Journal of Computational Physics 473 (2023) 111768

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Theory-guided physics-informed neural networks for boundary layer problems with singular perturbation

Amirhossein Arzani [a,b,*], Kevin W. Cassel [c], Roshan M. D'Souza [d]

[a] *Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, USA*
[b] *Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA*
[c] *Department of Mechanical, Materials and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL, USA*
[d] *Department of Mechanical Engineering, University of Wisconsin–Milwaukee, Milwaukee, WI, USA*

ARTICLE INFO

Article history:
Received 3 June 2022
Received in revised form 30 September 2022
Accepted 8 November 2022
Available online 14 November 2022

Keywords:
Scientific machine learning
Deep learning
Data-driven modeling
Asymptotic expansion
Convective transport

ABSTRACT

Physics-informed neural networks (PINNs) are a recent trend in scientific machine learning research and modeling of differential equations. Despite progress in PINN research, large gradients and highly nonlinear patterns remain challenging to model. Thin boundary layer problems are prominent examples of large gradients that commonly arise in transport problems. In this study, boundary-layer PINN (BL-PINN) is proposed to enable a solution to thin boundary layers by considering them as a singular perturbation problem. Inspired by the classical perturbation theory and asymptotic expansions, BL-PINN is designed to replicate the procedure in singular perturbation theory. Namely, different parallel PINN networks are defined to represent different orders of approximation to the boundary layer problem in the inner and outer regions. In different benchmark problems (forward and inverse), BL-PINN shows superior performance compared to the traditional PINN approach and is able to produce accurate results, whereas the classical PINN approach could not provide meaningful solutions. BL-PINN also demonstrates significantly better results compared to other extensions of PINN such as the extended PINN (XPINN) approach. The natural incorporation of the perturbation parameter in BL-PINN provides the opportunity to evaluate parametric solutions without the need for retraining. BL-PINN demonstrates an example of how classical mathematical theory could be used to guide the design of deep neural networks for solving challenging problems.

1. Introduction

Thin boundary layers with large gradients are a common feature of high Reynolds number flows and high Peclet number heat/mass transfer. The aerodynamic problem of drag reduction in turbulent boundary layers [46], convective heat transfer in cooling [16], and biotransport in concentration boundary layers [1] are a few important examples. Prandtl's boundary layer theory proposed during the start of the 20th century has sparked continuing research in this area over the past 120 years [21]. Modeling thin boundary layers is computationally challenging due to the inherently large gradients. In momentum transport analysis, thin boundary layers in practice are typically turbulent, and therefore numerically expensive to model. In heat and mass transport, thin boundary layers can also occur in the laminar regime due to reduced diffusivity.

* Corresponding author at: Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, USA.
E-mail address: amir.arzani@sci.utah.edu (A. Arzani).

https://doi.org/10.1016/j.jcp.2022.111768
0021-9991/

For example, cardiovascular mass transport problems have very thin concentration boundary layers due to the very small diffusion coefficients of biochemicals in blood, which make numerical modeling very challenging [24].

In recent years, data-driven modeling and scientific machine learning approaches have gained considerable interest in fluid flow and transport modeling [12,13]. Perhaps the earliest such work in the context of boundary layers was done by Thwaites in 1949 where a solution to the boundary layer momentum-integral equation was found by using a collection of available experimental and analytical results to fit a term in the momentum-integral equation and enable a closed-form analytical solution [49,56]. The correlation method of Thwaites was an early example of hybrid data-driven and physics-based modeling in fluid mechanics and specifically boundary layers.

Physics-informed neural networks (PINN) are a trending topic in scientific machine learning and enable hybrid physics-based and data-driven modeling within a deep learning setting [43,29]. PINN has been applied to various fluid mechanics [13] and heat transfer [14] problems. However, the robustness of PINN in certain problems remains an issue [29]. PINN has limited accuracy in complex and highly nonlinear flow patterns such as turbulence, vortical structures, and boundary layers [29]. Developing robust and reliable models has been identified as a priority in scientific machine learning research [5]. Boundary layers are one of the topics that challenge the robustness of PINNs. In current PINN models, after a sufficient reduction of the boundary layer thickness (e.g., reduction in the diffusion coefficient), PINN will suffer from convergence issues. Such difficulty also poses a challenge for operator learning approaches such as DeepONet [33], which might not be able to learn parametric variations in the solution across all parameters, and therefore robustness will be challenging to achieve.

Over the past couple of years, various variants of the original PINN approach have been proposed that attempt to overcome certain PINN limitations. Fourier feature networks have been developed within PINN to overcome spectral bias in deep neural networks, which limits how well high-frequency functions could be learned [55]. Conservative PINN (cPINN) [28], extended PINN (XPINN) [27], and other similar domain decomposition techniques [53] have been proposed to leverage localized neural networks in regions of high gradient or complex patterns to enable efficient learning of complex functions. Alternatively, other approaches have used an enhanced local sampling of the collocation or training points near high gradient regions to improve convergence [34,39]. However, none of these techniques studied thin boundary layers. We demonstrate that domain decomposition without special treatment cannot resolve the issues with learning thin boundary layers due to their highly localized abrupt behavior. Additionally, we show that increasing the resolution of the collocation points within the boundary layer does not resolve PINN training issues. PINN has been applied to various advection-diffusion transport problems [19,26,18,38] including boundary layers [3,58,7]. These studies investigated optimal weighting of the loss terms and mainly focused on low Peclet numbers to enable a solution to these challenging problems. However, thin boundary layers (the limit of vanishing viscosity/diffusivity) remain an elusive target for PINNs.

In this manuscript, we present a theory-guided and model-driven machine learning approach for learning thin boundary layer behavior. Our framework is inspired by the singular perturbation and asymptotic expansions method for solving differential equations [9,50]. The singular perturbation theory is a well-established approach in applied mathematics and much of its developments have been inspired by the fluid dynamics community [40]. In singular perturbation problems, a small perturbation parameter (e.g., viscosity in momentum transport or diffusivity in heat/mass transport) is multiplied by the highest order derivative. The singular nature of the problem makes the behavior of the system in the limit of vanishing perturbation very different from a zero value of the perturbation parameter. A very thin boundary layer is created in such problems, and the resulting abrupt change in the solution is even difficult to resolve using traditional and established numerical techniques such as the finite element method (FEM) [24]. Singular perturbation solutions are tailor made for such situations as they actually become increasingly accurate as the boundary layer thins and the gradients increase. For example, such asymptotic basis functions have been used as the basis functions in Galerkin projection in order to accurately capture and represent the singular behavior inherent in such solutions [15]. Inspired by perturbation theory and its use as asymptotic basis functions in projection methods, we propose boundary layer physics-informed neural network (BL-PINN) to overcome the current limitations of deep learning in resolving thin boundary layers. That is, through the lens of asymptotic expansions [15], our BL-PINN approach could be perceived as a PINN-driven reduced-order model (ROM) where unlike traditional ROM models (e.g., proper orthogonal decomposition or dynamic mode decomposition) our ROM approach is not data-driven but instead physics-driven. In summary, our study makes the following key contributions

- We provide a new BL-PINN approach for physics-informed neural network modeling of thin boundary layers. We demonstrate in benchmark problems that our approach overcomes the limitations of PINN in solving forward and inverse thin boundary layer problems.
- We demonstrate how classical mathematical theories (herein, perturbation methods) could be replicated with PINN in a theory-guided/model-driven approach.
- Our approach provides a reduced-physics model (RPM) within PINN. This approach is entirely driven by the governing mathematical equations and is in contrast with current data-driven ROM approaches, which rely on data to form their basis function. BL-PINN could be perceived as a combination of an RPM and PINN.
- Our asymptotic basis function approach in BL-PINN incorporates gauge functions (containing the perturbation parameter) and the spatial coordinates dependence distinctly, and therefore could be used to re-evaluate the solution as the small parameter (herein, diffusion coefficient) varies. This natural incorporation of parametric dependence is an improvement compared to traditional data-driven approaches. BL-PINN enables parametric PINN evaluation without the

need for retraining, therefore providing attractive advantages similar to operator learning approaches such as DeepONet. In fact, BL-PINN actually becomes more accurate with increasing Reynolds/Peclet number, which is the opposite of traditional PINN that fails as the boundary layer thins and corresponding gradients increase.

The rest of the manuscript is organized as follows. First, we overview the solution procedure to singularly perturbed differential equations. Next, we present the BL-PINN approach along with a few benchmark problems. We present the results and demonstrate the advantage of BL-PINN over the traditional PINN approach and other variants of PINN (local clustering of collocation points and XPINN). Finally, we discuss the results and present future directions and other applications for BL-PINN.

2. Methods

2.1. Problem statement: singularly perturbed differential equations

Consider a differential equation of the form

$$L_\epsilon \mathbf{u} = f(\mathbf{x}) \,, \tag{1}$$

subject to appropriate boundary conditions where ϵ is a small parameter appearing in the operator L_ϵ (e.g., a given small diffusion coefficient). We assume this is a singularly perturbed problem, which means that the solution found by the differential equation when $\epsilon = 0$ behaves very differently from that in the limit $\epsilon \to 0$. A common scenario is when ϵ is multiplied by the highest order derivative term. This will lead to a "boundary layer" where the solution varies rapidly in a small region. The thickness of this region approaches zero in the limit $\epsilon \to 0$. In perturbation theory [9,50,32], the solution to such a problem is written in terms of asymptotic expansions and the solution is divided into an inner and outer region, as shown in Fig. 1. The outer region (away from the boundary layer) is approximated with a regular expansion

$$\mathbf{u}_{\text{outer}}(\mathbf{x}) = \sum_{n=0}^{\infty} \delta_n(\epsilon)\phi_n(\mathbf{x}) \,, \tag{2}$$

where $\delta_n(\epsilon)$ are gauge functions representing the asymptotic sequence of the terms in the solution (e.g., ϵ^n) and $\phi_n(\mathbf{x})$ are functions of space that embed the solution for each order of ϵ. As this is a regular expansion, the leading order solution corresponds to $\epsilon = 0$. On the other hand, to approximate the boundary layer region a stretched variable is introduced as $\xi = \frac{\mathbf{x}-\mathbf{x}_0}{\delta(\epsilon)}$, which allows one to zoom into the thin boundary layer region and locally represent the solution as

$$\mathbf{u}_{\text{inner}}(\mathbf{x}) = \sum_{n=0}^{\infty} \delta_n(\epsilon)\psi_n(\mathbf{x},\epsilon) = \sum_{n=0}^{\infty} \delta_n(\epsilon)\bar{\psi}_n(\xi) \,, \tag{3}$$

where $\bar{\psi}_n(\xi)$ is the spatial function $\psi_n(\mathbf{x},\epsilon)$ written in terms of the stretched variable. Finally, the outer and inner solutions are matched in the overlap region using matched asymptotic expansions [50] to obtain the final solution. Briefly, the inner solution when $\xi \to \infty$ is enforced to match the outer solution when $\mathbf{x} \to 0$.

2.2. Boundary layer physics-informed neural networks (BL-PINN)

We propose to use PINN for solving boundary layer problems with the above perturbation framework, and therefore leverage the hybrid data-driven and model-driven deep learning framework that PINN offers. Details about PINNs could be found in [43]. In the proposed BL-PINN approach, we use separate neural networks to approximate each solution level in the outer and inner expansions and use the matching condition to obtain a consistent solution. An overview of the framework is sketched in Fig. 1. Multiple parallel PINNs are used to represent the different orders of approximation for the inner and outer representations. Each PINN network has its own physics loss function based on the PDE derived for the specified order of approximation and region (inner or outer). The final solution in the inner and outer regions is derived by forming a linear combination of each PINN output weighted by the known gauge functions $\delta_n(\epsilon)$. The final solution is only used in the training process if measurement data is provided and a data loss is defined. Finally, appropriate boundary conditions are imposed for each network and a matching condition is used to ensure the inner and outer solutions are consistent in the overlap region between them. Each neural network representing the outer layer solutions $\phi_n(\mathbf{x})$ and inner layer solutions $\bar{\psi}_n(\xi)$ is optimized using the following loss functions

$$\mathcal{L}^n_{outer}(\mathbf{W}^n_{i,outer}, \mathbf{b}^n_{i,outer}) = \mathcal{L}^n_{phys,outer} + \lambda_b \mathcal{L}^n_{BC,outer} + \lambda_d \mathcal{L}_{data,outer} \,, \tag{4a}$$

$$\mathcal{L}^n_{inner}(\mathbf{W}^n_{i,inner}, \mathbf{b}^n_{i,inner}) = \mathcal{L}^n_{phys,inner} + \lambda_b \mathcal{L}^n_{BC,inner} + \lambda_d \mathcal{L}_{data,inner} \,, \tag{4b}$$

$$\mathcal{L}_{tot} = \sum_n \mathcal{L}^n_{outer} + \sum_n \mathcal{L}^n_{inner} + \lambda_m \sum_n \mathcal{L}^n_{match} \,, \tag{4c}$$

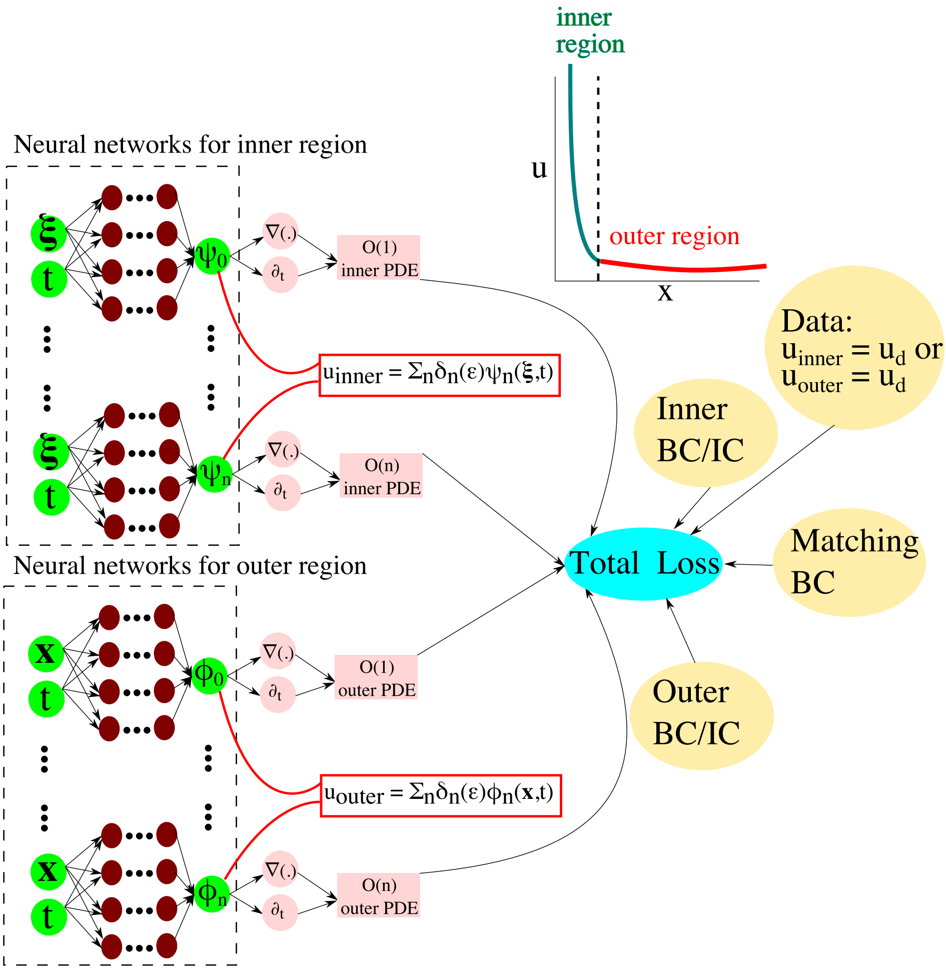

Fig. 1. An overview of the proposed boundary layer physics-informed neural network (BL-PINN) framework is sketched. The network architecture consists of two coupled networks: the inner and outer networks. These inner and outer regions are highlighted in a sample u(x) function shown, which exhibits a boundary layer. The inner and outer parts of BL-PINN provide an asymptotic expansion approximation to the solution in the boundary layer and outside of boundary layer regions, respectively. Each part (inner or outer) consists of multiple parallel PINN networks that each represent a certain order approximation to the solution. The final solution is derived by a combination of these parallel PINN networks. However, the final solution ($\mathbf{u}_{\text{inner}}$ or $\mathbf{u}_{\text{outer}}$) is not needed in the training process unless measurement data are provided and a data loss is needed. Each parallel PINN network is trained based on a PDE that is derived analytically for the desired order of approximation. The matching boundary condition (BC) loss enforces the coupling between the inner and outer networks.

where $n = 1, 2, \ldots$ represent the different orders of the asymptotic expansion solutions, each equipped with appropriate physics $\mathcal{L}^n_{phys}$ and boundary condition $\mathcal{L}^n_{BC}$ loss functions defined based on their domain (inner vs. outer) and order of approximation (n) in ϵ. The match loss function $\mathcal{L}^n_{match}$ is used as the matching condition for the inner and outer neural networks. The total loss $\mathcal{L}_{tot}$ is defined by summing the inner and outer loss functions over their order of approximation together with the matching condition. Finally, if data is available, the data loss function $\mathcal{L}_{data}$ is defined and backpropogated based on the final output produced by a linear composition of all solutions as shown in Fig. 1. The λ hyperparameters are set to weight the contribution of each loss term. A standard stochastic gradient descent algorithm (Adam) is used to find the optimal weights $\mathbf{W}_i$ and biases $\mathbf{b}_i$ for each layer i and each inner/outer network n.

The matching condition will require the $\xi \to \infty$ output of the inner PINNs to match with the $\mathbf{x} \to 0$ output of the corresponding outer PINNs. However, the infinity limit is not possible as neural network inputs should be ideally normalized. To overcome this issue, a new variable $0 < z < 1$ is defined as $z = \frac{\xi}{A}$ and the inner equation is rescaled using this variable. The constant A is set to a sufficiently large value and $\xi \to \infty$ is approximated as $z = 1$. This approach was inspired by the classical similarity solutions in boundary layer theory where an appropriately large value is estimated based on the equations to approximate infinity [56]. Below we discuss the choice of the constant A.

In summary, BL-PINN leverages the observation that the perturbation theory is nothing but a series of differential equations that are solved with appropriate boundary/matching conditions and the solutions are added to form the final solution.

Therefore, one can use different PINN networks to solve each one of these differential equations and subsequently linearly add these predictions to form the final solution.

2.3. Boundary layer test cases

In this section, we explain the different singular perturbation problems that were used to test the proposed BL-PINN approach. In each case, BL-PINN is compared to the original PINN approach (with similar network parameters). Analytical solutions or high-resolution numerical models are considered as the reference for comparison. No data was used ($\lambda_d = 0$) in the problems below with the exception of the inverse problem (test case 5). In all of these examples, ϵ represents a small value that appears in the given equation and leads to boundary layer formation. We treat ϵ as the perturbation parameter. 100 collocation points were uniformly placed (equidistant) in the 1D and 2D problems producing 100 and 10,000 total collocation points, respectively. In the 3D problem (test case 7), 80 points were used in each dimension producing 512,000 total collocation points.

2.3.1. Test case 1: 1D linear advection-diffusion-reaction transport

First, we consider a simple 1D advection-diffusion-reaction equation presented in [9,32]

$$\epsilon \frac{\partial^2 u}{\partial x^2} + (1+\epsilon)\frac{\partial u}{\partial x} + u = 0\,, \tag{5}$$

where ϵ is a small parameter set to 5×10^{-4}, $x \in [0,1]$, and the boundary conditions are given as $u(0) = 0$ and $u(1) = 1$. Similar models known as Friedrichs' boundary layer models are commonly used to illustrate the difficulties associated with modeling viscous flow boundary layers [56]. The above equation could be analytically solved, which will be used for evaluating the PINN solution accuracy:

$$u(x) = \frac{e^{-x} - e^{\frac{-x}{\epsilon}}}{e^{-1} - e^{\frac{-1}{\epsilon}}}\,. \tag{6}$$

An asymptotic analysis of the differential equation (5) in the limit as $\epsilon \to 0$ reveals that the distinguished limit is $\delta(\epsilon) = \epsilon$ based on the dominant balance between terms in the differential equation. The outer problem is then derived by substituting Eq. (2) with the gauge function $\delta(\epsilon) = \epsilon$ into the governing equation. The leading order approximation in the outer region with $\epsilon = 0$ (away from the boundary layer) becomes

$$\frac{\partial u_{outer}}{\partial x} + u_{outer} = 0\,. \tag{7}$$

To derive the inner problem, the gauge function $\delta(\epsilon) = \epsilon$ is used and the stretched variable is defined as $\xi = \frac{x}{\epsilon}$. The leading inner problem reads

$$\frac{\partial^2 u_{inner}}{\partial \xi^2} + \frac{\partial u_{inner}}{\partial \xi} = 0\,. \tag{8}$$

The equation is rescaled to $z = \frac{\xi}{A}$ to make the matching condition possible

$$\frac{1}{A}\frac{\partial^2 u_{inner}}{\partial z^2} + \frac{\partial u_{inner}}{\partial z} = 0\,. \tag{9}$$

The boundary conditions are $u_{outer}(x=1) = 1$ and $u_{inner}(z=0) = 0$, and $u_{inner}(z=1) = u_{outer}(x=0)$ is the imposed matching condition. The parameter A needs to be appropriately selected. A very large parameter will create another undesirable singularly perturbed problem in Eq. (9), whereas a small parameter might not accurately represent infinity. To see how this parameter could be selected, we solve Eq. (8) to obtain $u = Ce^{-\xi} + D$. To approximate $\xi \to \infty$, we need $e^{-\xi} \to 0$. Selecting 1×10^{-4} as the tolerance leads to $e^{-\xi} < 1 \times 10^{-4}$, and $\xi = 10$ is thus sufficient to represent infinity with this tolerance; therefore, $A = 10$ was selected.

The networks had five hidden layers with 60 neurons per layer. $\lambda_b = 1$ and $\lambda_m = 10$ were used, and the learning rate was 1×10^{-4} with 2000 epochs.

2.3.2. Test case 2: nonlinear 1D transport problem

A nonlinear autonomous equation is considered [9]

$$\epsilon \frac{\partial^2 u}{\partial x^2} + 2\frac{\partial u}{\partial x} + e^{u} = 0\,, \tag{10}$$

where $u(0) = u(1) = 0$ are the boundary conditions and $\epsilon = 1 \times 10^{-3}$ was used. With $\epsilon = 0$, the leading order outer problem is

$$2\frac{\partial u_{outer}}{\partial x}+e^{u_{outer}}=0\,. \tag{11}$$

The inner problem is obtained with the $\delta(\epsilon)=\epsilon$ distinguished limit and $z=\frac{\xi}{A}$ rescaling

$$\frac{1}{A}\frac{\partial^2 u_{inner}}{\partial z^2}+2\frac{\partial u_{inner}}{\partial z}=0\,. \tag{12}$$

The neural network parameters were similar to the previous problem and $A=8$ was used here. The corresponding numerical simulation for comparison was performed with a fourth-order finite difference algorithm for boundary value problems [30]. The continuation method [52] was used to enable a solution for a small ϵ.

2.3.3. Test case 3: 2D advection-diffusion transport in Couette flow

Consider the 2D advection-diffusion equation representing high Peclet number mass transport

$$u\frac{\partial c}{\partial x}+v\frac{\partial c}{\partial y}=\epsilon\left(\frac{\partial^2 c}{\partial x^2}+\frac{\partial^2 c}{\partial y^2}\right)\,, \tag{13}$$

where $u=10y$ and $v=0$ are set as the velocity components (Couette flow), $\epsilon=1\times10^{-4}$ is selected as the diffusion coefficient, and the domain is selected as [0,1]×[0,1]. For boundary conditions, $\frac{\partial c}{\partial y}(x, y=0)=-10$ at the bottom wall, $c=0$ at the inlet, and a no-flux Neumann boundary condition at the other boundaries is prescribed. Performing the asymptotic expansion in y gives the following leading order outer problem

$$\frac{\partial c_{outer}}{\partial x}=0\,. \tag{14}$$

The leading inner problem with the distinguished limit $\delta(\epsilon)=\sqrt{\epsilon}$, and inner scaling $z=\frac{\xi}{A}$ becomes

$$u(\sqrt{\epsilon}Az)\frac{\partial c_{inner}}{\partial x}=\frac{1}{A^2}\frac{\partial^2 c_{inner}}{\partial z^2}\,. \tag{15}$$

The Neumann boundary condition at the wall becomes $\frac{\partial c}{\partial z}(x, z=0)=-10A\sqrt{\epsilon}$.

The neural networks had seven hidden layers with 128 neurons per layer. $\lambda_b=10$ and $A=8$ were used and the learning rate was 5×10^{-6} with 2000 epochs and a batch size of 128. The outer solution was simply set to $c=0$ based on Eq. (14) and the boundary conditions.

Finite element method (FEM) simulation was performed in the open-source PDE solver FEniCS to provide benchmark data for comparison. The stabilized SUPG method [11] was implemented, and the mesh had 152,000 triangular elements. To facilitate convergence in the challenging high Peclet number regime, the transport model (Eq. (13)) was treated as a transient problem and was integrated in time until a steady state was reached.

2.3.4. Test case 4: 2D advection-diffusion transport in the double gyre flow

We reconsider the 2D advection-diffusion equation above (Eq. (13)) with a more complicated velocity field. Namely, the double gyre flow [47] is considered, which is a commonly used benchmark problem in chaotic advection studies [6]. The velocity field is defined as

$$u=-\pi B\sin(2\pi x)\cos(\pi y)\,, \tag{16a}$$

$$v=2\pi B\cos(2\pi x)\sin(\pi y)\,, \tag{16b}$$

where $B=-0.1$ and the domain of interest is [0,1]×[0,1]. The diffusion coefficient is set to $\epsilon=1\times10^{-4}$. A Neumann boundary condition with $\frac{\partial c}{\partial y}(x, y=0)=-10$ is imposed at the bottom wall, $c=0$ is used at the left and right boundary, and zero flux is imposed on the top boundary. Similar to the previous test case, the leading order outer problem reads

$$u\frac{\partial c_{outer}}{\partial x}+v\frac{\partial c_{outer}}{\partial y}=0\,. \tag{17}$$

The diffusion term could be kept in the outer problem to improve the solution stability. The leading order inner problem could be derived similar to test case 3 with an additional term due to non-zero vertical velocity as follows:

$$u(x,\sqrt{\epsilon}Az)\frac{\partial c_{inner}}{\partial x}+v(x,\sqrt{\epsilon}Az)\frac{\partial c_{inner}}{\partial z}/(\sqrt{\epsilon}A)=\frac{1}{A^2}\frac{\partial^2 c_{inner}}{\partial z^2}\,. \tag{18}$$

The neural network parameters were the same as test case 3 but with a variable learning rate between 2×10^{-4} and 6×10^{-6} during 65,000 epochs with a batch size of 256. The FEM solution was carried out similar to test case 3 but with a higher resolution mesh (318,000 triangular elements) and without stabilization.

2.3.5. Test case 5: inverse modeling to infer boundary flux in 2D transport

We reconsider the 2D transport problem in test case 3. We assume that the flux boundary condition at the bottom wall is unknown and use six sensors (shown in the results) to measure concentration in the boundary layer and define a data loss for inferring the unknown flux. The sensors were probed based on the FEM solution. The network parameters were set similar to test case 3 with 60000 epochs. $\lambda_d = 10$ was used to incorporate the data measurements into the total loss.

2.3.6. Test case 6: axisymmetric transport in 3D Burgers vortex

In this example, we consider a 3D velocity field. The Burgers vortex is considered as a canonical vortex flow. The Burgers vortex could be derived as an asymptotic steady solution to the momentum equation and represents viscous vortices with axial stretching [42,57]. In cylindrical coordinates (r, θ, x), the velocity field could be written as

$$v_r = -\frac{\gamma}{2} r \tag{19a}$$

$$v_\theta = \frac{\Gamma_0}{2\pi r}\left(1 - e^{-\beta r^2}\right) \tag{19b}$$

$$v_x = \gamma x \,, \tag{19c}$$

where the parameters are set to $\gamma = 0.2$, $\Gamma_0 = 2\pi$, and $\beta = 1$. We consider a cylindrical domain with a radius of 0.5 and a height of x = 0.3. The diffusion coefficient is set to $\epsilon = 1 \times 10^{-4}$ and the Neumann boundary condition at the bottom wall (x=0) is $\frac{\partial c}{\partial x} = -5$. Zero concentration is imposed on the side walls. Due to the symmetric nature of the transport problem, despite the 3D nature of the flow, the advection-diffusion equation could be simplified to a 2D problem in cylindrical coordinates

$$v_r \frac{\partial c}{\partial r} + v_x \frac{\partial c}{\partial x} = \epsilon \left(\frac{\partial^2 c}{\partial r^2} + \frac{1}{r}\frac{\partial c}{\partial r} + \frac{\partial^2 c}{\partial x^2} \right) . \tag{20}$$

The inner and outer problems are derived similar to test case 4. The neural network architecture was similar to test cases 3 and 4. $A = 8$ was used and the batch size was 512. The learning rate varied between 4×10^{-4} and 1×10^{-5} with 32,000 epochs. The FEM solution was performed with a full 3D discretization (4.6M tetrahedral elements) with local boundary layer refinement.

2.3.7. Test case 7: 3D transport near flow separation

In this example, a fully 3D mass transport problem is considered. The velocity field is defined to represent flow around a separation profile. Namely, we consider a saddle type fixed point in wall shear stress (WSS) vector field, which represents flow separation in steady flows [48]. Subsequently, the velocity field near the separation point is defined using a Taylor series expansion. Such topological analysis of fluid flow has been utilized in studying flow separation [48,57] and more recently near-wall mass transport [1,22].

In this example, a 3D box is used to define the domain as [-0.7,0.7]×[-0.3,0.3]×[0,0.3]. The bottom wall (z=0) is considered the separation region. The WSS vector field $\boldsymbol{\tau}$ in this wall is defined as $(\tau_x,\ \tau_y) = (-x + y,\ x - \frac{y}{4})$. Subsequently, using Taylor series expansion of the WSS vector field the velocity field is extrapolated to the rest of the domain [23,1]

$$\mathbf{v}_\pi = \frac{\boldsymbol{\tau} z}{\mu} = \left((-xz + yz)/\mu \ ,\ (xz - \frac{yz}{4})/\mu \right) \tag{21a}$$

$$v_z = -\frac{1}{2\mu} \nabla \cdot \boldsymbol{\tau}\, z^2 = 0.625 z^2/\mu \,, \tag{21b}$$

where $\mathbf{v}_\pi = (v_x . v_y)$ is the 2D velocity vector in the xy plane, v_z is the velocity component normal to this plane, and μ is the dynamic viscosity set to one in this non-dimensional example. The above velocity field is used to solve the 3D advection-diffusion equation where $\frac{\partial c}{\partial z} = -10$ is imposed at z=0 to generate a boundary layer and zero concentration is applied to the lateral walls. The inner and outer problems are derived similar to test case 4 as

$$v_x \frac{\partial c_{outer}}{\partial x} + v_y \frac{\partial c_{outer}}{\partial y} + v_z \frac{\partial c_{outer}}{\partial z} = 0 \tag{22a}$$

$$v_x(x, y, \sqrt{\epsilon} A z) \frac{\partial c_{inner}}{\partial x} + v_y(x, y, \sqrt{\epsilon} A z) \frac{\partial c_{inner}}{\partial y} + v_z(x, y, \sqrt{\epsilon} A z) \frac{\partial c_{inner}}{\partial z} / (\sqrt{\epsilon} A) = \frac{1}{A^2} \frac{\partial^2 c_{inner}}{\partial z^2} \,, \tag{22b}$$

where the diffusion term could be brought back to the outer problem to improve stabilization, and z in Eq. (22b) is the rescaled stretched variable as defined earlier (not to be confused with the physical z coordinates in all the other equations in this Section). The network architecture was similar to previous cases (test cases 3, 4, and 6). $A = 8$ and $\epsilon = 1 \times 10^{-4}$ were used and a large batch size of 8192 was used to enable efficient solution in 3D. The learning rate was varied between 4×10^{-4} and 2×10^{-5} during 24,000 epochs. The FEM simulation was performed with 2.7M tetrahedral elements with local refinement around the boundary layer region.

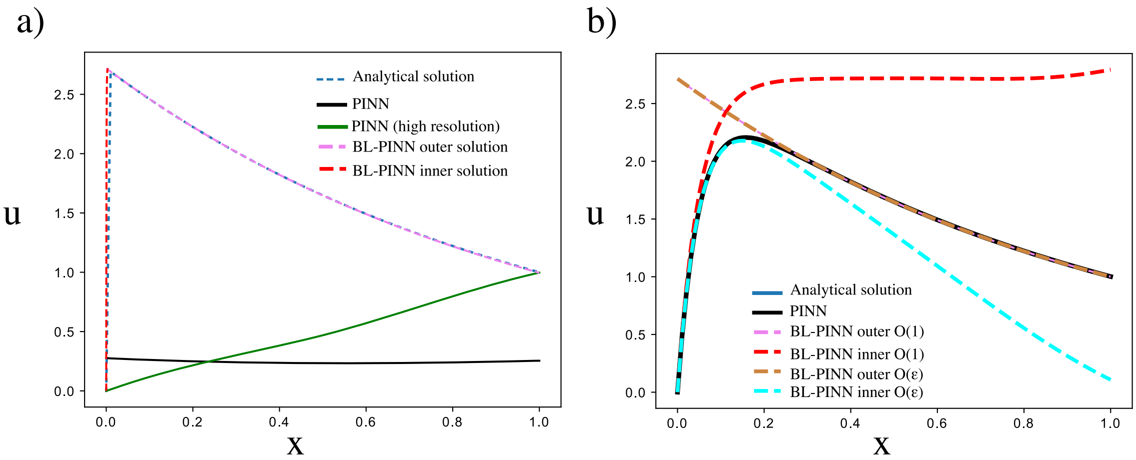

Fig. 2. Test case 1 (linear advection-diffusion-reaction) results are plotted and the inner and outer solutions approximated by BL-PINN are compared to the original PINN approach, a high-resolution local sampling approach, and the analytical solution. a) $\epsilon = 5 \times 10^{-4}$ and only the leading order approximation in BL-PINN is retained. b) $\epsilon = 0.05$ and the O(1) approximation (leading order) as well as the O(ϵ) approximation in BL-PINN are compared. PINN and analytical solutions are on top of each other in this case. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3. Results

The seven test case results are presented in this section. In all cases, only the O(1) networks, corresponding to the leading-order asymptotic solution, were considered in the simulations unless otherwise noted. In addition to comparison to the original PINN method for all test cases, the results are also compared to two other approaches: localized high-resolution clustering of collocation points (test case 1) and XPINN (test case 2).

The test case 1 results (linear advection-diffusion-reaction) are plotted in Fig. 2a. Observe that the traditional PINN approach does not converge to a reasonable solution or capture the singular boundary layer behavior near $x = 0$, whereas the inner and outer BL-PINN approximations match the exact analytical solution very well in their respective regions. An additional original PINN simulation was performed where an additional set of collocation points were seeded inside and in the vicinity of the boundary layer (1000 points). The results show that this high-resolution local sampling approach, which was suggested in prior work [34,39], still cannot find the correct solution. In Fig. 2b, the difference between O(1), leading order approximation, and O(ϵ) approximations is shown. To distinguish between these results, case 1 was repeated with a larger perturbation parameter ($\epsilon = 0.05$). In this case, due to the larger diffusion coefficient, the original PINN approach converges to the exact solution. In BL-PINN, increasing the asymptotic expansion order does not improve the outer solution, however, the O(ϵ) approximation provides notable improvement for the inner solution. Overall, the O(ϵ) approximation provides accurate results in both inner and outer regions but does not offer any advantage over the original PINN approach in this case due to the larger ϵ value, and the correspondingly less severe gradients.

Test case 2 extends the previous problem to a nonlinear differential equation and also presents a comparison with XPINN as shown in Fig. 3. Similar results could be seen where BL-PINN approximates the true solution very well, while the original PINN approach cannot converge to the correct solution. In this case, it could be seen that the original PINN approach seems to be learning a shifted version of only the outer layer solution. The reason for the shifted solution is the imposed $x = 0$ boundary condition, which is where the boundary layer is occurring. XPINN cannot provide much improvement over the original PINN approach. In XPINN, the domain was decomposed into boundary layer and outer regions, and continuity was imposed at the interface. We also investigated sensitivity to the choice of the size of the boundary layer domain of XPINN and confirmed similar results (not shown).

The 2D advection-diffusion transport result for the Couette flow problem (test case 3) is shown in Fig. 4 and Fig. 5. The first figure shows the contour plots of the concentration results. It could be seen that the original PINN approach does not capture the quantitative features in the boundary layer correctly, whereas BL-PINN produces results very similar to the reference FEM solution. To better visualize the quantitative features, the concentration on the bottom wall where the boundary layer is created is plotted in Fig. 5. It could be seen that BL-PINN captures the quantitative behavior much more accurately. Similar to the previous example, the original PINN solution shows a shifted behavior where in this case it predicts the qualitative trend away from $x = 0$ (the leading edge of the boundary layer) and only in a shifted fashion.

A more complicated advection-diffusion transport example is shown in Fig. 6 and Fig. 7 where test case 4 (double gyre flow) results are shown. The velocity vector field is sketched showing the two counterrotating vortices in the double gyre flow. In this example, we are interested in the boundary layer that forms at the bottom wall where the Neumann boundary condition is prescribed. The contour results shown in Fig. 6 show that the inner part of BL-PINN is capable of capturing the quantitative and qualitative behavior in the boundary layer. The original PINN approach does not provide results close to the reference FEM solution (note the different color bar range). In the outer region (outside of the boundary layer at

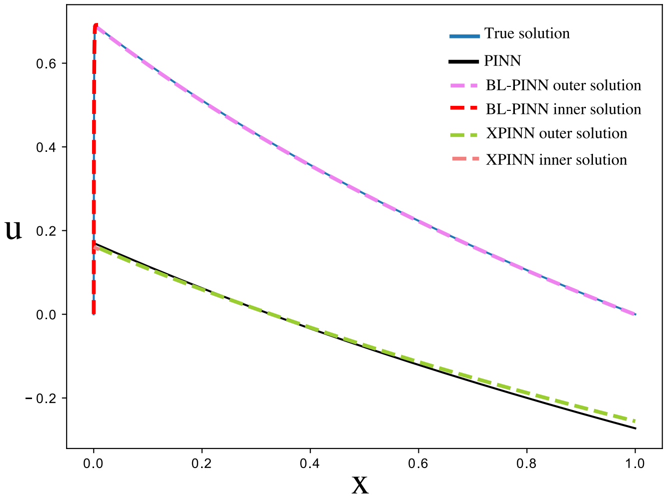

Fig. 3. Test case 2 (nonlinear advection-diffusion-reaction) results are plotted and the inner and outer solutions approximated by BL-PINN are compared to the original PINN approach, XPINN, and the true solution. The inner XPINN solution covers the very thin boundary layer region; however, it cannot discover the true solution and just continues the outer XPINN pattern based on XPINN's interface condition (continuity in solution and its flux).

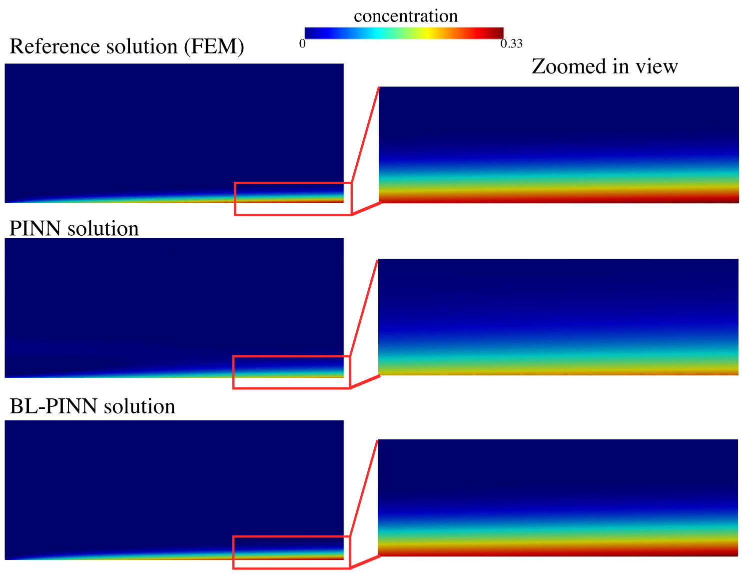

Fig. 4. Test case 3 (2D advection-diffusion in Couette flow) contour results are shown and the BL-PINN approach is compared to the original PINN approach and the reference FEM solution.

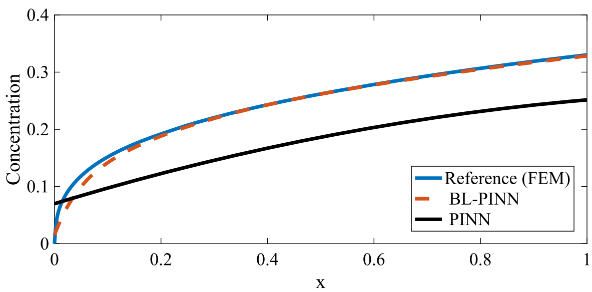

Fig. 5. Test case 3 (2D advection-diffusion in Couette flow) concentration results are plotted at the bottom wall ($y = 0$) where the boundary flux is imposed and the boundary layer is generated. The original PINN, BL-PINN, and reference FEM results are compared.

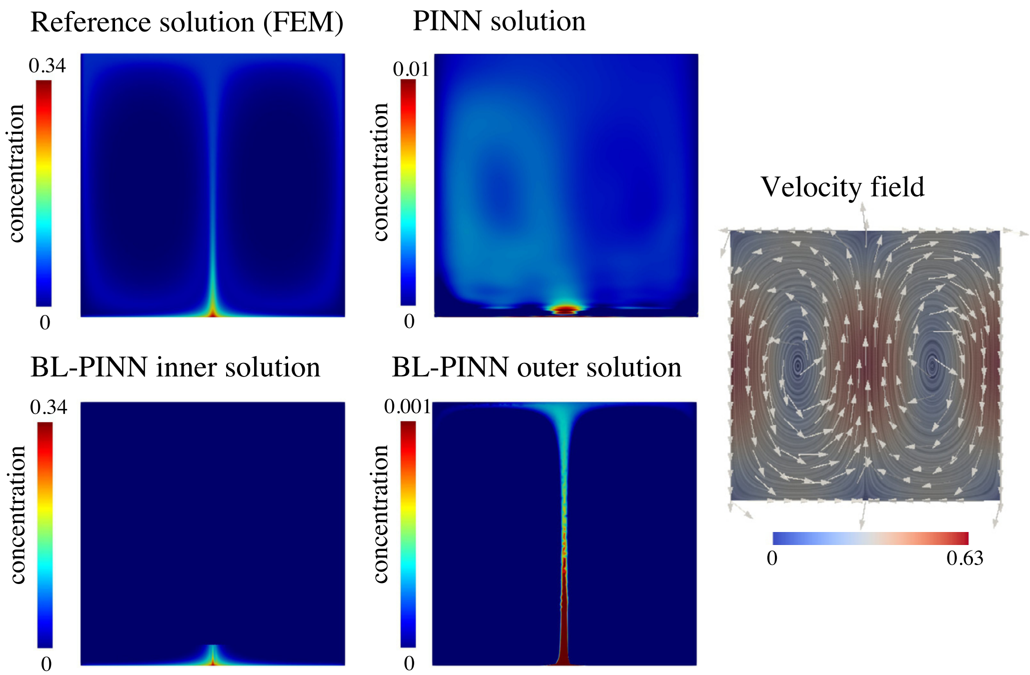

Fig. 6. Test case 4 (2D advection-diffusion in the double gyre flow) contour results are shown and the BL-PINN approach is compared to the original PINN approach and the reference FEM solution. In the BL-PINN panels, the entire solution is shown. However, the inner and outer solutions are only valid near and away from the bottom wall, respectively. To better demonstrate the qualitative behavior, different color bar ranges are used in some cases where the error was higher. The velocity vector field is shown on the right where normalized vector fields are superimposed on top of the streamlines to show the velocity direction.

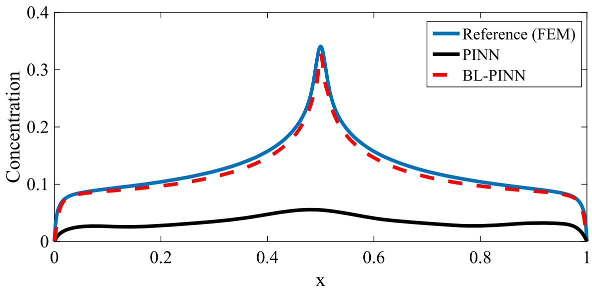

Fig. 7. Test case 4 (2D advection-diffusion in the double gyre flow) concentration results are plotted at the bottom wall ($y = 0$) where the boundary flux is imposed and the boundary layer is generated. The original PINN, BL-PINN, and reference FEM results are compared.

the bottom wall), the problem is more complicated due to the domination of advection. The BL-PINN outer network in this case cannot capture quantitative concentration patterns in the outer region and only captures the qualitative behavior. On the other hand, the original PINN approach completely misses the qualitative behavior and cannot find even a qualitatively meaningful solution. Interestingly, the outer part of BL-PINN can provide a correct quantitative approximation near the interface with the inner part of BL-PINN, and therefore the matching boundary condition is satisfied, which helps produce correct boundary layer results by the inner network. This is further shown in Fig. 7 where the concentration is plotted at the bottom wall. We can see that BL-PINN provides a very accurate quantitative prediction of the concentration pattern, while PINN cannot approximate the correct quantitative pattern.

The inverse problem (test case 5) results are shown in Fig. 8. The right panel shows the placement of the measurement sensors (the six gray spheres) within the boundary layer. The left panel shows the learned flux boundary condition, $\frac{\partial c}{\partial y}(y = 0)$, during different epochs of the deep learning training. It is seen that BL-PINN converges to the ground truth flux that was used to generate the data, whereas the original PINN approach cannot converge to the ground truth flux.

In the last two test cases (6 and 7), the BL-PINN approach did not provide accurate results in the outer region (similar to the double gyre flow problem), and therefore these results are not included. It should be highlighted that the boundary layer is the region of interest in our work, and therefore this is not a concern. In test case 7, we further substantiate this by demonstrating the success of a BL-PINN approach inspired by surface transport models where we completely omit the outer BL-PINN network in our approach. We further discuss these observations in the Discussion. The Burgers vortex (test

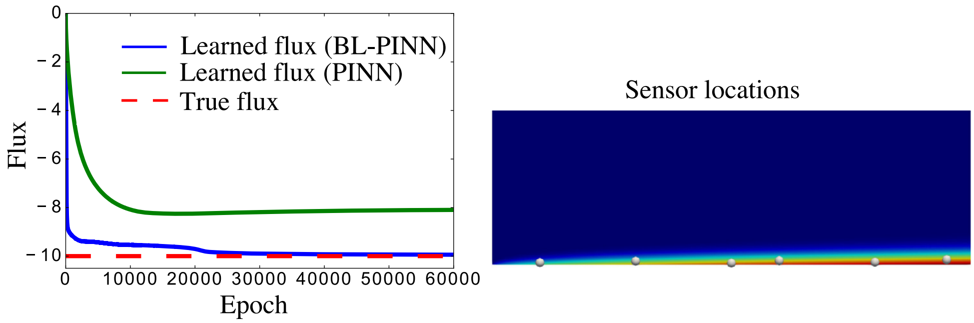

Fig. 8. Test case 5 (inverse modeling of flux in the Couette flow transport problem) results are shown on the left panel. The learned flux versus deep learning epochs is shown for the BL-PINN and original PINN approaches along with the true flux. The right panel demonstrates the location of the measurement sensors within the boundary layer that were used to define the data loss. The gray spheres mark the sensor locations.

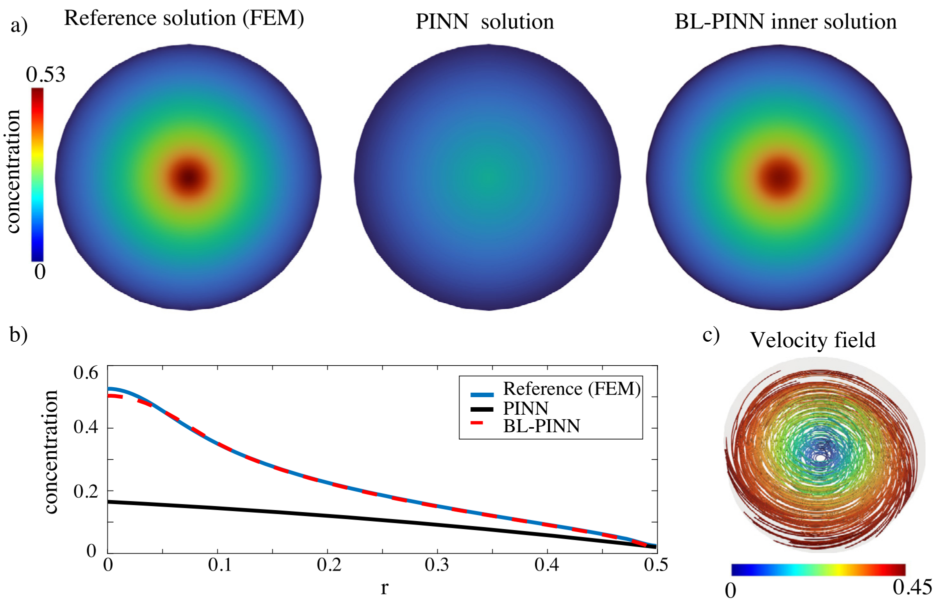

Fig. 9. Test case 6 (axisymmetric advection-diffusion in 3D Burgers vortex) results are shown. a) The BL-PINN approach is compared to the original PINN and reference FEM solution. The x=0 plane where the boundary layer forms is shown. b) The concentration results are quantitatively compared at the x=0 plane for different radial positions. c) The 3D velocity streamlines are shown in the cylindrical region of interest and are colored based on velocity magnitude.

case 6) results are shown in Fig. 9. We can see that the BL-PINN approach leads to considerable improvement in the wall concentration results compared to the original approach.

Finally, the results for test case 7 (3D transport around flow separation) are shown in Fig. 10. The original PINN cannot capture the qualitative (Fig. 10a) or quantitative (Fig. 10b) patterns. To assist with qualitative visualization of the patterns, the maximum color bar range for the original PINN approach is set to 0.03 and for the other approaches, this is 1.04. In this test case, we also present a new BL-PINN approach where we just consider the inner network and at the matching condition set zero concentration for the inner network. This approach was inspired by recent work on near-wall transport in the context of biomedical flows [25,1,22] where it has been shown that in thin concentration boundary layer problems one could reduce the problem to a surface transport model based on WSS and near-wall velocity, and therefore ignore transport away from the wall with minimal loss in accuracy for most problems. Interestingly, our results here demonstrate that the BL-PINN approach with just an inner network (inspired by surface transport models) produces very accurate results. As a relevant note, the region of high surface concentration (red region) corresponds to the unstable manifold of the WSS vector field. The unstable WSS manifold, also known as the attracting WSS Lagrangian coherent structure, has been shown to dominate near-wall concentration patterns in complex 3D problems [1,22,2].

4. Discussion

In this work, boundary layer PINN (BL-PINN) was proposed for solving thin boundary layer problems. One- and two-dimensional benchmark problems were presented as proof-of-concept where it was shown that BL-PINN overcomes PINN

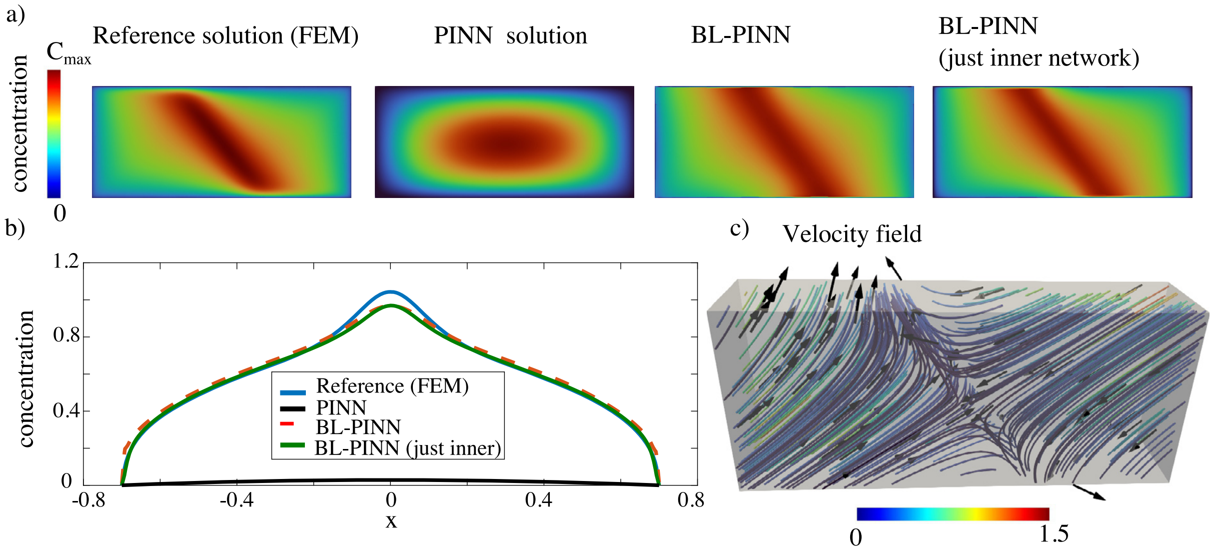

Fig. 10. Test case 7 (3D transport near flow separation) results are shown. a) The BL-PINN approach is compared to the original PINN and reference FEM solution. Additionally, a de-coupled BL-PINN solution (just the inner network) is shown where the outer network is not included during training and is replaced with a zero concentration matching condition. In the color bar, c_{max} is 0.03 for the original PINN panel and 1.04 for the other approaches. The z=0 plane where the boundary layer forms is shown. b) The concentration results are quantitatively compared at the z=0 plane for a line passing through the middle of the plane (-0.7 $< x <$ 0.7, y=0). c) The 3D velocity streamlines are shown in the cylindrical region of interest and are colored based on velocity magnitude. Normalized velocity vectors are also plotted to show the flow direction.

limitations in solving thin boundary layer problems. As illustrated here, only a small number of asymptotic basis functions is necessary to accurately capture the solution using BL-PINN. This is in marked contrast to traditional numerical methods that have increasing difficulty and require more small elements to capture high-gradient regions of a solution. It was also shown that prior extensions of PINN (XPINN and local collocation point clustering) were not able to resolve thin boundary layers.

Solutions of physical problems that contain large gradients give rise to numerical difficulties when solved using traditional numerical methods, such as FEM. Typically, such problems contain a parameter that becomes very small or very large, in which case perturbation methods are well suited to deciphering the solution's dependence on this parameter. Asymptotic basis functions are obtained directly from the governing differential equation. As such, they contain physical information about how the system depends on the small or large parameter. In fact, the accuracy of the asymptotic basis functions increases as the small parameter approaches zero (or the large parameter approaches infinity) or as additional terms are included in the expansion. This is in contrast to numerical methods attempting to capture the same solution. Their primary limitation is that they only apply when the parameter is very small or large. Consequently, incorporating these asymptotic basis functions into more general techniques holds great promise in combining the best of both into a robust solution framework that takes advantage of the flexibility of general methods and the model-driven, as opposed to data-driven, approach to capturing abrupt behavior in a solution. Galerkin projection with asymptotic basis functions is one approach for accomplishing this [15], and PINN offers an alternative framework. Such reduced-physics models (RPM) have the potential to dramatically reduce the computational requirements necessary for solving physical problems containing large gradients as compared to traditional numerical methods.

Whether for use in a projection method or PINN, the ideal basis functions would contain as much information as possible about the system and accommodate solutions for a range of parameter values. This is precisely what asymptotic basis functions offer. Perturbation (asymptotic) methods comprise a set of techniques for obtaining the solution in terms of an asymptotic series for problems having a very small or very large parameter. These methods allow for determination of the dependence of the system of the small or large parameter in a formal manner from the governing equation(s) itself without any need for data from the system. This dependence is contained in the gauge functions, which unlike most ROM approaches captures the system's dependence on the parameter.

BL-PINN shares similarities with other extensions of PINN and yet provides clear advantages for boundary layer problems. Similar to XPINN and cPINN, BL-PINN is based on a domain decomposition implementation of PINN where separate neural networks are used in different regions and matched at the interface. However, unlike the arbitrary nature of XPINN and cPINN, BL-PINN decomposes the domain into an inner region (boundary layer) and an outer region in a systematic fashion inspired by the perturbation theory. Additionally, the rescaling of the equation within the boundary layer enables an accurate solution to thin boundary layers, which is not possible with prior approaches. Similar to the recently proposed sparse, physics-based, and partially interpretable neural networks (SPINN) [45], BL-PINN leverages rescaling of the input variables to define the stretched variable ξ (similar to the mesh encoding layer in SPINN) and relies on parallel neural networks and their linear combination to build the solution. Therefore, similar to SPINN, BL-PINN is partially interpretable. However, BL-PINN extends SPINN's interpretability since its design is based on asymptotic expansions and therefore in the context of asymptotic basis functions [15], BL-PINN could be interpreted as a physics-based reduced-order model representation with PINN where parametric dependence is naturally considered in its design. In theory, defining kernels that represent boundary layer behavior (similar to FEM enrichment of basis functions [10]) could be implemented in SPINN for modeling thin boundary layers, however, the exponential nature of such kernels in boundary layers poses a challenge for effective training of the neural networks.

A key advantage of BL-PINN is that it becomes more accurate as the perturbation parameter becomes smaller, and therefore it is suitable for thin boundary layer problems. Interestingly, this is in contrast with existing PINN methods that lose accuracy as the perturbation parameter decreases. Another advantage of BL-PINN compared to other PINN approaches is its natural incorporation of the perturbation parameter (e.g., diffusion coefficient) into the solution. That is, one can re-evaluate the solution without retraining with new parameters. In addition, BL-PINN can add parallel networks as higher order approximations to the solution instead of increasing the degrees of freedom in each network. Each of these higher order approximation networks is trained based on a different equation and could have an arbitrary architecture independent of the other networks. This could be somewhat compared to p-refinement in finite element method as opposed to an h-refinement analogy where one would use more collocation points. One disadvantage of BL-PINN is the higher computational cost. For instance, in most examples shown in this paper, two neural networks (inner and outer) were used to approximate the solution. However, similar to XPINN, these neural networks could possess independent architectures and accuracy based on the region of interest (inner vs outer). In terms of computational cost, BL-PINN requires at least two neural networks to be trained (more networks if higher order approximation is required), and therefore has roughly twice the computational cost of PINN for the same number of epochs. Nevertheless, one has the freedom to reduce the outer network size to improve computational cost. For instance, in the limit where the outer network is dropped (Fig. 10), BL-PINN will just need to train one neural network similar to PINN.

An interesting observation in our results was that BL-PINN was capable of finding accurate surface concentration patterns in the boundary layer (our region of interest) even without producing necessarily accurate results in the outer region. While this might be surprising at first, our group has previously shown similar results in the context of high Peclet and high Schmidt number mass transport problems where thin boundary layers are formed [1,22]. That is, such mass transport problems could be reduced to a surface transport problem where the surface concentration patterns are determined by the WSS (a scale of near-wall velocity) vector field. To further investigate this scenario, we performed a simulation in test case 7 where we only considered the inner neural network and at the matching interface forced the neural network to be equal to zero (instead of coupling it to the outer network). This could be perceived as a near-wall transport model in PINN where we are just studying transport within the boundary layer and assuming the outer region to have minimal influence on the results. Interestingly, Fig. 10 shows promising results for this approach where the surface concentration patterns are very similar to the original BL-PINN approach. We should highlight that solving high Peclet mass transport problems even with well established numerical methods such as finite element method is challenging and it is not surprising to see inaccurate PINN results. For instance, various stabilization methods have been proposed in the finite element literature for overcoming these numerical difficulties [11,17,24].

There are several areas where our study could be improved. Compared to the original PINN approach, BL-PINN only demonstrates significant improvement once the boundary layer thickness is sufficiently reduced, i.e. for small ϵ. An example could be seen in Fig. 2b where the original PINN can solve the problem due to the boundary layer size. Due to this reason, we did not present thin boundary layer problems in the Navier-Stokes equations. If the momentum boundary layer thickness is sufficiently reduced (Reynolds number increased), transition to turbulence will occur. Therefore, special treatment of turbulence within PINN will be needed [20]. In the double gyre flow, BL-PINN could not provide quantitatively accurate concentration patterns in the outer region (Fig. 6). This is a well-known problem in advection-dominated transport modeling with PINN and could be improved with other approaches such as curriculum learning [31]. Alternatively, a hybrid FEM-PINN approach [37] could be developed where a traditional numerical solver such as FEM solves the outer region. Interestingly, BL-PINN is capable of correctly resolving the boundary layer region as well as the interface, however, it struggles to find the correct solution in the outer region where the original advection-diffusion equation is solved without any special treatment. Finally, we demonstrated an example of inverse modeling with BL-PINN (test case 5). More complicated inverse modeling examples such as finding velocity fields from concentration [44] could be investigated for boundary layers in future work.

5. Conclusion and future directions

We presented BL-PINN, a new theory-guided/model-driven extension of PINN for solving thin boundary layer problems. In the benchmark problems investigated, BL-PINN demonstrated excellent results and significantly outperformed prior PINN approaches, which could not provide any meaningful results for solutions containing large gradients. BL-PINN was designed based on asymptotic expansions and singular perturbation theory, and therefore the designed network is partially interpretable. Finally, thanks to the analytical incorporation of the perturbation parameter in asymptotic expansions, BL-PINN naturally incorporates the perturbation parameter of interest (e.g., diffusion coefficient) and does not need to be retrained during parametric evaluations.

There are several additional problems for which BL-PINN could potentially be utilized. DeepONets [33] and physics-informed DeepONets [54] have been recently introduced for learning operators and parametric solutions. Theory-guided and model-driven designs similar to BL-PINN could be used to facilitate parametric learning of problems where variation in parameters leads to extreme behavior in the solution and large gradients. Boundary layer control is another application area where flow measurement and data-driven modeling within boundary layers are necessary [4,8]. Unsteady boundary layers could occur for systems of differential equations with multiscale temporal behavior, where the solution rapidly changes in time [51,32]. BL-PINN could be applied to such dynamical systems problems. Characterizing multiple time-scale behavior in chaotic dynamical systems with perturbation methods is another relevant example [36]. Singular perturbation problems also occur in systems of reaction-diffusion or advection-diffusion-reaction equations that are commonly used in modeling the spatiotemporal dynamics of disease [41]. Finally, similar singular perturbation methods could be used in modeling low Reynolds number hydrodynamics [35].

CRediT authorship contribution statement

Amirhossein Arzani: Conceptualization, Funding acquisition, Methodology, Software, Writing – original draft, Writing – review & editing. **Kevin W. Cassel:** Methodology, Writing – review & editing. **Roshan M. D'Souza:** Funding acquisition, Writing – review & editing.

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.

Data availability

Codes available on https://github.com/amir-cardiolab/BL-PINN/

Acknowledgement

This work was supported by the National Science Foundation ECCS-CCSS program (Award No. 2103434 and 2103560) and an NSF CAREER award (Award No. 2247173).

References

[1] A. Arzani, A.M. Gambaruto, G. Chen, S.C. Shadden, Lagrangian wall shear stress structures and near-wall transport in high-Schmidt-number aneurysmal flows, J. Fluid Mech. 790 (2016) 158–172.
[2] A. Arzani, A.M. Gambaruto, G. Chen, S.C. Shadden, Wall shear stress exposure time: a Lagrangian measure of near-wall stagnation and concentration in cardiovascular flows, Biomech. Model. Mechanobiol. 16 (3) (2017) 787–803.
[3] A. Arzani, J.X. Wang, R.M. D'Souza, Uncovering near-wall blood flow from sparse data with physics-informed neural networks, Phys. Fluids 33 (7) (2021) 071905.
[4] S. Bagheri, L. Brandt, D.S. Henningson, Input–output analysis, model reduction and control of the flat-plate boundary layer, J. Fluid Mech. 620 (2009) 263–298.
[5] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, et al., 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), 2019.
[6] S. Balasuriya, N.T. Ouellette, I.I. Rypina, Generalized Lagrangian coherent structures, Physica D 372 (2018) 31–51.
[7] H. Bararnia, M. Esmaeilpour, On the application of physics informed neural networks (PINN) to solve boundary layer thermal-fluid problems, Int. Commun. Heat Mass Transf. 132 (2022) 105890.
[8] B.A. Belson, O. Semeraro, C.W. Rowley, D.S. Henningson, Feedback control of instabilities in the two-dimensional Blasius boundary layer: the role of sensors and actuators, Phys. Fluids 25 (5) (2013) 054106.
[9] C.M. Bender, S.A. Orszag, Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory, vol. 1, Springer Science & Business Media, 1999.
[10] R. Borker, C. Farhat, R. Tezaur, A discontinuous Galerkin method with Lagrange multipliers for spatially-dependent advection–diffusion problems, Comput. Methods Appl. Mech. Eng. 327 (2017) 93–117.
[11] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1–3) (1982) 199–259.
[12] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (2020) 477–508.
[13] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, Acta Mech. Sin. (2022) 1–12.
[14] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, J. Heat Transf. 143 (6) (2021).
[15] K.W. Cassel, Projection-based model reduction using asymptotic basis functions, in: International Conference on Computational Science, Springer, 2019, pp. 465–478.
[16] L. Chen, K. Asai, T. Nonomura, G. Xi, T. Liu, A review of Backward-Facing Step (BFS) flow mechanisms, heat transfer and control, Therm. Sci. Eng. Prog. 6 (2018) 194–216.
[17] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, Comput. Methods Appl. Mech. Eng. 156 (1–4) (1998) 185–210.
[18] T. de Wolff, H. Carrillo, L. Martí, N. Sanchez-Pi, Towards optimally weighted physics-informed neural networks in ocean modelling, arXiv preprint, arXiv:2106.08747, 2021.

[19] V. Dwivedi, B. Srinivasan, Physics informed extreme learning machine (PIELM)–a rapid method for the numerical solution of partial differential equations, Neurocomputing 391 (2020) 96–118.
[20] H. Eivazi, M. Tahani, P. Schlatter, R. Vinuesa, Physics-informed neural networks for solving Reynolds–averaged Navier–Stokes equations, arXiv preprint, arXiv:2107.10711, 2021.
[21] P. Erhard, D. Etling, U. Muller, U. Riedel, K.R. Sreenivasan, J. Warnatz, Prandtl-Essentials of Fluid Mechanics, vol. 158, Springer, 2010.
[22] A. Farghadan, A. Arzani, The combined effect of wall shear stress topology and magnitude on cardiovascular mass transport, Int. J. Heat Mass Transf. 131 (2019) 252–260.
[23] A.M. Gambaruto, D.J. Doorly, T. Yamaguchi, Wall shear stress and near-wall convective transport: comparisons with vascular remodelling in a peripheral graft anastomosis, J. Comput. Phys. 229 (14) (2010) 5339–5356.
[24] K.B. Hansen, A. Arzani, S.C. Shadden, Finite element modeling of near-wall mass transport in cardiovascular flows, Int. J. Numer. Methods Biomed. Eng. 35 (1) (2019) e3148.
[25] K.B. Hansen, S.C. Shadden, A reduced-dimensional model for near-wall transport in cardiovascular flows, Biomech. Model. Mechanobiol. 15 (3) (2016) 713–722.
[26] Q. He, A.M. Tartakovsky, Physics-informed neural network method for forward and backward advection-dispersion equations, Water Resour. Res. 57 (7) (2021) e2020WR029479.
[27] A.D. Jagtap, G.E. Karniadakis, 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) (2020) 2002–2041.
[28] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, Comput. Methods Appl. Mech. Eng. 365 (2020) 113028.
[29] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[30] J. Kierzenka, L.F. Shampine, A BVP solver based on residual control and the Maltab PSE, ACM Trans. Math. Softw. 27 (3) (2001) 299–316.
[31] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M.W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, Adv. Neural Inf. Process. Syst. 34 (2021).
[32] J.N. Kutz, Advanced differential equations: asymptotics & perturbations, arXiv preprint, arXiv:2012.14591, 2020.
[33] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (3) (2021) 218–229.
[34] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Eng. 360 (2020) 112789.
[35] H. Masoud, H.A. Stone, The reciprocal theorem in fluid dynamics and transport phenomena, J. Fluid Mech. 879 (2019).
[36] K.D. Mease, U. Topcu, E. Aykutluğ, M. Maggia, Characterizing two-timescale nonlinear dynamics using finite-time Lyapunov exponents and subspaces, Commun. Nonlinear Sci. Numer. Simul. 36 (2016) 148–174.
[37] S.K. Mitusch, S.W. Funke, M. Kuchta, Hybrid FEM-NN models: combining artificial neural networks with the finite element method, J. Comput. Phys. 446 (2021) 110651.
[38] R. Mojgani, M. Balajewicz, P. Hassanzadeh, Lagrangian PINNs: a causality-conforming solution to failure modes of physics-informed neural networks, arXiv preprint, arXiv:2205.02902, 2022.
[39] M.A. Nabian, R.J. Gladstone, H. Meidani, Efficient training of physics-informed neural networks via importance sampling, Comput.-Aided Civ. Infrastruct. Eng. 36 (8) (2021) 962–977.
[40] R.E. O'Malley Jr, Singular perturbation theory: a viscous flow out of Göttingen, Annu. Rev. Fluid Mech. 42 (2010) 1–17.
[41] A.V. Panfilov, H. Dierckx, V. Volpert, Reaction–diffusion waves in cardiovascular diseases, Physica D 399 (2019) 1–34.
[42] R.L. Panton, Incompressible Flow, John Wiley & Sons, 2006.
[43] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[44] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, Science 367 (6481) (2020) 1026–1030.
[45] A.A. Ramabathiran, P. Ramachandran, SPINN: sparse, physics-based, and partially interpretable neural networks for PDEs, J. Comput. Phys. 445 (2021) 110600.
[46] W. Schoppa, F. Hussain, A large-scale control strategy for drag reduction in turbulent boundary layers, Phys. Fluids 10 (5) (1998) 1049–1051.
[47] S.C. Shadden, F. Lekien, J.E. Marsden, Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows, Physica D 212 (3–4) (2005) 271–304.
[48] A. Surana, O. Grunberg, G. Haller, Exact theory of three-dimensional flow separation. Part 1. Steady separation, J. Fluid Mech. 564 (2006) 57–103.
[49] B. Thwaites, Approximate calculation of the laminar boundary layer, Aeronaut. Q. 1 (3) (1949) 245–280.
[50] M. Van Dyke, Perturbation methods in fluid mechanics, NASA STI/Recon Technical Report A 75, 46926, 1975.
[51] F. Verhulst, Methods and Applications of Singular Perturbations, Springer, 2005.
[52] V.G. Vetekha, Parameter continuation method for ordinary differential equations, in: Proceedings of the Second ISAAC Congress, Springer, 2000, pp. 737–742.
[53] H. Wang, R. Planas, A. Chandramowlishwaran, R. Bostanabad, Train once and use forever: Solving boundary value problems in unseen domains with pre-trained deep learning models, arXiv e-prints, arXiv–2104, 2021.
[54] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, Sci. Adv. 7 (40) (2021), eabi8605.
[55] S. Wang, H. Wang, P. Perdikaris, On the eigenvector bias of Fourier feature networks: from regression to solving multi-scale PDEs with physics-informed neural networks, Comput. Methods Appl. Mech. Eng. 384 (2021) 113938.
[56] F.M. White, Viscous Fluid Flow, vol. 3, McGraw-Hill, New York, 2006.
[57] J.Z. Wu, H.Y. Ma, M.D. Zhou, Vorticity and Vortex Dynamics, Springer Science & Business Media, 2007.
[58] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. 425 (2021) 109913.