Proceedings of the 1st international Symposium on AI and Fluid Mechanics
Paper No S10 P4

Solving high-dimensional parametric flow problems around airfoils using neural network solvers

[1]Wenbo Cao*; [1]Weiwei Zhang

[1] *School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China*

Abstract

Recent years have witnessed increasing research interests of physics-informed neural networks (PINNs) in solving forward, inverse, and parametric problems governed by partial differential equations (PDEs). Despite their promise, PINNs still face significant challenges in many scenarios due to ill-conditioning, especially in the context of flow around airfoils.

We propose two methods to enhance the solution of flow around airfoils. First, we construct a controlled system with an adjustable condition number of the Jacobian matrix, validating that the ill-conditioning of PINNs is significantly associated with the Jacobian matrix. As the condition number of the Jacobian matrix decreases, PINNs achieve faster convergence and higher accuracy. Based on this understanding and as a natural extension of the controlled system, we propose time-stepping-oriented neural network (TSONN), which transforms the ill-conditioning optimization problem of PINNs into a series of well-conditioned sub-optimization problems. Second, we combine neural network solvers with mesh transformation, using neural network to learn the flow in the uniform computational space instead of physical space. Mesh transformation is employed to amplify the region near the airfoil by using an appropriate mesh distribution, thereby facilitating the neural network's ability to capture large-gradient flow and thin boundary layers.

Using these methods, we successfully solve inviscid and high-Reynolds-number flows around airfoils. Furthermore, we address a parametric problem involving flow conditions and airfoil shapes, covering nearly all laminar flow scenarios in engineering. The model is trained on hundreds of millions of flow conditions and airfoil shapes, resulting in a surrogate model with strong generalization capability that does not require labeled data. This highlights its potential for high generalizability and cost-effectiveness in solving high-dimensional parametric problems and surrogate modeling.

Keywords: PINNs; TSONN; Parametric problem; High-dimensional; Surrogate modeling.

1 Introduction

Neural network solvers, such as physics-informed neural networks (PINNs) [1], the deep Ritz method [2], and the deep Galerkin method [3], have revolutionized PDE solving by optimizing network parameters to minimize PDE losses while enforcing boundary conditions or fitting data labels. Various PINN-like methods have since emerged, achieving remarkable success in computational science and engineering. One of the key advantages of these solvers is their efficiency in solving parametric problems. Instead of repeatedly solving the PDE for different conditions, they learn a unified mapping that directly predicts flow fields under varying flow conditions, geometries, or initial conditions, making them a powerful tool for surrogate modeling.

Despite these advantages, neural network solvers still face challenges, particularly in solving airfoil flows. For instance, while PINNs can handle inviscid flow around a cylinder, they fail for airfoils [4]. Moreover, most studies focus on low-Reynolds-number cases, limiting practical applications. In this study, we analyze the ill-conditioning of PINNs and propose the TSONN method for mitigation. Additionally, we integrate mesh transformation to better capture steep gradients and thin boundary layers near the airfoil.

2 Techniques for Enhancing Neural Network Solvers

2.1 An analysis of ill-conditioning in PINNs

The ill-conditioning of PINNs significantly limits their application in complex engineering problems. This study examines PINNs' ill-conditioning, beginning with an analysis of its presence in traditional numerical methods. In these methods, solving the dynamic system $f(q) = 0$ involves discretizing it into a large system $\boldsymbol{f}(\boldsymbol{q}) = 0$, where the

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn

Jacobian matrix $J = \partial \boldsymbol{f} / \partial \boldsymbol{q}$ characterizes key properties of the PDE system. The eigenvalues of J reflect the stability of the system, while its condition number serves as an indicator of ill-conditioning, which can impede convergence in iterative methods due to the matrix's spectral properties.

In the context of PINNs, obtaining and analyzing the explicit Jacobian matrix is challenging, hindering the connection between convergence issues and the ill-conditioning of the Jacobian matrix. To address this, we construct a controlled system that modifies the Jacobian's eigenvalues by adding a linear forcing term based on control theory. The modified system is written as $f_c(q) = f(q) - \gamma(q - q_s) = 0, \gamma > 0$, where q_s is the solution of the system, and γ is the gain. The controlled system shares the same solution as the original but with its Jacobian eigenvalues shifted by the gain. As the gain increases, the eigenvalues move further negative, improving the system's conditioning. By observing the convergence and accuracy of PINNs at different gains, we establish the relationship between PINNs' ill-conditioning and the Jacobian matrix.

We consider a one-dimensional convection problem $u_t + u_x = 0$, a well-conditioned problem that PINNs can solve effectively. By solving controlled systems with varying gains using PINNs, we observe how gains affect convergence. Figure 1 (a) and (b) show the convergence history for systems with different positive and negative gains. Table 1 presents the condition numbers of the Jacobian matrices in FDM, offering a rough estimation for PINNs. We find that as positive gains increase and the condition number decreases, PINNs converge faster. However, for positive gains, rapid convergence is driven by two factors: the linear forcing term alleviates the Jacobian's ill-conditioning, and it directly drives q to approach q_s. These factors are intertwined and cannot be easily separated. In contrast, for negative gains, the forcing term worsens the ill-conditioning (Table 1), leading to opposing effects. As shown in Figure 1(b), increasing the gain magnitude from 0 to 5 worsens the Jacobian's ill-conditioning, causing larger convergence errors, although the forcing term drives q to approach q_s with greater weight. When the gain magnitude becomes very large, the system behaves like a supervised learning task, resulting in minimal error. These results highlight the negative impact of ill-conditioning on PINNs' convergence.

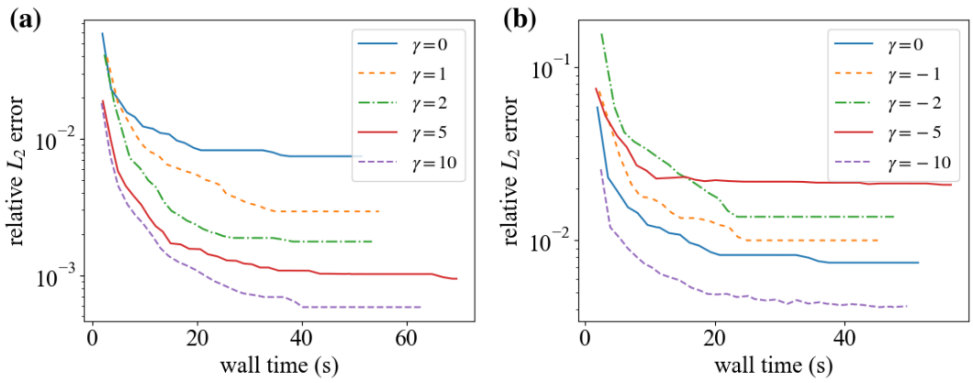

Figure 1. Convergence history of controlled systems with (a) positive gains and (b) negative gains.

Table 1. Condition numbers of Jacobian matrices for controlled systems with different gains, computed in FDM.

gain	0	1	2	5	10
condition number	129	124	119	107	90
gain	/	-1	-2	-5	-10
condition number	/	135	140	160	204

2.2 Time-stepping-oriented neural network

As observed in the previous subsection, mitigating the ill-conditioning of the Jacobian matrix is crucial to enabling PINN to solve complex problems. In fact, the controlled system not only provides a visualization of the ill-conditioning in PINNs but also presents a solution. The key challenge lies in how to substitute known quantities for q_s in the controlled system, as it is always not available in practice. A simple strategy is to substitute the output $q_n = \hat{q}(\cdot; \boldsymbol{\theta}_n)$

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn

of the neural network at the *n*th optimization step for q_s, yielding $f(q)-\gamma(q-q_n)=0$. For the initial condition operator or the Dirichlet boundary condition operator, q_n can be directly replaced by the known target value. We minimize the residual of the equation, referred to as the inner iteration. Once it achieves appropriate convergence after *K* iterations, we proceed to the next optimization step and update q_n with the latest network output, referred to as the outer iteration. Algorithm 1 provides detailed steps. We propose using the LBFGS optimizer and setting the maximal number of iterations per optimization step to *K*, thereby easily implementing inner iterations. At each outer iteration, we restart the optimizer to adapt to changes of the loss function, while randomly resampling collection points. Algorithm 1 is referred to as time-stepping-oriented neural network because the equation $f(q)-\gamma(q-q_n)=0$ can be readily transformed into the implicit pseudo time-stepping scheme $(q-q_n)/\Delta\tau=f(q)$, where the gain γ is equivalent to $1/\Delta\tau$. Thus, TSONN essentially decomposes the originally ill-conditioned problem of PINNs into a series of well-conditioned sub-problems over given pseudo time intervals.

Algorithm 1: Time-stepping-oriented neural network (TSONN)

Input: Initial $\boldsymbol{\theta}$, outer iterations *N*, inner iterations *K*, pseudo time step $\Delta\tau$.

1: for $n=1,2,\cdots,N$ do
2: (a). Randomly sample collocation points and initialize an LBFGS optimizer with *K* iterations per step.
3: (b). Get the state $\boldsymbol{q}_n=\boldsymbol{q}(\cdot;\boldsymbol{\theta})$
4: (c). Perform an optimization step using the optimizer to minimize the loss $\mathrm{L}(\boldsymbol{\theta})=\left\|\boldsymbol{f}(\boldsymbol{q}(\cdot;\boldsymbol{\theta}))-(\boldsymbol{q}(\cdot;\boldsymbol{\theta})-\boldsymbol{q}_n)/\Delta\tau\right\|^2$
5: end

Output: $\hat{q}(\cdot;\boldsymbol{\theta})$

2.3 Mesh Transformation Technique

In aerodynamic flow problems, large computational domains are used to handle far-field boundary conditions, but flow gradients are concentrated near the aircraft, making it difficult for neural networks to capture variations. This work combines neural network solvers with mesh transformation, utilizing the neural network to learn the flow within a uniform computational space rather than the physical space, as shown in Figure 2. The mesh transformation amplifies the airfoil region, enabling the network to capture large gradients and thin boundary layers within the computational space. Mesh transformation convert non-uniform grids into uniform rectangular grids with mappings $\xi=\xi(x,y)$ and $\eta=\eta(x,y)$. When using mesh transformation, the first-order derivatives of q in the physical space are evaluated through the transformation given by Eq.(1). The terms q_ξ and q_η are evaluated using automatic differentiation. The terms x_ξ, y_ξ, x_η and y_η are called metrics, which are obtained through finite differences of the grid. For simplicity, only the calculation of first-order derivatives is provided. For second-order derivatives, please refer to [5].

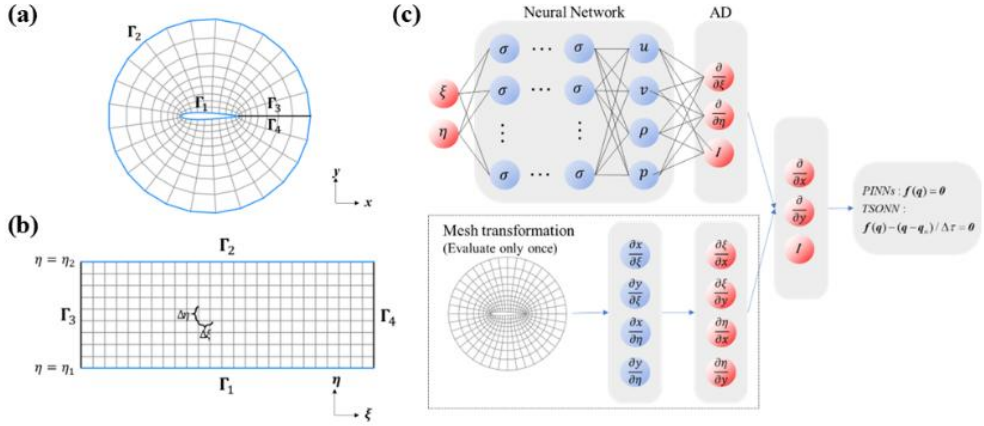

Figure 2. Schematic of the mesh transformation technique. (a) Physical space, (b) Computational space, (c) Neural network solver combined with mesh transformation.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix}=\begin{bmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{bmatrix}\begin{bmatrix} q_\xi \\ q_\eta \end{bmatrix},\quad \begin{bmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{bmatrix}=\begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix}^{-1}=J^{-1} \tag{1}$$

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn

3 Results

3.1 The laminar flow around the NACA0012 airfoil

We first consider the flow around the classical NACA0012 airfoil with Re = 5000. We solve this system using TSONN with varying pseudo time steps $\Delta\tau$ using a network with 5 hidden layers and 128 neurons per hidden layer. In addition, we use volume-weighted residuals. For more settings and details, please refer to [6]. Figure 3 present the relative L_2 error for different pseudo time steps. We observe that PINNs fail to achieve any meaningful result, highlighting its ill-conditioning. When the time step size is large ($\Delta\tau = 1/\gamma = 10$), TSONN obtains convergence histories almost identical to PINNs, validating PINNs as a special case of TSONN when the pseudo time step tends to infinity (i.e., gain tends to 0). As the time pseudo step decreases (i.e., the gain increases), the PDE system becomes increasingly well-conditioned, thus achieving more stable convergence. However, smaller pseudo time step implies more iterations required for the flow to evolve from an initial value to its steady-state solution, potentially resulting in slower convergence.

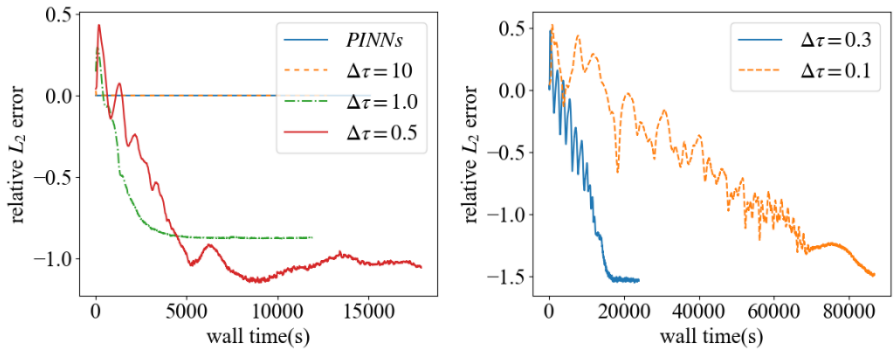

Figure 3. Convergence history of the error obtained by PINNs and TSONN with different pseudo time steps.

3.2 The turbulence around the NACA0012 airfoil

We second consider the turbulence around the NACA0012 airfoil with $Re = 1\times10^5, \alpha = 5°$ to validate the effectiveness of the mesh transformation. We use the C-type mesh to perform the mesh transformation. When the mesh transformation is not applied, we employ the mesh points as collocation points. Figure 4 presents the convergence curves with and without mesh transformation, where the error is evaluated using the relative L_2 error of the wall pressure coefficient distribution. We observe that although some convergence can be achieved without mesh transformation, the convergence is very slow, and the final error stagnates at 10%. In contrast, with mesh transformation, the error rapidly reduces to 3%. Figure 5 shows the contour plots of velocity and pressure after convergence. We observe that the results with mesh transformation are more consistent with the reference solution. For more settings and details, please refer to [7].

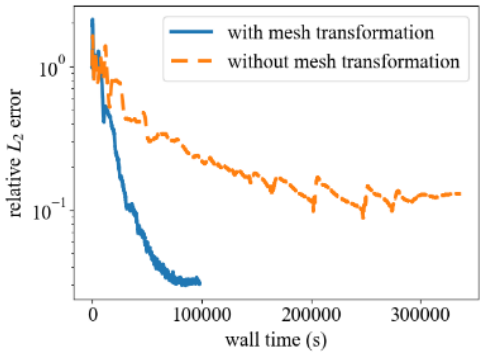

Figure 4. The convergence curves obtained by TSONN with mesh transformation and without mesh transformation.

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn

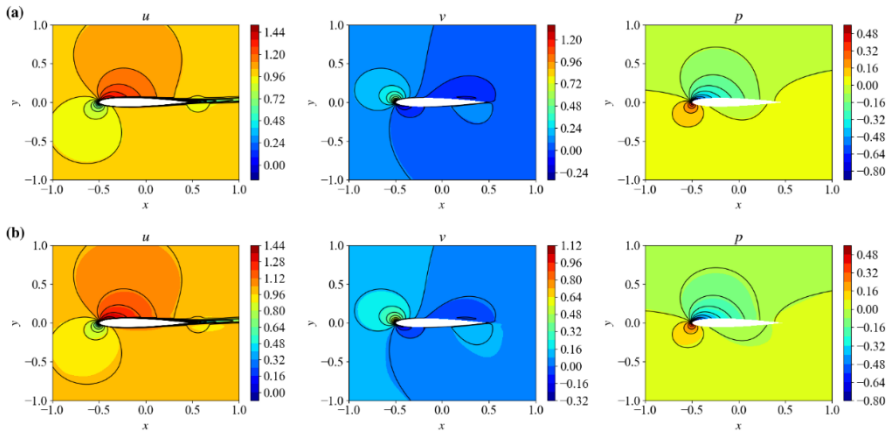

Figure 5. The contour of velocity and pressure obtained by TSONN (a) with mesh transformation, (b) without mesh transformation. The solid black lines are the contour lines of the reference solution.

3.3 The parametric problem involving laminar flow around airfoils

Finally, we combine TSONN and mesh transformation to solve a parametric problem involving flow conditions and airfoil shapes, covering nearly all laminar flow around airfoils in engineering. The goal is to develop a surrogate model with strong generalization capability. The model inputs include all relevant state parameters, such as the computational space coordinates, Reynolds number, angle of attack, and shape parameter vector. We use 6 Class-Shape Transform (CST) parameters for both the upper and lower surfaces. The parameter space for Reynolds number and angle of attack is $Re \in [10, 7000]$, $\alpha \in [-8^{\circ}, 18^{\circ}]$. The shape parameter space is the union of spaces generated by randomly perturbing each shape in the UIUC airfoil database, a rich and professional database containing approximately 1600 airfoils. Specifically, for each collocation point, we randomly select a shape from the UIUC database and add ±35% random perturbation to its CST parameters, using the perturbed CST parameters as input. Figure 6(a) and (b) list the cases used to validate the parametric solver. Figure 6 (c) shows the convergence history for solving the parametric problem. We perform gradient descent involving 10,000 outer iterations, a total of $3.1*10^8$ different collocation points, meaning the model training considers $3.1*10^8$ different flow conditions and shapes. The training achieves stable convergence.

(a)

	Re	α	Airfoil
Case1	100	15°	NACA0012
Case2	200	11°	NACA4412
Case3	500	7°	RAE2822
Case4	1000	3°	RAE5214
Case5	2500	-1°	S2050
Case6	5000	-5°	S9000

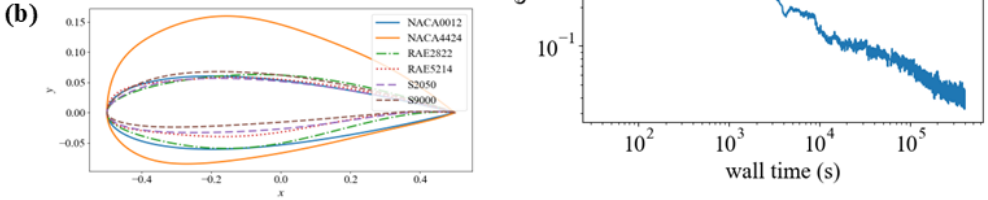

Figure 6. (a) Case configurations and (b) airfoils for evaluation. (c) Error curves for solving the parametric problem, showing the average relative L_1 error of the wall pressure coefficient distributions across the six cases.

By solving the parametric problem, surrogate models are constructed, enabling near real-time prediction of any flow within the parameter space. Figure 7 shows the predicted wall pressure coefficient and skin-friction coefficient distributions. The results demonstrate that the model maintains high consistency with the reference solutions.

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn

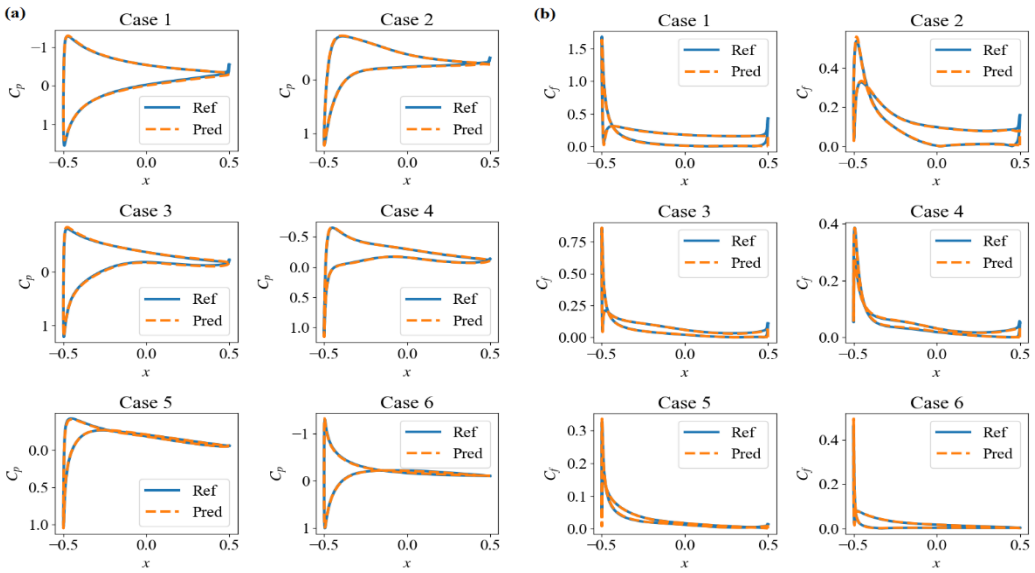

Figure 7. The wall pressure coefficient distributions and skin friction coefficient distributions predicted by the model.

4 Conclusion

This study presents two innovative approaches to enhance the accuracy and efficiency of neural network solvers for airfoil flow simulations. First, we establish a controlled system and reveal that PINNs' ill-conditioning is linked to the Jacobian matrix's condition number. Based on this, we develop TSONN, which transforms the ill-conditioned optimization problem into a series of well-conditioned sub-problems. Second, we introduce a mesh transformation technique, enabling the network to learn the flow in the computational space. This approach effectively amplifies the boundary layer region near the airfoil, improving the network's ability to capture large-gradient flows. Numerical results show that these methods successfully solve inviscid and high-Reynolds-number flows. Furthermore, we extend this solver to high-dimensional parametric engineering problems, covering nearly all laminar flows of interest around airfoils. The results demonstrate that it can efficiently obtain all solutions within the given parameter space all-at-once, verifying its significant advantage. As the efficiency and accuracy of neural network solvers in solving forward problems continue to improve, they are expected to show even greater potential in handling parametric problems.

References

1 Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics* **378**, 686-707 (2019).

2 Weinan, E. & Yu, B. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Communications in Mathematics and Statistics* **6**, 1-12 (2018).

3 Sirignano, J. & Spiliopoulos, K. Dgm: A Deep Learning Algorithm for Solving Partial Differential Equations. *Journal of Computational Physics* **375**, 1339-1364 (2018).

4 Cao, W., Song, J. & Zhang, W. A Solver for Subsonic Flow around Airfoils Based on Physics-Informed Neural Networks and Mesh Transformation. *Physics of Fluids* **36** (2024).

5 Cao, W., Shan, X., Tang, S., Ouyang, W. & Zhang, W. Solving Parametric High-Reynolds-Number Wall-Bounded Turbulence around Airfoils Governed by Reynolds-Averaged Navier-Stokes Equations Using Time-Stepping-Oriented Neural Network. *Physics of Fluids* **31** (2025).

6 Cao, W. & Zhang, W. An Analysis and Solution of Ill-Conditioning in Physics-Informed Neural Networks. *Journal of Computational Physics* **520**, 113494 (2025).

7 Cao, W., Shan, X., Tang, S., Ouyang, W. & Zhang, W. Solving Parametric High-Reynolds-Number Wall-Bounded Turbulence around Airfoils Governed by Reynolds-Averaged Navier–Stokes Equations Using Time-Stepping-Oriented Neural Network. *Physics of Fluids* **37** (2025).

*Corresponding Author, Wenbo Cao caowenbo@mail.nwpu.edu.cn