

PHYSICS-INFORMED NEURAL OPERATORS FOR DISTURBANCE FIELD PREDICTION IN NONPARALLEL BOUNDARY-LAYER STABILITY ANALYSIS

Geovanna Dalinajara Abreu Smith Borges[1], Vinicius Silva Figueredo[1],
Gustavo Luiz Olichevis Halila[1], José M. Magalhães Júnior[2] & João Luiz F. Azevedo[3]

[1]Mechanical Engineering Graduate Program (PPGEM), Pontifícia Universidade Católica do Paraná, 80215-901 Curitiba, PR, Brazil.
[2]Georgia Institute of Technology, Aerospace Engineering Department, 30332 Atlanta, GA, USA.
[3]Instituto Tecnológico de Aeronáutica, 12228-900 São José dos Campos, SP, Brazil.

Abstract

Data-driven neural operators have emerged as promising surrogate models for approximating complex nonlinear dynamics, potentially enabling orders-of-magnitude reductions in computational cost once trained, which makes them attractive for computational fluid dynamics applications involving repeated evaluations or parametric analyses. However, the successful application of neural operators often depends on the availability of high-quality and representative datasets in sufficient quantity to accurately describe the underlying physical process. In this context, incorporating physical principles through governing equations emerges as a promising strategy to improve data efficiency and model robustness. The Physics-Informed Neural Operator (PINO) architecture combines the physics-constrained learning paradigm of PINNs with the operator-learning and generalization capabilities of neural operators. PINOs embed physical knowledge through the partial differential equations (PDEs) governing the phenomenon and, unlike conventional PINNs, which are typically trained for a specific PDE realization, learn solution operators across parameterized families of problems. In the present work, we evaluate the capability of PINO to recover disturbance fields associated with nonparallel hydrodynamic stability analyses in incompressible flat-plate boundary layers. PINO predictions are compared against reference solutions obtained using a Parabolized Stability Equations (PSE) solver over a range of Reynolds numbers to assess predictive accuracy, robustness, and generalization capability. Our results demonstrate that the proposed framework accurately reproduces the principal quantities of interest in hydrodynamic stability analysis, including the complex streamwise wavenumber, disturbance amplitude fields, energy-based growth rate, and N-factor.

Keywords: Physics Informed Neural Operator, Flow Stability Analysis, Parabolized Stability Equations

1. Introduction

The integration of data-driven modeling and machine learning has gained increasing attention in computational fluid dynamics (CFD), particularly for problems governed by nonlinear partial differential equations (PDEs) whose numerical solution can be computationally expensive [1]. Among recent developments, neural operators have emerged as promising surrogate models capable of learning mappings between input conditions and solution fields for parameterized PDE systems. Once trained, these models can significantly reduce computational costs in applications requiring repeated evaluations or parametric analyzes.

Physics-Informed Neural Networks (PINNs), originally introduced in Refs. [2, 3, 4], incorporate governing PDEs into the training process to constrain the solution space and reduce reliance on large training datasets. Despite their success across a range of scientific applications, conventional PINNs

are typically trained to solve individual PDE realizations and may exhibit limitations when addressing complex, multi-scale, or stiff problems, including training instabilities and spectral bias [5]. A complementary line of research seeks to overcome some of the limitations of conventional PINNs through operator learning approaches. In this context, neural operators have emerged as a promising framework for learning mappings between function spaces associated with parameterized partial differential equations (PDEs).

Among neural operator approaches, the Fourier Neural Operator (FNO) introduces a mesh independent framework for learning mappings between infinite-dimensional function spaces using neural networks [6, 7]. By learning operators rather than discretization-specific solutions, FNO enables a single set of network parameters to generalize across different meshes and problem discretizations. Furthermore, the use of spectral convolutions significantly reduces the computational burden associated with operator learning, making FNO particularly effective for approximating complex nonlinear PDE systems [8].

The FNO framework has demonstrated a strong capability in approximating complex operators arising from highly nonlinear PDE systems, including problems characterized by high-frequency modes and slow energy decay [9].

By incorporating governing PDE constraints into the neural operator framework, the Physics-Informed Neural Operator (PINO) combines the generalization capability of FNO-based architectures with the physics-based learning paradigm of PINNs, enabling improved robustness and generalization across varying PDE instances. In practice, PINO integrates supervised training data with a physics-based residual loss derived from the governing PDEs, promoting physically consistent predictions while preserving the operator-learning capability of FNO architectures. This hybrid formulation also helps alleviate some of the optimization challenges commonly encountered in conventional PINNs, making PINO particularly attractive for complex and time-dependent PDE systems [10].

In this work, we evaluate the capability of PINO to predict pressure and velocity disturbance fields computed using a Parabolized Stability Equations (PSE) solver for incompressible boundary-layer stability analysis [11, 12]. Model performance is assessed across a range of Reynolds numbers to quantify predictive accuracy, robustness, and generalization capability. The implementation follows the foundational PINO framework proposed by [10]. The present study focuses on incompressible boundary-layer stability analysis using a PSE solver formulated for spatially developing disturbances [13], although recent efforts have also explored compressible PSE-based mode amplification data [14, 15]. In particular, we investigate the amplification of Tollmien–Schlichting (TS) waves [16, 17, 18, 19] over a flat plate under zero-pressure-gradient conditions. The present work aims to predict the spatial evolution of disturbance fields using a trained PINO framework. Automating flow stability analysis through PINO is particularly attractive because conventional stability tools often require substantial user expertise and careful numerical setup. In contrast, data-driven approaches offer the potential for more accessible workflows and straightforward integration with high-fidelity CFD solvers for transition prediction and aerodynamic analysis. Our results demonstrate that the FNO-based operator accurately predicts the complex streamwise wavenumber, disturbance amplitude fields, the energy-based growth rate, and the N-factor throughout the trained parameter space.

This paper is organized as follows. Section 2. introduces the Parabolized Stability Equations (PSE) theory. Section 3. provides an overview of the mathematical theory behind the Fourier Neural Operator, while Sect. 4. presents the reconstruction of the Physics-Informed Neural Operator for the PSE. Numerical results are presented in Sect. 5.. We end the paper with final remarks in Sect. 6..

2. Parabolized Stability Equations

In modal linear stability theory, the stability problem is solved by using a set of wave modes, each of which is solved independently. Both PSE and LST are examples of modal linear stability methods. The overall principle is based on the decomposition of the flow state vector $\mathbf{q}$ into a steady base flow state $\bar{\mathbf{q}}$ and an unsteady disturbance component, $\tilde{\mathbf{q}}$. The decomposition reads $\mathbf{q}(\mathbf{x},t) = \bar{\mathbf{q}}(\mathbf{x}) + \varepsilon\tilde{\mathbf{q}}(\mathbf{x},t)$, where $\mathbf{x}$ is the space coordinate vector, t is the time, and $\mathbf{q} = (u,w,v,p)$ is the flow state vector, and $\varepsilon \ll 1$.

The PSE method is appropriate for the analysis of streamwise disturbance growth in slowly varying

shear flows such as boundary layers [11, 20, 12]. Mathematically, this is expressed as

$$\partial_x \bar{\mathbf{q}} \ll \partial_y \bar{\mathbf{q}}; \quad \partial_z \bar{\mathbf{q}} = \mathbf{0}, \tag{1}$$

$$\bar{\mathbf{q}}(\mathbf{x}) = \bar{\mathbf{q}}(x^*, y), \tag{2}$$

where x^* is a scaled version of x used to represent the base flow slow variation in the streamwise direction, while y represents the wall-normal direction.
The base flow velocity components $\bar{u}$ and $\bar{v}$, aligned with the streamwise and spanwise directions, respectively, exhibit small variations in the streamwise (x) direction and are constant along the spanwise (z) direction. We introduce the local Reynolds number, $Re = U_e \delta(x)/\nu$ where ν is the kinematic viscosity and $\delta(x)$ is a length scale proportional to the boundary layer thickness, $\delta(x) = \sqrt{\nu x/U_e}$, where U_e is the unperturbed boundary layer edge velocity. The length scale, $\delta(x)$, is typically used in PSE analysis [21, 13]. The reference length scale used to nondimensionalize the equations is δ evaluated at the first marching streamwise coordinate [11]. The wall-normal component, $\bar{w}$, is nonzero and scales with $1/Re$. Formally defining the slowly varying scale $x^* = x/Re$, the scalings are:

$$\begin{aligned} \bar{w} &\sim \frac{1}{Re}, \\ \frac{\partial}{\partial x^*} &\sim \frac{1}{Re}, \\ \alpha &= \alpha(x^*), \\ \hat{\mathbf{q}} &= \hat{\mathbf{q}}(x^*, y). \end{aligned} \tag{3}$$

In a linear PSE analysis, the perturbation vector is expanded in terms of a single mode, truncated Fourier component assuming time-periodicity,

$$\tilde{\mathbf{q}}(x,y,x,t) = \hat{\mathbf{q}}(x,y) \exp\left[i \left(\int_x \alpha\left(x'\right) dx' + \beta z - \omega t \right) \right], \tag{4}$$

where $\hat{\mathbf{q}}(x,y)$ has a slow variation in x. The flow disturbance amplitudes, $\hat{\mathbf{q}}$, present the three velocity components even when a two-dimensional base flow is considered.
To obtain the linear PSE equations, we replace the flow state vector decomposition, $\mathbf{q}(\mathbf{x},t) = \bar{\mathbf{q}}(\mathbf{x}) + \varepsilon \tilde{\mathbf{q}}(\mathbf{x},t)$, in the linearized Navier–Stokes equations and neglect terms of $\mathscr{O}(\varepsilon^2)$. We also consider the scaling from Eq. (3) and neglect higher derivatives with respect to x in the viscous terms (noting that $\frac{\partial}{\partial x}\frac{1}{Re} \sim \varepsilon^2$). The resulting linear PSE equations, in compact form, read

$$\mathbf{A}\hat{\mathbf{q}} + \mathbf{B}\frac{1}{h_y}\frac{\partial \hat{\mathbf{q}}}{\partial y} + \mathbf{C}\frac{1}{h_y^2}\frac{\partial^2 \hat{\mathbf{q}}}{\partial y^2} + \mathbf{D}\frac{1}{h_x}\frac{\partial \hat{\mathbf{q}}}{\partial x} = \mathbf{0}, \tag{5}$$

where h_x and h_y are curvature metrics. The entries for the incompressible PSE operators **A**, **B**, **C**, and **D** are detailed by Ref. [13]. The compressible PSE operators are detailed in Ref. [21]. The boundary conditions are:

$$\hat{u} = \hat{v} = \hat{w} = 0 \quad \text{at} \quad y = 0, \tag{6}$$

$$\hat{u} = \hat{v} = \hat{w} = \hat{p} = 0 \quad \text{as} \quad y \to \infty. \tag{7}$$

In the PSE framework, changes in amplitude along the slowly varying spatial direction are contained both in the amplitude function $\hat{\mathbf{q}}$ and in the phase function defined in Eq. (4). To remove such ambiguity, a normalization condition is required. One possibility for the normalization is to impose that the total kinetic energy of the shape function be independent of x so that fast disturbance variations in the streamwise direction are absorbed into the phase function [12],

$$\int_0^\infty \hat{\mathbf{u}}^\dagger \frac{\partial \hat{\mathbf{u}}}{\partial x} dy = 0, \tag{8}$$

where the superscript † represents the complex conjugate. This normalization condition also ensures an assumed scaling of $\partial\hat{\mathbf{q}}/\partial x$ with $1/Re$.

The disturbance kinetic energy is used to measure the disturbance growth,

$$E(x) = \frac{1}{2}\int_0^\infty \left(|\hat{\mathbf{u}}|^2 + |\hat{\mathbf{v}}|^2 + |\hat{\mathbf{w}}|^2\right) dy. \tag{9}$$

The disturbance kinetic energy-based disturbance growth ratio reads

$$\sigma_E = \frac{1}{h_x}\left(-\alpha_i + \frac{d}{dx}\ln\sqrt{E(x)}\right). \tag{10}$$

To compute the onset of the transition region, the e^N method is usually employed, where N is the amplification factor, or N-factor, defined as

$$N = \ln\left(\frac{A}{A_0}\right) = \int_{x_0}^{x_n} \sigma_E(x)\, dx, \tag{11}$$

where A_0 is the disturbance amplitude at the first neutral-stability point location, denoted by x_0, and x_n the final station. The N-factor envelope is obtained as the superposition of different N-factor curves at each station, computed by the PSE code for a range of frequencies and spanwise wavenumbers.
The linear PSE (5) are intended to be parabolic. Therefore, it is possible to treat the streamwise direction as a pseudo-time and then to implement a marching strategy in this spatial direction. Numerical instabilities appear when the streamwise integration step is too small [13]. The reason for that, as explained by Herbert [12], is that there are traces of ellipticity that inject ill-posed characteristics. One remedy for this is the use of a first-order backward difference scheme with a lower integration step limit $\Delta x > 1/|\alpha_r|$. To relax this limit, Andersson et. al. [22] propose a stabilization procedure leading to $\Delta x > 1/|\alpha_r| - 2s$, where s is a small number.
Due to the predominantly parabolic character of the linear PSE (5), the disturbance evolution is influenced by both local and upstream flow information and, therefore, the PSE method is recognized as a *nonlocal* approach, in contrast to, for instance, the Orr–Sommerfeld equation, which is a *local* formulation.

3. Neural Operators

Classical neural networks approximate functions between finite dimensional Euclidean spaces by mapping input vectors to output vectors [10]. In contrast, neural operators learn mappings between infinite dimensional function spaces [23]. In this framework, the operator represents the solution map of a partial differential equation (PDE), associating an input function, such as an initial or boundary condition, with the corresponding solution field. This continuous formulation alleviates limitations associated with fixed discretizations and enables predictions across different meshes and resolutions [9]. Moreover, neural operators can naturally accommodate variations in physical parameters, including viscosity, temperature, and material properties, while generalizing across different boundary conditions and time dependent forcings [10].

3.1 The Neural Operator Formulation

A neural operator aims to capture the mapping between a continuous input function $\mathscr{X}$ and its corresponding continuous output function $\mathscr{Y}$ [24]. To perform complete training, the pair of functions $(\mathscr{X},\mathscr{Y})$ is discretized into a pair of instances (x,y) during the training process. The goal is to learn a mapping $\mathscr{G}$ between x and y, denoted by:

$$y = \mathscr{G}(x). \tag{12}$$

The mapping $\mathscr{G}$ is built as an iterative architecture and involves sequential steps. Initially, a lifting transformation that increases the dimensionality of the input channel, denoted by $\mathscr{P}$, is performed. This step is followed by the mapping itself through $\mathscr{L}$ layers $(\mathscr{H}_1,\mathscr{H}_2,...,\mathscr{H}_\mathscr{L})$. Finally, a projection back to the original channel dimensionality through $\mathscr{Q}$ [25] takes place. Thus, the operator can be expressed as:

$$\mathscr{G} = \mathscr{Q}\circ\mathscr{H}_\mathscr{L}\circ\cdots\circ\mathscr{H}_2\circ\mathscr{H}_1\circ\mathscr{P}. \tag{13}$$

The transformations $\mathscr{P}$ and $\mathscr{Q}$ are applied pointwise (pixel-wise) and can be implemented using models such as the Multilayer Perceptron (MLP). In Eq. (13), the circle operator ($\circ$) denotes function

composition. For two operators or mappings A and B, the composition $A \circ B$ indicates that B is applied first, followed by A.
In each iteration, the update is defined as the composition of a non-local integral kernel operator $\mathscr{K}$ and a local, nonlinear activation function σ [24]. Each layer is therefore composed of a local (pixel-wise) linear transformation W applied in the spatial domain; an integral kernel operator $\mathscr{K}$; and the activation σ [10]. Formally, a layer can be expressed as:

$$H(x) = \sigma(Wx + \mathscr{K}(x)). \tag{14}$$

This composition is the central mechanism that allows the architecture to approximate highly nonlinear operators: even though the integral operator is linear and global, combining it with local nonlinear activations enables the representation of complex nonlinear dependencies, in close analogy with how standard neural networks compose linear multiplications and nonlinear activations.
The integral kernel operator $\mathscr{K}$ acts on a representation v of the input function [10], incorporating non-local data across the spatial domain D:

$$(\mathscr{K} v)(x) = \int_D k_\phi\big(x, y, \mathscr{X}(x), \mathscr{X}(y)\big)\, v(y)\, dy, \tag{15}$$

where k_ϕ is a learnable kernel, parameterized by a neural network with parameters ϕ. In its most general form, this kernel depends not only on the pair of points (x, y) in the spatial domain, but also on the values of the input function $\mathscr{X}(x)$ and $\mathscr{X}(y)$, which makes the operator highly expressive. This generality, however, comes at a cost: evaluating the integral directly in physical space is computationally expensive, which historically limited the efficiency of neural operators [26].

3.2 The Fourier Neural Operator

The Fourier Neural Operator (FNO) addresses this limitation by replacing the kernel integral operator ($\mathscr{K}$) with a convolutional operator in Fourier space [9]. Let $\mathscr{F}$ denote the Fourier transform of a function $f : D \to \mathbb{R}^{d_v}$, and $\mathscr{F}^{-1}$ its inverse, such that:

$$(\mathscr{F} f)_j(k) = \int_D f_j(x) e^{-2i\pi\langle x,k\rangle} dx, \qquad (\mathscr{F}^{-1} f)_j(x) = \int_D f_j(k) e^{2i\pi\langle x,k\rangle} dk, \tag{16}$$

where $j = 1, 2, ..., d_v$ and the imaginary unit is $i = \sqrt{-1}$. Here, d_v denotes the dimensionality of the latent feature representation v, corresponding to the number of channels over which the Fourier transform is applied. Assuming that $k_\phi\big(x, y, \mathscr{X}(x), \mathscr{X}(y)\big) = k_\phi(x - y)$ and by the spectral convolution theorem [10], convolutions in the spatial domain correspond to pointwise products in the frequency domain. Therefore, the kernel integral can be rewritten as:

$$(\mathscr{K} v)(x) = \mathscr{F}^{-1}(\mathscr{F}(k_\phi) \cdot \mathscr{F}(v))(x), \qquad \forall x \in D. \tag{17}$$

The kernel k_ϕ must be parameterized in Fourier space. A Fourier integral operator is then defined as:

$$(\mathscr{K} v)(x) = \mathscr{F}^{-1}(\mathscr{R}_\phi \cdot \mathscr{F}(v))(x), \qquad \forall x \in D, \tag{18}$$

where R_ϕ is the Fourier transform of a periodic function $\kappa_\phi : \bar{D} \to \mathbb{R}^{d_v \times d_v}$ parameterized by ϕ. For frequency mode $k \in D$, it's accepted that $(\mathscr{F} v)(k) \in \mathbb{C}^{d_v}$ and $R_\phi \in \mathbb{C}^{d_v \times d_v}$ [9]. Given that κ is assumed to be periodic, it admits a Fourier series expansion; thus, it is valid to work with discrete modes $k \in \mathbb{Z}^d$ [10]. A finite-dimensional parameterization is selected by truncating the Fourier series at a maximum number of modes $k_{max} = |Z_{k_{max}}| = |\{k \in \mathbb{Z}^d : |k_j| \leq k_{max,j}, for j = 1, 2, ..., d\}|$.
Then, R_ϕ is directly parameterized as a complex-valued $(k_{max} \times d_v \times d_v)$ tensor, comprising a collection of truncated Fourier modes, thereby removing ϕ from the notation. Because κ_ϕ is real-valued, conjugate symmetry can be imposed. The set $Z_{k_{max}}$ is not a canonical choice for the low-frequency modes of v, which are typically defined by an upper bound on the 1-norm of k [9]. The choice of $Z_{k_{max}}$ is motivated by its allowance for greater implementation efficiency [10].
For the discretization, it is assumed that the domain D is discretized with $n \in \mathbb{N}$ points, $v \in \mathbb{R}^{n \times d_v}$, and $\mathscr{F}(v) \in \mathbf{C}^{n \times d_v}$ [9]. Since the convolution of v is performed with a function that possesses only

k_{max} Fourier modes, the operation can be simplified by truncating the higher modes, resulting in $\mathscr{F}(v) \in \mathbf{C}^{k_{max} \times d_v}$. The multiplication by the weight tensor $R \in \mathbf{C}^{k_{max} \times d_v \times d_v}$ takes the form:

$$(R \cdot (\mathscr{F}))_{k,l} = \sum_{j=1}^{d_v} R_{k,l,j} (\mathscr{F} v)_{k,j}, \quad k = 1, \ldots, k_{\max}, \quad j = 1, \ldots, d_v. \tag{19}$$

The discretization is uniform with a resolution $s_1 \times ... \times s_d = n$, so that $\mathscr{F}$ can be replaced by the Fast Fourier Transform (FFT) [9]. For $f \in \mathbf{R}^{n \times d_v}, k = (k_1, ..., k_d) \in \mathbf{Z}_{s_1} \times \cdots \times \mathbf{Z}_{s_d}$, and $x = (x_1 \cdots, x_d) \in D$, the FFT $\hat{\mathscr{F}}$ and its inverse $\hat{\mathscr{F}}^{-1}$ are defined as:

$$(\hat{\mathscr{F}} f)_l(k) = \sum_{x_1=0}^{s_1-1} \cdots \sum_{x_d=0}^{s_d-1} f_l(x_1, \ldots, x_d) e^{-2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{s_j}}, \qquad (\hat{\mathscr{F}}^{-1} f)_l(x) = \sum_{k_1=0}^{s_1-1} \cdots \sum_{k_d=0}^{s_d-1} f_l(k_1, \ldots, k_d) e^{2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{s_j}}, \tag{20}$$

for $l = 1, ...d_v$. In this specific case, the set of truncated modes becomes:

$$Z_{k_{\max}} = (k_1, \ldots, k_d) \in \mathbf{Z}_{s1} \times \cdots \times \mathbf{Z}_{sd} \mid k_j \leq k_{\max,j} \text{ or } s_j - k_j \leq k_{\max,j}, ; j = 1, \ldots, d. \tag{21}$$

In the implementation, R is treated as an $s_1 \times \cdots \times s_d \times d_v \times d_v$-tensor, and the above definition of $Z_{k_{max}}$ corresponds to the "corners" of R, which enables a direct parallel implementation of Eq. (19) via matrix-vector multiplication [9].

4. Physics-Informed Neural Operator

The Physics-Informed Neural Operator (PINO) combines supervised training data with a physics-based residual loss derived from the governing partial differential equations (PDEs) [27]. By integrating operator learning with physics-constrained training, PINO alleviates some of the optimization difficulties commonly encountered in purely data-driven neural operators while reducing reliance on large training datasets. In contrast to conventional Physics-Informed Neural Networks (PINNs), which are typically trained to approximate the solution of a single PDE realization, PINO learns the solution operator $\mathscr{G}$ associated with a parameterized family of PDE problems [27]. As a result, once trained, the learned operator generalizes across varying problem configurations without requiring retraining or problem-specific optimization.

The training process is commonly divided into two phases. The first phase, referred to as operator learning, trains a neural operator to approximate the solution operator associated with a family of PDE problems using available data, synthetically generated PDE instances, or a combination of both [10]. An appealing strategy consists of combining low-resolution training data with high-resolution PDE residual constraints, enabling accurate extrapolation to finer scales [27].

The second phase, known as instance-wise fine-tuning, refines the pretrained neural operator for a specific PDE realization by minimizing the governing PDE residual. In this stage, an additional *anchor loss* regularization term is introduced to stabilize the optimization and preserve proximity to the pretrained operator [10]. Because the learned operator parametrizes the solution in function space, fine-tuning efficiently adapts the model to individual PDE instances without requiring complete retraining [9].

4.1 PINO Reconstruction for PSE

In this work, we consider a two-dimensional, incompressible base flow over a flat plate. Considering the incompressible PSE operators **A**, **B**, **C**, and **D**, and assuming unitary curvature metrics, $h_x = h_y = 1$, Eq. (5) is written as:

$$i\alpha\hat{u} + i\beta\hat{w} + \frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{v}}{\partial y} = 0, \tag{22}$$

$$\left(\zeta + \frac{\partial U}{\partial x}\right)\hat{u} + \frac{\partial U}{\partial y}\hat{v} + i\alpha\hat{p} + V\frac{\partial \hat{u}}{\partial y} - \frac{1}{Re}\frac{\partial^2 \hat{u}}{\partial y^2} + U\frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{p}}{\partial x} = 0, \tag{23}$$

$$\left(\zeta + \frac{\partial V}{\partial y}\right)\hat{v} + V\frac{\partial \hat{u}}{\partial y} - \frac{1}{Re}\frac{\partial^2 \hat{v}}{\partial y^2} + U\frac{\partial \hat{v}}{\partial x} + \frac{\partial \hat{p}}{\partial y} = 0, \tag{24}$$

$$\frac{\partial W}{\partial x}\hat{u}+\frac{\partial W}{\partial y}\hat{v}+\zeta\hat{w}+i\beta\hat{p}+V\frac{\partial\hat{w}}{\partial y}-\frac{1}{Re}\frac{\partial^2\hat{w}}{\partial y^2}+U\frac{\partial\hat{w}}{\partial x}=0, \tag{25}$$

where

$$\zeta=-i\omega+i\alpha U+i\beta W+\frac{1}{Re}(\alpha^2+\beta^2). \tag{26}$$

In Eq. (26), U, V, and W are the base flow velocity components, i is the imaginary unit, α is the streamwise wave number, ω is the angular frequency, and β is the spanwise wave number.
We employ the PINO framework to learn the solution operator associated with the PSE-based flow stability analysis. The operator $\mathscr{G}_\theta$, parametrized by θ and built upon the Fourier Neural Operator architecture introduced in Section 3.2, maps an input function a to the output function $q=(\hat{u},\hat{v},\hat{w},\hat{p},\alpha)$. The input function comprises the base flow velocity components (U,V,W), their gradients (U_x,U_y,V_y,W_x,W_y), and the physical parameters (Re,ω,β), evaluated over the spatial domain (x,y), where subscripts denote differentiation. Sampling the input function from a distribution μ over the parameter and base flow space yields a family of PSE instances, $a\sim\mu$, over which the operator is trained. The base flow velocity gradients U_x,U_y,V_y,W_x,W_y are computed directly by the PSE flow solver and supplied to the operator as part of the input function. We are interested in learning the disturbance amplitude fields $\hat{u}(x,y)$, $\hat{v}(x,y)$, and $\hat{w}(x,y)$, the pressure field $\hat{p}(x,y)$, and the streamwise wavenumber $\alpha(x)$. Figure 1 describes the operator structure adopted in this work.

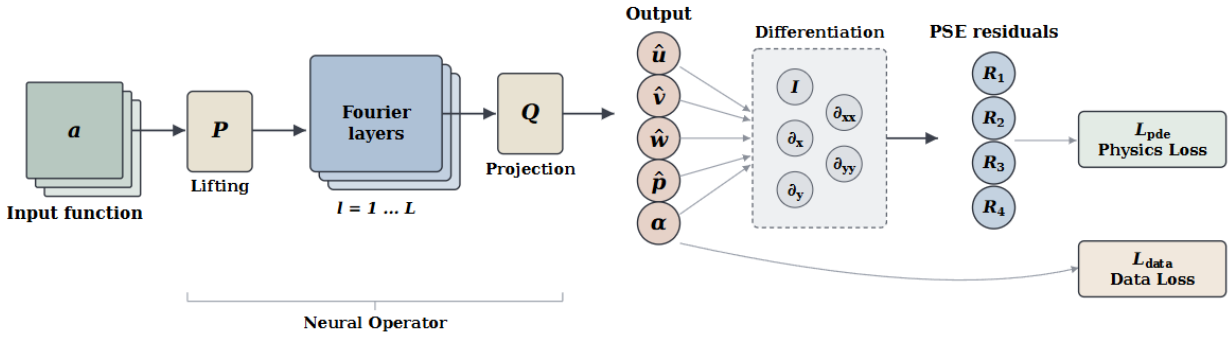


Figure 1 – Physics-Informed Neural Operator (PINO) framework adopted for the reconstruction of disturbance fields obtained from the Parabolized Stability Equations (PSE).

We write the PSE equations as

$$i\alpha\hat{u}+i\beta\hat{w}+\hat{u}_x+\hat{v}_y=0, \tag{27}$$

$$(\zeta+U_x)\hat{u}+U_y\hat{v}+i\alpha\hat{p}+V\hat{u}_y-Re^{-1}\hat{u}_{yy}+U\hat{u}_x+\hat{p}_x=0, \tag{28}$$

$$(\zeta+V_y)\hat{v}+V\hat{u}_y-Re^{-1}\hat{v}_{yy}+U\hat{v}_x+\hat{p}_y=0, \tag{29}$$

$$W_x\hat{u}+W_y\hat{v}+\zeta\hat{w}+i\beta\hat{p}+V\hat{w}_y-Re^{-1}\hat{w}_{yy}+U\hat{w}_x=0, \tag{30}$$

where subscripts indicate differentiation. From these equations, we define the residuals R_1, R_2, R_3, and R_4,

$$R_1:=i\alpha\hat{u}+i\beta\hat{w}+\hat{u}_x+\hat{v}_y, \tag{31}$$

$$R_2:=(\zeta+U_x)\hat{u}+U_y\hat{v}+i\alpha\hat{p}+V\hat{u}_y-Re^{-1}\hat{u}_{yy}+U\hat{u}_x+\hat{p}_x, \tag{32}$$

$$R_3:=(\zeta+V_y)\hat{v}+V\hat{u}_y-Re^{-1}\hat{v}_{yy}+U\hat{v}_x+\hat{p}_y, \tag{33}$$

$$R_4:=W_x\hat{u}+W_y\hat{v}+\zeta\hat{w}+i\beta\hat{p}+V\hat{w}_y-Re^{-1}\hat{w}_{yy}+U\hat{w}_x, \tag{34}$$

which measure the deviation from zero obtained when the operator outputs are substituted into the differential operators of Eqs. (27)–(30). The derivatives of the operator outputs required to evaluate the residuals are obtained through Fourier-space differentiation, intrinsic to the FNO architecture; since the wall-normal direction y is non-periodic, the Fourier continuation technique is applied to embed the domain into a periodic space prior to differentiation.

In the operator-learning phase, the operator is trained over the family of instances by minimizing a composite objective that combines a data loss and a physics loss. The data loss measures the discrepancy between the operator predictions and the reference fields, while the physics loss enforces the PSE residuals,

$$\mathscr{J}(\mathscr{G}_\theta) = \mathscr{J}_{\text{data}}(\mathscr{G}_\theta) + \mathscr{J}_{\text{pde}}(\mathscr{G}_\theta), \tag{35}$$

with

$$\mathscr{J}_{\text{data}}(\mathscr{G}_\theta) = \mathbb{E}_{a\sim\mu}\left[\frac{1}{N_d}\sum_{k=1}^{N_d}\left(|\hat{u}^k_{ref}-\hat{u}^k_{pred}|^2+|\hat{v}^k_{ref}-\hat{v}^k_{pred}|^2+|\hat{w}^k_{ref}-\hat{w}^k_{pred}|^2+|\hat{p}^k_{ref}-\hat{p}^k_{pred}|^2+|\alpha^k_{ref}-\alpha^k_{pred}|^2\right)\right], \tag{36}$$

$$\mathscr{J}_{\text{pde}}(\mathscr{G}_\theta) = \mathbb{E}_{a\sim\mu}\left[\frac{1}{N_c}\sum_{k=1}^{N_c}\left(|R^k_1|^2+|R^k_2|^2+|R^k_3|^2+|R^k_4|^2\right)\right], \tag{37}$$

where the subscripts d and c denote, respectively, the reference data and the collocation points, N_d is the number of data points, N_c is the number of collocation points, and the expectation $\mathbb{E}_{a\sim\mu}[\cdot]$ is taken over the sampled PSE instances. The data loss in Eq. (36) is evaluated on the available reference fields, which may be provided at low resolution, whereas the physics loss in Eq. (37) is imposed at a higher resolution, thereby enabling accurate reconstruction at scales finer than those of the training data.

In the instance-wise fine-tuning phase, the pretrained operator $\mathscr{G}_{\theta_0}$ is refined for a single PSE instance by minimizing the physics loss together with the anchor loss,

$$\mathscr{L}(\mathscr{G}_{\theta_i}) = \mathscr{L}_{\text{pde}}(\mathscr{G}_{\theta_i}) + \lambda\,\mathscr{L}_{\text{op}}(\mathscr{G}_{\theta_i},\mathscr{G}_{\theta_0}), \tag{38}$$

where

$$\mathscr{L}_{\text{pde}}(\mathscr{G}_{\theta_i}) = \frac{1}{N_c}\sum_{k=1}^{N_c}\left(|R^k_1|^2+|R^k_2|^2+|R^k_3|^2+|R^k_4|^2\right), \tag{39}$$

and the anchor loss

$$\mathscr{L}_{\text{op}}(\mathscr{G}_{\theta_i},\mathscr{G}_{\theta_0}) = \|\mathscr{G}_{\theta_i}(a)-\mathscr{G}_{\theta_0}(a)\|^2 \tag{40}$$

constrains the fine-tuned operator $\mathscr{G}_{\theta_i}$ at the i-th training epoch to remain close to the pretrained operator $\mathscr{G}_{\theta_0}$, and $\lambda > 0$ weights the regularization.

Collocation points constitute a fundamental component of the physics-informed formulation. These points are specific coordinates, selected or generated within the domain, at which the governing PDEs are explicitly enforced. Their primary function is to embed physical constraints by requiring the operator output to satisfy the PSE system. This enforcement is achieved by minimizing the PDE residuals defined in Eqs. (31)–(34), which quantify the deviation from zero when the approximated solution is substituted into the differential operators. The minimization of these residuals across the collocation points acts as a regularization mechanism that penalizes solutions violating the prescribed physics, thereby biasing the optimization toward physically admissible functions. Because the physics loss is evaluated on the operator outputs over the entire family of instances, this constraint promotes accurate interpolation and extrapolation, allowing the operator to achieve robust generalization even when trained on limited or sparse observational data.

5. Numerical Results

In this section, we present results obtained with a Physics-Informed Neural Operator (PINO) framework applied to flow stability analysis. The problem addresses the evolution of Tollmien–Schlichting (TS) wave disturbances in an incompressible, zero-pressure-gradient flat plate boundary layer. The training, test, and validation data were generated using a Parabolized Stability Equations (PSE) solver [13], which serves as the high-fidelity ground truth for this investigation.

The dataset consists of 100 PSE solutions covering a parametric space defined by the inlet displacement thickness Reynolds number, Re_δ, ranging from 300 to 500, the outlet displacement-thickness Reynolds number ranging from 1,000 to 2,000, and the reduced frequency, F, ranging from 5.6×10^{-5} to 1.24×10^{-4}, with $\beta = 0$ throughout. Each solution is computed on a grid of 80 streamwise stations

and 250 points in the wall-normal direction. The dataset is split such that 80% of the cases are used for training, 10% for testing, and the remaining 10% for validation.

We select a Fourier Neural Operator, which is set to operate on the full two-dimensional spatial field of the PSE solution. The architecture consists of 4 Fourier layers with 64 hidden channels, retaining 20 modes in the wall-normal direction and 12 modes in the streamwise direction. The input to the network comprises 13 channels: the spatial coordinates (x, y), the global flow parameters $(Re_\delta, \omega, \beta)$, and eight mean flow quantities $(u, v, w, u_x, u_y, v_y, w_x, w_y)$. The network output consists of 10 channels corresponding to the real and imaginary parts of the disturbance eigenfunctions $(\hat{u}, \hat{v}, \hat{w}, \hat{p})$ and the streamwise wavenumber, α. The total loss function is composed of two components: a weighted data-matching term, and a physics-informed PSE residual. The data loss employs channel-specific weights, with the α channels weighted twenty times higher than the eigenfunction channels to improve the prediction accuracy of the growth rate, which is critical for stability analysis. The model is trained using the Adam optimizer with a learning rate of 3×10^{-4}, reduced by a factor of 0.5 when the validation loss shows no improvement for 100 consecutive epochs, with a minimum learning rate of 10^{-6}. Training is halted early if no improvement is observed over 200 epochs. The loss weights are set to $W_{\text{data}} = W_{\text{PSE}} = 1$.

We focus on disturbances aligned with the streamwise direction ($\beta = 0$), thereby disregarding the spanwise disturbance velocity component $\hat{w}$, which vanishes in this case. The flow conditions are characterized by the local Reynolds number based on the displacement thickness δ^*, spanning from 400 at the inlet to 1,000 at the outlet. The reduced frequency parameter is set to $F = \omega / Re_{\delta^*} = 1.0 \times 10^{-4}$. The base flow represents a laminar, zero-pressure-gradient flat plate boundary layer. The comparative results presented in the following sections assess the ability of the PINO framework to predict disturbance field evolution and recover key flow stability features.

As shown in Fig. 2, the PINO framework accurately reproduces the streamwise evolution of the complex wavenumber α in close agreement with the reference PSE solution. The real component, α_r, is predicted with a relative L_2 error of $[2.28 \times 10^{-3}]$. The imaginary component, α_i, which directly governs the disturbance amplification and is highly sensitive to the flow conditions, is also captured with good accuracy, yielding an error of $[4.12 \times 10^{-2}]$. The agreement observed for both components indicates that the PINO successfully learns the underlying linear dispersion relation, $\alpha_r(\omega, Re_\delta)$, accurately capturing the dependence of the perturbation phase velocity on the flow parameters.

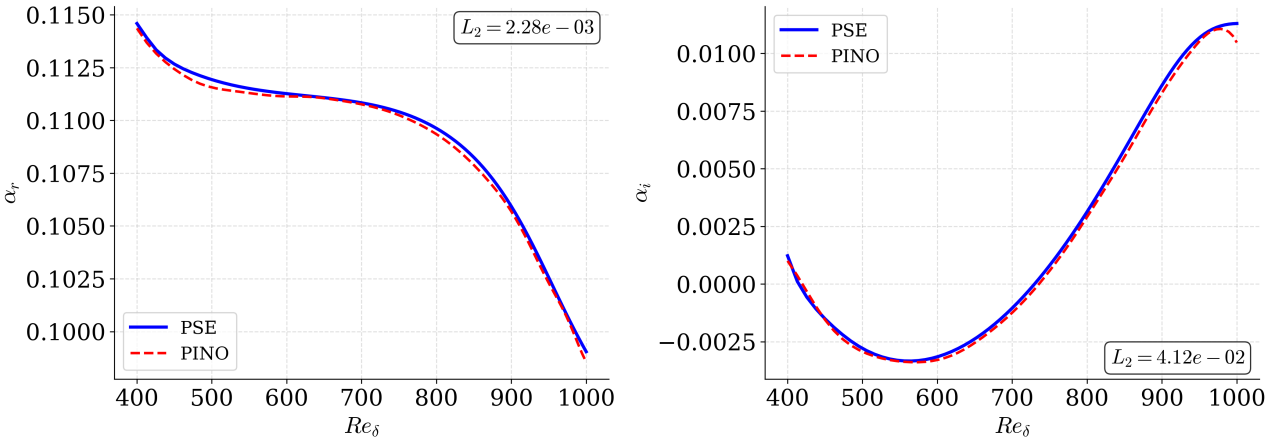


Figure 2 – Real and imaginary parts of the streamwise wavenumber α as predicted by PINO and the PSE solver.

The two-dimensional disturbance amplitude fields for the velocity components $\hat{u}$, $\hat{v}$, and the pressure $\hat{p}$ are shown in Figs. 3, 4, and 5, respectively. Each figure presents, from left to right, the PSE reference solution, the PINO prediction, and the pointwise difference (PSE $-$ PINO). The operator accurately reproduces the spatial structure of the disturbance fields across the entire domain, capturing both the streamwise amplification and the wall-normal eigenfunction shape.

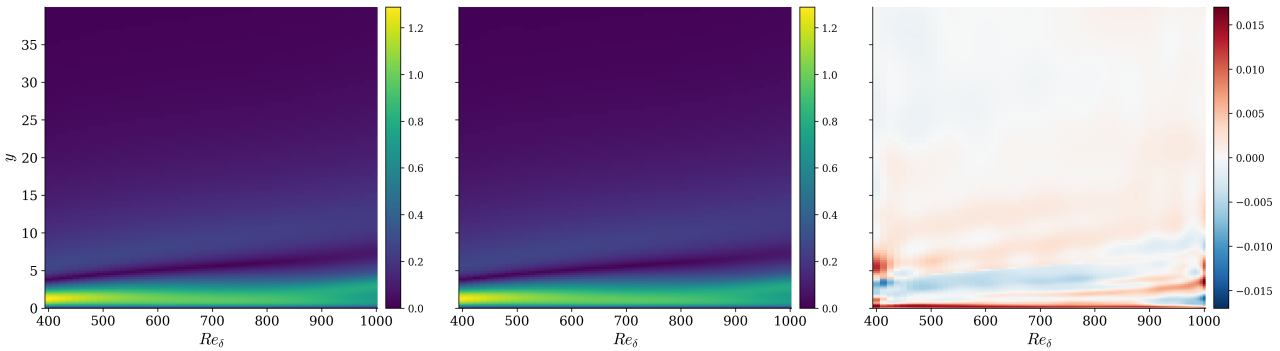


Figure 3 – Disturbance amplitude field $|\hat{u}|$ obtained from the reference PSE solution (left) and predicted by the PINO framework (center), together with the corresponding pointwise error field (right).

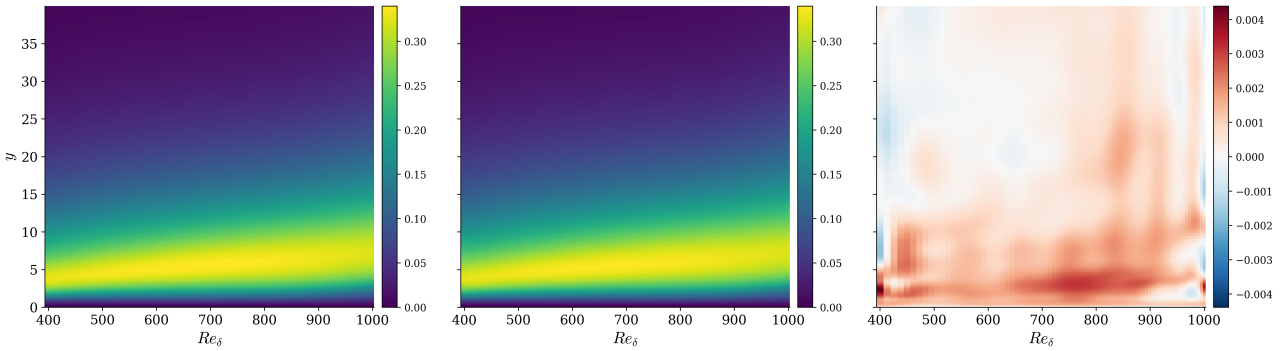


Figure 4 – Disturbance amplitude field $|\hat{v}|$ obtained from the reference PSE solution (left) and predicted by the PINO framework (center), together with the corresponding pointwise error field (right).

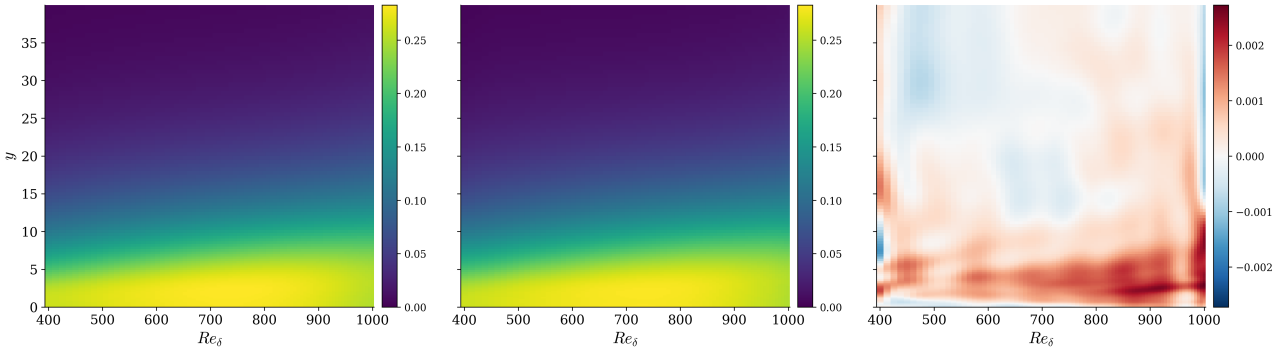


Figure 5 – Disturbance amplitude field $|\hat{p}|$ obtained from the reference PSE solution (left) and predicted by the PINO framework (center), together with the corresponding pointwise error field (right).

The streamwise evolution of the energy-based growth rate, σ_E, is presented in Fig. 6. The PINO reproduces the characteristic shape of the growth rate curve, including the damped region near the inlet, the transition through the first branch of the neutral stability curve, and the amplified region at higher Reynolds numbers, with a relative L_2 error of $[4.27 \times 10^{-2}]$.

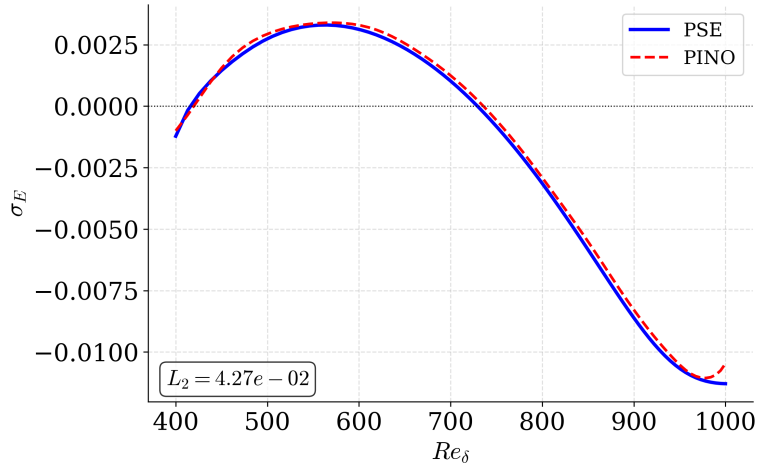


Figure 6 – Streamwise evolution of the energy-based growth rate, σ_E, predicted by the PINO framework and the reference PSE solver.

The amplification factor, or N-factor, shown in Fig. 7, is directly relevant to laminar-turbulent transition prediction via the e^N method. The PINO prediction follows the PSE reference closely throughout the domain, yielding a relative L_2 error of $[1.34 \times 10^{-1}]$. The accuracy of the N-factor prediction confirms that the cumulative errors in the growth rate remain bounded, which is essential for reliable transition location estimates.

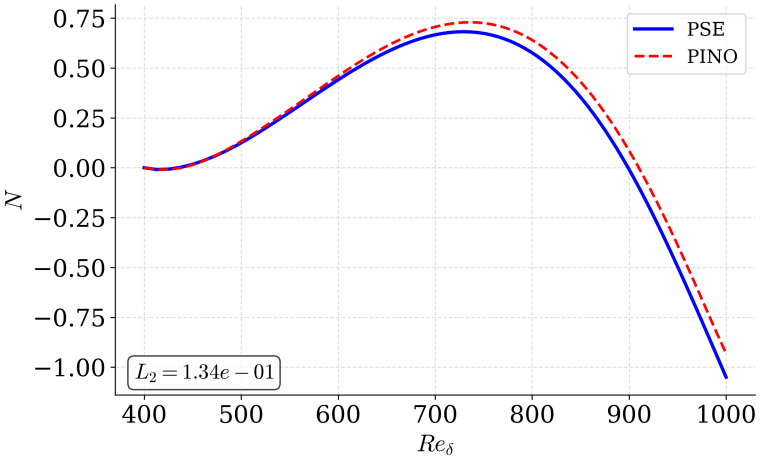


Figure 7 – Streamwise evolution of the amplification factor, N, predicted by the PINO framework and the reference PSE solver.

The results presented above demonstrate that the PINO framework successfully reproduces the PSE solution across all quantities of interest, including the complex streamwise wavenumber, the disturbance amplitude fields of the velocity and pressure components, the energy-based growth rate, and the N-factor, all with low relative errors. Notably, the operator accurately captures both the wall-normal eigenfunction structure and the streamwise amplification behavior, without requiring explicit integration of the governing stability equations during inference. The accurate prediction of the N-factor is particularly important, as it indicates that cumulative errors in the growth rate remain bounded, a necessary condition for reliable transition-location estimates based on the e^N method.

6. Final Remarks

This work presents a Physics-Informed Neural Operator (PINO) framework for approximating the solution operator of the Parabolized Stability Equations (PSE) for incompressible boundary-layer stability analysis over a prescribed range of flow parameters. The framework combines the operator-learning capability of the Fourier Neural Operator (FNO) with physics-informed constraints derived

directly from the governing PSE system, enabling the reconstruction of disturbance fields associated with spatially developing instability waves.

The results demonstrate that the proposed framework accurately reproduces the principal quantities of interest in hydrodynamic stability analysis, including the complex streamwise wavenumber, disturbance amplitude fields, energy-based growth rate, and N-factor. In particular, the operator successfully captures both the wall-normal eigenfunction structure and the streamwise amplification behavior of Tollmien–Schlichting disturbances. The accurate prediction of the growth-rate-critical $alpha_i$ component leads to reliable estimates of the disturbance growth rate and bounded cumulative errors in the N-factor, which is essential for transition prediction based on the e^N method.

A fundamental distinction between the present approach and a conventional Physics-Informed Neural Network (PINN) lies in the nature of the learned mapping. While PINNs are generally formulated as instance-specific solvers, the proposed PINO learns an approximation of the PSE solution operator over a parameterized family of flow conditions. As a result, once trained, the framework produces predictions for unseen flow configurations within the training domain through a single forward pass, without requiring explicit marching or integration of the governing stability equations. This characteristic makes the approach particularly attractive for applications involving repeated stability evaluations, including parametric studies, uncertainty quantification, transition prediction workflows, and gradient-based aerodynamic design optimization.

7. Acknowledgments

The authors gratefully acknowledge the support provided by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, Brazil, under the Research Grants No. 88887.282158/2026-00 and 88887.217284/2025-00. The authors also acknowledge the support for the present research provided by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under the Research Grants No. 2013/07375-0 and 2021/11258-5. Further support for the present research was provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grant No. 315411/2023-6.

8. Contact Author Email Address

Geovanna Dalinajara Abreu Smith Borges: geovannadalinajara7@gmail.com.
Vinicius Silva Figueredo: viniciussifigueredo@gmail.com.
Gustavo Luiz Olichevis Halila: gustavo.halila@pucpr.br
José M. Magalhães Júnior: jose.magalhaes@gatech.edu
João Luiz F. Azevedo: joaoluiz.azevedo@gmail.com

9. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[2] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part i): Data-Driven Solutions of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10561*, 2017.

[3] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part ii): Data-Driven Discovery of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10566*, 2017.

[4] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 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.

[5] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

[6] Lu Chi, Borui Jiang, and Yadong Mu. Fast Fourier Convolution. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4479–4488. Curran Associates, Inc., 2020.

[7] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021.

[8] Nicholas H. Nelsen and Andrew M. Stuart. The Random Feature Model for Input-Output Maps between Banach Spaces. *SIAM Journal on Scientific Computing*, 43(5):A3212–A3243, 2021.

[9] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations (ICLR)*, page 16, May 2021.

[10] Valentin Duruisseaux, Jean Kossaifi, and Anima Anandkumar. Fourier Neural Operators Explained: A Practical Perspective. *ArXiv*, abs/2512.01421, 2025.

[11] Fabio P. Bertolotti and Thorwald Herbert. Analysis of the Linear Stability of Compressible Boundary Layers using PSE. *Theoretical and Computational Fluid Dynamics*, 3:117–124, 1991.

[12] Thorwald Herbert. Parabolized Stability Equations. *Annual Review of Fluid Mechanics*, 29:245–283, 1997.

[13] Matthew P. Juniper, Ardeshir Hanifi, and Vassilis Theofilis. Modal Stability Theory. Lecture Notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Aplied Mechanics Reviews*, 66:1–22, March 2014.

[14] Gustavo L. O. Halila, Guodong Chen, Yayun Shi, Krzysztof J. Fidkowski, Joaquim R. R. A. Martins, and Márcio Mendonça. High-Reynolds Number Transitional Flow Prediction using a Coupled Discontinuous-Galerkin RANS PSE Framework. *Aerospace Science and Technology*, 91:321–336, August 2019.

[15] Gustavo L. O. Halila, Krzysztof J. Fidkowski, and Joaquim R. R. A. Martins. Toward Automatic Parabolized Stability Equation-Based Transition-to-Turbulence Prediction for Aerodynamic Flows. *AIAA Journal*, 59(2):462–473, February 2021.

[16] Walter Tollmien. Über die Entstehung der Turbulenz. In *Nachr. Ges. Wiss. Göttingen, Math.-Phys. Klasse, 1. Mitteilung*, pages 21–44, 1929.

[17] Hermann Schlichting. Zur Entstehung der Turbulenz bei der Plattenströmung. In *Nachr. Ges. Wiss. Göttingen, Math.-Phys. Klasse, Fachgruppe II*, volume 38, pages 181–208, 1933.

[18] Galen B. Schubauer and Harold K. Skramstad. Laminar Boundary Layer Oscillations and Transition on a Flat Plate. NACA Report 909, NACA, 1948.

[19] Peter G. Baines, Sharan J. Majumdar, and Humio Mitsudera. The Mechanics of the Tollmien-Schlichting Wave. *Journal of Fluid Mechanics*, 312:107–124, 1996.

[20] Fabio P. Bertolotti. *Turbulence and Transition Modelling,* Ch. 8, *Transition Modelling Based on the PSE*. Kluwer Academic Publishers, Netherlands, 1996.

[21] Ardeshir Hanifi, Dan Henningson, Stefan Hein, Fabio P. Bertolotti, and Martin Simen. Linear Nonlocal Instability Analysis – The Linear NOLOT Code. Technical Report FFA-TN 1994-54, The Aeronautical Research Institute of Sweden, Aerodynamics Department, 1994.

[22] Paul Andersson, Dan S. Henningson, and Ardeshir Hanifi. On a Stabilization Procedure for the Parabolic Stability Equations. *Journal of Engineering Mathematics*, 33:311–332, February 1998.

[23] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps between Function Spaces with Applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[24] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *9th International Conference on Learning Representations (ICLR 2021)*, Virtual Event, Austria, May 2021. OpenReview.net.

[25] Shaoxiang Qin, Fuyuan Lyu, Wenhui Peng, Dingyang Geng, Ju Wang, Xing Tang, Sylvie Leroyer, Naiping Gao, Xue Liu, and Liangzhu Leon Wang. Toward a Better Understanding of Fourier Neural Operators from a Spectral Perspective. Report in arXiv, https://arxiv.org/abs/2404.07200, 2024.

[26] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps between Function Spaces with Applications to PDEs. *J. Mach. Learn. Res.*, 24(1), January 2023.

[27] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-Informed Neural Operator for Learning Partial Differential Equations. *ACM / IMS J. Data Sci.*, 1(3), May 2024.