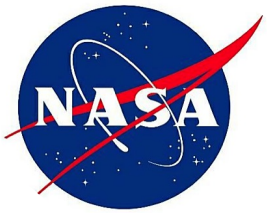
DeepONet-Assisted Optimization of Surface Topography for Transition Delay in a Mach 4.5 Boundary Layer

Nathaniel Hildebrand, Vishal Srivastava (AMA), and Meelan M. Choudhari
Computational AeroSciences Branch,
NASA Langley Research Center, Hampton, VA

Tamer A. Zaki
Johns Hopkins University, Baltimore, MD

14th International ERCOFTAC Symposium on
Engineering Turbulence Modelling and Measurements (ETMM14)
Mini Symposia: Machine Learning for Turbulence
September 8, 2023



Motivation

- Design optimization is critical to enable radically new aerospace concepts, especially in applications that require high-fidelity information based on nonlinear time-accurate flow behavior, e.g., aeroacoustics, laminar-turbulent transition, aerothermodynamics, etc.

- Current methods based on direct applications of DNS, LES, etc., are computationally expensive and slow
  - Repeated evaluations of the design metric and sensitivities to different control parameters (around 30+)

- Our goal is to reduce the computational cost for aerodesign requiring high-fidelity time-accurate data, while encapsulating the knowledge base generated during the design task into an accurate and efficient post-design reduced-order model
  - Novel machine-learning architectures based on Deep Operator Networks (DeepONets) that encode nonlinear operators and complex systems of governing equations (Lu, Jin, and Karniadakis 2021)

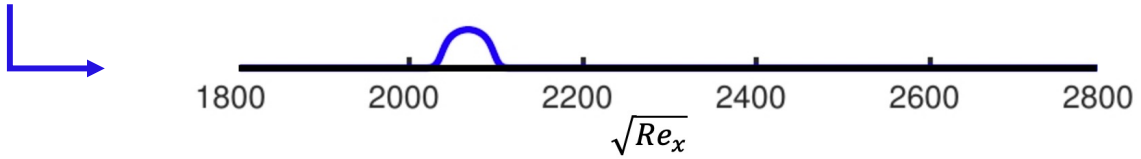
Outline

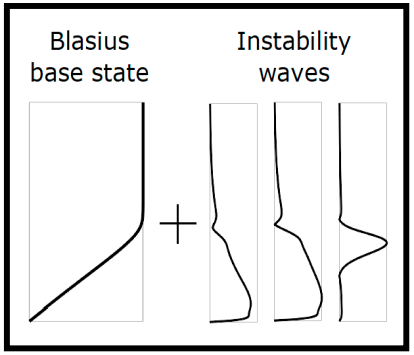
- Example use case: Delay transition in high-speed boundary layer via surface protuberance ("roughness" element) by combining Ensemble-Variational (EnVar) approach with DeepONets and DNS.
  - Inspired by recent application of EnVar + DNS by Jahanbakhshi & Zaki (2023)
  - Alternate problem choices, objective function, etc., also possible, but main emphasis is to evaluate the merit of DeepONets to reduce the optimization cost
- Validation of automated Python interface for EnVar against results in Jahanbakhshi and Zaki (2023)
- Parametric study using EnVar and DNS for different roughness parameters
  - Training data for DeepONet model with 3 design variables
- Best training practices for DeepONet model
- Demonstrate of EnVar + DeepONets with no DNS (except for training) for transition delay in Mach 4.5 boundary layer with a single roughness element
  - Computational speedup by a factor of 5–6 so far
- Summary and future work

Flat Plate Configuration

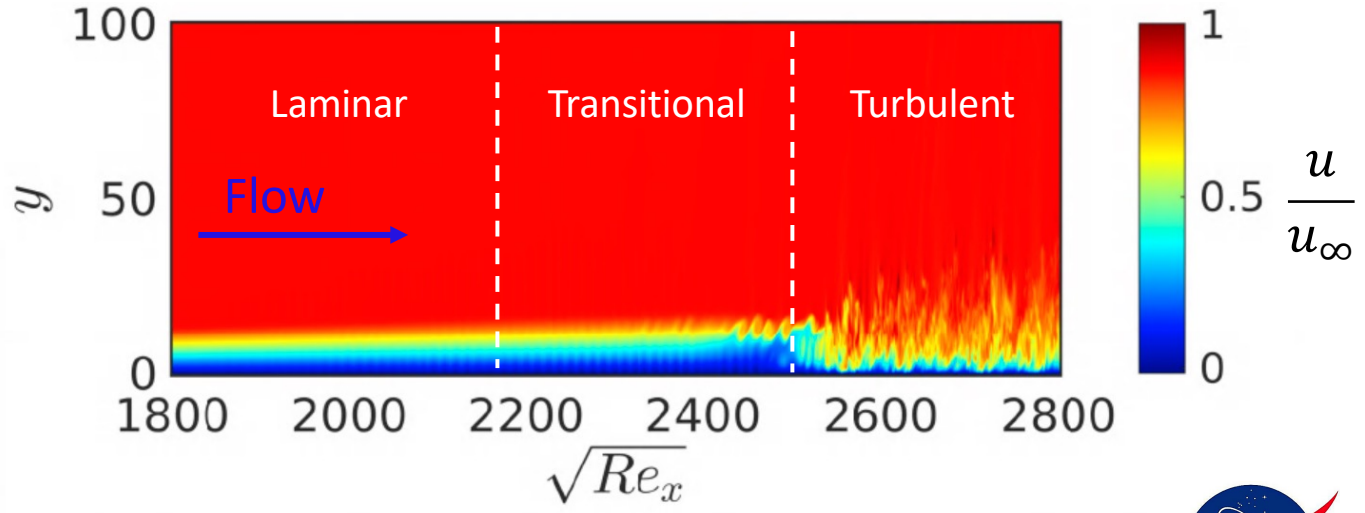
(Jahanbakhshi and Zaki 2023)

- Mach 4.5 boundary layer with $\mathrm{Re}_{x_0} = 3.24\times10^6$
- Mesh consists of $N_x \times N_y \times N_z = (2984 \times 189 \times 151)$ grid points
- Flow control via transition delay with a local surface deformation
  - Roughness defined by the height (H_r), width (W_r), and abruptness (L_r)





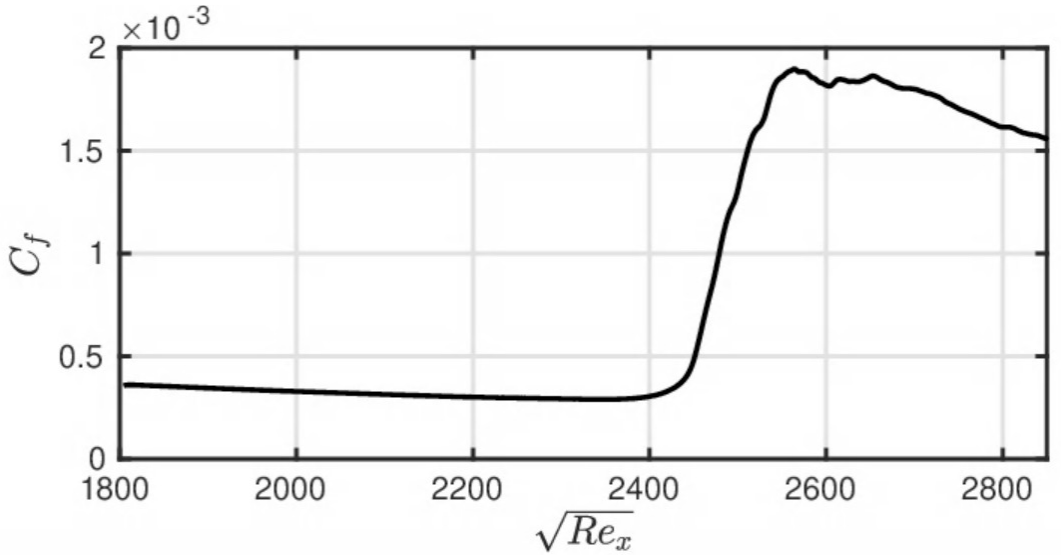
Inflow

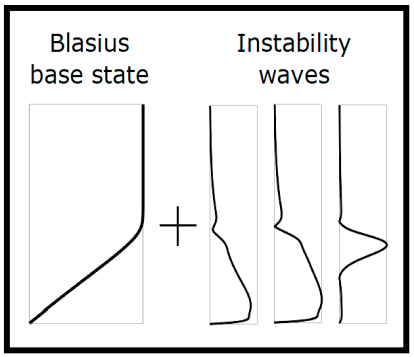


Baseline DNS (No Roughness Element)

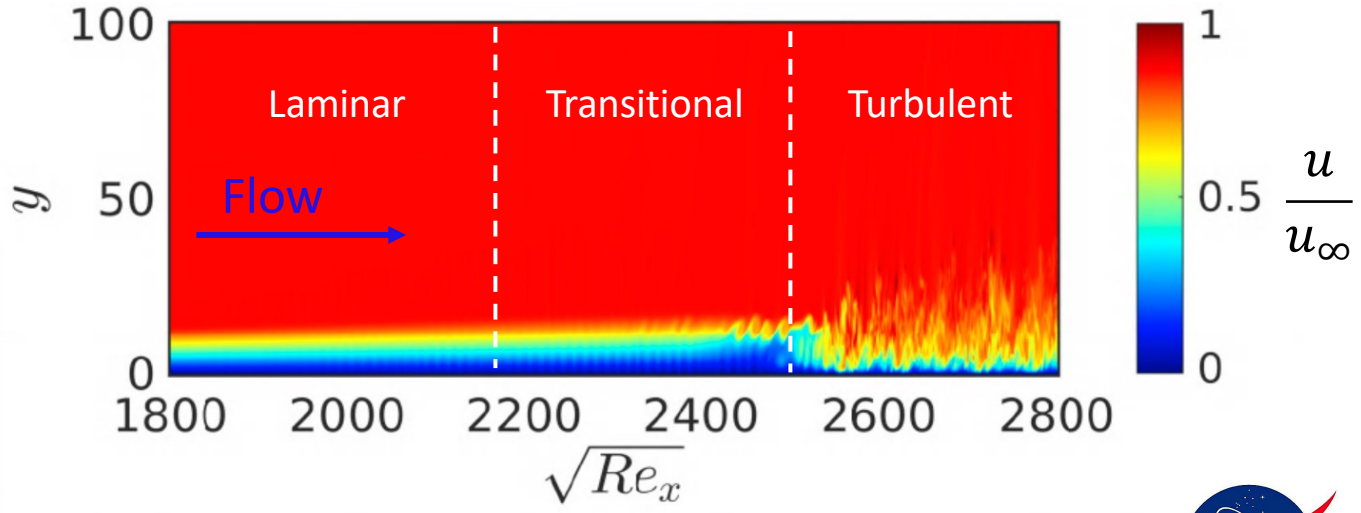
(Jahanbakhshi and Zaki 2023)

Transition occurs around $\sqrt{Re_x} = 2500$





Inflow



EnVar Algorithm

$\boldsymbol{i = 0}$

- Estimate the initial control vector $\boldsymbol{c}_0 = [H_r, W_r, L_r]^{\mathrm{T}}$
- Obtain the mean observations via DNS/DeepONet
- Evaluate the initial cost function
  - $\Im = \frac{1}{2}\left\|\boldsymbol{c} - \boldsymbol{c}^{(e)}\right\|^2_{\boldsymbol{B}^{-1}} + \frac{1}{2}\left\|C_f\sqrt{dx/L_x}\right\|^2_{\boldsymbol{R}^{-1}}$

while *flow still transitions in domain* **then**

- Generate ensemble members by sampling normal distribution
- Acquire ensemble member observations via DNS/DeepONet
- Construct observation matrix
- Compute gradient $\nabla\Im$ and Hessian
- Acquire optimal weights
- Construct next ensemble mean $\boldsymbol{c}_{i+1} = [H_r, W_r, L_r]^{\mathrm{T}}$
- Acquire mean observations via DNS/DeepONet
- Compute the cost $\Im$ and check the convergence criteria

$\boldsymbol{i = i + 1}$

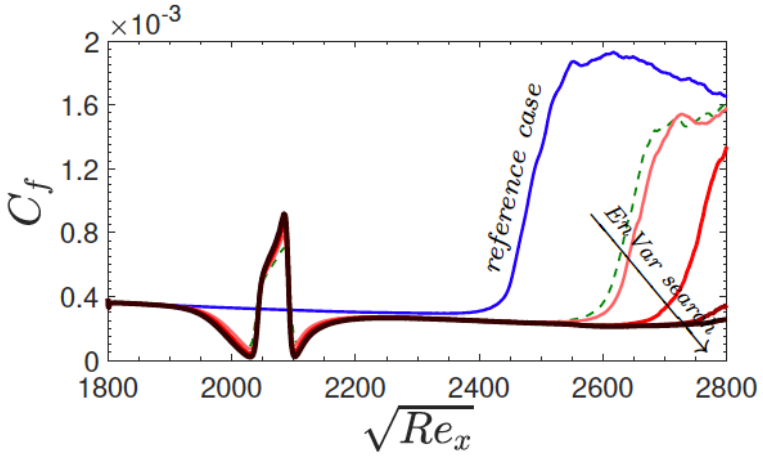
end

- Optimal $\mathbf{c} = [H_r, W_r, L_r]^{\mathrm{T}}$, $C_f(x)$ with delayed onset of transition

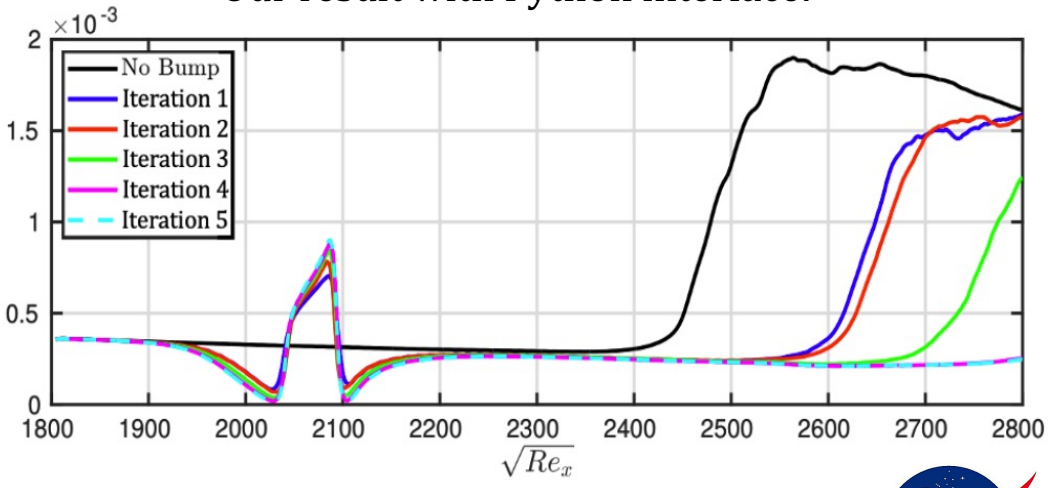
Verification of Automated Python Interface for EnVar

- Find the optimal combination of roughness parameters (height, width, and abruptness) that delays transition using EnVar and DNS
- After four iterations of EnVar, the roughness element successfully delays transition beyond the computational domain
- Results agree very closely with Jahanbakhshi and Zaki 2023
  - Any subtle changes due to random seeding of ensemble members

Jahanbakhshi and Zaki 2023:

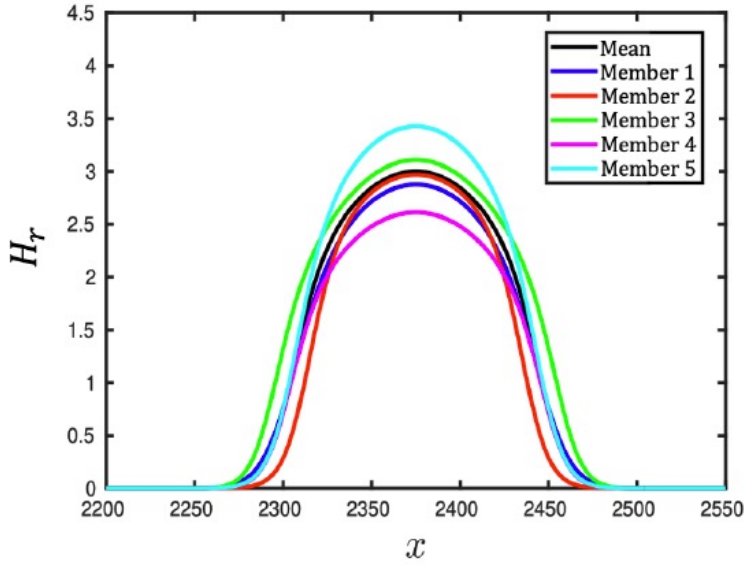


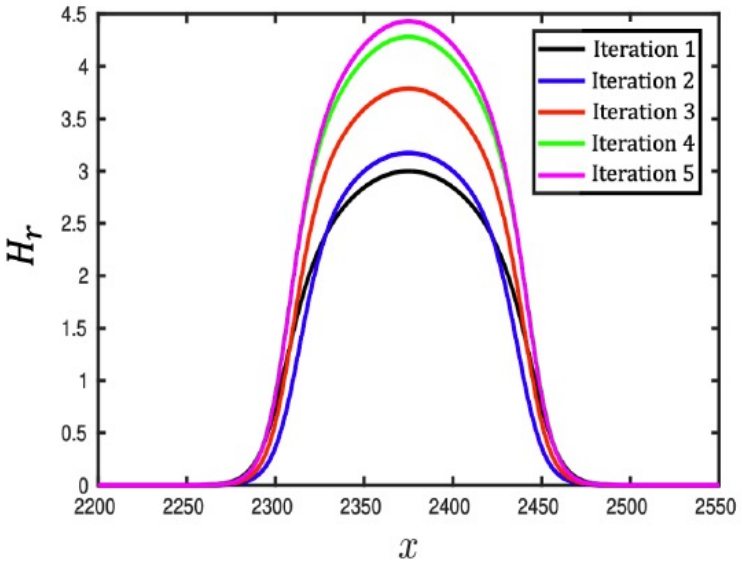
Our result with Python interface:



Influence of Roughness Parameters

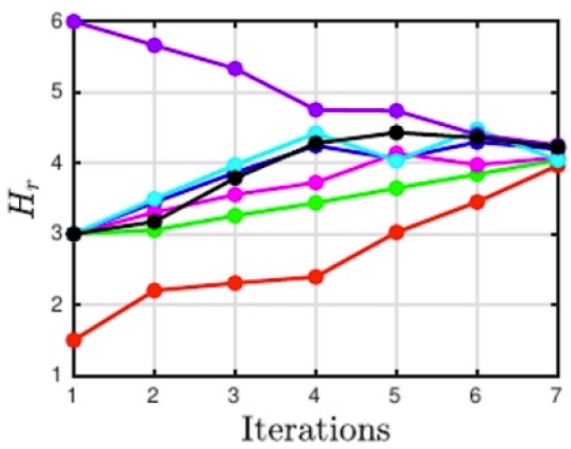
- Initial parameters: $H_r = 3.0$, $W_r = 72.5$, and $L_r = 0.06$
- Each iteration has one mean and five random ensemble members
- Roughness height becomes much larger as the iteration number increases from 1 to 5 and transition is delayed
- Roughness width and abruptness do not change significantly

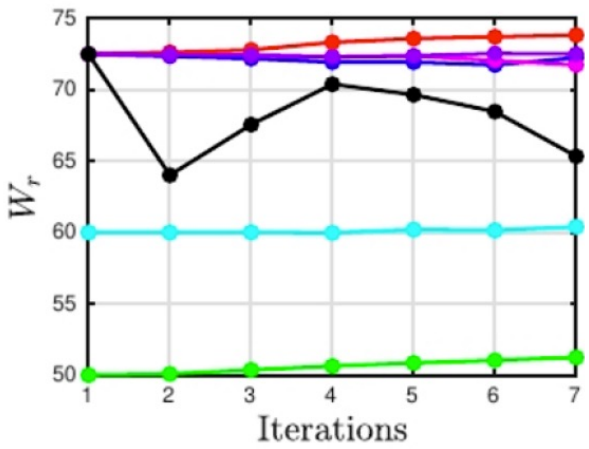


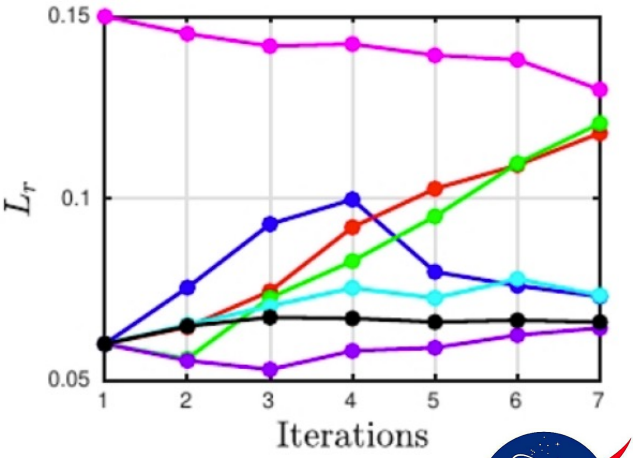


Results of Parametric Study

- Further examine the influence of H_r, W_r, and L_r on transition delay using EnVar and DNS with Python interface
- Roughness height converges to a value around H_r = 4.2 (1.1 mm or $0.31\delta_{x_0}$)
  - Most important parameter for transition delay
- Roughness width and abruptness are not very significant and converge to different values
- All simulations resulted in fully laminar flow

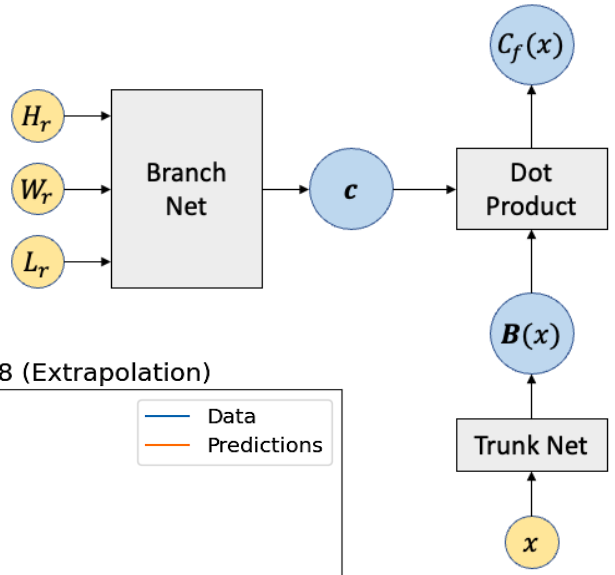


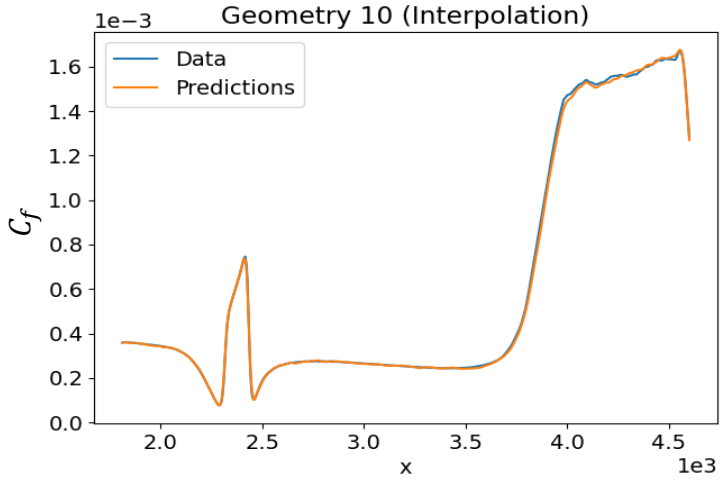


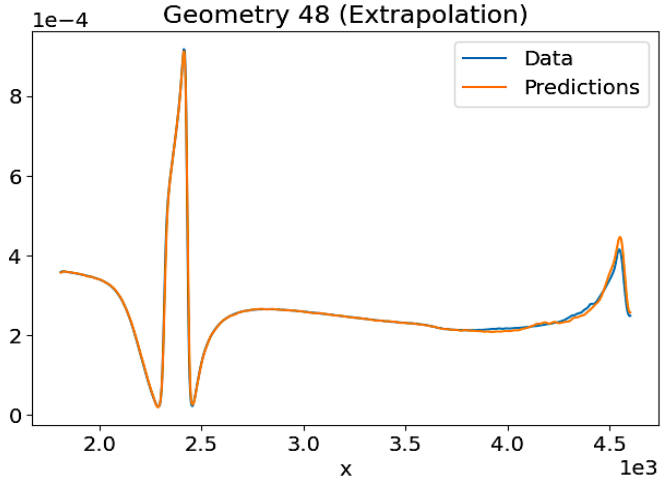


DeepONet Training

- Consists of a branch net that generates the input function space and a trunk net that generates the basis coefficients for the output
- Prediction of skin-friction distribution given low-dimensional parameterization
  - Roughness height, width, and abruptness
- 14 DNS used for training the DeepONet model
- Good agreement between DNS and DeepONet



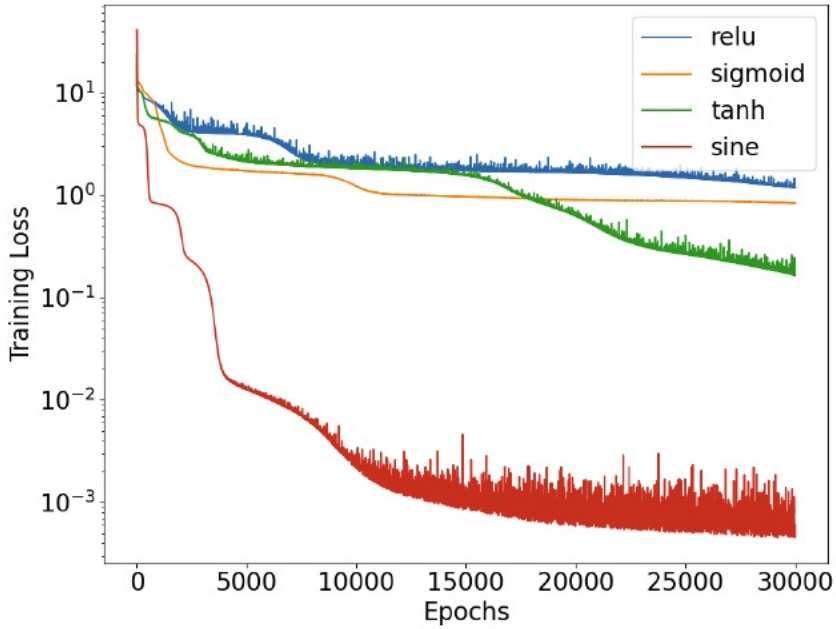




Activation Functions for DeepONet Training

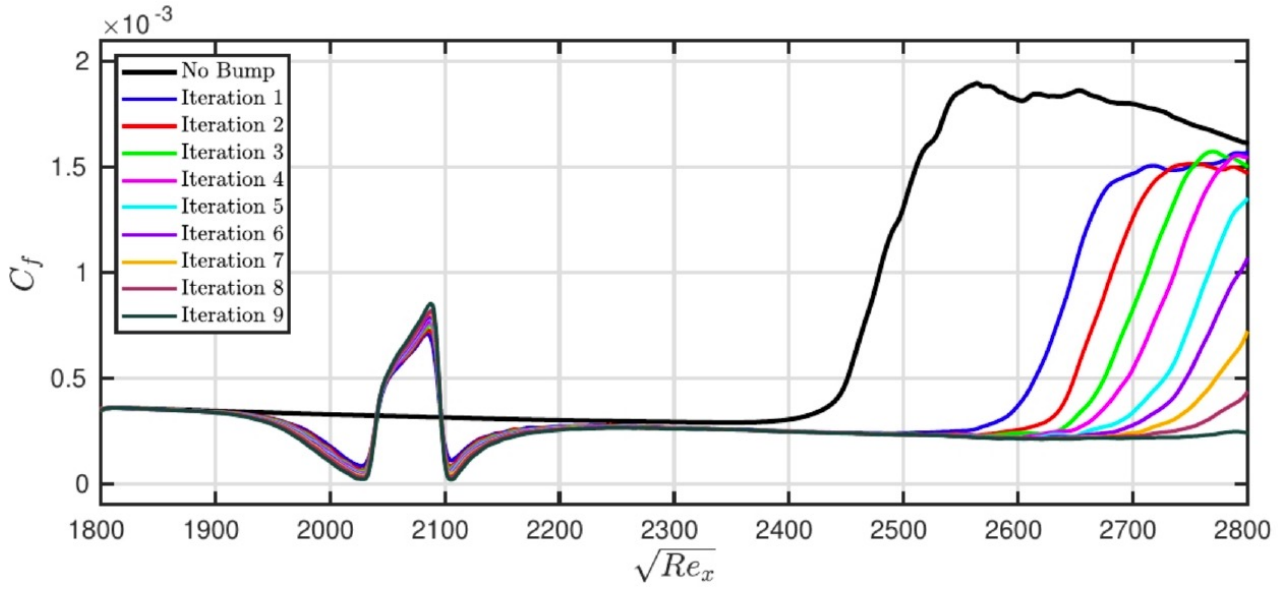
- DeepONet has 2 hidden layers with 30 neurons each
  - Sigmoid activation function is used for hidden layers in branch net
  - Linear activation function used for output layer

- Trunk net takes x as input and contains 1 hidden layer with 100 neurons
  - Sinusoidal activation function is used for trunk net
  - Outperforms relu, sigmoid, and tanh functions



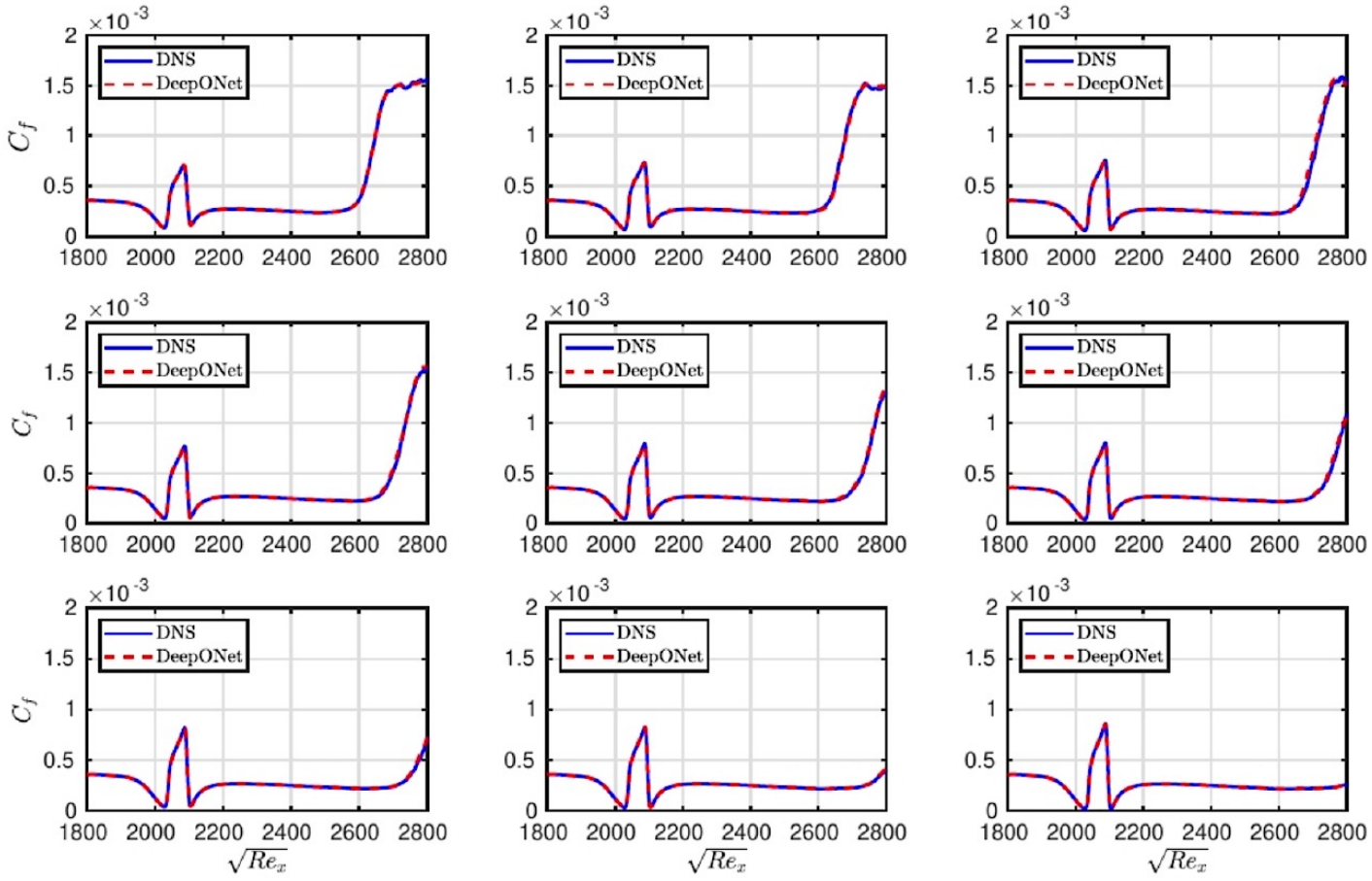
EnVar + DeepONet Optimization

- Repeat optimization problem using well-trained DeepONet model (no DNS)
  - Initial parameters: $H_r = 3.0$, $W_r = 72.5$, and $L_r = 0.06$
- Requires more iterations, but pushes transition out of domain
  - Final parameters: $H_r = 4.30$, $W_r = 73.1$, and $L_r = 0.073$
- Still working on algorithms to leverage both DNS and DeepONet to speed up convergence and computational cost



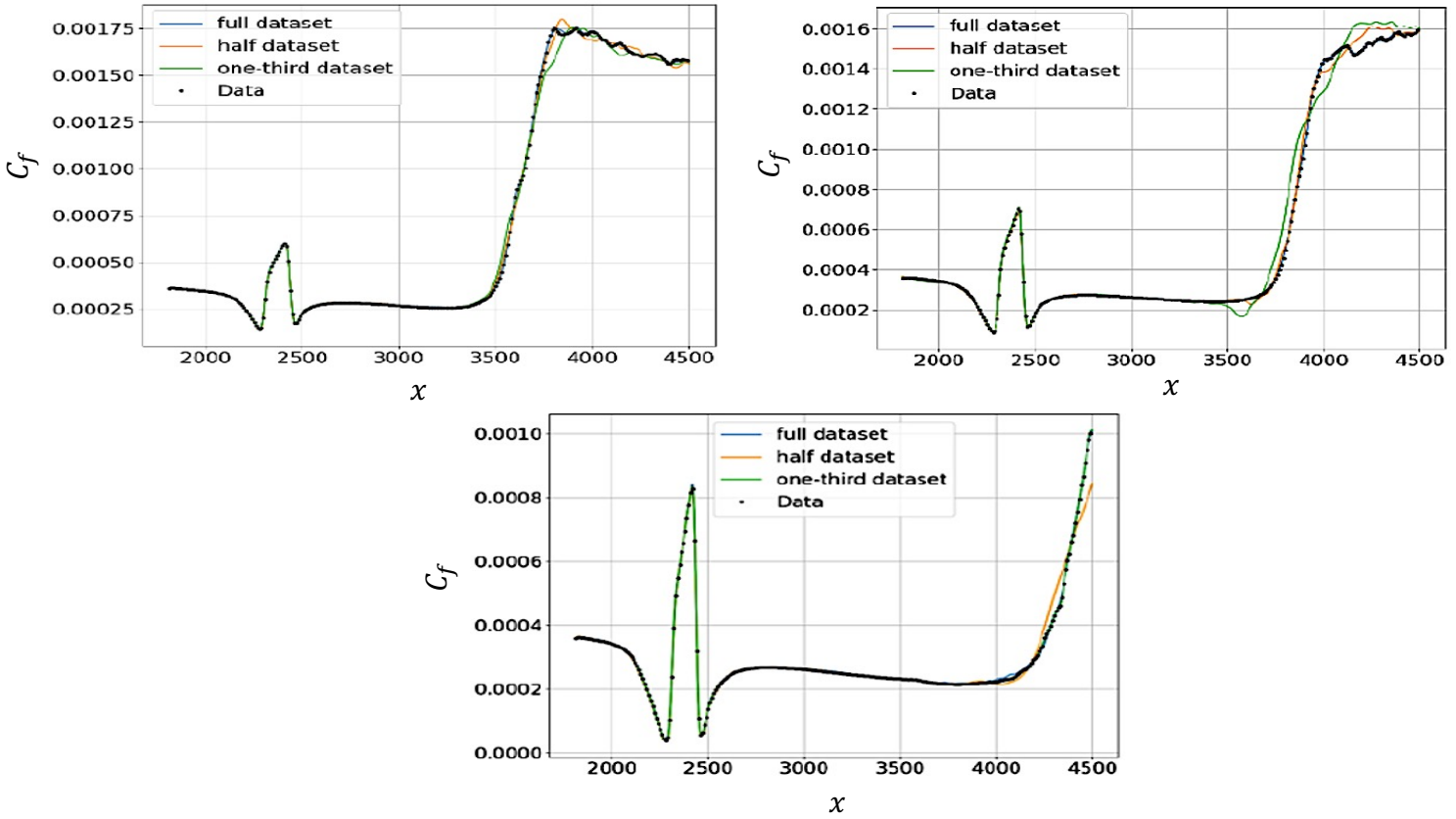
DNS and DeepONet Comparisons

- Excellent agreement between DNS and DeepONet results
  - Every case besides the initial one (top left) is outside of training data



Can the DeepONet Model be Trained with Less Data?

- 18 total DNS used for this exercise, along with one half of the full dataset (i.e., 9 DNS) and one-third (6 DNS)
- $H_r = 2.4$ to 5.8 with increments of 0.2, $W_r = 72.5$, and $L_r = 0.06$



Computational Cost of DeepONets

- 1 DNS = 15,260 CPU hours
- 1 EnVar iteration requires 6 DNS, i.e., 92,160 CPU hours
- 1 Python aerodesign procedure takes 6 iterations, i.e., 552,960 CPU hours

- DeepONet model is trained for 30,000 epochs on 1 GPU for 4 hours

- EnVar + DeepONet results are instantaneous and run on a laptop

- From the last exercise, one third of the total training dataset or 6 DNS is enough to obtain an accurate DeepONet model that can predict skin friction and high-speed boundary-layer transition

- Demonstrated computational speedup by a factor of 5 to 6 so far

- Improvements to reach a computational speedup equal to an order of magnitude will involve algorithmic improvements, extension to large degrees-of-freedom (DoF) cases, and extrapolation of same DeepONet model to different flow conditions with very minimal or no additional training

Summary and Future Work

- Aerodesign problem of delaying transition in a Mach 4.5 flat-plate boundary layer using a single roughness element parameterized by (H_r, W_r, L_r)
- Verified automated Python framework that incorporates EnVar design and DNS with results from Jahanbakhshi and Zaki 2023
- Parametric study demonstrated that roughness height was the most important parameter in terms of delaying transition
- DeepONet model allowed successful delay of transition and the skin-friction distributions agreed with the DNS results
- Computational speedup of 5 to 6 seems possible with current training process for DeepONet model compared to state-of-the-art transition predictions (Jahanbakhshi & Zaki 2023) for a 2D boundary layer
- Ongoing efforts aim to improve the optimization algorithm to better leverage DNS and DeepONets, while extending to more complex geometries with larger number of design variables (e.g., cone with more general surface topography)

Acknowledgments

The computations described in this presentation were carried out in support of the NASA Langley Research Center FY23 CIF/IRAD Program under “Machine-Learning-Based Reduced-Order Modeling for Aerodynamic Design”.

We appreciate computational resources from both the NASA Langley Research Center K Cluster and the NASA High-End Computing Program through the NASA Advanced Supercomputing Division at the NASA Ames Research Center.