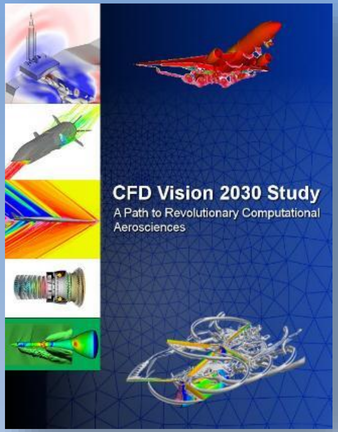
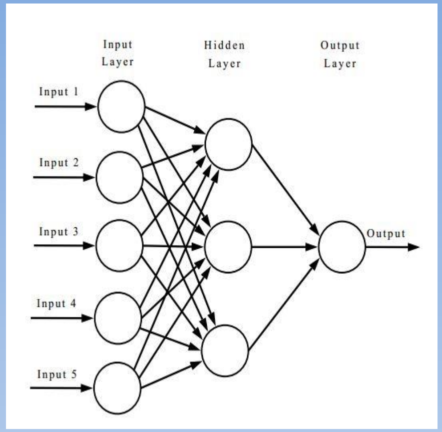
CFD Vision 2030, and the Potential for Machine Learning



Mujeeb R. Malik
NASA Langley Research Center
Hampton, VA 23681



NSF Workshop on Exuberance of Machine Learning in Transport Phenomena

Dallas, TX
February 10-11, 2020

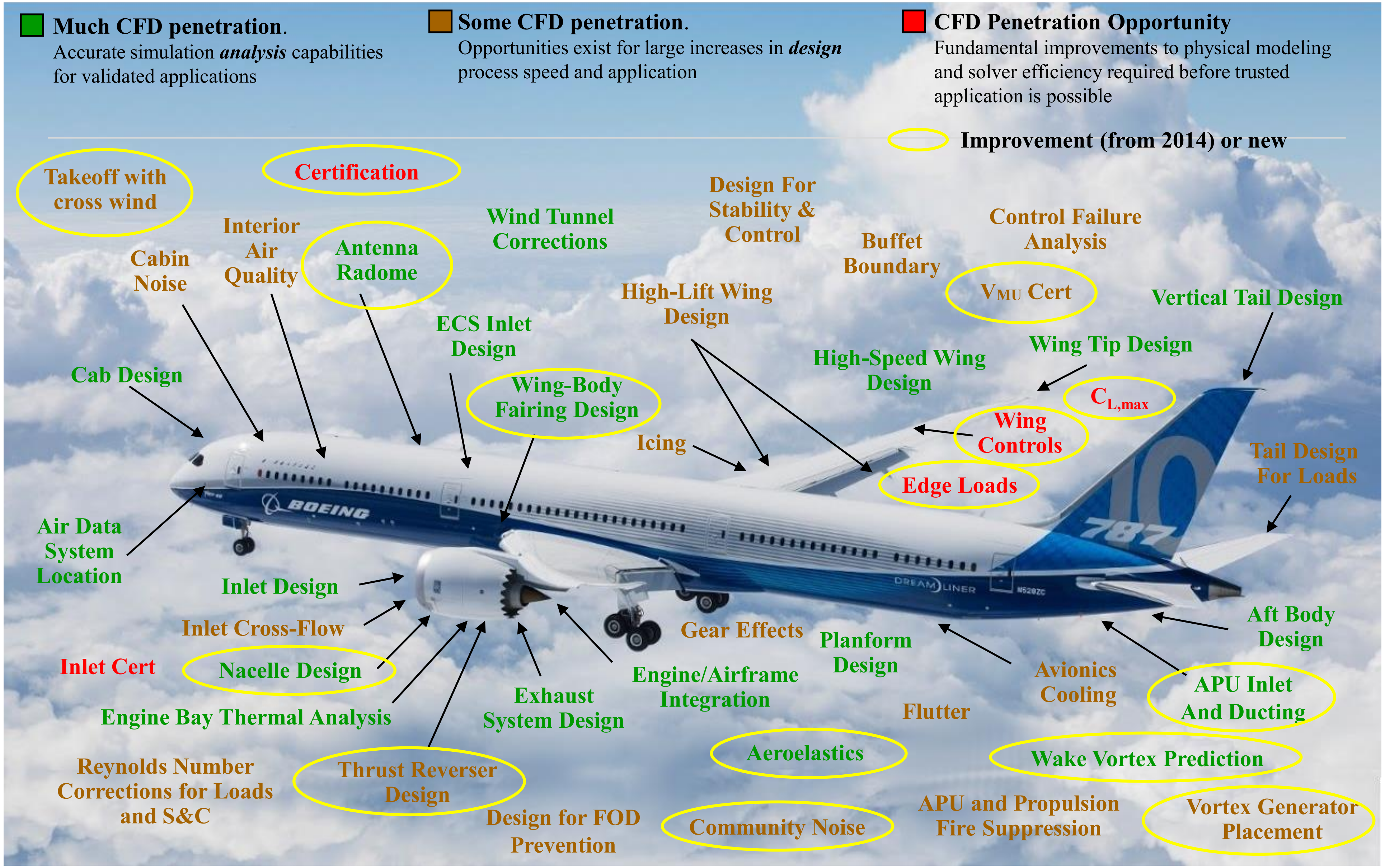
Outline

- **Introduction**
  - Impact of CFD on aerospace vehicle design
  - NASA's CFD Vision 2030
- **Role of machine learning (ML) in physical modeling**
  - Turbulence modeling
  - Transition modeling
- **Announcement of a NASA turbulence modeling/ML workshop**
- **Summary**

Acknowledgements

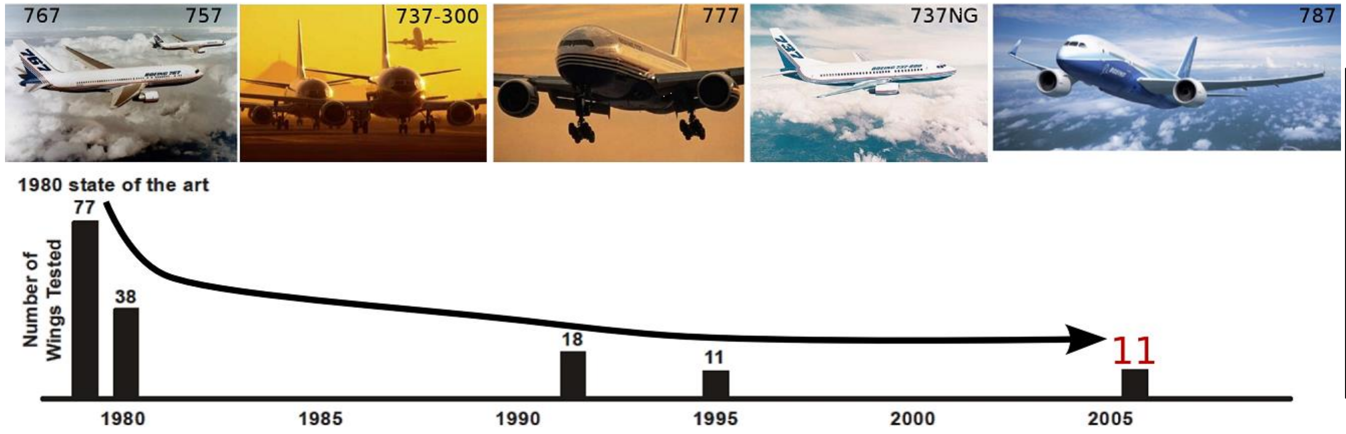
- **Turbulence**
  - Chris Rumsey
  - Gary Coleman
- **Transition**
  - Chau-Lyan Chang
  - Pedro Parades
  - Meelan Choudhari

CFD is used for virtually every airplane configuration component



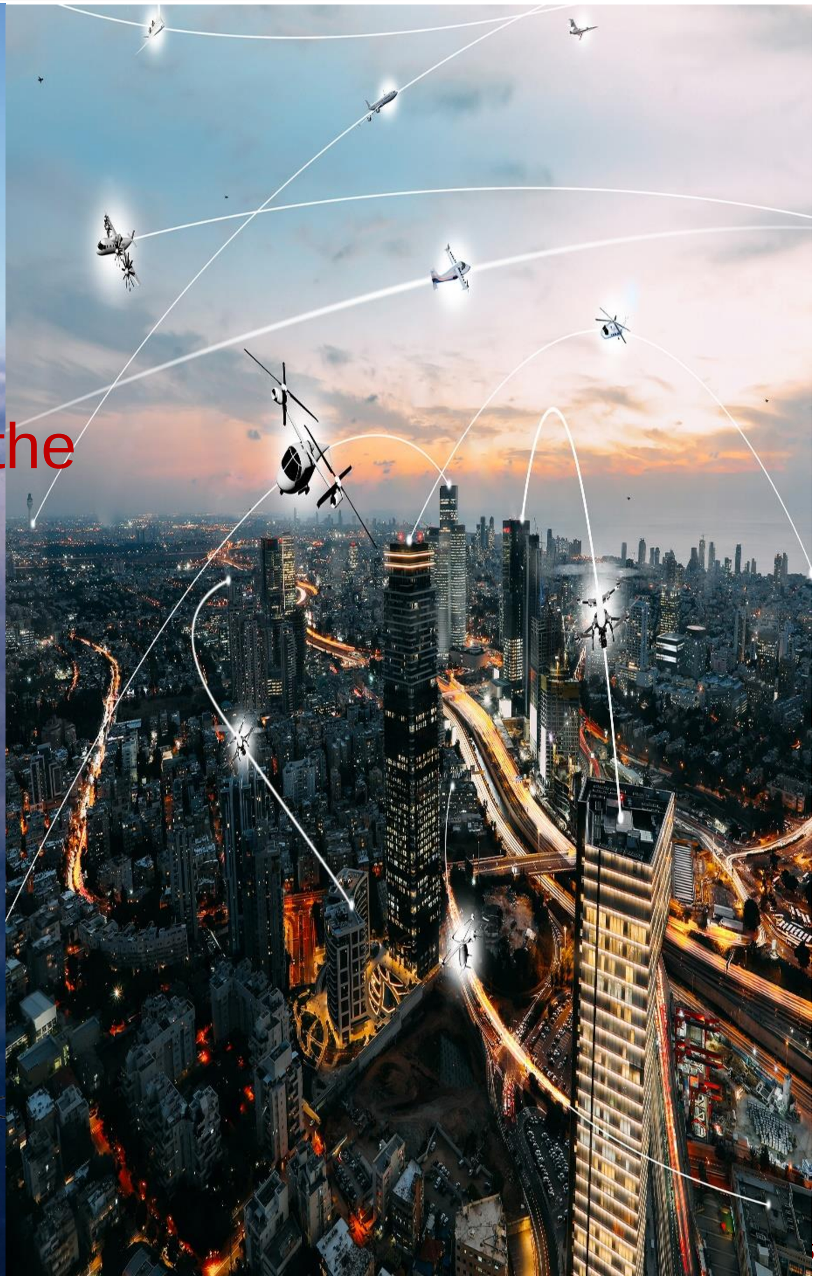
Courtesy, Boeing

There is 50+ years of experience designing similar planes…



- Number of tests continually decreased due to computational tools and experience base with same type of configurations
- Further reduction in number of tests, and design of novel configurations, requires overcoming of the deficiencies in computational tools

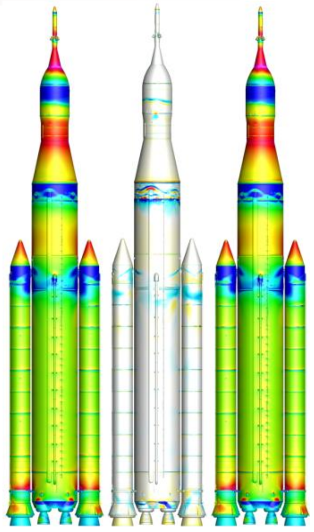
Air Vehicles of the Future, Much More Integrated Need Prediction Capability for NASA Aeronautics

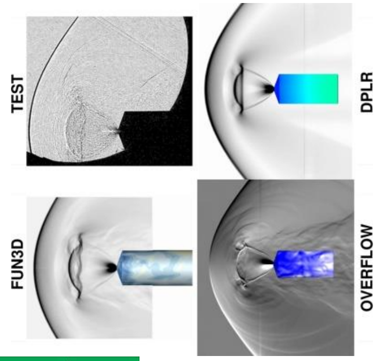


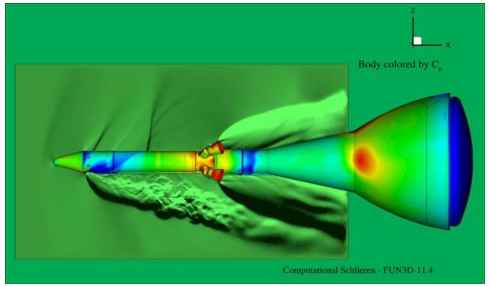
Prediction of separated flows is the key technical challenge

CFD Applications to Space Missions

- **Key Applications** :
  - Launch vehicles
  - Entry, Descent, & Landing (Earth, Mars, etc.)
  - Launch environments (acoustics, heating, vibration, etc.)
  - Spacecraft Decelerators
  - Separation events
  - Supersonic Retropropulsion
- **CFD Challenges:**
  - Separated flows, *plus…*
  - Aero/plume interaction
  - Aerothermal predictive capability (with chemistry at extreme conditions)
  - Fluid/structure interactions (parachute systems)



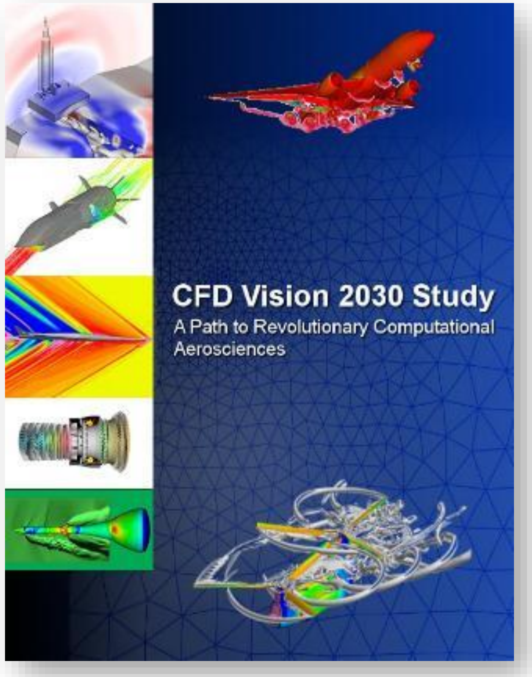




CFD Vision 2030 Study

- **The Study Objective** was "…to provide a knowledge-based forecast of the future computational capabilities required for turbulent, transitional, and reacting flow simulations…"

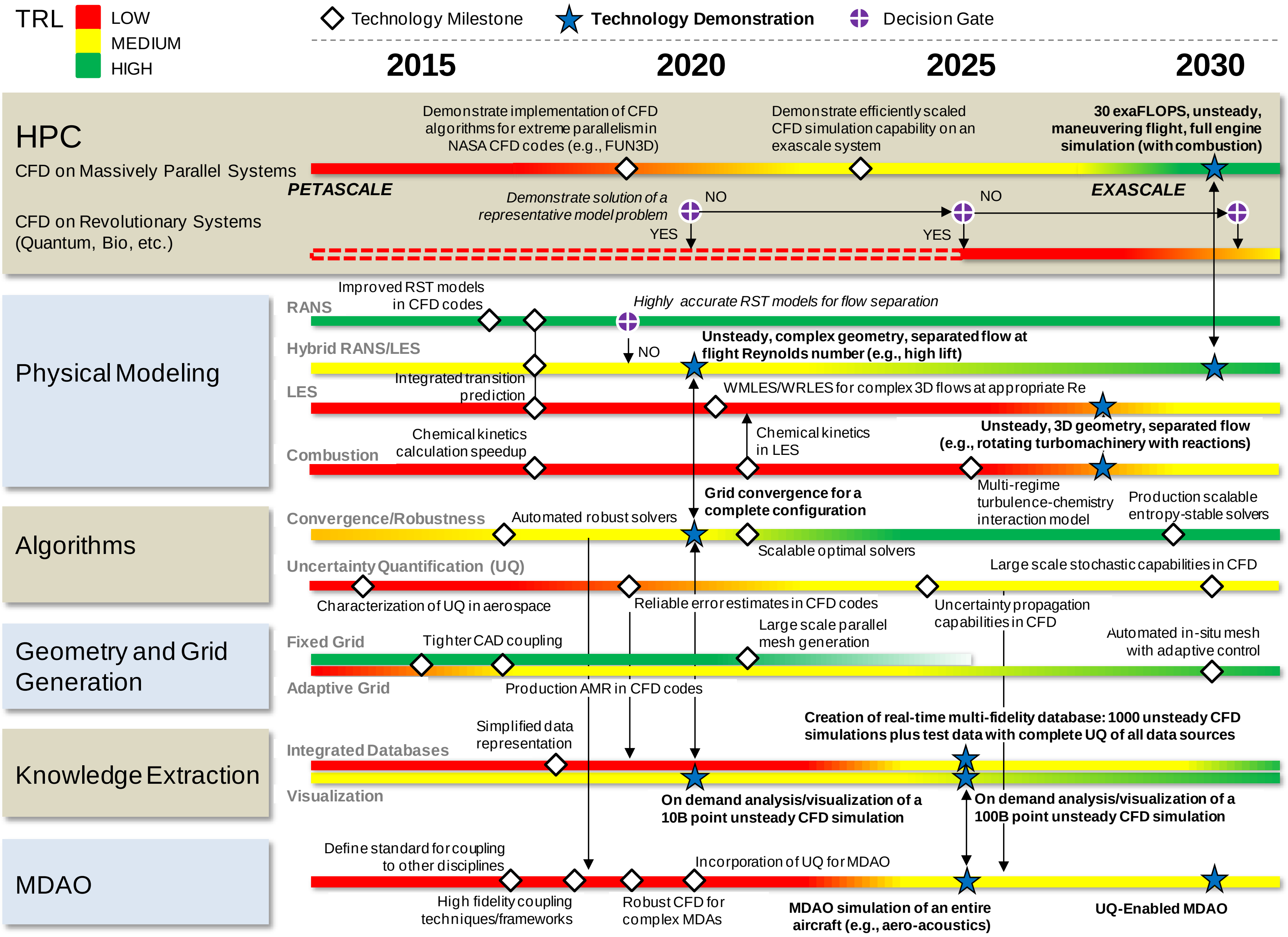
- "Perhaps the single, most critical area in CFD simulation capability that will remain a pacing item by 2030 in the analysis and design of aerospace systems is the ability to adequately predict viscous turbulent flows with possible boundary layer transition and flow separation present." (The Vision 2030 Report)



NASA CR 2014-218178
Report published: 3/19/2014

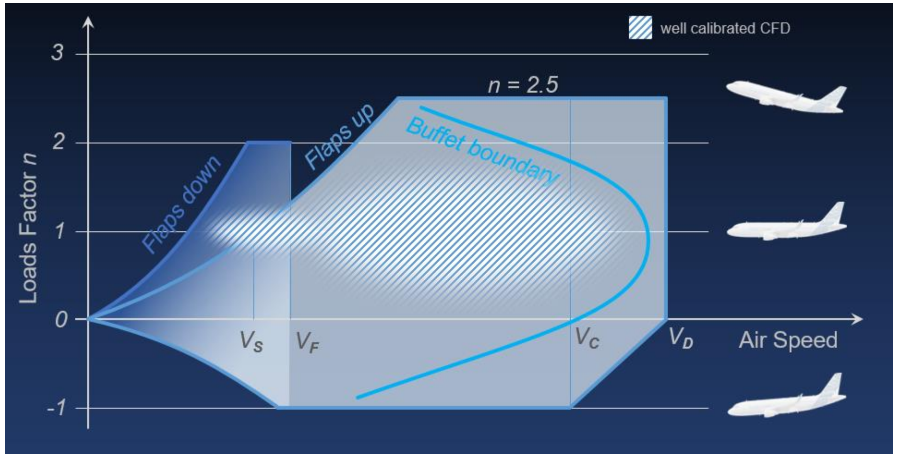
Report available at:
https://www.nasa.gov/aeroresearch/programs/tacp/ttt/cfd-vision-2030-study

Technology Development Roadmap



CFD Technical Challenge

- “Develop and demonstrate computationally efficient, eddy-resolving modeling tools that predict maximum lift coefficient ($C_{L,max}$) for transport aircraft with the same accuracy as certification flight tests” (Level 1 milestone due 9/30/2025).
  - High-lift certification flight tests account for roughly 2/3 of flight certification test points
  - 50% of flight test points replaced by computations would save about $300 million from each aircraft development program
- Promise to expand capability of CFD to predict unsteady phenomena such as stall, buffet and flutter
- Eddy-resolving methods require an order of magnitude more computational resources than RANS



Courtesy, Boeing

- CFD has been calibrated only in **relatively small regions of the operating envelope** where the external flow is well modeled by current RANS methods
  - High-speed cruise (aero design)
  - Low-speed at nominal attitude with moderate flap settings

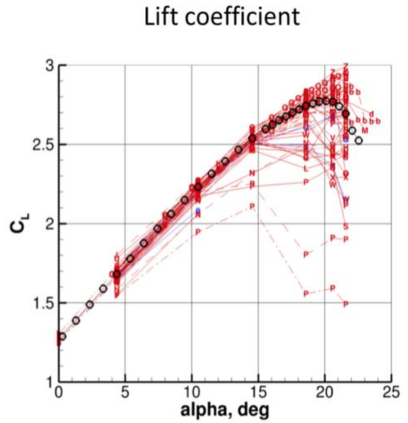
Euler Equations (inviscid)	*Reynolds-averaged Navier-Stokes: “RANS” (turbulence modeled)*	*Large-Eddy Simulations: “LES” (large scales resolved)*	*Direct Numerical Simulations: “DNS” (all scales resolved)*

Increasing physics, increasing cost →

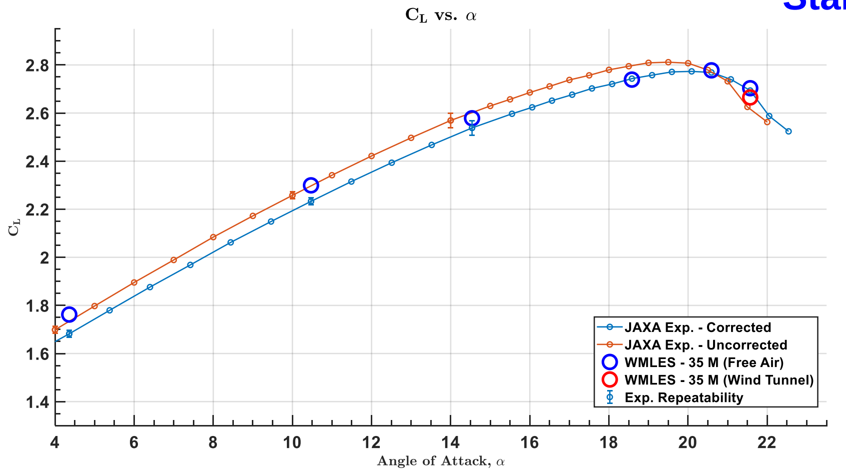
$C_{L,max}$ Prediction for JAXA Standard Model

Wind Tunnel: Lift Coefficient

Including the wind tunnel in the calculation recovers the observed experimental trend in C_L



RANS

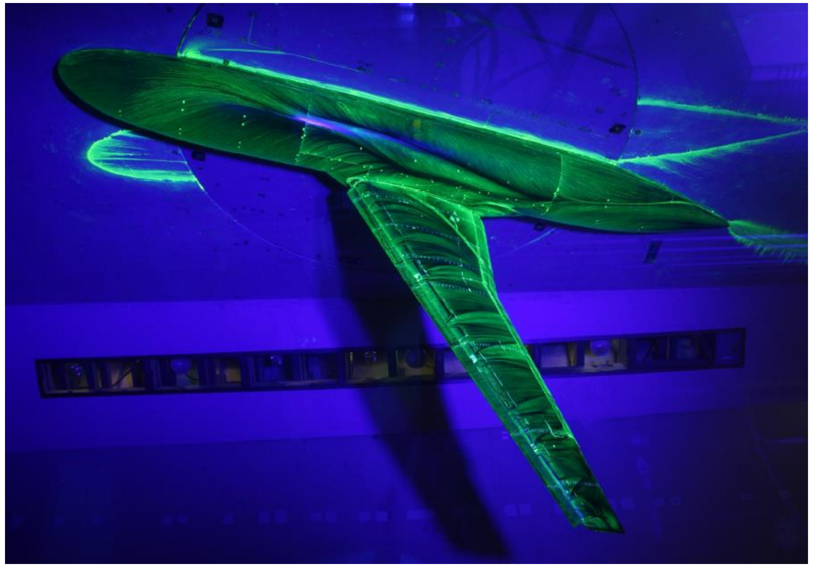


WMLES

Work underway to further evaluate and mature WMLES approaches for NASA applications

Stanford NRA

Wind Tunnel Simulation



Wind Tunnel Experiment

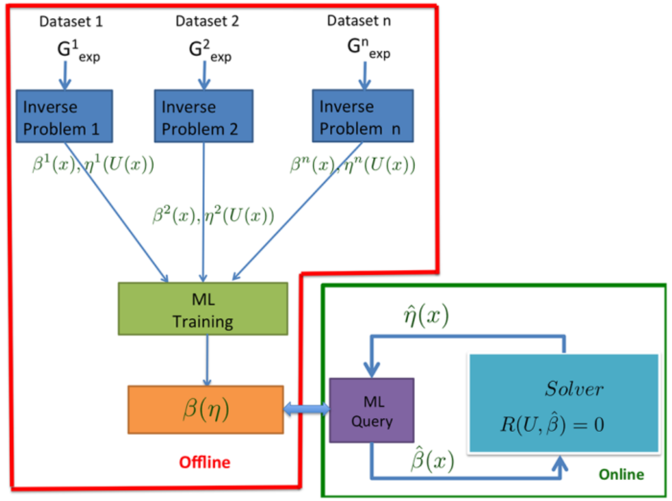
Motivation/Potential for Machine Learning (ML) in Prediction of Turbulent Flows

- Current RANS turbulence models are generally unreliable for predicting separated flows
  - Accurate prediction of separation is required for $C_{L,max}$ and for many off-design conditions
  - Progress in RANS has largely stagnated in this arena

- Data-driven/ML methods are currently emerging as a possible path forward for RANS
  - They attempt to use the capabilities of machine learning to search for patterns that may enable RANS turbulence model improvements
  - There has been a significant recent uptick in worldwide research in machine learning for CFD modeling
    - e.g., EU's HiFi-TURB Project (initial emphasis on big data analytics: feature detection and advanced analysis of LES/DNS data)
- If successful, these methods may provide improved models that will run at a fraction of the cost of scale-resolving simulations
  - Even relatively minor improvements to existing RANS models could mean a great deal to the aerospace field

Current Strategy

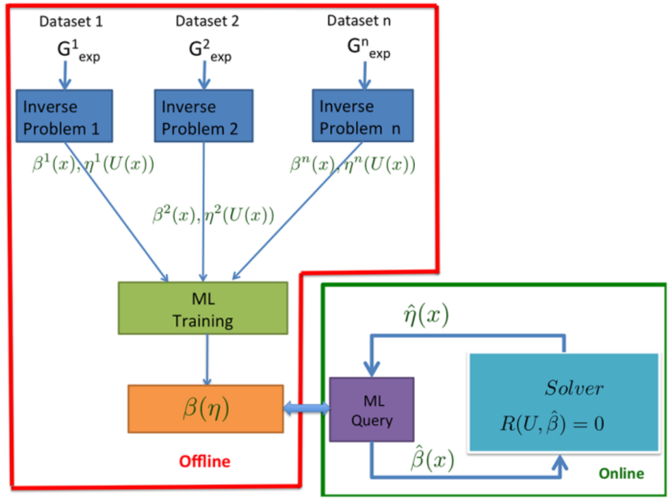
- NASA funded University of Michigan to explore potential of ML for turbulence modeling improvement: Singh, Medida, and Duraisamy (AIAA J 55(7):2215-2227, 2017)
  - Field inversion (find an optimized "β" field that improves the Spalart-Allmaras (SA) model for given training cases)
  - Machine learning (use the training cases to learn β as a function of local nondimensional field variables)
- Current NASA (under RCA) effort is to internalize and evaluate the above approach
- Make use of NASA CFD code FUN3D's existing adjoint capability to perform the field inversion
  - Implement new adjoint capability in FUN3D to find β
- Leverage existing Python and/or Matlab/Octave machine-learning tools toward development of a new SA-ML model version
- Note: there are other ML-based efforts described in the literature
  - This particular approach was chosen to make a start; there is no guarantee that it is the best path forward



Singh et al. (2017)

Field Inversion Step

- Baseline SA model:
  - $\frac{D\nu}{Dt} = P(\nu, U) - D(\nu, U) + T(\nu, U)$
- Introduce a multiplier (β) on the production term:
  - $\frac{D\nu}{Dt} = \beta(x, y, z)P(\nu, U) - D(\nu, U) + T(\nu, U)$
  - β is a new field variable that exists at every grid point
  - (Location of β on the RHS does not matter, since it is the total RHS imbalance that defines ν)



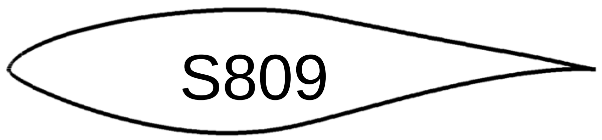
Singh et al. (2017)

- Find β by minimizing an objective function. Currently using C_l for airfoil cases:
  - $\min[C_{l,\,specified} - Cl(\beta)]^2$ (this objective function is already available in FUN3D)
    - ➢ extend FUN3D adjoint capabilities to enable calculation of sensitivity derivatives, df/dβ
    - ➢ linearize the flow residual with respect to the field design variable, β
    - ➢ implement steepest descent optimization method to optimize β (with positivity constraint)
  - perform gradient-based optimization for various flow conditions

Current status

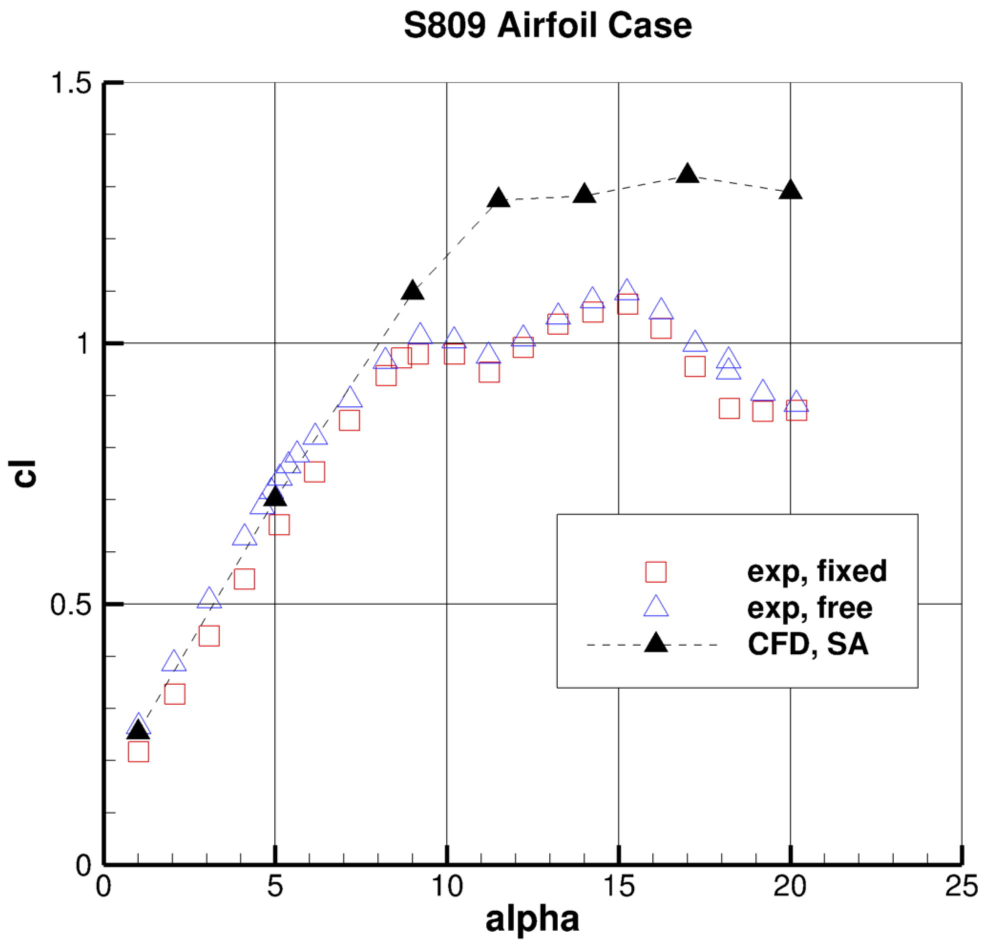
- Field inversion step for SA-ML has been implemented in FUN3D
  - Test case: S809 2-D airfoil (same as used by Singh et al. (2017))

- Machine learning step (which occurs outside of FUN3D) has been initiated very recently
  - It is taking time to come up to speed in this new area



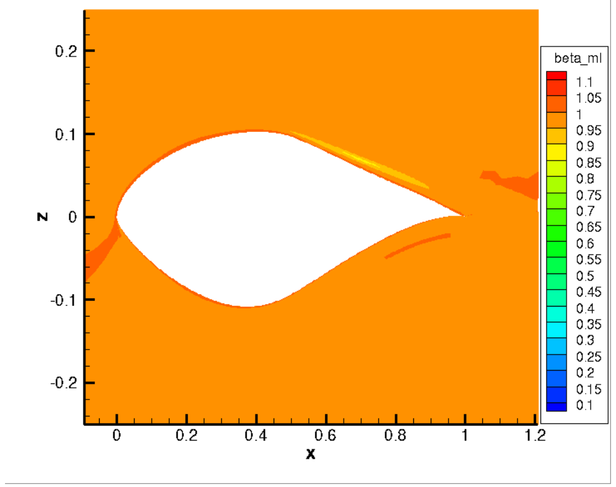
Wind turbine blade

Goal: improve model predictions at high alpha (separated flow)



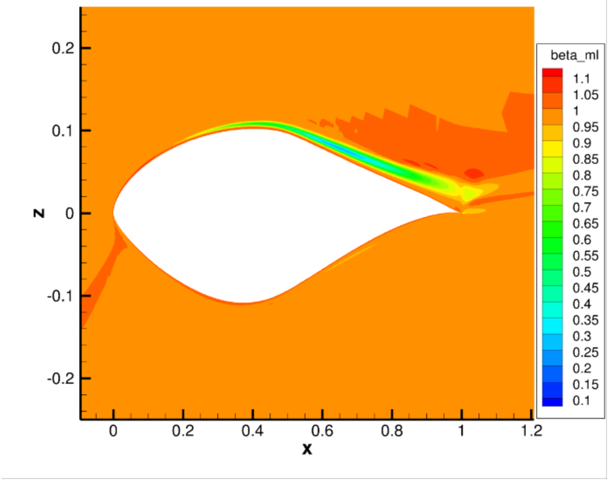
Computed β Fields

AoA=5 deg



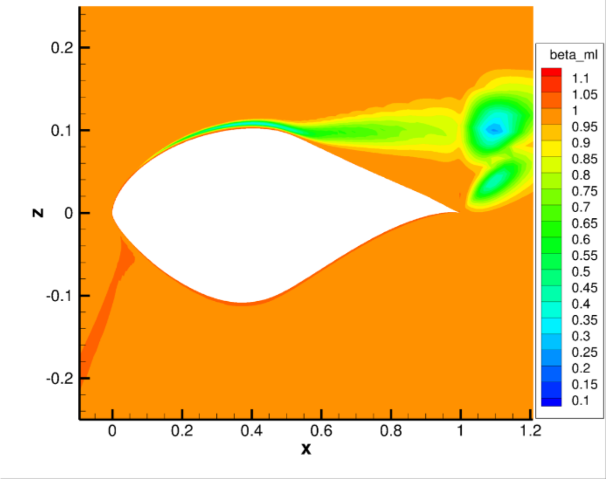
min=0.80, max=1.02

AoA=9 deg



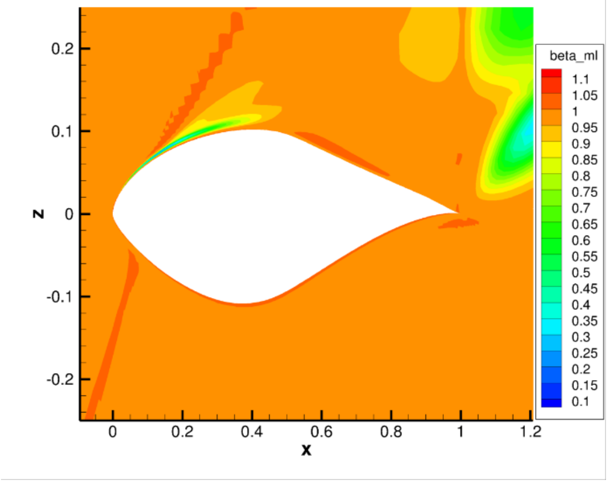
min=0.24, max=1.13

AoA=14 deg



min=0.23, max=1.12

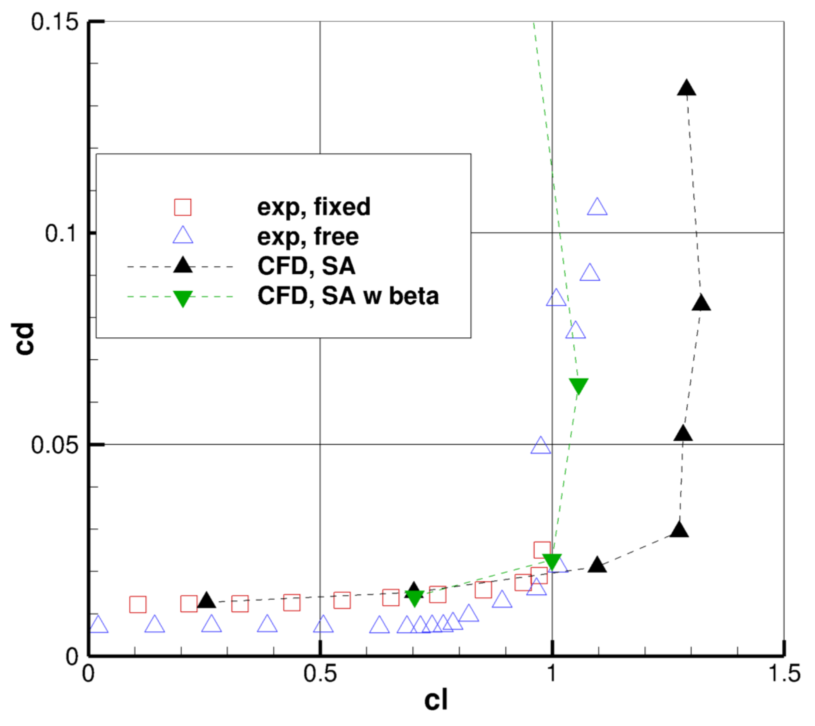
AoA=20 deg

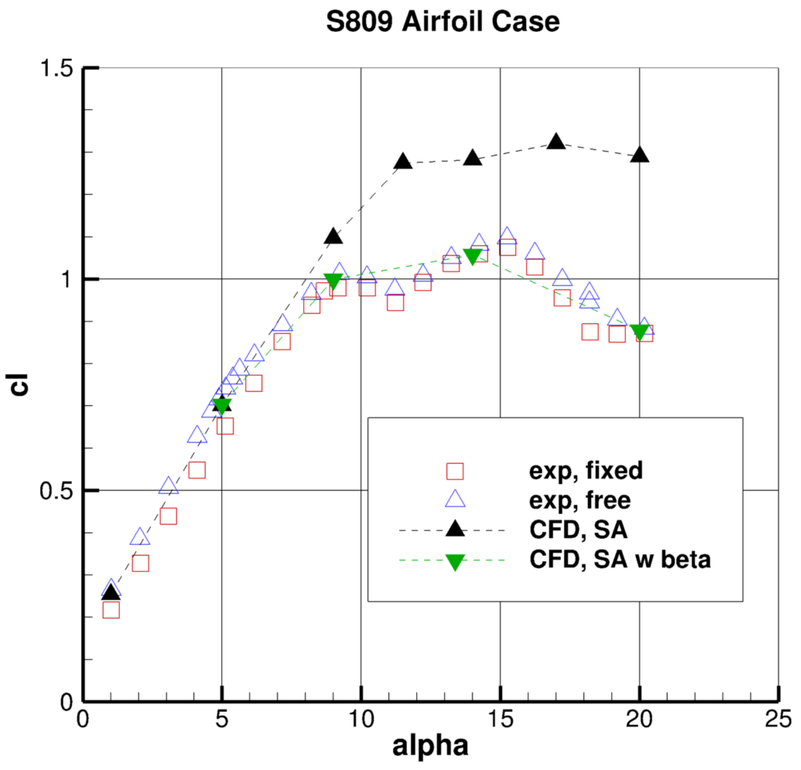


min=0.21, max=1.25

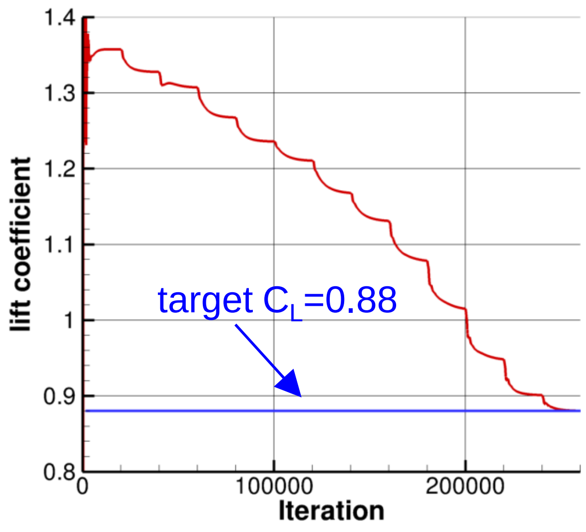
Results

- C_l in good agreement with the experimental results
- Even the drag prediction significantly improves



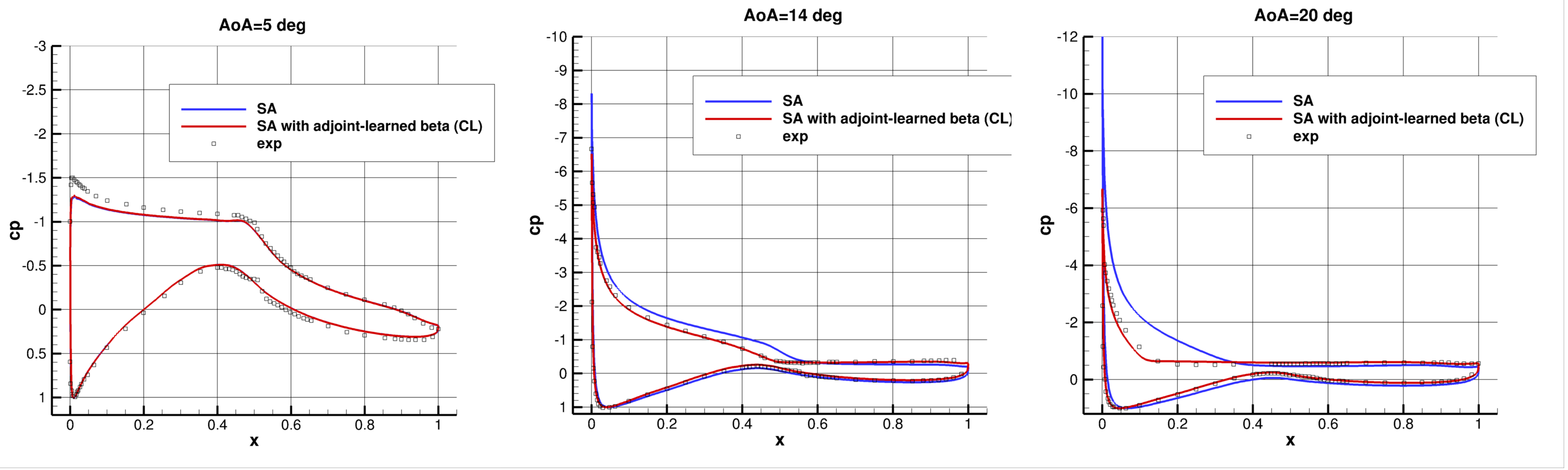


C_L progress toward target



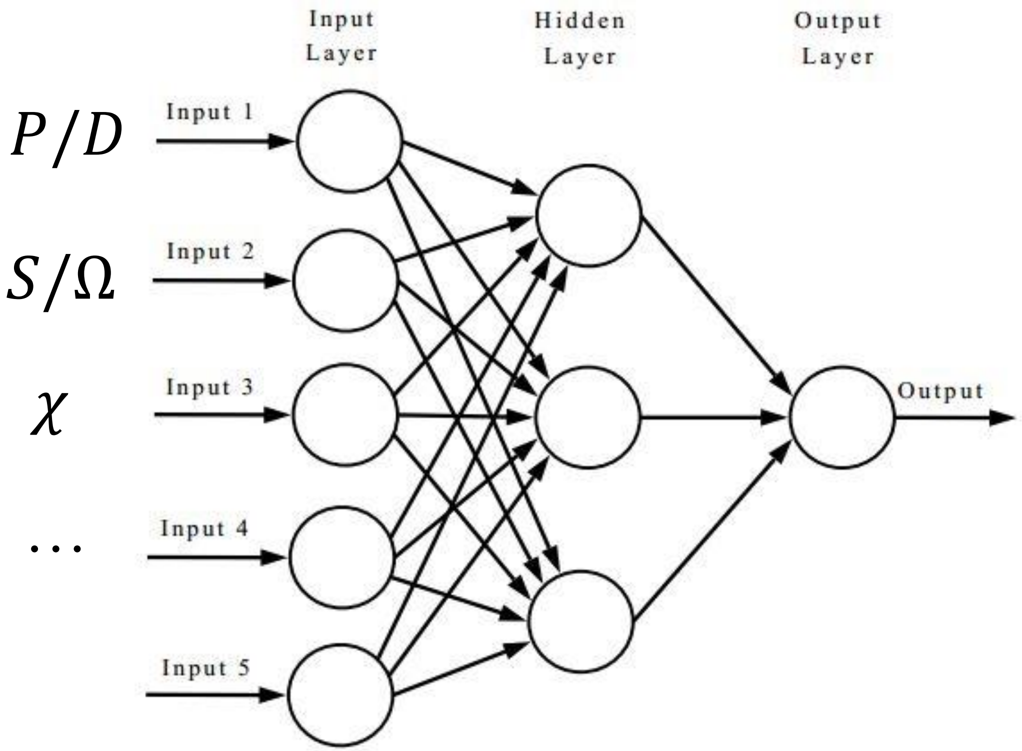
C_P Results

- C_P significantly improves for large angles of attack
- As expected, no change in C_P in the linear (lower AoA) regime



Machine learning step (preliminary)

- Select nondimensional field “features”
  - e.g., P/D, S/Ω, χ, …
- Employ a neural network (NN) algorithm to “learn” β as a function of these features



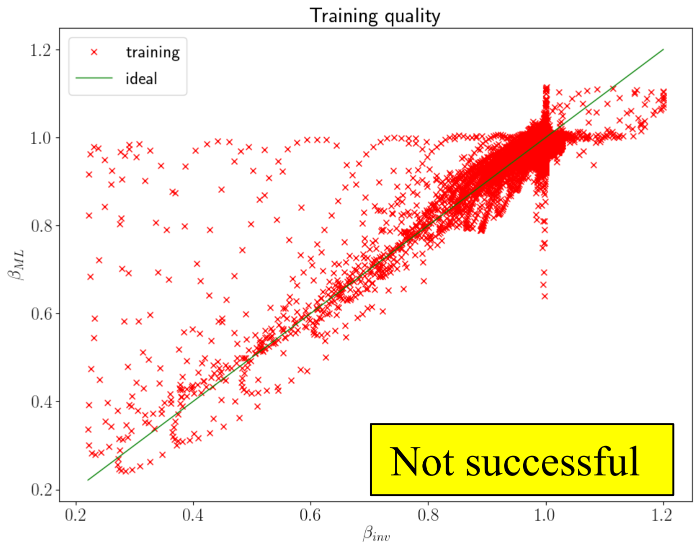
Choices:
- Which/how many training cases to use?
- Which features to use?
- How many hidden layers?
- How many nodes per layer?

Currently somewhat of an art

Machine Learning Step (preliminary)

- Features selected for this airfoil case:
  - D/P, $\frac{\rho\Omega d^2}{\mu_t+\mu}$, χ, d/l_{ref}, $(u'v')/(U_{ref}^2)$
  - Partly from experience, partly at random
- Masked β values located beyond *d*=1 from body
- Normalized and shuffled all features
- Used NN with two hidden layers and 20 nodes in each layer
- Using python based ML tools
- Very early stages of implementation in FUN3D, work in progress
- Note: Singh et al. (2017) did successfully construct a predictive model that computed stall in good agreement with experimental data at a Reynolds number not used in the NN training (see next slide)

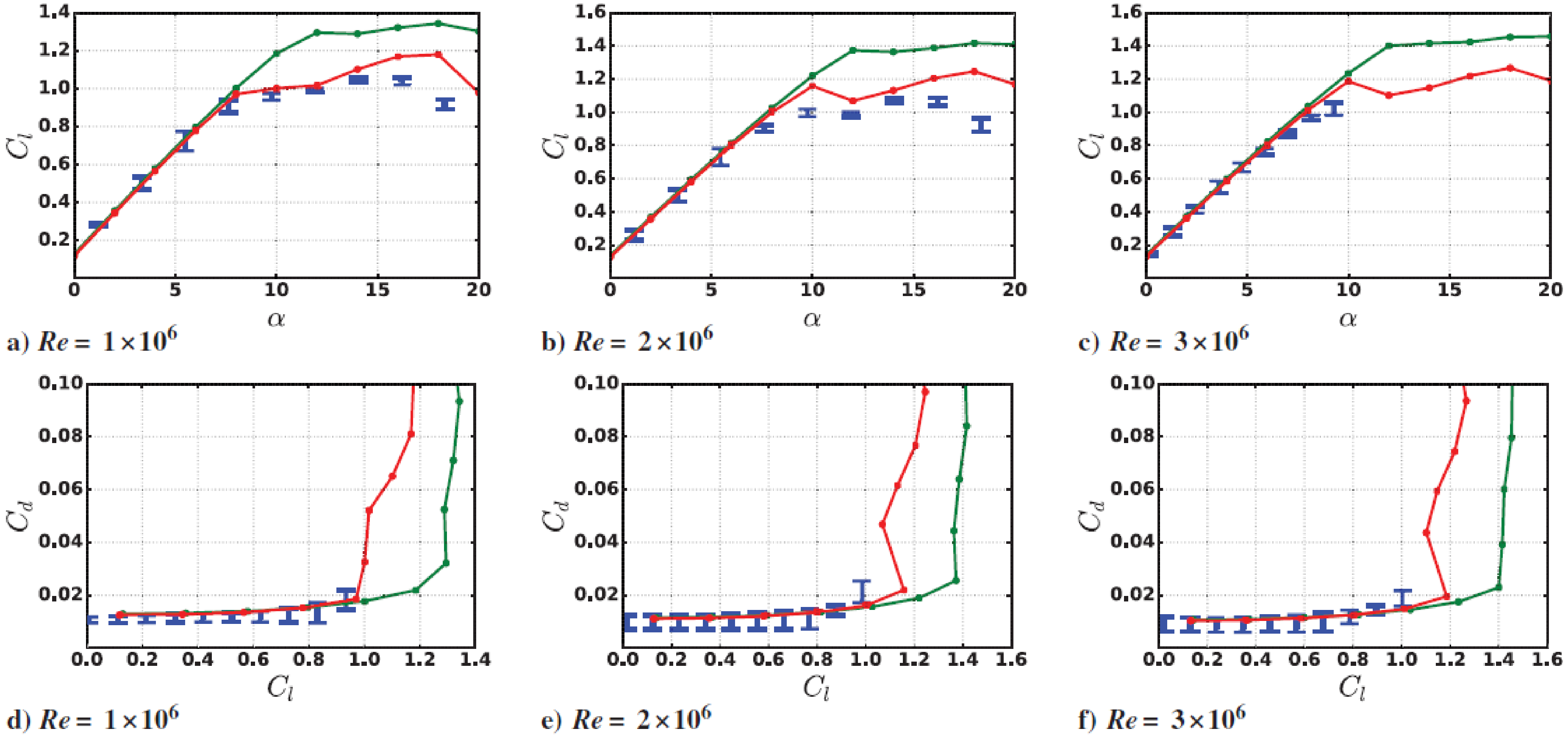
Results trained on one case (AoA=20 deg)



Results trained on two cases (AoA=5 and 20 deg)

Comparison of Base SA and NN Model

Singh et al. (2017)



Green: Base SA Model

Red: NN Model

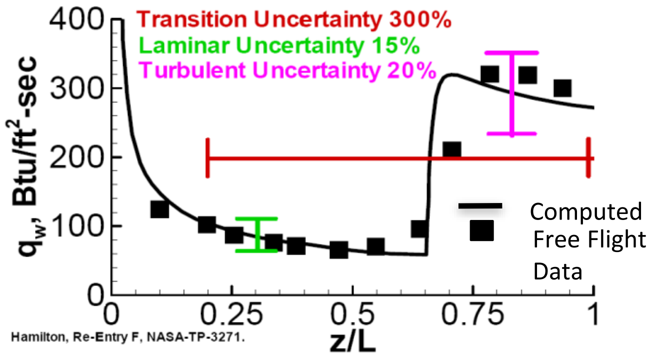
Note: The 3 million Re case was not included in the training data

Turbulence Modeling and ML: Next steps

- Implement neural-network-learned model back into FUN3D for predictive step
- Open questions
  - How does the choice of features, hidden layers, nodes, etc. influence the training?
  - What causes success/failure in the training step?
  - Which features are most/least important?
  - How does one decide when a trained model is “good enough”?
  - How “universal” is a given trained model?
- Where does the data come from?
- Testing, testing, testing
- Gain experience, and possibly see path toward new ideas

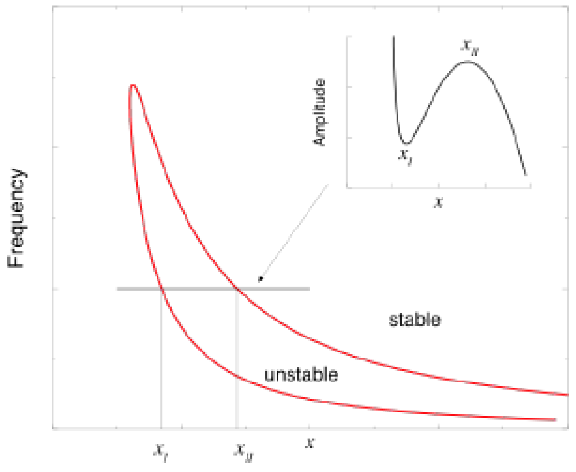
Boundary Layer Transition

- Transition important for design of aerospace vehicles
- It is difficult to predict, because myriad factors influence onset
- For low disturbance environments, e^N offers a good empirical approach
  - Compute accurate mean flow and derivatives
  - Analyze flow field using linear stability theory (LST)
  - Compute disturbance growth rate, and integral growth ratio (e^N)
  - Different modes of instability
    - Tolmien-Schlichting (TS)
    - First mode
    - Second mode
    - Crossflow
    - Gortler

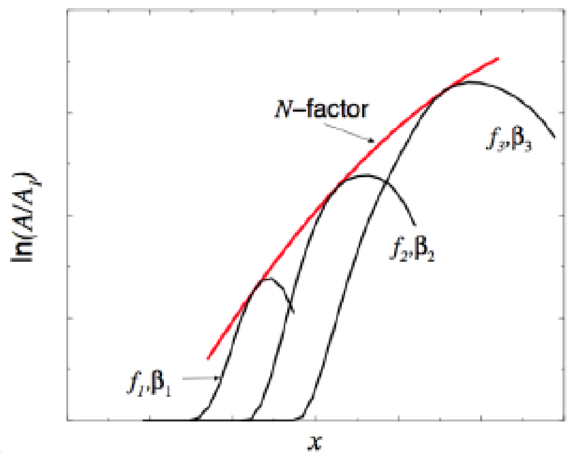


Schneider, Prog. in Aerospace Sciences, 40, 2004

- Neutral instability curve

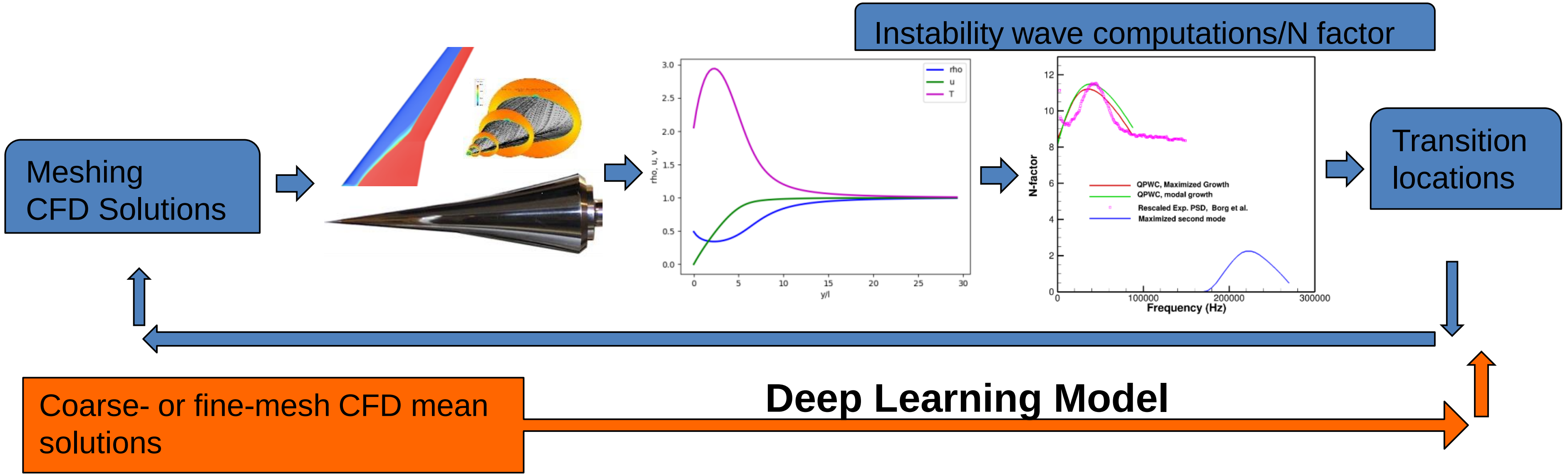


- *N*-factor envelope



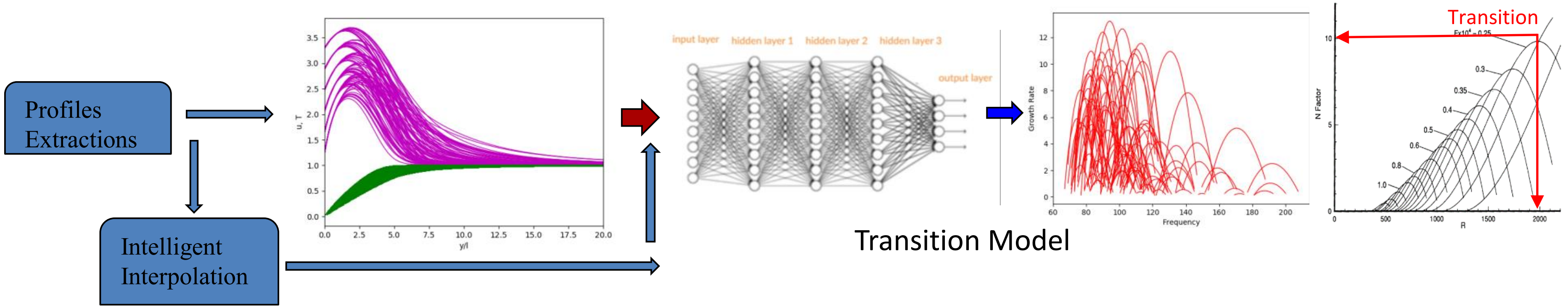
$$N = \max_{f,\beta}\left[\ln\left(\frac{A}{A_I}\right)\right]$$

Physics-Based Transition Prediction Paradigm



- State-of-the-art physics-based transition prediction solvers
  - Stringent mesh resolution requirements
    - Need accurate wall normal profiles, as well as derivatives
  - Instability wave solver requires expert knowledge
  - Multiple iterations may be needed to re-mesh and/or recompute instability waves to assure convergence
- Need an ML model to encompass the above process
  - Robustly and very efficiently compute N factor and transition onsets even with relatively coarse CFD mesh

Deep Learning Model for Transition Onset Prediction



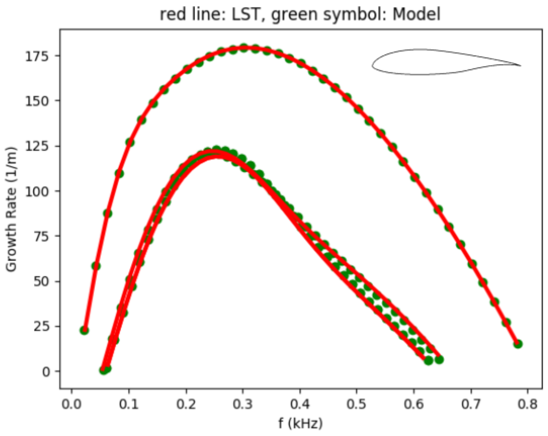
- Profile extraction by coupling CFD codes with a dynamically linked library (DLL)
  - Under-resolved boundary layers need intelligent interpolation
- Vector inputs $[\mathrm{y}, \rho, u, v, w, T, Re_x, \frac{dp}{dx}, M_\infty \ldots]$ to the deep learning model (typically 3-6 layers)
- Vector outputs $[f, \sigma]$ represents instability wave spectra for the given profile input
- Simple integration (along streamlines) routines to map N factors on the body surface
  - Transition locations determined and RANS models turned on accordingly
- Training data using LASTRAC for instability wave computations with meanflows
  - From boundary layer codes for weak inviscid-viscous interactions
  - From Navier-Stokes codes for strong inviscid-viscous interactions (e.g. blunt cones)
- Neural Network models
  - TensorFlow 2
  - Keras library

Number of mean flows computed for training data (200-400 stations):
- Blunt cone 160
- Sharp cone 15
- Flat plate 30
- 2D wing 40
- 3D wing 45

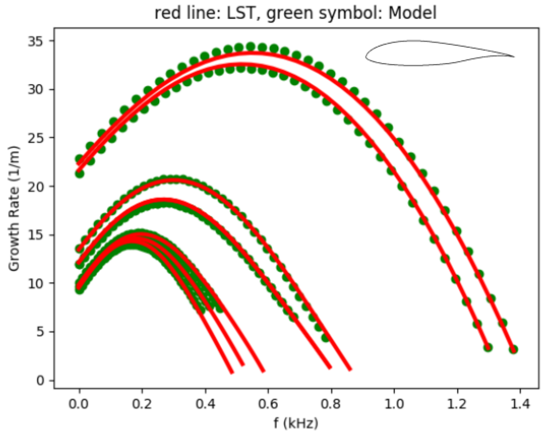
Developed Transition Model Predictions for Various Instability Mechanisms

- Model predictions compared against LASTRAC linear stability theory (LST) solutions

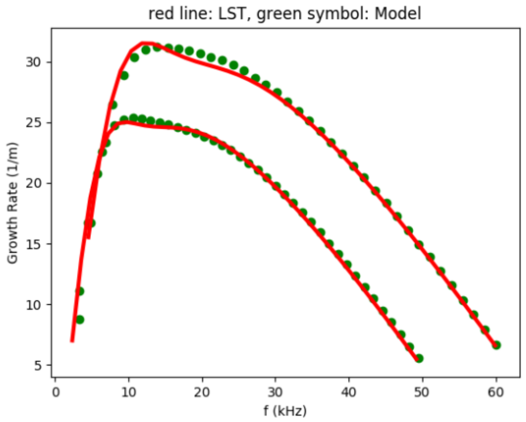
Lines: LST solutions
Symbols: Model predictions



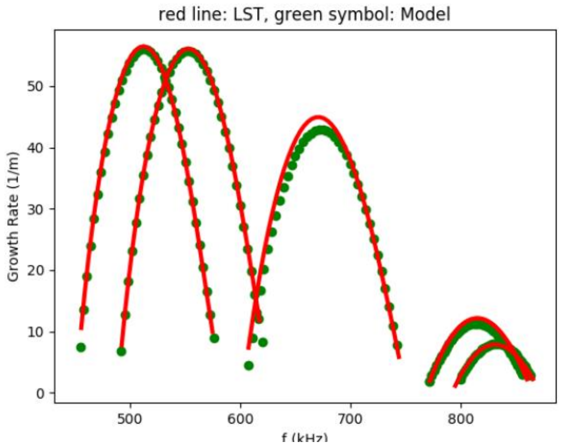
TS wave: incompressible flow over a 2D wing



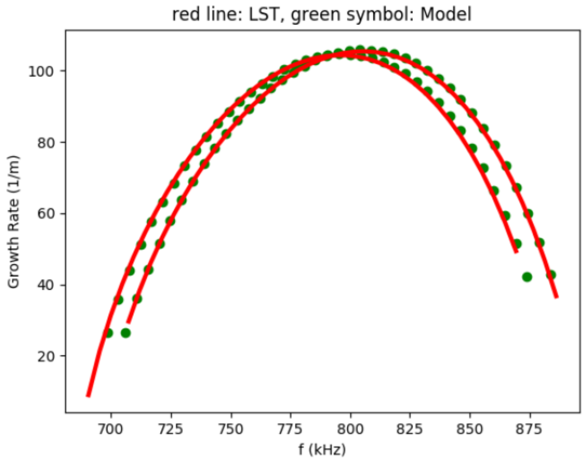
Optimized **cross flow instability** : 45° swept wing



Optimized **first-mode instability** : Mach 2 flat plate

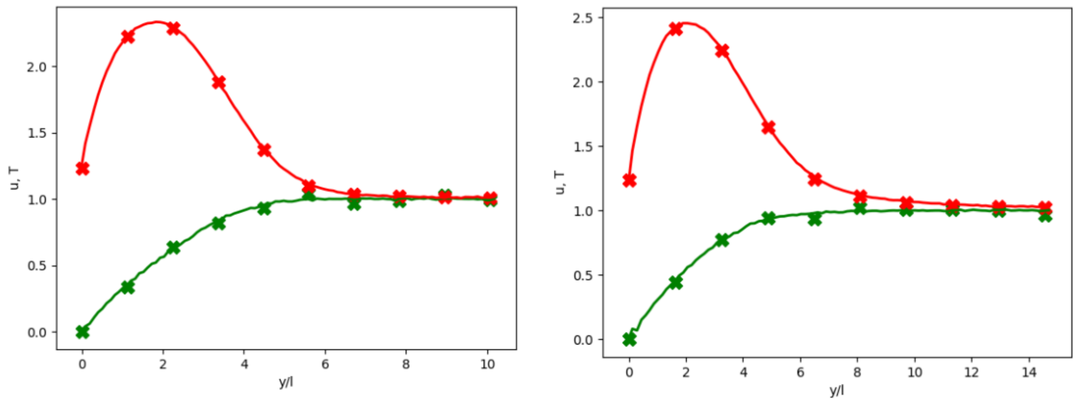


Second-mode instability : Mach 5 over 7° blunt cone

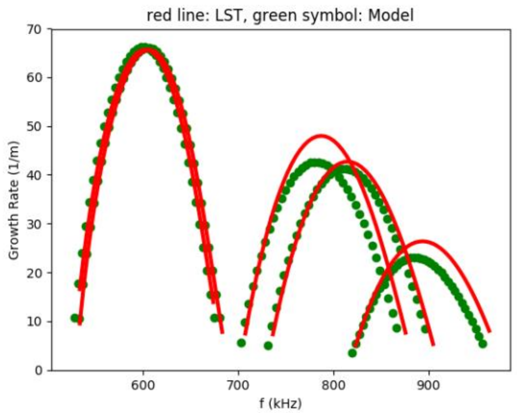


Second-mode instability : Mach 6 over 7° sharp cone

A Deep Learning Model for Intelligent Profile Interpolation

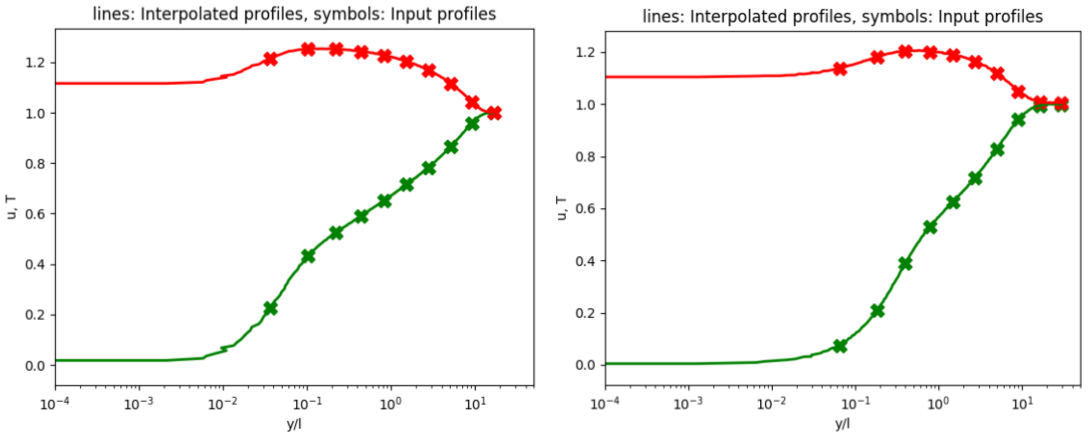


Mach 6
blunt cone

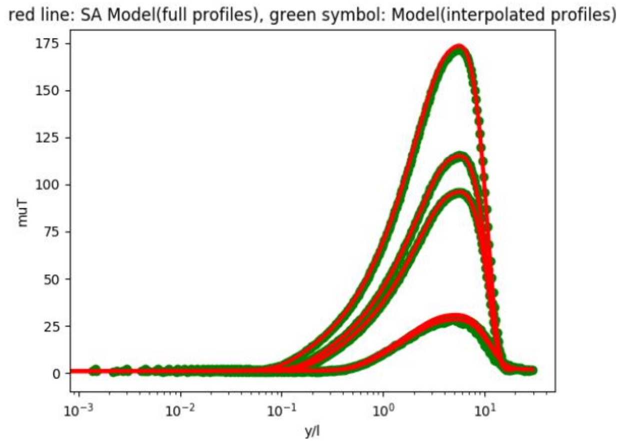


Symbols are input discrete profiles (11 points, subsampled and shifted)
Lines are interpolated 101-point profiles (red: temperature, green: velocity)

Symbols are predictions using interpolated profiles
Lines are LST or RANS solutions using full original profiles



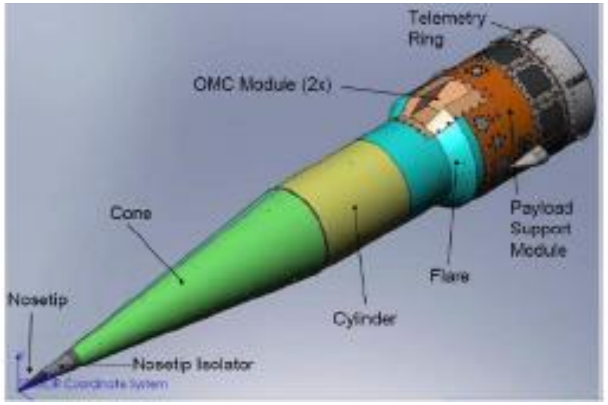
Mach 2 flat
plate



- Using a multiple fully-connected-layer model, trained by data from laminar or RANS Navier-Stokes solutions
- Input layer consists of a set of 11-point (ρ, u, v, w, T) profiles subsampled and shifted (to mimic under resolution) from a full profile and then interpolated to 101 or 201 points using the deep learning model
- Plug in interpolated profiles to the transition or turbulence model for predictions
- Good accuracy in predicted instability wave or turbulent eddy viscosity, given only 11 points in input profiles

Transition Prediction – A Case Study

- Nonablating, 7deg half-angle cone with r_n = 2.5 mm
- Cold wall, Mach number = 3.8 – 5.5., AoA = 0 deg
- Data analyzed by F. Li et al. 2005(J. Spacecraft & Rockets **52**, pp. 1283-1293)
  - N = 13.5 (second mode) correlated well with transition onset location



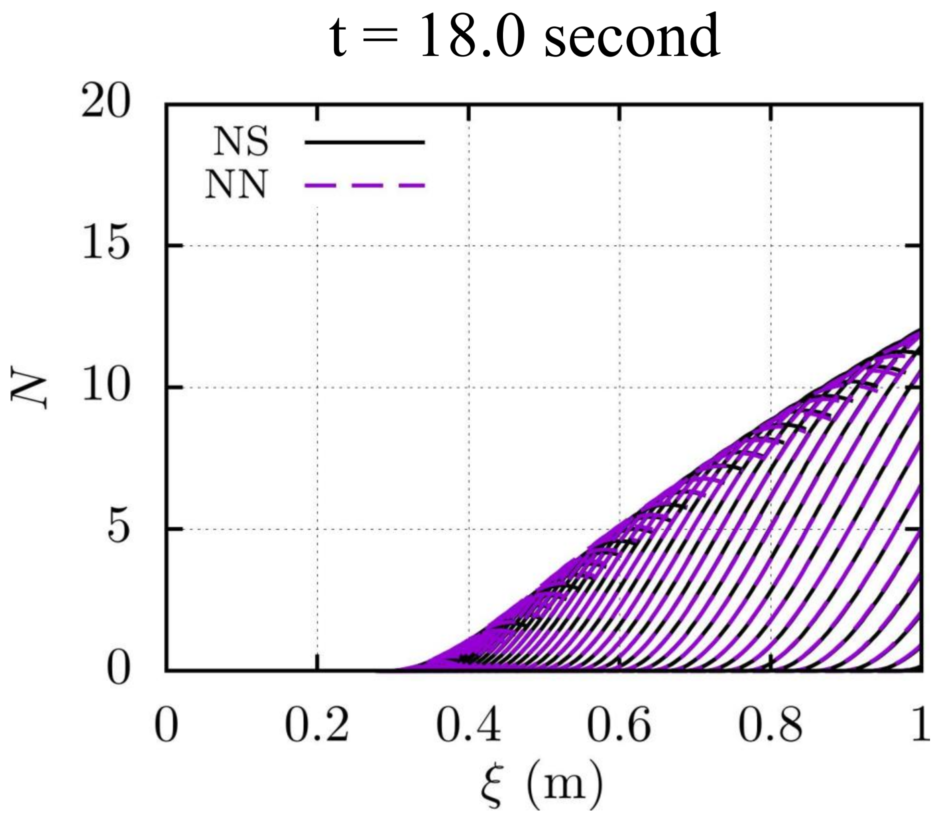
HIFiRE-1 Flight Transition Experiment

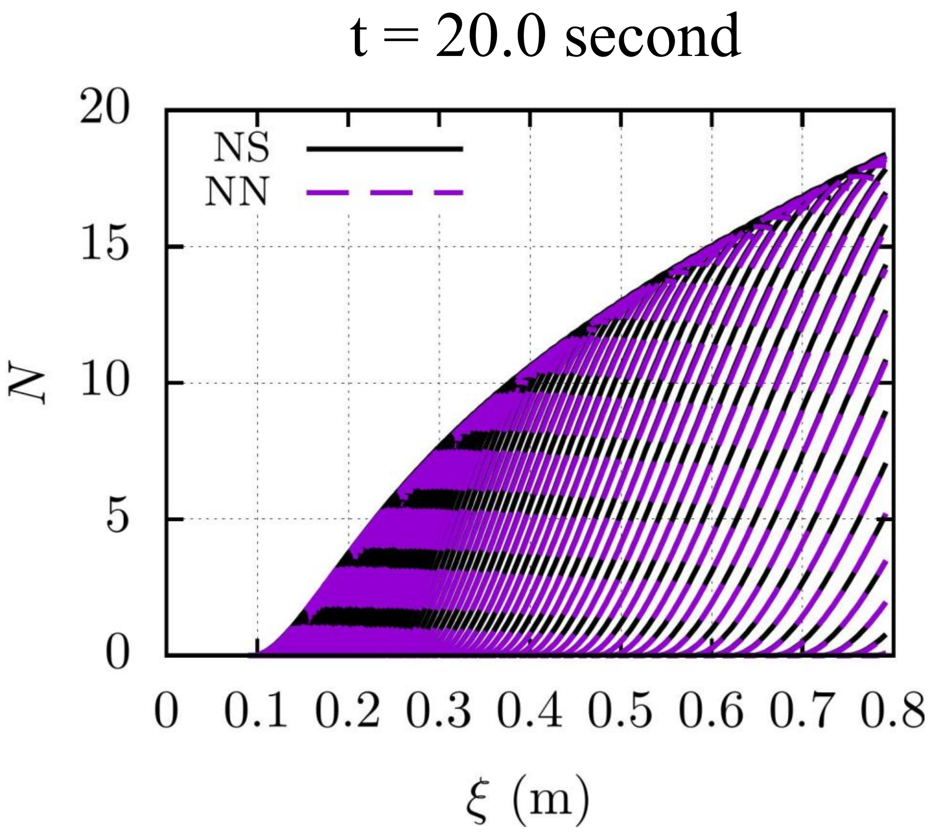
- **ML Objectives:**
  - Use NN to develop a transition model
  - Compare model predictions with direct LST computations

- **Training Data:**
  - LST results for 693 basic states (calculated using VULCAN-CFD solver)
    - 7° half-angle cone with $r_n = 2.5$ mm and $L_c = 1.1$ m at $AoA = 0^\circ$
    - Grid of 673×513 points with 2^{nd} FV and shock adaptation
    - Conditions: $M_\infty = 4.0 : 0.25 : 6.0$, $\bar{T}_\infty = 200$ K, $\bar{p}_\infty = 6000 : 2000 : 18000$
    - Isothermal wall: $\bar{T}_w = 300 : 50 : 800$ K
  - Together with boundary layer data points and disturbance parameters, total number of data entries are 4.41×10^6

Performance of NN Model

- Excellent agreement for N factor results computed using the Neural Network model with direct linear stability computations

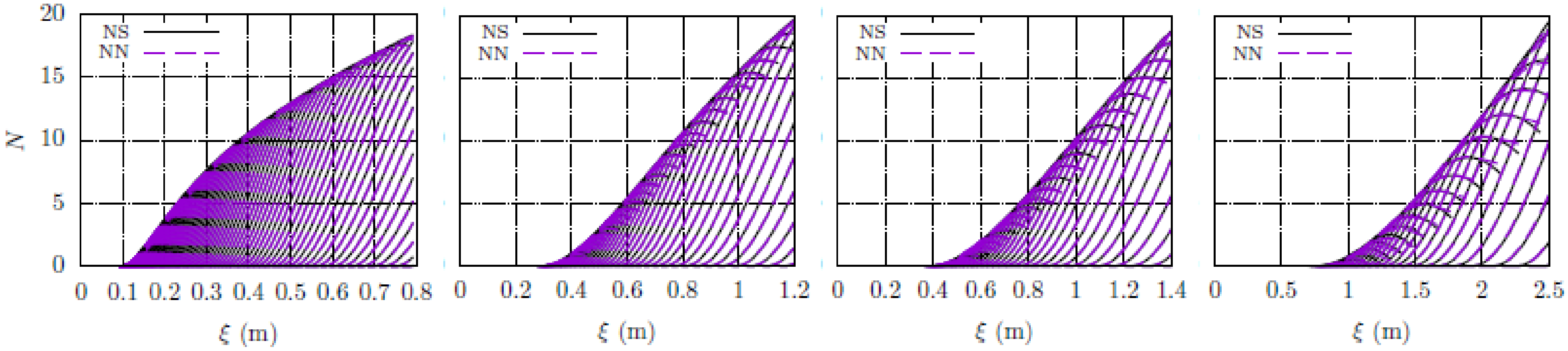




NN Model Prediction for Varying Nose Bluntness

- Applicability of NN evaluated for off-database radius values ($r_{n,0} = 2.5$ mm)
  - Cone at $AoA = 0°$ with $7°$ half-angle and conditions of HIFiRE-1 at $t = 20$ s

- $r_n = 0.5 r_{n,0}$
- $r_n = 1.5 r_{n,0}$
- $r_n = 2.0 r_{n,0}$
- $r_n = 4.0 r_{n,0}$



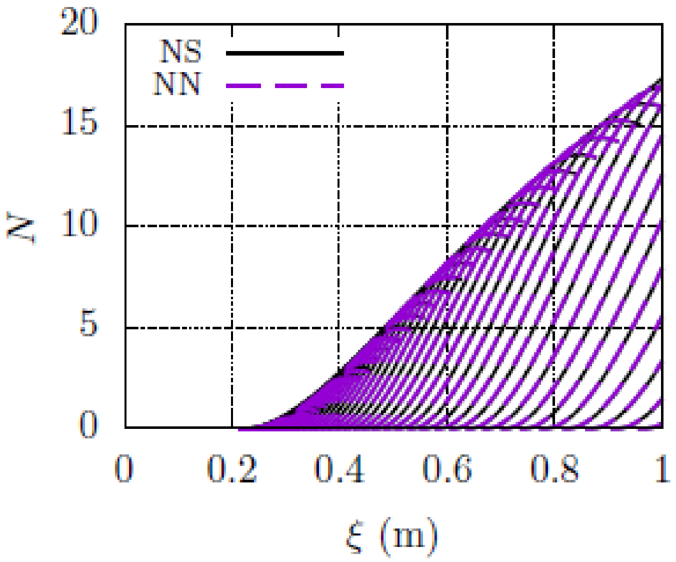
- Excellent agreement between stability calculations and NN predictions
- NN based on constant $r_{n,0}$ able to predict effect of bluntness

Nose bluntness is known to stabilize second mode disturbances, and the NN model predicts this trend well

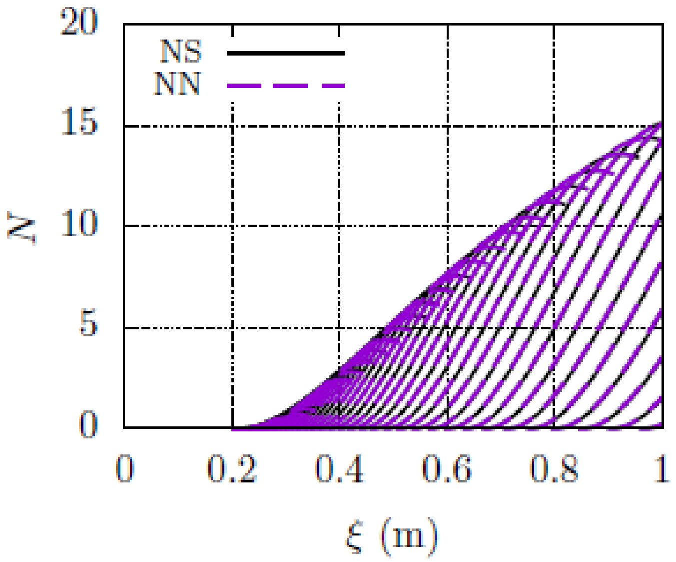
Applicability of NN Model for Varying Cone Angle

- Applicability of NN evaluated for off-database cone half-angle
  - Cone at $AoA = 0°$ with $5°$ half-angle and $r_n = 2.5$ mm

- Conditions: HIFiRE-1 at $t = 20.0$ s



- Conditions: HIFiRE-1 at $t = 21.5$ s



- Excellent agreement between stability calculations and NN predictions
- NN based on constant half-angle of $7°$ able to predict effect of half-angle

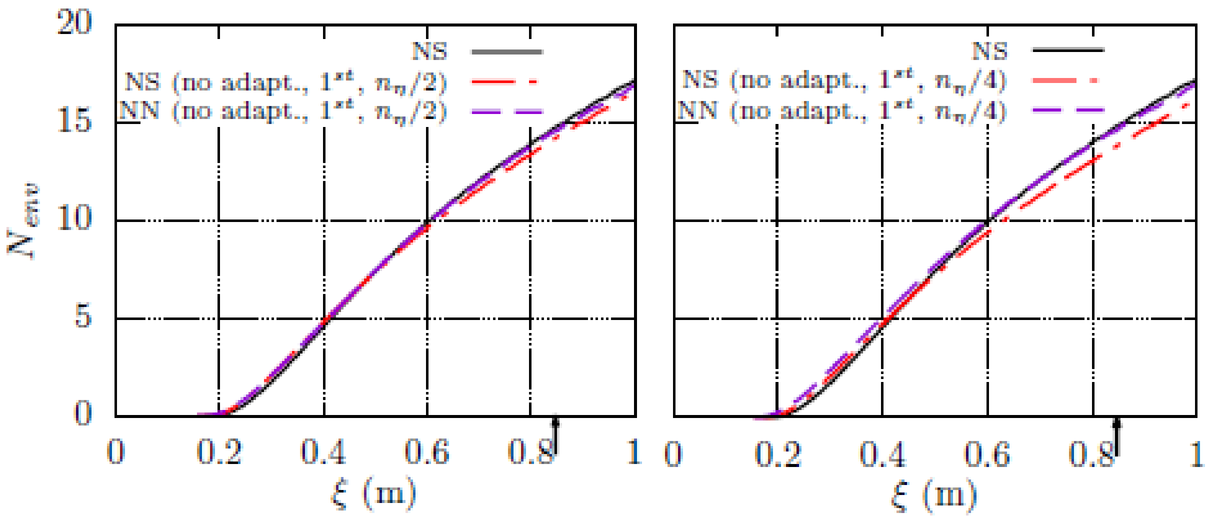
NN model predicts the effect of changing cone angle, although it lies outside the training data

NN Model Robustness

Underresolved basic state

- Robustness of NN evaluated for underresolved basic state
  - $AoA = 0°$, $7°$ half-angle, $r_n = 2.5$ mm, conditions of HIFiRE-1 at $t = 21.5$ s
  - "no adapt.": switched off shock adaptation
  - "1^{st}": reduced discretization order from second to first order
  - "$n_\eta/2$" & "$n_\eta/4$": reduced wall-normal points by half and by four

- no adapt., 1^{st}, $n_\eta/2$
- no adapt., 1^{st}, $n_\eta/4$



- Performance of NN model vs LST

Numerical Method	ϵ_{LST} (%)	ϵ_{NN} (%)
no adapt.	0.24	0.25
no adapt., 1^{st}	2.60	1.48
no adapt., 1^{st}, $n_\eta/2$	4.13	1.43
no adapt., 1^{st}, $n_\eta/4$	7.67	1.14

- LST error increases as the accuracy of the basic state solution decreases
- NN predictions remains below 2% for all the cases

Planned Turbulence Modeling/ML Workshop

- Turbulence Modeling: Roadblocks, and the Potential for Machine Learning
  - March 24-26, 2021
  - LMCO “Lighthouse”, Suffolk, VA
- Scope of the Workshop
  - Identification of critical issues for RANS models, and possible ways forward
  - A "Collaborative Testing Challenge" for data-driven RANS models
    - Apply your ML-assisted model to canonical NASA test cases
  - https://turbmodels.larc.nasa.gov/turb-prs2021.html
- Points of Contact
  - Chris Rumsey c.l.rumsey@nasa.gov
  - Gary Coleman g.n.coleman@nasa.gov

Summary

- Implementation of data-driven modeling/ML has been initiated in FUN3D
  - There are many questions
    - Choice of features, hidden layers, nodes, etc. influence the training?
    - How does one decide when a trained model is “good enough”?
    - Most importantly, the source of training data (experiments, high-fidelity simulations)?
    - How “universal” is a given trained model?
- There seems to be a clear path toward developing a robust NN model for transition because training data can be easily generated
  - A multi-layer deep learning model gives accurate predictions for all instability wave mechanisms
- Physics expert in the loop is essential for developing transition and turbulence models using machine learning