



## Letter

## Physics-informed deep learning for incompressible laminar flows

Chengping Rao<sup>a</sup>, Hao Sun<sup>b,c,\*</sup>, Yang Liu<sup>a,\*</sup><sup>a</sup> Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA<sup>b</sup> Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, USA<sup>c</sup> Department of Civil and Environmental Engineering, MIT, Cambridge, MA 02139, USA

## HIGHLIGHTS

- Proposed a mixed-variable physics-informed deep learning scheme for modeling incompressible laminar flows.
- Used the general continuum and constitutive formulations.
- Demonstrated effectiveness for modeling both steady and transient flows.

## ARTICLE INFO

## Article history:

Received 14 January 2020

Received in revised form 10 April 2020

Accepted 20 April 2020

Available online 22 May 2020

## Keywords:

Physics-informed neural networks  
(PINN)

Deep learning

Fluid dynamics

Incompressible laminar flow

## ABSTRACT

Physics-informed deep learning has drawn tremendous interest in recent years to solve computational physics problems, whose basic concept is to embed physical laws to constrain/inform neural networks, with the need of less data for training a reliable model. This can be achieved by incorporating the residual of physics equations into the loss function. Through minimizing the loss function, the network could approximate the solution. In this paper, we propose a mixed-variable scheme of physics-informed neural network (PINN) for fluid dynamics and apply it to simulate steady and transient laminar flows at low Reynolds numbers. A parametric study indicates that the mixed-variable scheme can improve the PINN trainability and the solution accuracy. The predicted velocity and pressure fields by the proposed PINN approach are also compared with the reference numerical solutions. Simulation results demonstrate great potential of the proposed PINN for fluid flow simulation with a high accuracy.

©2020 The Authors. Published by Elsevier Ltd on behalf of The Chinese Society of Theoretical and Applied Mechanics. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Deep learning (DL) has attracted tremendous attentions in recent years in the field of computational mechanics due to its powerful capability in nonlinear modeling of complex spatiotemporal systems. According to a technical report [1] by U.S. Department of Energy, a DL-based approach should be featured with the domain-aware, interpretable and robust to be a general approach for solving the science and engineering problems. Recent studies of leveraging DL to model physical system include, just to name a few [2-6]. These applications can be categorized into two types based on how the DL model is constructed: in either data-driven or physics-informed manner. In a data-driv-

en framework, the DL model is constructed as a black-box to learn a surrogate mapping from the formatted input  $\mathbf{x} \in \mathbb{R}^m$  to the output  $\mathbf{y} \in \mathbb{R}^n$ . The exceptional approximation ability of deep neural network (DNN) makes possible to learn the mapping even when the dimensionality  $m$  and  $n$  are very high. The training dataset  $\{\mathbf{x}, \mathbf{y}\}$ , typically very rich, can be obtained by conducting high-fidelity simulations using exact solvers (e.g., see Refs. [3, 4, 7]). Nevertheless, obtaining a rich and sufficient dataset from simulations for training a reliable DL model is computationally expensive and requires careful case design. To address this fundamental challenge, physics-informed DL explicitly embed the physical laws (e.g., the governing partial differential equations (PDEs), initial/boundary conditions, etc.) into the DNN, constraining the network's trainable parameters within a feasible solution space. The objective of exploiting physical laws

\* Corresponding author.

E-mail address: [h.sun@northeastern.edu](mailto:h.sun@northeastern.edu) (H. Sun), [yang1.liu@northeastern.edu](mailto:yang1.liu@northeastern.edu) (Y. Liu).

in DNN is assumed to (1) reduce the large dependency of the model on available dataset in terms of both quality and quantity, and (2) improve the robustness and interpretability of the DL model. In this regard, DNN essentially has the capacity of approximating the latent solutions for PDEs [8, 9], with distinct benefits summarized as follows: (1) the superior interpolation ability of DNN, (2) the approximated solution has a close form with its infinite derivative continuous, and (3) state-of-the-art hardware advances make the numerical implementation and parallelization extremely convenient.

More recently, Raissi et al. [5, 6] introduced a general framework of PINN and demonstrated its capacity in modeling complex physical systems such as solving/identifying PDEs. A huge difference from some of the previous studies is that, in addition to the physical laws, the PINN can also exploit the available measurement data, making it possible to discover the systems whose physics are not fully understood. In particular, seminal contributions of using PINN to model fluid flows have been made recently. For example, Kissas et al. [10] employed PINN to predict the arterial blood pressure based on the MRI data and the conservation laws. Sun et al. [11] proposed a PINN approach for surrogate modeling of fluid flows without any simulation data. Zhu et al. [12] proposed a physics-constrained convolutional encoder-decoder network and a generative model for modeling of stochastic fluid flows.

In this paper, we formulate a mixed-variable PINN scheme for simulation of viscous incompressible laminar flows without any measurement data. The remaining this paper is organized as follows. We will introduce the methodology of the mixed-variable PINN and the mathematical formulation for fluid dynamics. The steady and transient laminar flows passing a circular cylinder will be modeled using the proposed PINN scheme without any simulation data. A comparison study is made to demonstrate the improved solution accuracy and network trainability by the proposed scheme.

Let us consider the incompressible Newtonian flow governed by the following Navier–Stokes equations:

$$\nabla \cdot \mathbf{v} = 0, \quad (1)$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{v} + \mathbf{g}, \quad (2)$$

where  $\nabla$  is the Nabla operator,  $\mathbf{v} = (u, v)$  is the velocity vector,  $p$  is the pressure,  $\mu$  is the viscosity of the fluid,  $\rho$  is the density of fluid and  $\mathbf{g}$  is the gravitational acceleration. When leveraging PINN to solve the aforementioned PDEs, minimizing a complex residual loss resulted from Eq. (2) is intractable due to its complex form with multiple latent variables (e.g.,  $\mathbf{v}$  and  $p$ ) and high-order derivatives (e.g.,  $\nabla^2$ ). In order to design an easily trainable PINN, we convert the Navier–Stokes equation in Eq. (2) to the following continuum and constitutive formulations:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} + \mathbf{g}, \quad (3)$$

$$\boldsymbol{\sigma} = -p \mathbf{I} + \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T), \quad (4)$$

where  $\boldsymbol{\sigma}$  is the Cauchy stress tensor and  $p = -\text{tr}\boldsymbol{\sigma}/2$ . The benefits of using the continuum-mechanics-based formulation are two-

fold: (1) reducing the order of derivatives when a mixed-variable scheme in PINN is used and (2) improved trainability of DNN as found in the comparison of numerical results.

The proposed mixed-variable scheme is used in this paper to solve the aforementioned PDEs (see Eqs. (1), (3) and (4)) that govern the laminar flow dynamics. The salient feature of PINN is that the physical fields are approximated globally by a DNN. In particular, the DNN maps the spatiotemporal variables  $\{t, \mathbf{x}\}^T$  to the mix-variable solution  $\{\psi, p, \boldsymbol{\sigma}\}$ , where the stream function  $\psi$  is employed rather than the velocity  $\mathbf{v}$  to ensure the divergence free condition of the flow. In this way, the continuity equation will be satisfied automatically. For a two-dimensional problem, the velocity components can be computed by  $[u, v, 0] = \nabla \times [0, 0, \psi]$ . Note that  $\mathbf{v} = [u, v]$  is taken as the latent variable. The automatic differentiation is used to obtain the partial derivatives of the DNN output regarding the time and space (e.g.,  $t$ ,  $x$  and  $y$ ). The loss function is composed of the data loss  $J_d$  (if measurements are available) and the physics loss  $J_p$ . The physics loss  $J_p$  is the summation of the governing equation loss  $J_g$  and the initial and boundary condition loss  $J_{i/bc}$ , given by

$$J_g = \frac{1}{N_g} \sum_{i=1}^{N_g} \|\mathbf{r}_g(\mathbf{x}^i, t^i)\|^2, \quad (5)$$

$$J_{i/bc} = \frac{1}{N_I} \sum_{i=1}^{N_I} \|\mathbf{r}_I(\mathbf{x}^i, 0)\|^2 + \frac{1}{N_{nb}} \sum_{i=1}^{N_{nb}} \|\mathbf{r}_{nb}(\mathbf{x}^i, t^i)\|^2 + \frac{1}{N_{db}} \sum_{i=1}^{N_{db}} \|\mathbf{r}_{db}(\mathbf{x}^i, t^i)\|^2, \quad (6)$$

where  $\mathbf{r}_{(\cdot)}$  denotes the residual,  $\|\cdot\|$  denotes the  $\ell_2$  norm and  $N_{(\cdot)}$  denotes the number of collocation points (subscripts  $g$  for governing equation,  $I$  for initial condition,  $nb$  for Neumann boundary, and  $db$  for Dirichlet boundary). The total physics loss  $J_p$  is defined as

$$J_p = J_g + \beta J_{i/bc}, \quad (7)$$

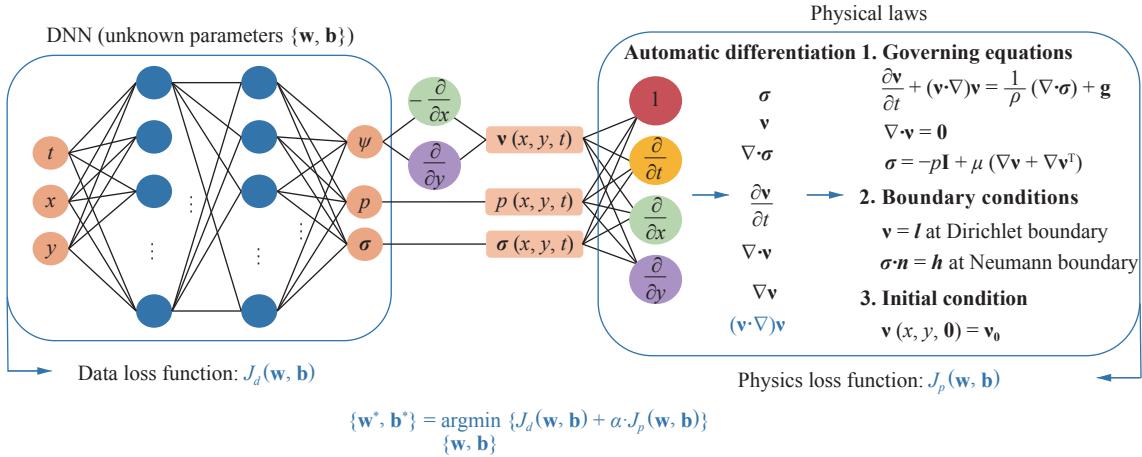
where  $\beta > 0$  is a user-defined weighting coefficient for initial and boundary condition loss. Noteworthy, having the measurement data makes the fluid flow modeling data-driven, which is however not a prerequisite. The architecture of the proposed PINN for fluid dynamics simulation is presented in Fig. 1. In this paper, no measurement data from simulations or physical experiments is used for training the PINN.

In this section, we employ the proposed PINN to model the steady and transient flows passing a circular cylinder. A parabolic velocity profile is applied on the inlet while the zero pressure condition is applied on the outlet, as shown in Fig. 2. Non-slip conditions are enforced on the wall and cylinder boundaries. The gravity is ignored in both two cases. The proposed PINN is implemented on the TensorFlow [13] and the source codes can be found in <https://github.com/Raoocp/PINN-laminar-flow>.

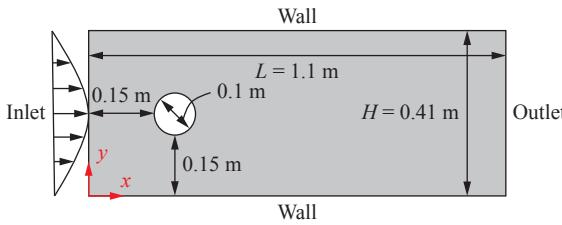
For the steady case, the dynamic viscosity and density of the fluid is  $2 \times 10^{-2}$  kg/(m·s) and 1 kg/m<sup>3</sup> respectively. The normal velocity profile is defined as

$$u(0, y) = 4U_{\max}(H - y)/H^2, \quad (8)$$

with  $U_{\max}$  equal to 1.0 m/s which results in a small Reynolds number so that the flow is dominated by laminar flow. A total



**Fig. 1.** Architecture of the physics-informed neural network for fluid dynamics. Note that  $\alpha$  is a user-defined weighting coefficient.  $\mathbf{w}$  and  $\mathbf{b}$  are weights and biases for the DNN. The constraint of initial and boundary conditions can be converted as residuals adding to the loss function based on Lagrangian multipliers. The data loss  $J_d$  is present only when data is available.



**Fig. 2.** Diagram of the computation model

number of  $N_g = 50000$  collocation points, which includes  $N_{db} = 1200$  Dirichlet boundary (cylinder, wall, inlet) points and  $N_{nb} = 200$  Neumann boundary (outlet) points, are generated using Latin hypercube sampling (LHS) for the training the network. It should be noted that the collocation points are refined near the cylinder to better capture the details of the flow. A grid search strategy is used to find an optimal combination of depth and width for the network. The relative  $\ell_2$  error defined by

$$\epsilon = \frac{\sqrt{\sum_{i=1}^M \|\mathbf{f}_{\text{pred}}^i - \mathbf{f}_{\text{ref}}^i\|^2}}{\sqrt{\sum_{i=1}^M \|\mathbf{f}_{\text{ref}}^i\|^2}}, \quad (9)$$

is used as the metric for comparison, where  $\mathbf{f}$  is the physical quantity of interest, and  $M$  is the total number of reference points. Adam [14] and Limited-memory BFGS (L-BFGS) optimizer [15] is employed to train the DNN due to their good convergence speed demonstrated in the tests. We also implement the traditional scheme for fluid dynamics employed in [5,6] where the stream function and pressure  $\{\psi, p\}$  act as the output variables. From the relative  $\ell_2$  errors of the velocity field (see Table 1), it can be seen that the network of 8×40 achieves the best result among all the configurations. The mixed-variable PINN improves the accuracy of numerical results over the traditional PINN.

The predicted velocity and pressure fields by the PINNs with mixed-variable and traditional scheme are shown in Fig. 3(b, c).

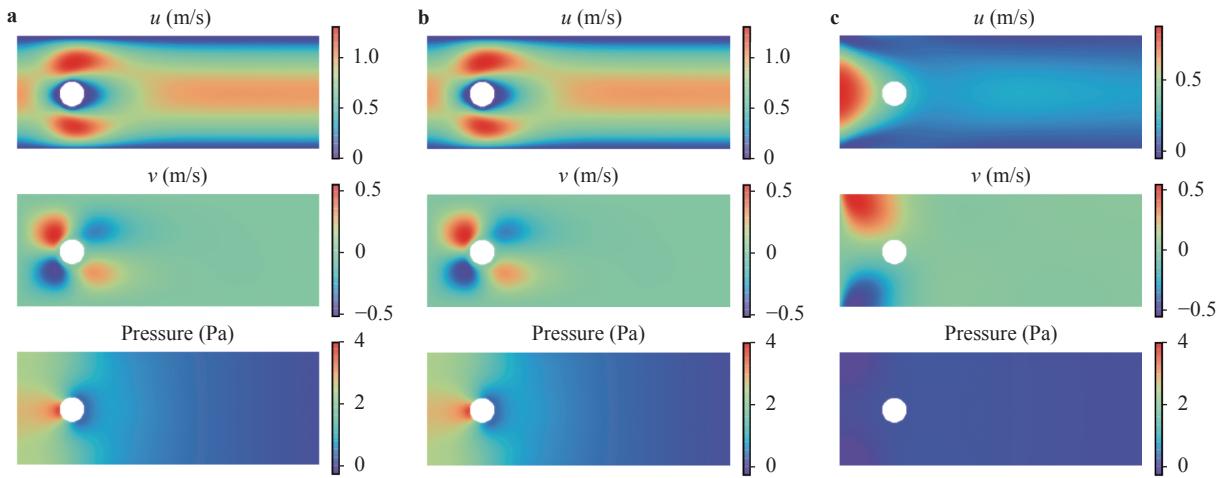
**Table 1** Relative  $\ell_2$  errors (unit:  $10^{-2}$ ) of the velocity field for different DNN configurations with  $\beta = 2$  (left: the traditional scheme; right: the proposed mixed-variable scheme)

Width depth	6	7	8
40	76.1/4.2	69.1/4.7	73.8/1.8
50	74.0/2.6	81.1/2.4	76.9/2.3
60	74.5/2.3	75.2/2.4	75.4/1.9

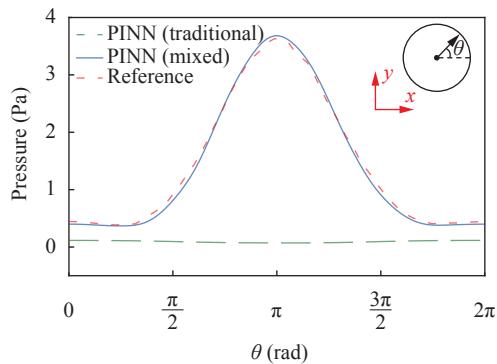
The reference solution is obtained from the ANSYS Fluent 18.1 package (finite volume-based) [16] (see Fig. 3a). It can be observed that the PINN with the traditional scheme fails to model the flow. In particular, the traditional scheme fails to enforce the non-slip condition on the lower and upper boundaries. However, the steady velocity and pressure fields are well reproduced by the PINN with mixed-variable scheme. It is worth mentioning that the pressure distribution on the cylinder surface is typically of interest for computing the resultant drag and lift forces. Therefore, we compare the pressure distributions obtained by two types of PINN and ANSYS Fluent as shown in Fig. 4. The overall agreement between the mixed-variable PINN and ANSYS Fluent is very good.

We also compare the performance of these two schemes under various  $\beta$  which controls the weight of the boundary condition loss. As shown in Fig. 5, the convergence of the traditional scheme is significantly affected by  $\beta$ , though the final loss can be reduced by increasing the value of  $\beta$  up to 10. However, the mixed-variable scheme yields consistent results for various  $\beta$ . The improvement by the mixed-variable scheme is thanks to the reduced order of derivatives required to construct the loss function, in comparison with the traditional scheme [5, 6], which makes the optimization problem easier.

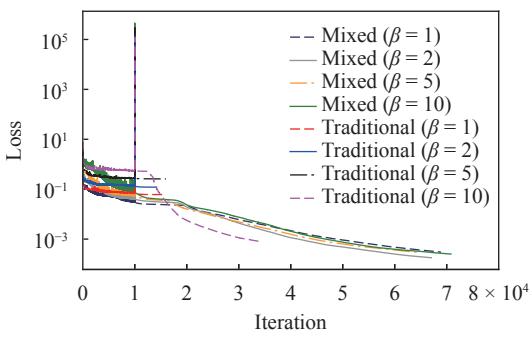
The transient flow with the same computation domain depicted in Fig. 2 is considered in this case. The dynamic viscosity of the flow is  $\mu = 5 \times 10^{-3} \text{ kg}/(\text{m} \cdot \text{s})$  while the density is  $\rho = 1 \text{ kg}/\text{m}^3$ . The time duration for the modeling is 0.5 s. Three virtual pressure probes  $P_1(0.15, 0.2) \text{ m}$ ,  $P_2(0.2, 0.25) \text{ m}$  and



**Fig. 3.** Velocity and pressure fields of the steady flow passing a circular cylinder. **a** Reference solution from ANSYS Fluent. **b** Mixed-variable scheme solution with  $8 \times 40$  network. **c** Traditional scheme solution with  $8 \times 40$  network. The hyperparameters and collocation points for training these two PINNs are kept same.



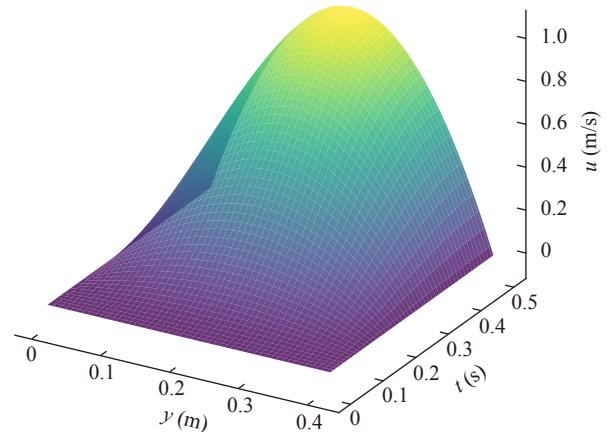
**Fig. 4.** Distribution of pressure on cylinder. Network of  $8 \times 40$  and  $\beta = 2$  are used.



**Fig. 5.** Comparison of convergence curves with respect to coefficient  $\beta$ . Network of  $8 \times 40$  is used in all the cases. 10000 iterations trained with Adam optimizer followed by L-BFGS optimizer

$P_3(0.25, 0.2)$  m are installed on the surface of the cylinder. The flow is initially still while a time-varying parabolic inlet velocity profile is applied subsequently, which is defined as

$$u(0, y) = 4 \left[ \sin\left(\frac{2\pi t}{T} + \frac{3\pi}{2}\right) + 1 \right] U_{\max}(H - y)y/H^2, \quad (10)$$

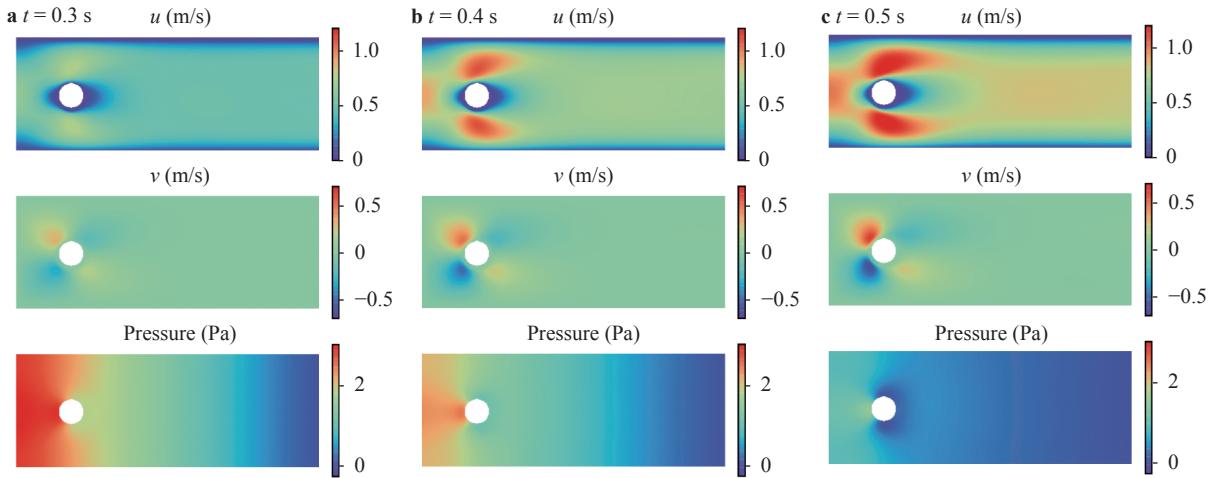


**Fig. 6.** Transient normal velocity profile.

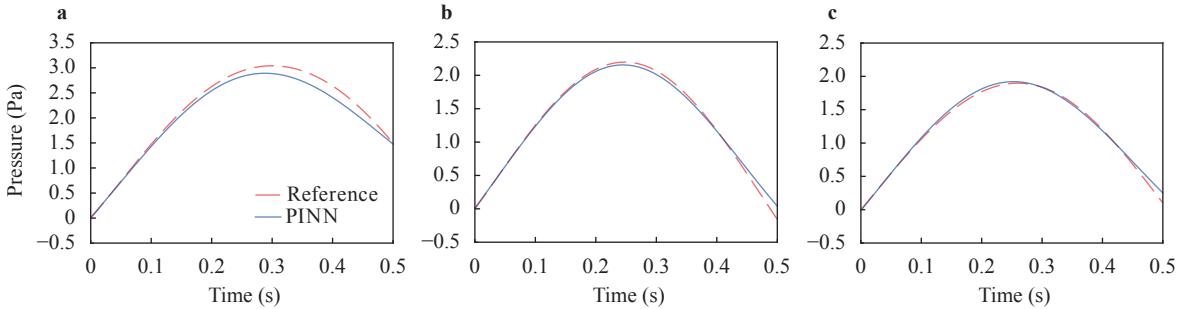
where  $U_{\max}$  equals to 0.5 m/s and the period  $T$  is 1.0 s. The remaining boundary conditions are the same as those in the previous example. The inflow velocity as a function of  $t$  and  $y$  is visualized in Fig. 6. The width and depth of the network are selected to be 50 and 7 respectively while the coefficient  $\beta$  is set to be 2. A total number of  $N_g = 120000$  collocation points, which include  $N_{db} = 9600$  points on cylinder, wall and inlet boundaries,  $N_{nb} = 3200$  points on outlet, and  $N_i = 3500$  points at initial time, are used to train the network.

Three snapshots of the predicted flow fields are presented in Fig. 7 which shows the evolution of the flow as the inlet velocity increases over time. The reference flows obtained by ANSYS Fluent are not shown here since the PINN-predicted result matches very well with them. The pressure time histories on three probes obtained from the proposed PINN are compared with those from ANSYS Fluent, as depicted in Fig. 8. It can be seen that the proposed PINN approach can well predict the pressure time histories in a transient flow.

We propose a mixed-variable PINN scheme for modeling fluid flows, with particular applications to incompressible laminar



**Fig. 7.** Snapshots of the PINN-predicted transient flow fields passing a circular cylinder



**Fig. 8.** Pressure time histories on **a**  $P_1$ , **b**  $P_2$  and **c**  $P_3$  probes.

flows. The salient features of the proposed scheme include (1) employing the general continuum equations together with the material constitutive law rather than the derived Navier-Stokes equations, and (2) using stream function to ensure the divergence free condition of the flow in a mixed-variable setting. The comparison study indicates the benefits (high accuracy and good trainability) of the proposed mixed-variable scheme. In both the steady and transient flow cases, the result produced by the PINN shows a good agreement with the reference numerical solutions.

It is notable that the applications in this paper are limited to the laminar flows at low Reynolds numbers, although the approach is in theory applicable to turbulent flows at large Reynolds numbers. However, it requires discretizing the computation domain with much finer collocation points which will lead to computer memory issues and drastically increase the computational cost. Our future work aims to address this challenge by developing a "divide-and-conquer" training scheme in the context of transfer learning, that is to divide the time domain into multiple steps and re-train the network partially while fixing the weights and the biases from the previous step [17].

## References

- [1] S. Lee, N. Baker, Basic research needs for scientific machine learning: Core technologies for artificial intelligence, Tech.

- Rep., USDOE Office of Science (SC), USA (2018).
- [2] R. Zhang, Z. Chen, S. Chen, et al., Deep long short-term memory networks for nonlinear structural seismic response prediction, *Computers & Structures* 220 (2019) 55–68.
- [3] C. Yang, X. Yang, X. Xiao, Data-driven projection method in fluid simulation, *Computer Animation and Virtual Worlds* 27 (2016) 415–424.
- [4] J. Tompson, K. Schlachter, P. Sprechmann, et al., Accelerating eulerian fluid simulation with convolutional networks, in: *Proceedings of the 34th International Conference on Machine Learning* 70 (2017) 3424–3433.
- [5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physicsinformed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* 378 (2019) 686–707.
- [6] M. Raissi, Z. Wang, M.S. Triantafyllou, et al., Deep learning of vortex-induced vibrations, *Journal of Fluid Mechanics* 861 (2019) 119–137.
- [7] N. Geneva, N. Zabaras, Quantifying model form uncertainty in reynolds-averaged turbulence models with bayesian deep neural networks, *Journal of Computational Physics* 383 (2019) 125–147.
- [8] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* 9 (1998) 987–1000.

- [9] I.E. Lagaris, A.C. Likas, D.G. Papageorgiou, Neural network methods for boundary value problems with irregular boundaries, *IEEE Transactions on Neural Networks* 11 (2000) 1041–1049.
- [10] G. Kissas, Y. Yang, E. Hwang, et al., Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow mri data using physics-informed neural networks, *Computer Methods in Applied Mechanics and Engineering* 358 (2020) 112623.
- [11] L. Sun, H. Gao, S. Pan, et al., Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, *Computer Methods in Applied Mechanics and Engineering* 361 (2020) 112732.
- [12] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, et al., Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *Journal of Computational Physics* 394 (2019) 56–81.
- [13] M. Abadi, P. Barham, J. Chen, et al., Tensorflow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation, (2016) 265–283.
- [14] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980.
- [15] R. Byrd, P. Lu, J. Nocedal, et al., A limited memory algorithm for bound constrained optimization, *SIAM Journal on Scientific Computing* 16 (1995) 1190–1208.
- [16] ANSYS Fluent 18.1, ANSYS FLUENT Theory Guide, ANSYS Inc., PA, USA, 2018.
- [17] S. Goswami, C. Anitescu, S. Chakraborty, et al., Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, *Theoretical and Applied Fracture Mechanics* 106 (2020) 102447.