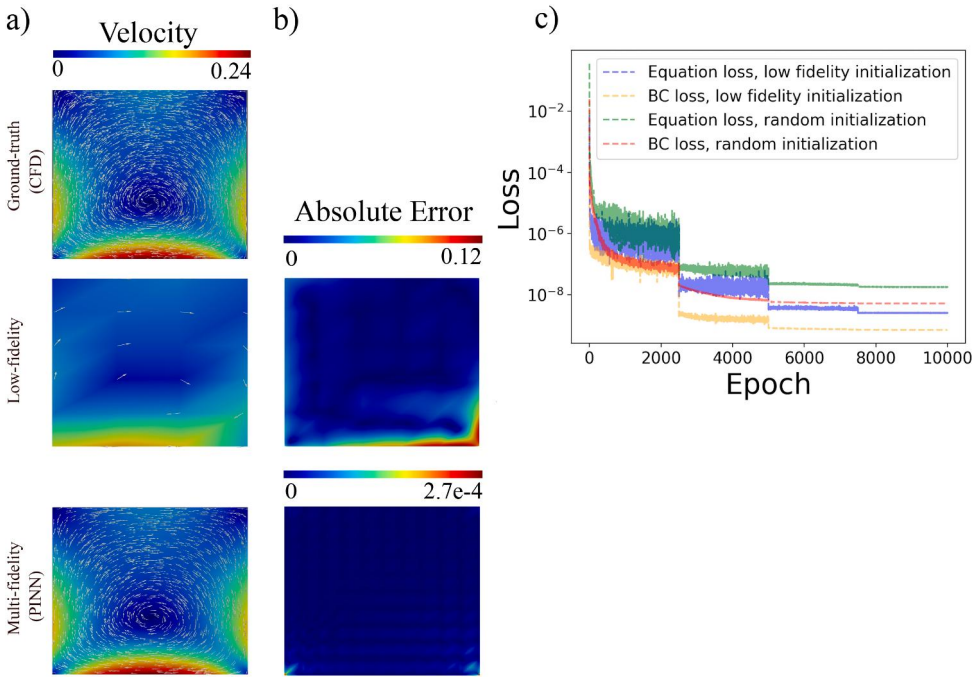

Fig. 6. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) velocity results for the rotating heterogeneous porous medium problem (test case 4) are shown. Normalized velocity vectors are also plotted to show the flow direction. b) The absolute errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation and boundary condition loss contributions are shown separately.

coupled with heat convection in the fluid) are dependent on the solution to the Navier–Stokes equations. However, according to the graph in Fig. 8c, this complexity does not prevent the low fidelity data to accelerate PINN convergence.

3.7. The effects of neural network size and freezing layers

The effect of the number of hidden layers and neurons is shown in Fig. 9. Figs. 9a and Figs. 9b show this effect for test cases 5 and 6, respectively. It could be seen that interestingly in these examples increasing the network size reduced the accuracy of the results.

During transfer learning, an efficient way to keep some information during training is to freeze some layers (Aggarwal, 2018). In this case, the optimizer does not update the parameters in the frozen layers. To study this effect, two initial layers were frozen in PINN and only later layers were updated for test cases 1 and 4. The equation and BC loss are plotted in Fig. 9c and Fig. 9d for test case 1 and 4, respectively. In both cases, freezing layers does not lead to a notable change in PINN's accuracy. However, since weights and biases of fewer layers need to get updated, the total simulation run time is reduced.

4. Discussion

In this work, we presented a multi-fidelity physics-informed neural network approach for solving various multiphysics partial differential equations. The proposed approach consisted of low-fidelity data coming from low-fidelity solvers combined with transfer learning and PINN. In particular, the proposed approach consists of three steps. First, low-fidelity simulations are carried out with multiphysics CFD solvers (herein, Fluent and FEniCS). Then, deep neural networks are used to fit the low-fidelity data. The last step uses the transfer learning method to initialize the deep neural networks in PINN and execute PINN with these initializations. Six problem sets were presented to demonstrate the performance of the proposed approach. Our problems consisted of fluid flow, heat transfer, porous media transport, and their multiphysics combinations. To illustrate the performance of the proposed approach, the results obtained from our approach were compared with ground-truth (high-fidelity) CFD simulations. Additionally, a traditional PINN solution (random initialization) was conducted to compare the convergence rate. For all of the considered problems, the proposed approach was able to enhance prediction accuracy and also speed up and improve PINN convergence. We had fixed the total number of epochs for the two

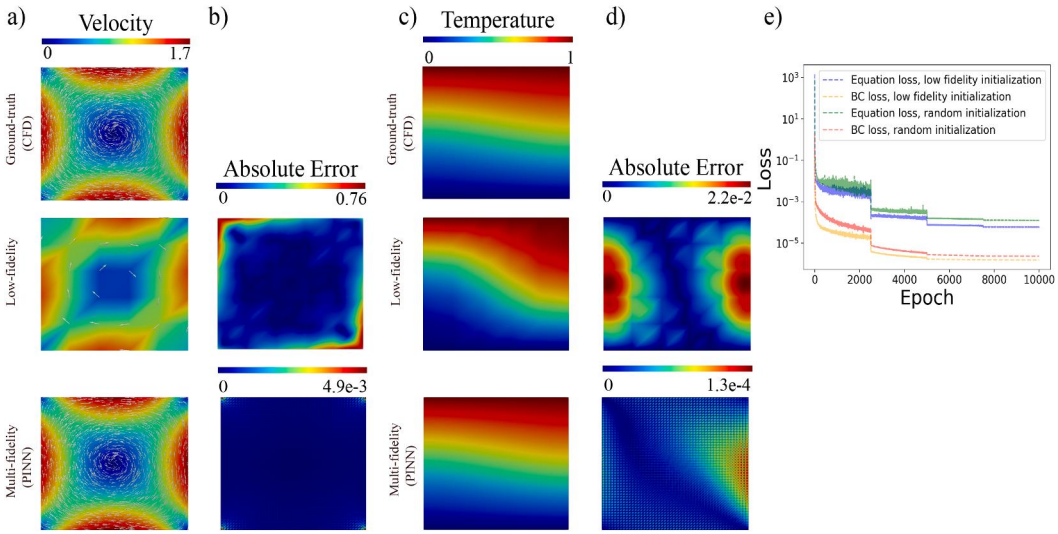

Fig. 7. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) velocity results for the rotating porous medium with heat transfer problem (test case 5) are shown. Normalized velocity vectors are also plotted to show the flow direction. b) The absolute velocity errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The corresponding temperature results and d) absolute errors in temperature are shown. e) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation and boundary condition loss contributions are shown separately where the equation loss contains all governing equations.

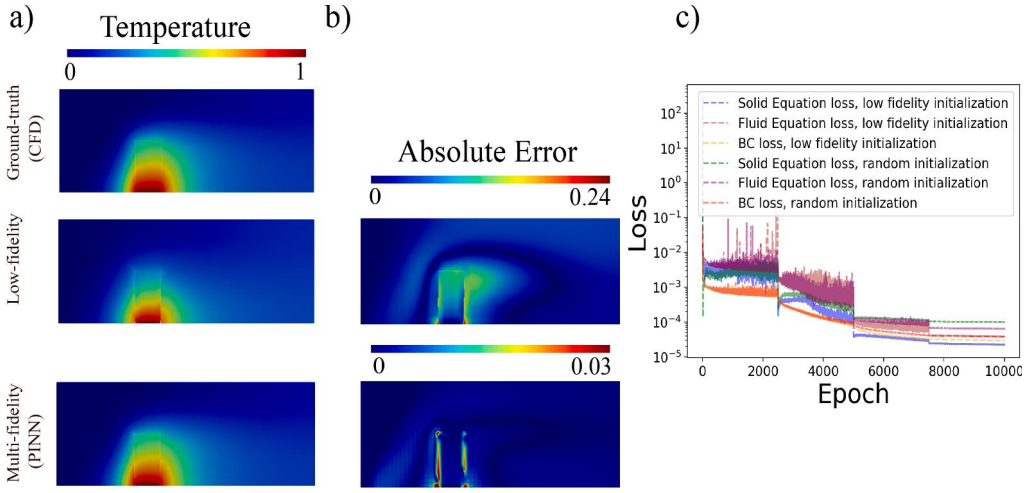

Fig. 8. a) The ground truth (high-fidelity CFD), low-fidelity CFD, and multi-fidelity (PINN) temperature results for the heat transfer in a fin problem (test case 6) are shown. b) The absolute errors with respect to the ground-truth data are shown for the low-fidelity and PINN results. c) The mean square error (MSE) loss versus the epoch (iteration) number is plotted to compare PINN convergence with random initialization and low-fidelity data initialization. The equation loss for the fluid and solid domains and boundary condition loss contributions are shown separately.

PINN approaches, therefore the total runtime was similar. However, the multi-fidelity approach reached a lower MSE loss in fewer epochs, which could be leveraged to reduce the multi-fidelity total runtime.

Our proposed approach was able to improve traditional PINN convergence. Currently, PINN's training cost and accuracy (compared to machine precision) are two main hurdles in applying PINN to complex problems. We were able to use low-fidelity and therefore computationally cheap CFD simulations to accelerate PINN. Additionally, the number of epochs that were used to fit the low-fidelity CFD data into the neural networks was low (~200 epochs). The low number of epochs in the pre-training step is required not only to justify the speedup gained in the multi-fidelity approach but also the early stopping helps avoid overfitting. As the low-fidelity data are just a crude approximation to the final solution, an approximated fitting of these data is attractive for our transfer learning method. It should be noted that various other methods such as adaptive loss (Bischof and Kraus, 2021), XPINN (Jagtap and Karniadakis, 2020), and optimal sampling (Nabian et al., 2021) have been recently proposed to improve PINN convergence. These methods could be used along with our multi-fidelity approach to further improve PINN training efficiency and accuracy.

Other studies have incorporated low-fidelity physics solvers into neural network models using different approaches. Low-fidelity models have been added to intermediate layers of a deep neural network as an input to augment network training, where the panel method (inviscid flow theory) was used as a low-fidelity model to improve the learning of aerodynamics models (Pawar et al., 2021). As another example, parareal PINN has been proposed to accelerate time-integration in unsteady problems (Meng et al., 2020). Parareal PINN uses a similar concept to our work but in the time domain. Specifically, it leverages coarse-grained time integrators (low-fidelity time integrators) to obtain approximate solutions, and subsequently, PINN corrects the low-fidelity predictions. It is also possible to use low-fidelity equations directly in PINN. For example, Reynolds-averaged Navier–Stokes (RANS) equations have been used to model turbulence with PINN (Eivazi et al., 2021).

Transfer learning has been used in PINN problems beyond multi-fidelity modeling. A popular application is to use transfer learning to accelerate PINN solutions to problems with variations in parameters or boundary conditions (Goswami et al., 2020; Jin et al., 2021). Curriculum learning, which leverages transfer learning has been used to

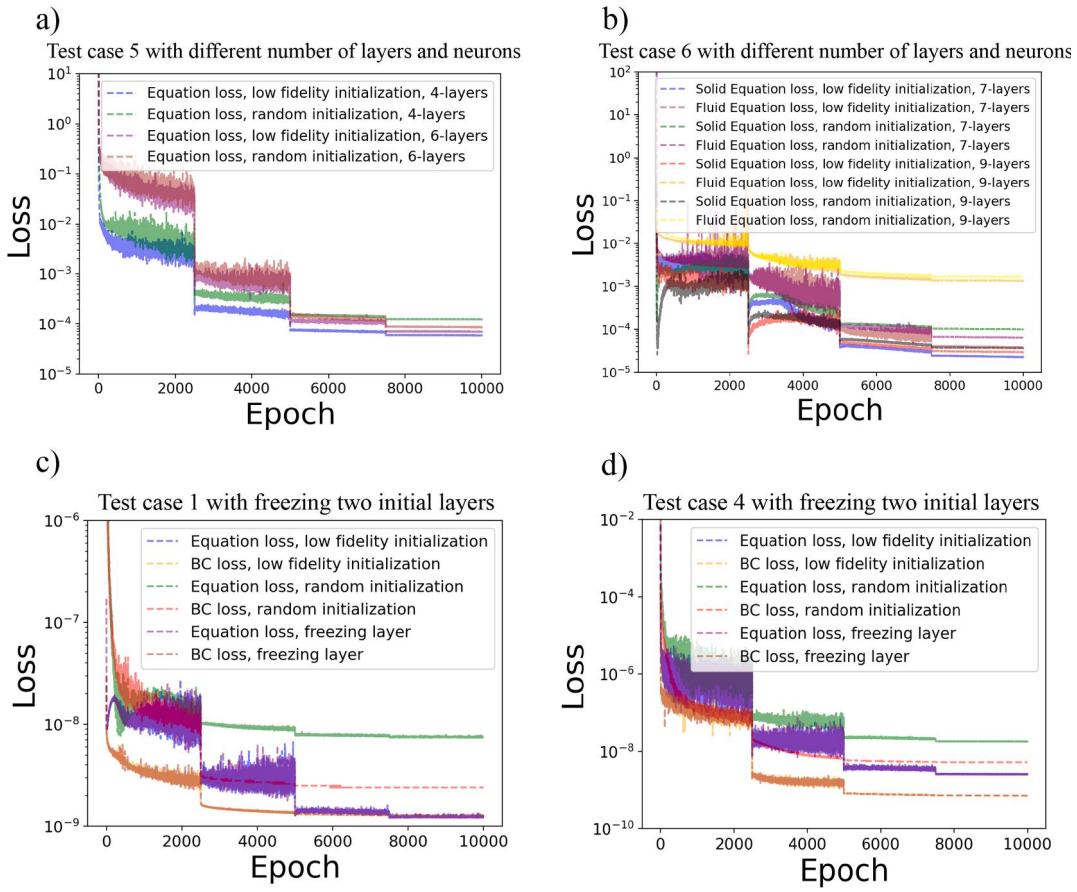

Fig. 9. The effect of network size and freezing layer are shown in select test cases. For each case, the mean square error (MSE) loss versus the epoch number is plotted. a) The rotating porous media with heat transfer problem loss is plotted. The 4-layers case had 70 neurons per layer and the 6-layers case had 100 neurons per layer. b) The multiphysics heat transfer in a fin problem loss is plotted. The 7-layers case had 150 neurons per layer and the 9-layers case had 220 neurons per layer. c) The effect of freezing two initial layers is shown for the lid-driven cavity problem. d) The effect of freezing two initial layers is shown for the rotating heterogeneous porous media problem.

progressively add complications to the PINN loss equation and therefore facilitate PINN training in complex differential equations (Krishnapriyan et al., 2021). For example, the contribution of advection to modeling transport is increased during different iterations to facilitate the final solution (Krishnapriyan et al., 2021). This could be compared to our approach where a higher diffusion coefficient (lower Peclet number) was used in our low fidelity solver. Finally, similar models in the context of experimental fluid mechanics could be developed (Eivazi and Vinuesa, 2022) where robustness to noise becomes an important issue.

In conclusion, the proposed multi-fidelity approach shows great promise for increasing the accuracy of low-fidelity solvers and improving PINN convergence. The proposed approach was used for solving 2D steady state problems with simple geometries. Future investigations are needed to check the efficiency of this method for 3D time-dependent problems. Currently, PINN training is still significantly slower than CFD modeling (approximately 1–2 days vs. 20 min for the results in the current study). With improvements in newer generations of PINNs and improved GPU hardware, the proposed framework in this study could lead to novel cyberinfrastructure where the proposed framework could be executed parallel to a low-fidelity and fast CFD solver to provide accurate results with reduced computational cost.

Data availability

The Pytorch codes and data used to generate the results presented are available on Githubhttps://github.com/amir-cardiolab/PINN_multiphysics_multifidelity.

CRediT authorship contribution statement

Maryam Aliakbari: Methodology, Software, Writing - original draft. **Mostafa Mahmoudi:** Methodology, Software. **Peter Vadasz:** Methodology, Writing - review & editing, Supervision. **Amirhossein Arzani:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

AA acknowledges support from the National Science Foundation ECCS-CCSS Award No. 2103434.

References

Aggarwal, C.C., 2018. Neural Networks and Deep Learning: A Textbook. Springer.
Arzani, A., Shadden, S.C., 2012. Characterization of the transport topology in patient-specific abdominal aortic aneurysm models. Phys. Fluids 24 (8), 081901.
Arzani, A., Wang, J.X., D'Souza, R.M., 2021. Uncovering near-wall blood flow from sparse data with physics-informed neural networks. Phys. Fluids 33 (7), 071905.

Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., et al., 2019. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Tech. rep., USDOE Office of Science (SC), Washington, DC (United States).

Bischof, R., Kraus, M., 2021. Multi-objective loss balancing for physics-informed deep learning. arXiv preprint arXiv:2110.09813.

Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2022. Physics-informed neural networks (pinns) for fluid mechanics: A review. Acta. Mech. Sin. 1–12.

Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E., 2021. Physics-informed neural networks (pinns) for heat transfer problems. J. Heat Transfer.

Chakraborty, S., 2021. Transfer learning based multi-fidelity physics informed deep neural network. J. Comput. Phys. 426, 109942.

Eivazi, H., Tahani, M., Schlatter, P., Vinuesa, R., 2021. Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations. arXiv preprint arXiv: 2107.10711.

Eivazi, H., Vinuesa, R., 2022. Physics-informed deep-learning applications to experimental fluid mechanics. arXiv preprint arXiv:2203.15402.

Fathi, M.F., Perez-Raya, I., Baghaie, A., Berg, P., Janiga, G., Arzani, A., D'Souza, R.M., 2020. Super-resolution and denoising of 4D-Flow MRI using physics-informed deep neural nets. Comput. Methods Programs Biomed. 105729.

Gasmi, C.F., Tchelepi, H., 2021. Physics informed deep learning for flow and transport in porous media. arXiv preprint arXiv:2104.02629.

Ghia, U., Ghia, K.N., Shin, C., 1982. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. J. Comput. Phys. 48 (3), 387–411.

Goswami, S., Anitescu, C., Chakraborty, S., Rabczuk, T., 2020. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theoret. Appl. Fract. Mech. 106, 102447.

Jagtap, A.D., Karniadakis, G.E., 2020. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Commun. Comput. Phys. 28 (5), 2002–2041.

Jia, X., Willard, J., Karpatne, A., Read, J.S., Zwart, J.A., Steinbach, M., Kumar, V., 2021. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. ACM/IMS Trans. Data Sci. 2 (3), 1–26.

Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. J. Comput. Phys. 426, 109951.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nature Rev. Phys. 1–19.

Khan, M.O., Valen-Sendstad, K., Steinman, D.A., 2015. Narrowing the expertise gap for predicting intracranial aneurysm hemodynamics: impact of solver numerics versus mesh and time-step resolution. Am. J. Neuroradiol. 36 (7), 1310–1316.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks. Adv. Neural Inform. Process. Syst. 34.

Liu, D., Wang, Y., 2019. Multi-fidelity physics-constrained neural network and its application in materials modeling. J. Mech. Des. 141 (12).

Meng, X., Karniadakis, G.E., 2020. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. J. Comput. Phys. 401, 109020.

Meng, X., Li, Z., Zhang, D., Karniadakis, G.E., 2020. PPINN: Parareal physics-informed neural network for time-dependent PDEs. Comput. Methods Appl. Mech. Eng. 370, 113250.

Nabian, M.A., Gladstone, R.J., Meidani, H., 2021. Efficient training of physics-informed neural networks via importance sampling. Comput.-Aided Civil Infrastruct. Eng. 36 (8), 962–977.

Pawar, S., San, O., Aksoylu, B., Rasheed, A., Kvamsdal, T., 2021. Physics guided machine learning using simplified theories. Phys. Fluids 33 (1), 011701.

Peherstorfer, B., Willcox, K., Gunzburger, M., 2018. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Rev. 60 (3), 550–591.

Penwarden, M., Zhe, S., Narayan, A., Kirby, R.M., 2021. Multifidelity modeling for physics-informed neural networks (PINNs). J. Comput. Phys. 110844.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.

Schreiber, R., Keller, H.B., 1983. Driven cavity flows by efficient numerical techniques. J. Comput. Phys. 49 (2), 310–333.

Vadasz, P., 1993. Fluid flow through heterogeneous porous media in a rotating square channel. Transp. Porous Media 12 (1), 43–54.

Vadasz, P., 1993. Three-demensional free convection in a long rotating porous box: Analytical solution. J. Heat Transfer 115 (3), 639–644.

Vadasz, P., 2021. Centrifugal buoyancy in a rotating fluid layer next to and distant from the rotation axis. Phys. Fluids 33 (3), 034123.

Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V., 2020. Integrating physics-based modeling with machine learning: A survey. arXiv preprint arXiv:2003.04919 1 (1), 1–34.