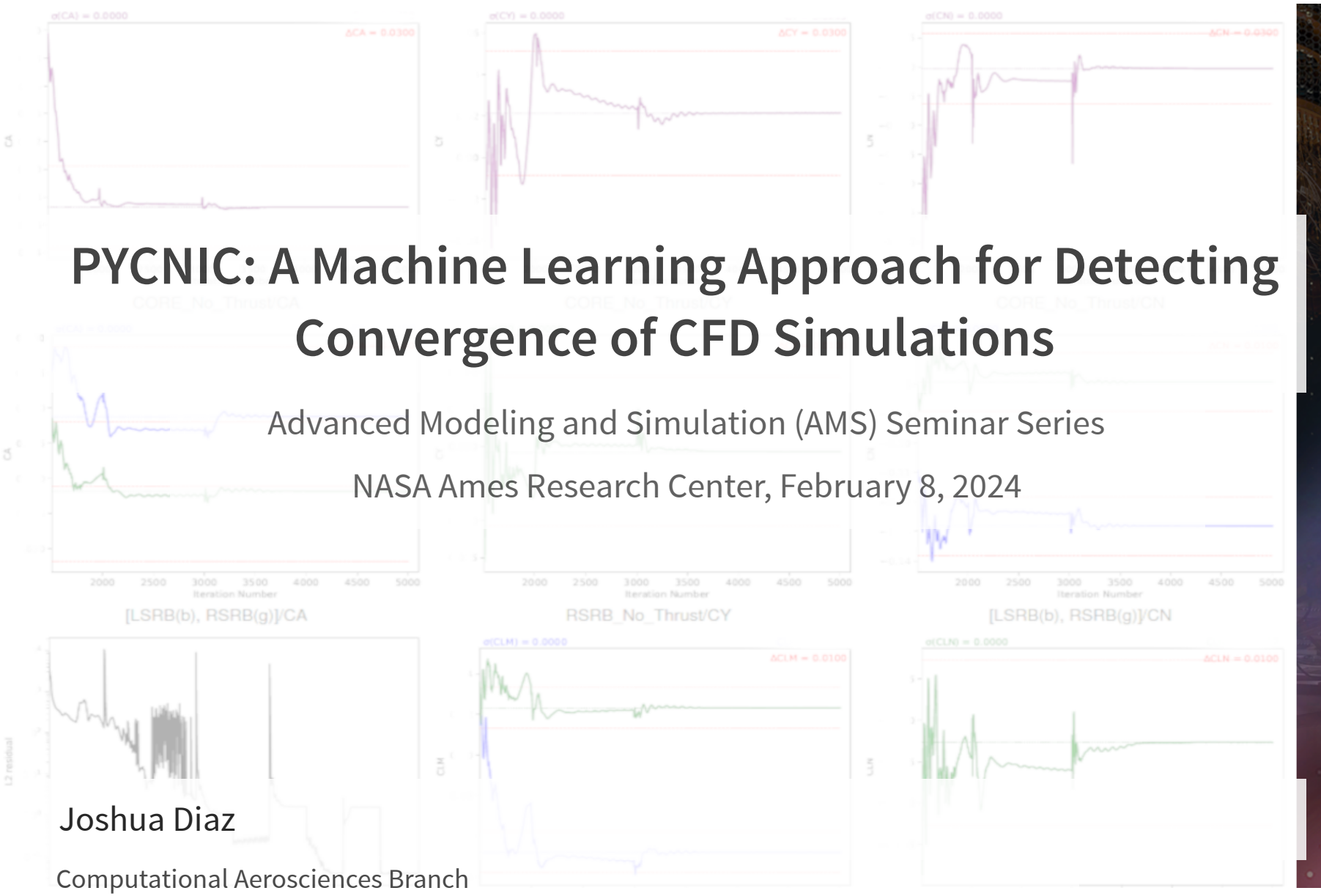
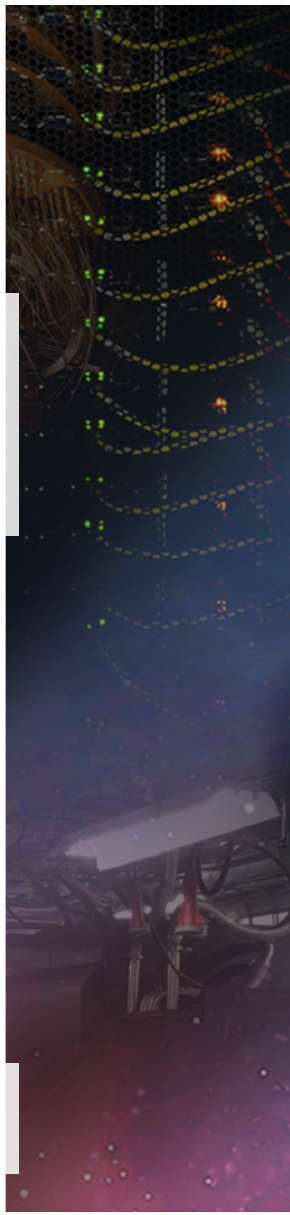






PYCNIC: A Machine Learning Approach for Detecting Convergence of CFD Simulations

Advanced Modeling and Simulation (AMS) Seminar Series

NASA Ames Research Center, February 8, 2024

Joshua Diaz

Computational Aerosciences Branch

NASA Ames Research Center, Moffett Field, CA

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Presentation Outline

1. Motivation
2. Introduction to PYCNIC
3. Look under the hood
4. Validation
5. Future work

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

What I do…

Ascent and **Booster Separation** modeling of NASA's Space Launch System (SLS) vehicle.

These are **LARGE** aerodynamic databases using: Cart3D, FUN3D, and OVERFLOW

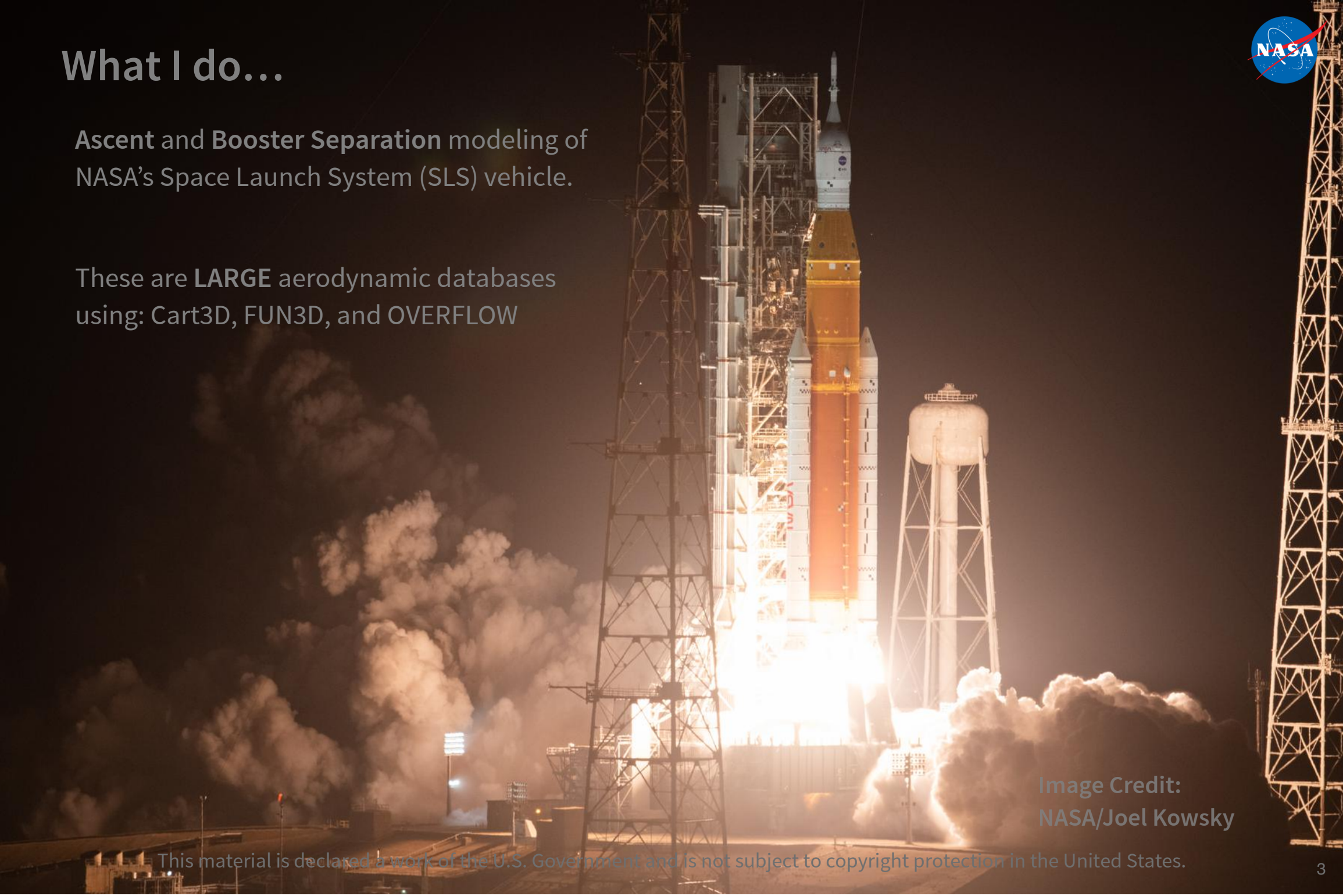


Image Credit:
NASA/Joel Kowsky

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c

Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
Case Config/Run Directory            Status  Iterations  Que CPU Time
---- ------------------------------ ------- ----------- --- --------
0    poweron02-ML/m0.50a0.0r000.0_a DONE    12500/12500 .   145224.7
1    poweron02-ML/m0.50a1.0r000.0   RUN     4233/4500   .    29998.9
2    poweron02-ML/m0.50a2.0r000.0   DONE    7000/7000   .    33468.2
3    poweron02-ML/m0.50a6.0r000.0   DONE    5500/5500   .    37828.9
4    poweron02-ML/m0.70a0.0r000.0   DONE    10500/10500 .    75882.0
5    poweron02-ML/m0.70a1.0r000.0   DONE    6500/6500   .    46045.2
6    poweron02-ML/m0.70a2.0r000.0   DONE    4750/4750   .    31409.3
7    poweron02-ML/m0.70a6.0r000.0   ERROR   3500/4500   .    19231.8
8    poweron02-ML/m0.80a0.0r000.0   DONE    8500/8500   .    60833.1
9    poweron02-ML/m0.80a1.0r000.0   DONE    5500/5500   .    38438.4
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10

Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
poweron02-ML/m0.50a0.0r000.0_a
  SRB_AftBSMPod_CA: New subfig at iteration 12500
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-jfdiaz3'
  SRB_AftBSMPod_CN: New subfig at iteration 12500
  SRB_MainTun_CA: New subfig at iteration 12500
  STACK-phi090-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')
  STACK-phi000-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')
  STACK_Aft-phi090-mach-large: New subfig at iteration 12500
> tec360 -b -p export-lay.mcr -mesa
    (PWD = 'poweron02-ML/m0.50a0.0r000.0_a/')
    (STDOUT = 'tecmcr.out')

    ...
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10
```

3) Transfer report to local machine

```
1  pfe27:f3_vac1$ putsup report/report-f3d-poweron02-ML.pdf
2  nasmac3329:28008$ getsup
```

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

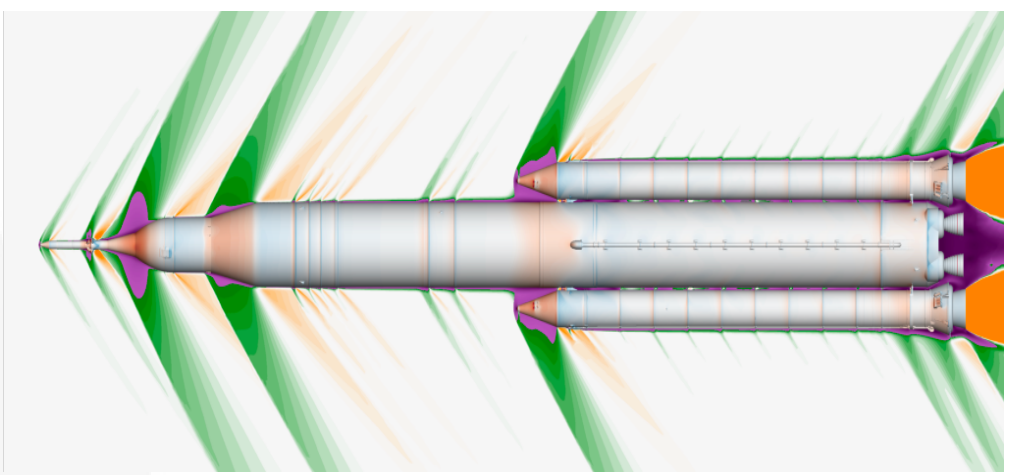
```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10
```

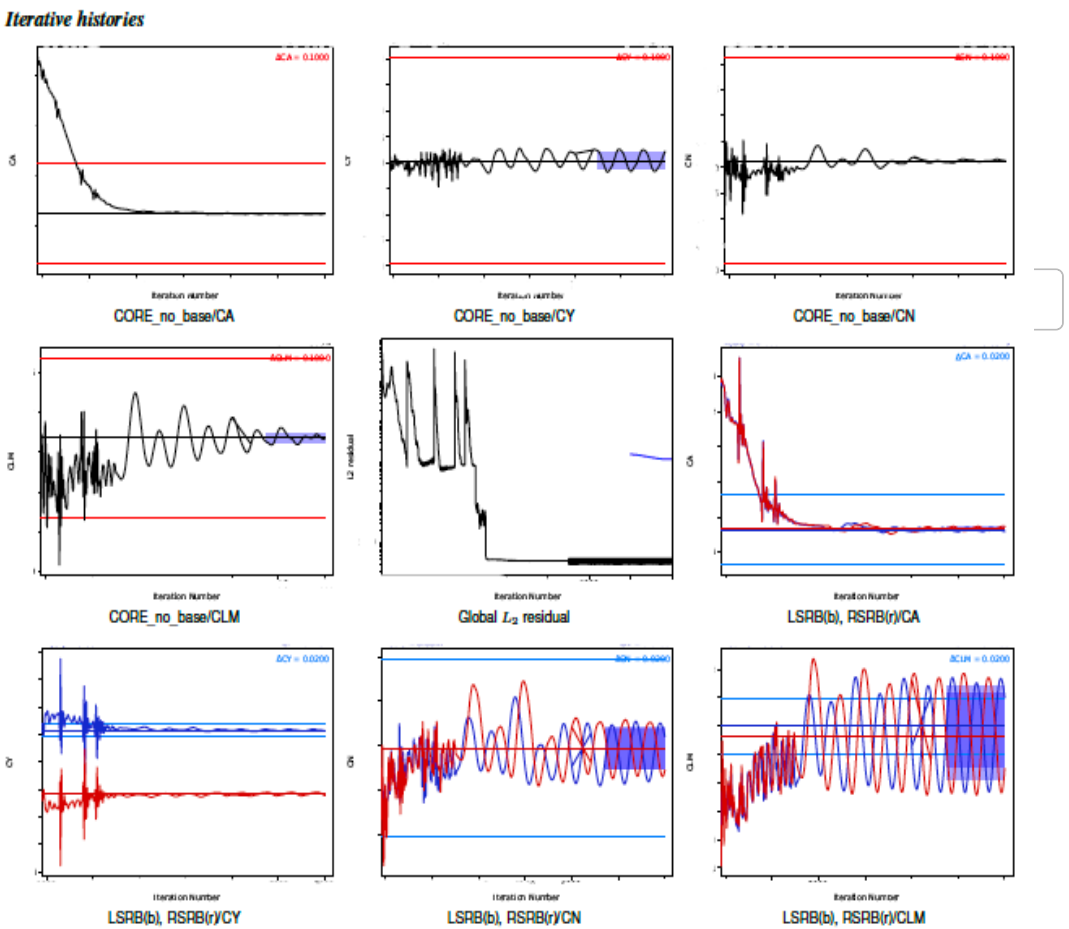
3) Transfer report to local machine

```
pfe27:f3_vac1$ putsup report/report-f3d-poweron02-ML.pdf
nasmac3329:28008$ getsup
```

4) Assess each case for convergence

```
1  nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```





This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current Workflow

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Generate case report

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10
```

3) Transfer report to local machine

```
pfe27:f3_vac1$ putsup report/report-f3d-poweron02-ML.pdf
nasmac3329:28008$ getsup
```

4) Assess each case for convergence

```
nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```

5) PASS or EXTEND cases on front end

```
1  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0,2:7 --PASS
2  pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 8:11 --extend
```

SLS Block 1B Ascent aerodynamic database is **1242** CFD cases.

There were an average of **4.5 manual checks** for convergence per case.

For 100 CFD cases: **HOURS**

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Workflow with PYCNIC

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

~~2) Generate case report~~

```
~~pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  --report -I 0,2:7,8:10~~
```

~~3) Transfer report to local machine~~

```
~~pfe27:f3_vac1$ putsup report/report-f3d-poweron02-ML.pdf~~
~~nasmac3329:28008$ getsup~~
```

4) Assess each case for convergence

```
nasmac3329:28008$ open report/report-f3d-poweron02-ML.pdf
```

5) PASS or EXTEND cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0,2:7 --PASS
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 8:11 --extend
```

SLS Block 1B Ascent aerodynamic database is **1242** CFD cases.

There were an average of **4.5 manual checks** for convergence per case.

For 100 CFD cases: ~~**HOURS**~~ **MINUTES**

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Workflow with PYCNIC

1) Check cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0:10 -c
```

2) Assess each case for convergence on front end

```
pfe27:f3_vac1$ pycnic -f run/poweron02-ML.json -I 0,2 --comp CORE_no_base
Importing module 'f3drunasc'
Importing module 'f3dnas'
Importing module 'tnas45pal'
  InitFunction: f3dnas.init_config()
  InitFunction: tnas45pal.add_triqfm()
Component - CORE_no_base
Prepping CA Iterative History 1/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  1.64it/s]
Prepping CY Iterative History 2/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.56it/s]
Prepping CN Iterative History 3/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.76it/s]
Prepping CLM Iterative History 4/4
Writing cases...: 100%|████████████████████| 2/2 [00:00<00:00,  2.79it/s]

 Loading Machine Learning Models...

Case 0: Sampling Iterations 10500-12500
CORE_no_base - Time Accurate Binary ML: EXTEND

Case 2: Sampling Iterations 5000-7000
CORE_no_base - Time Accurate Binary ML: PASS
```

3) PASS or EXTEND cases on front end

```
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 0 --extend
pfe27:f3_vac1$ pyfun -f run/poweron02-ML.json  -I 2 --PASS
```

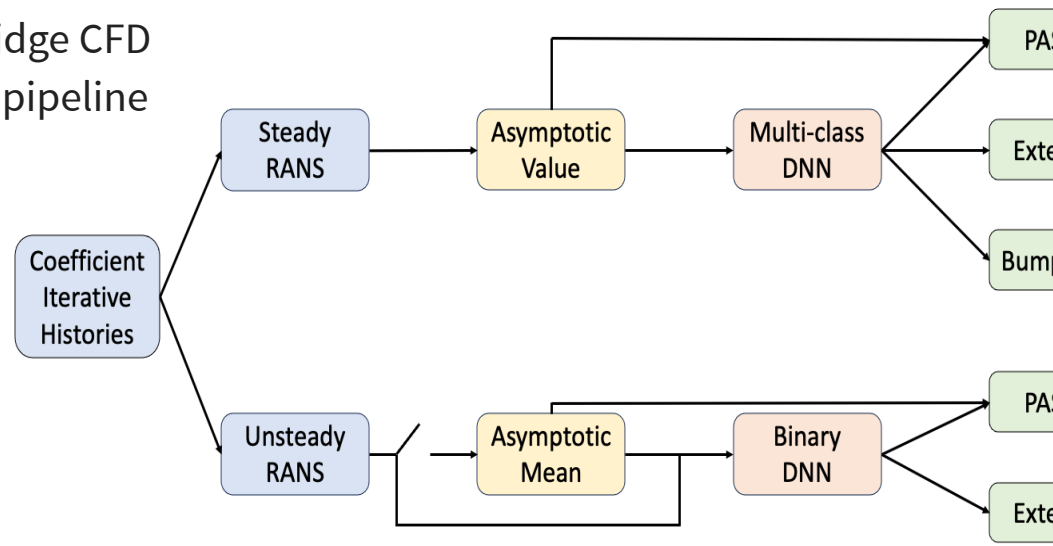
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

What is PYCNIC

PYCNIC stands for Python Classifier for Numerical Iterative Convergence

and acts as an automated convergence criteria.

- Built atop CAPE, an open-source end-to-end framework for running CFD simulations
- Implements both **heuristic convergence criteria** as well as **supervised deep learning**
- Goal: To not only offer a convergence detection tool, but bridge CFD and machine learning libraries through a machine learning pipeline for developing unique models

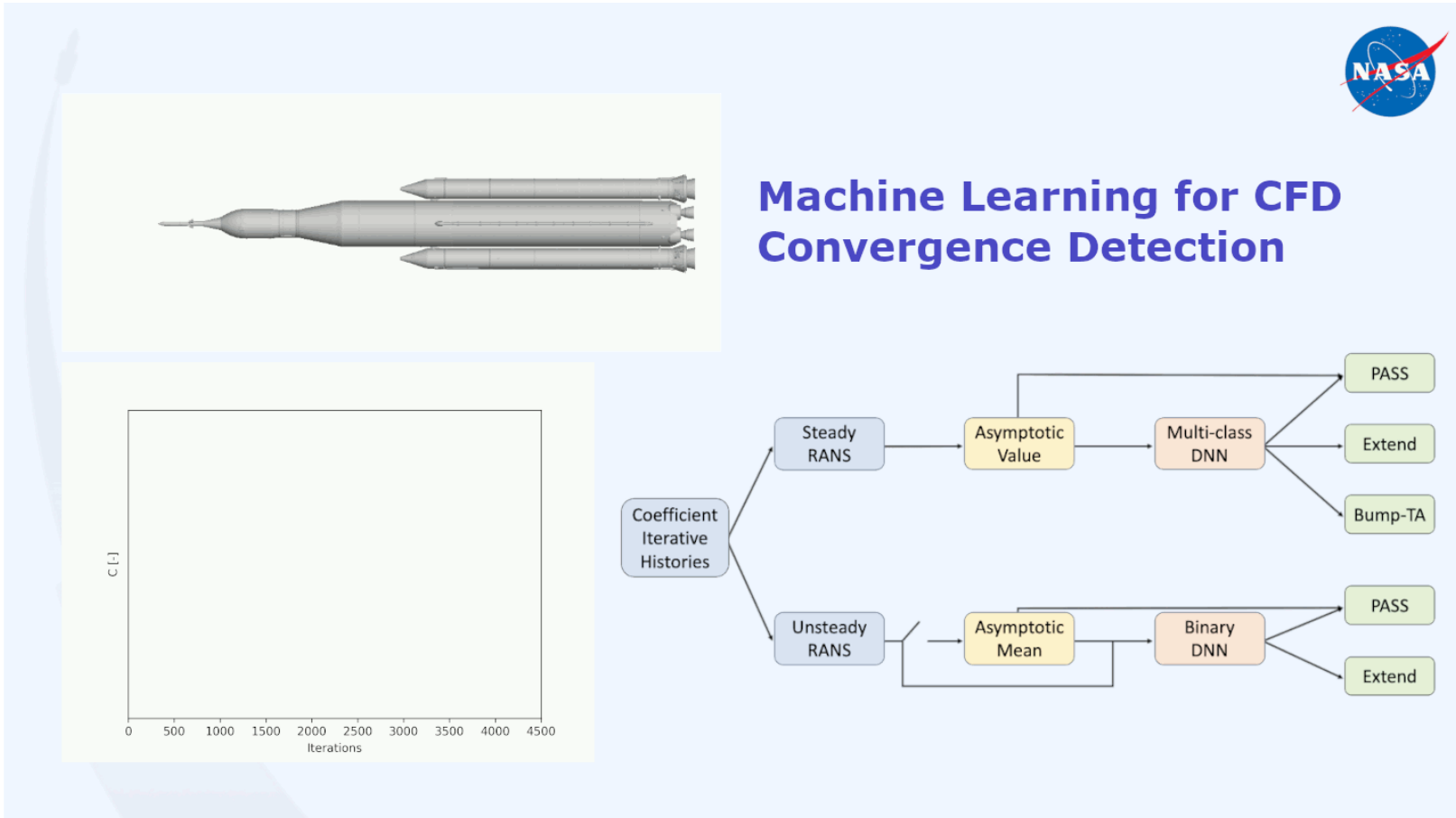


Why do we need PYCNIC

1. Expedite
    - Reduces the user's workload
    - Quickens the completion of high-volume run matrices
2. Standardize
    - Humans can have poor repeatablility when classifying the same data
    - Multiple practioners contribute with varying experience and opinions

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

What is PYCNIC

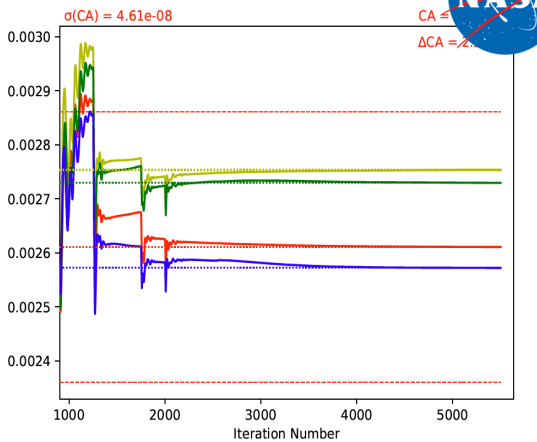


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Asymptotic Filters

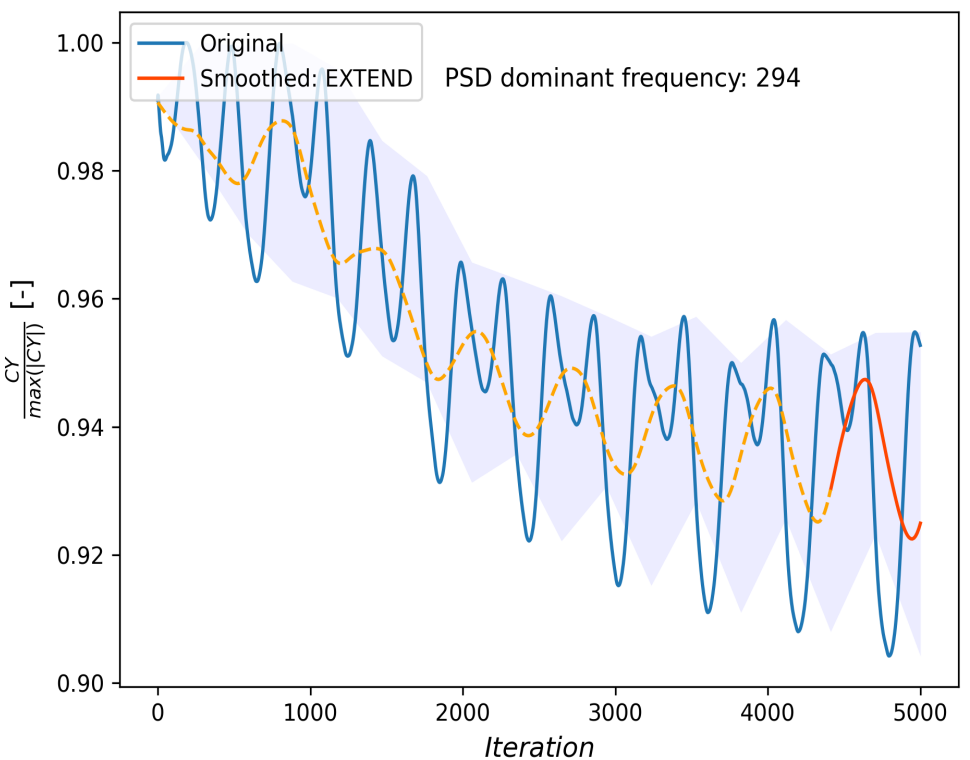
Steady RANS

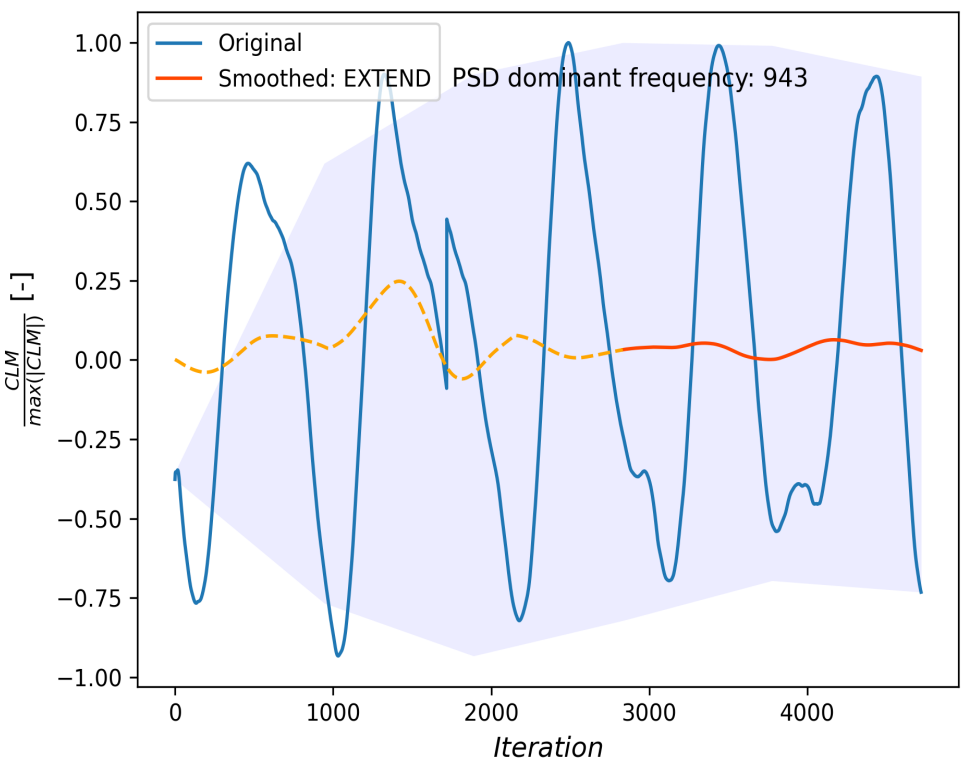
$$\begin{cases} \text{PASS}, & \text{if } |\max(C) - \min(C)| \leq X \\ \rightarrow \text{DNN}, & \text{if } |\max(C) - \min(C)| > X \end{cases}$$



Unsteady RANS

Smoothing function walks through an averaging window based on the dominant frequency determined by **Power Spectral Density** (scipy.signal.periodogram).

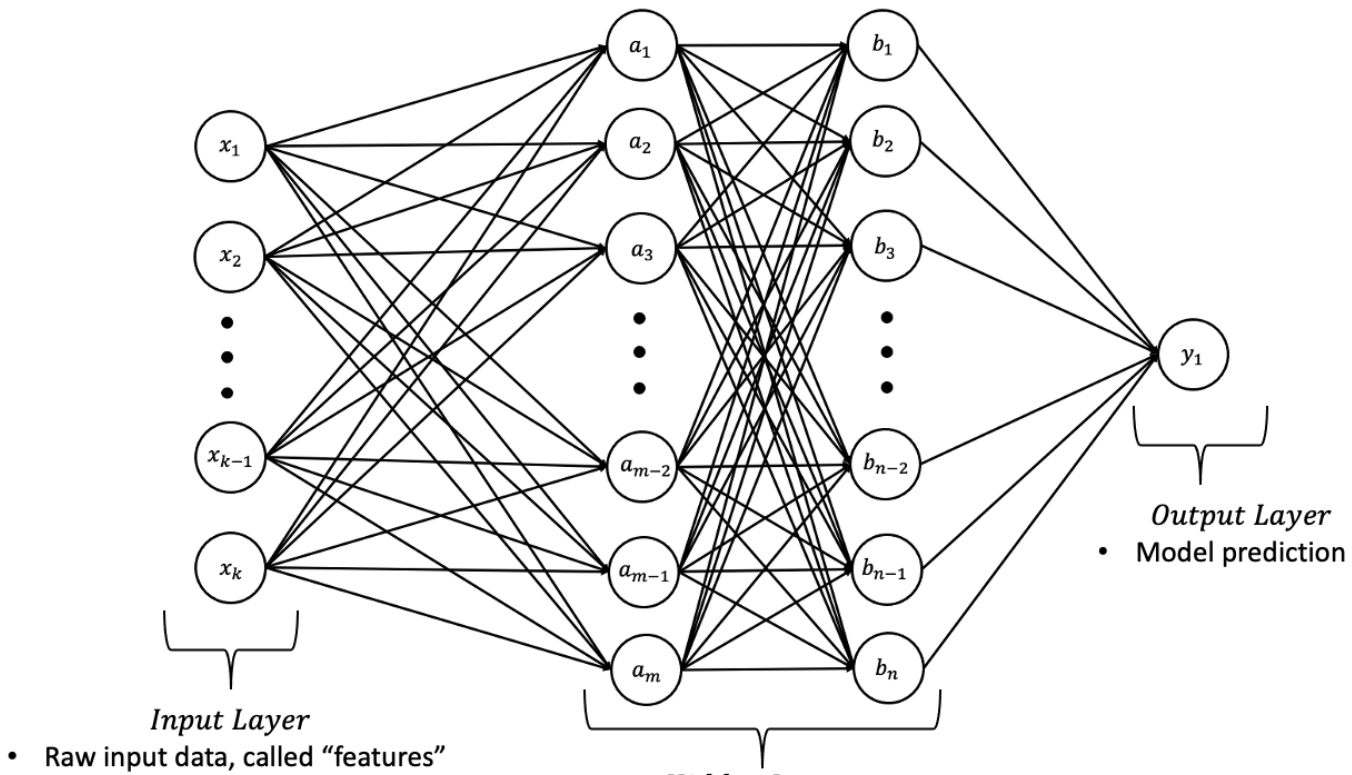




This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Deep Neural Network

- A Deep Neural Network (DNN) is a mathematical model inspired by the human brain which uses multiple layers to obtain a high-dimensional representation of raw input data to make a prediction.

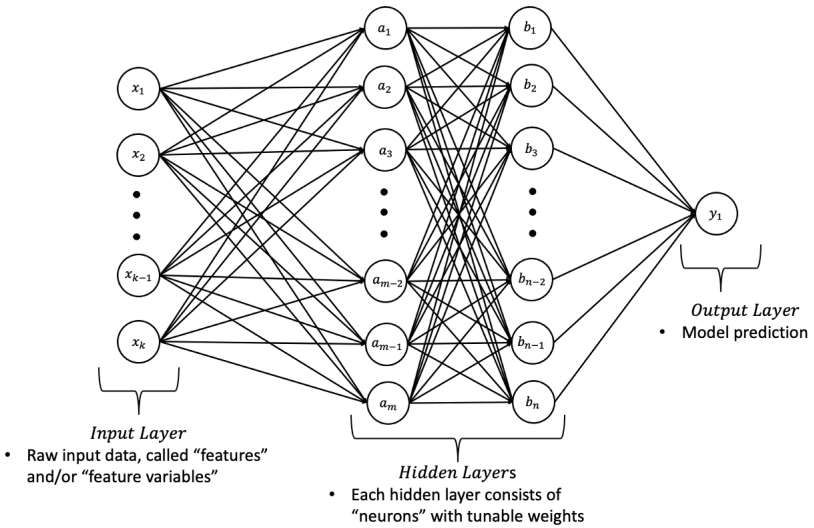


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Deep Neural Network

1. Define the input vector $\mathbf{x}$ with k features:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$



2. Initialize weights in each hidden layer by drawing from a uniform distribution between $[-1, 1]$:

$$\mathbf{W^{(a)}} = \begin{bmatrix} w_{11}^{(a)} & w_{12}^{(a)} & \cdots & w_{1k}^{(a)} \\ w_{21}^{(a)} & w_{22}^{(a)} & \cdots & w_{2k}^{(a)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^{(a)} & w_{m2}^{(a)} & \cdots & w_{mk}^{(a)} \end{bmatrix},$$

3. The output of the hidden layer is defined by first a linear transformation and then a nonlinear transformation.

$$\mathbf{z}^{(a)} = \mathbf{W}^{(a)}\mathbf{x}$$
$$\mathbf{a} = \varphi\left(\mathbf{z}^{(a)}\right)$$

4. Repeat for next hidden layer.

$$\mathbf{z}^{(b)} = \mathbf{W}^{(b)}\mathbf{a}$$
$$\mathbf{b} = \varphi\left(\mathbf{z}^{(b)}\right)$$

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

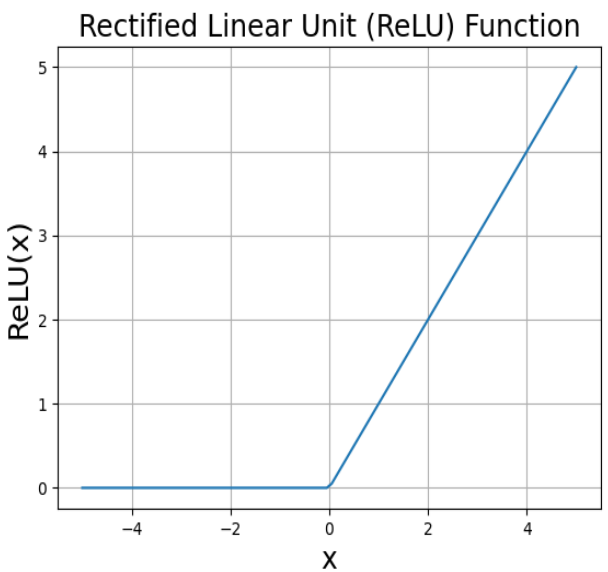
Deep Neural Network

Activation Functions

- **Rectified Linear Unit (ReLu)**

$$\varphi(\mathbf{z}) = \begin{cases} 0, & \text{if } z_i < 0 \\ z_i, & \text{if } z_i \geq 0 \end{cases}$$

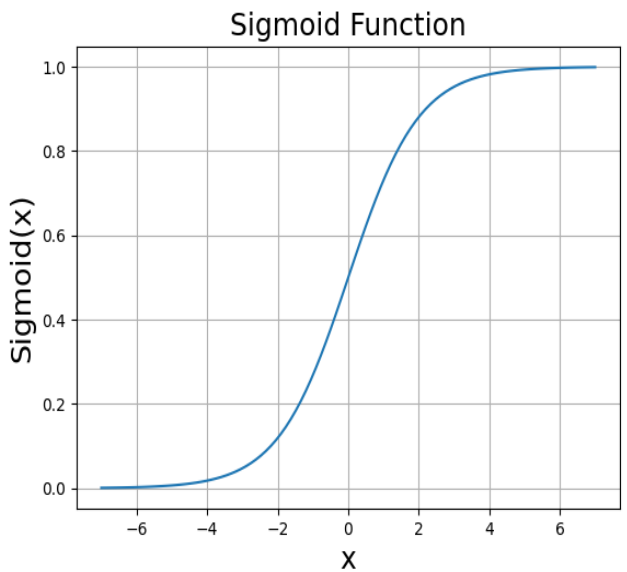
Used for all **hidden layers**.



- **Sigmoid**

$$\sigma(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}}$$

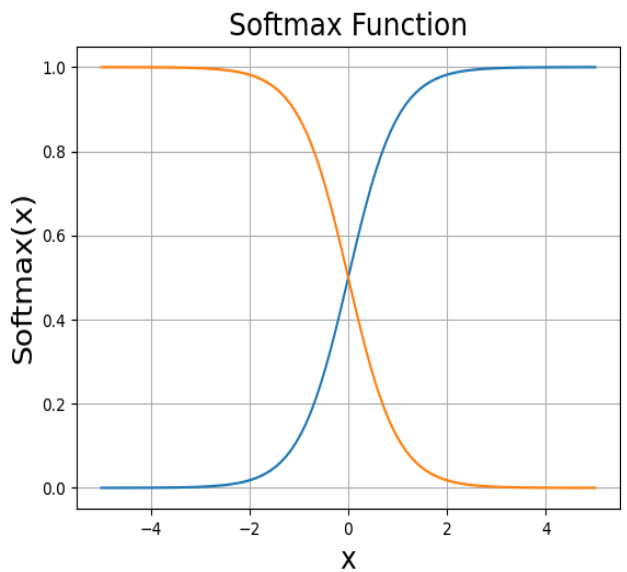
Used for **binary** model output layer. Returns a value between 0 and 1.



- **Softmax**

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}},$$

Used for **multi-class** model output layer. Returns a vector of values whose sum is equal to 1.



This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Deep Neural Network

Weights are optimized through backpropagation and a stochastic gradient descent method, minimizing a loss function over multiple training epochs.

Binary Cross-entropy Loss Function

$$L_{Binary} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right)$$

where, $p_i = \frac{1}{1+e^{-z}}$

Sparse Categorical Cross-entropy Loss Function

$$L_{Multi-class} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i)$$

where, $p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$

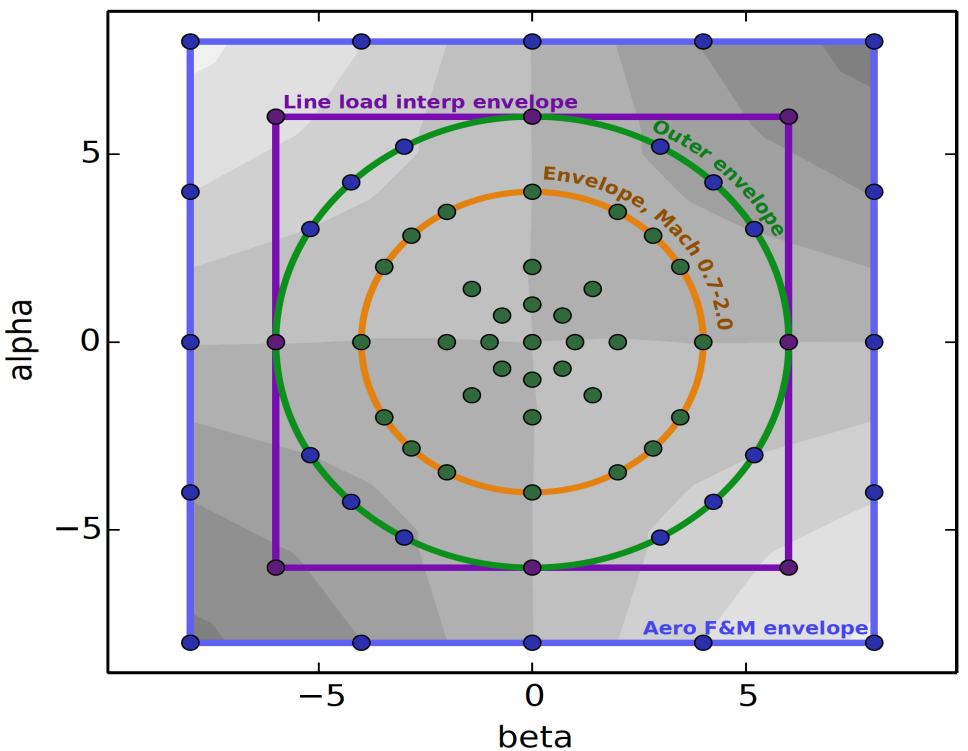
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Deep Neural Network

Supervised learning approach where the training data consists of features and labels.

Ascent databases perfectly lend themselves to supervised learning because they are:

1. high volume
2. cover multiple flight regimes
3. labeled



This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

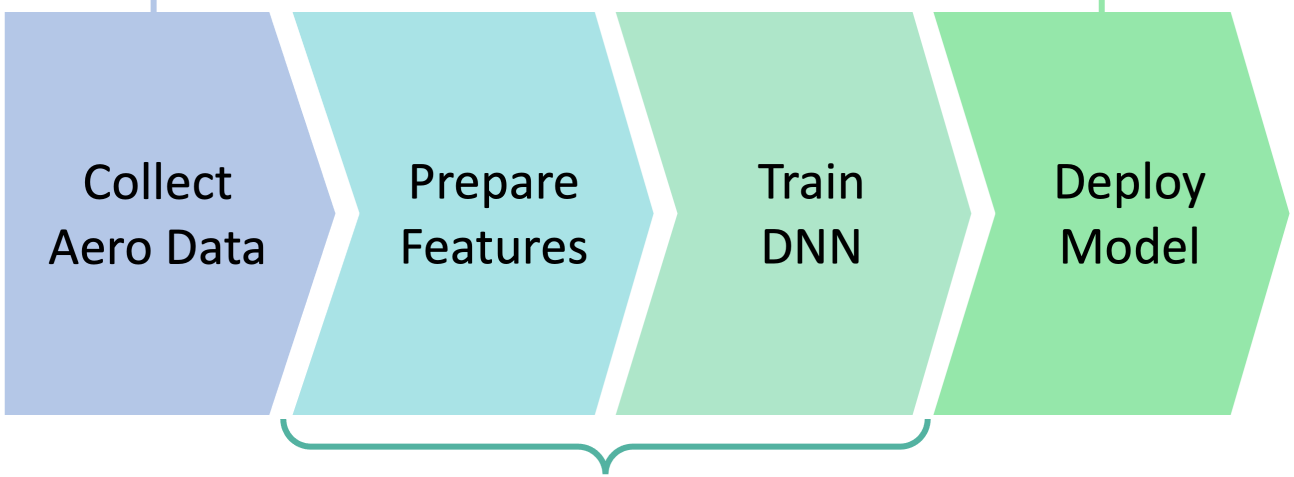
Deep Neural Network

PYCNIC: ML Pipeline

```
$ pycnic –f run/matrix.json --collect
$ ls data/
CA-comp.mat   CY-comp.mat
CN-comp.mat   CLM-comp.mat
```

Local Deployment

```
$ ls /home4/jfdiaz3/share/models/
TF-MC-4FV-300iters-2000epochs-100comb-
28008f3vac1-1700895508
```




```
from pycnic import read, prep, buildmodel
.
.
.
```

31 lines of code to go from `.mat` files to trained DNN

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Deep Neural Network

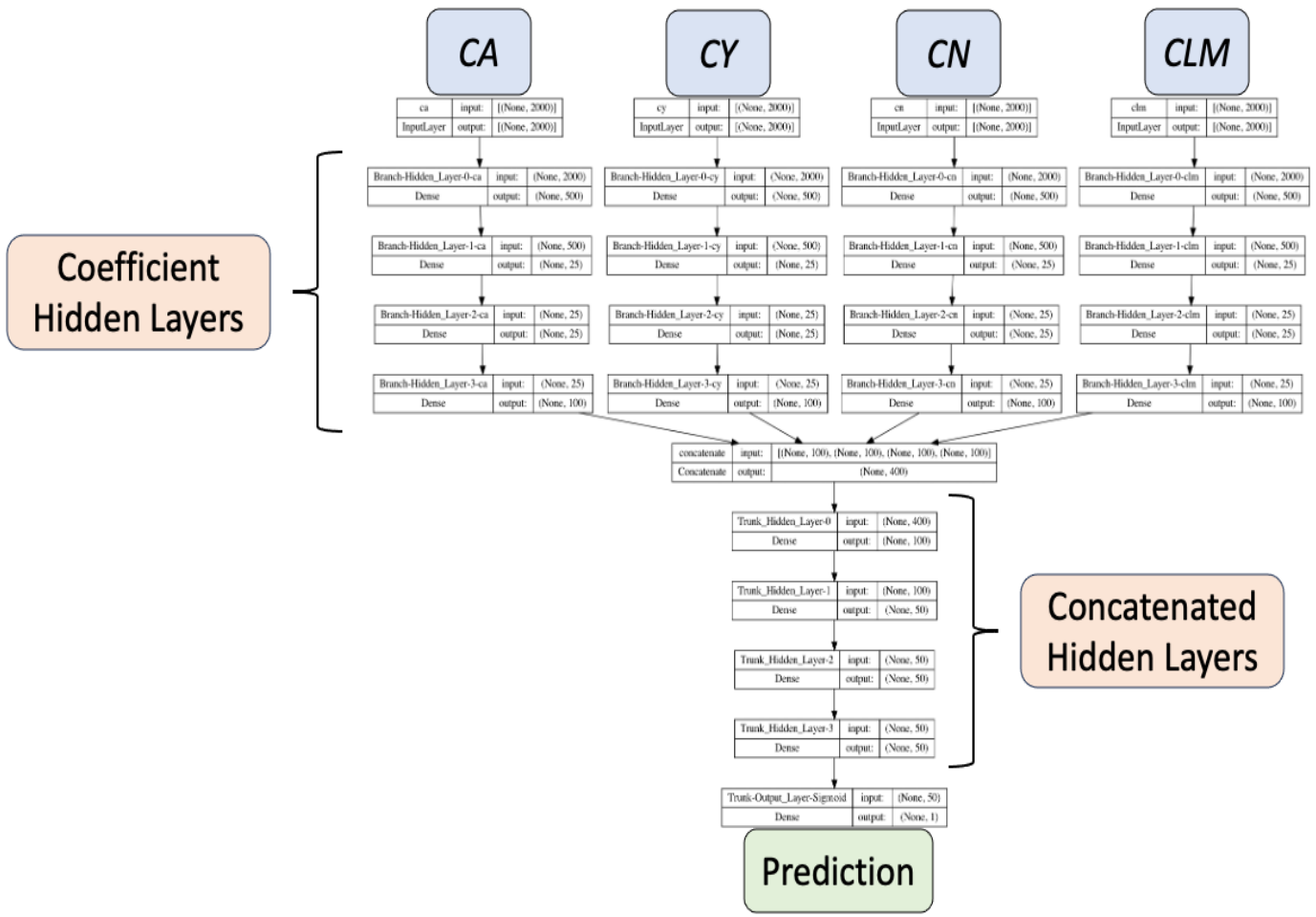
DNN Models were coded in Python using Google's machine learning framework **TensorFlow**, and optimized using the open-source library **Keras**

Input data

- 2000 features per feature variable
- Feature Variables: CA, CY, CN, CLM

Training data

- ~300 cases per label
- 3 classes: PASS, EXTEND, BUMPTA
- optimized hyperparameters
- ~1e6 parameters
- ~36 gpu hours to train

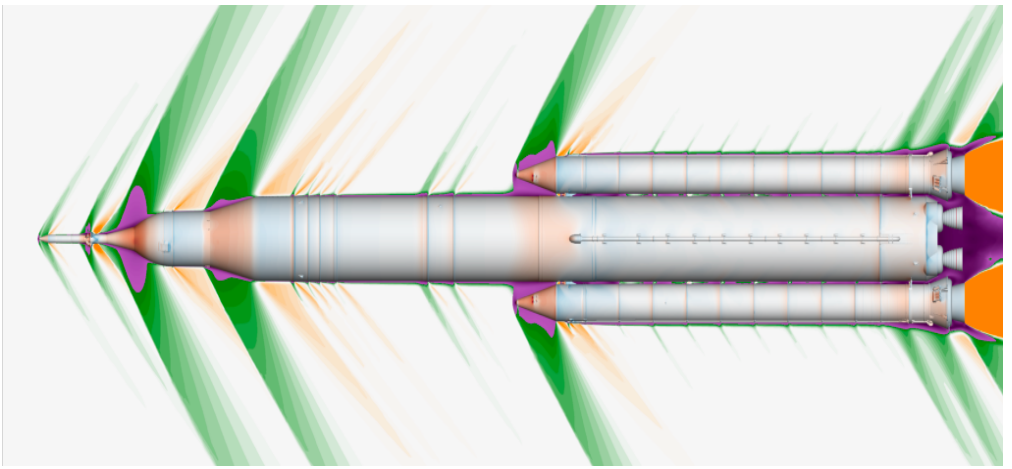


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

Model independently* re-ran 72 ascent cases for SLS Block 1B

- Mach = [0.5 : 4.50]
- alpha = [0.0, 1.0, 2.0, 6.0]
- beta = 0.0
- sASYM = 0.001, uASYM = 0.01
- Default: 2500 Steady RANS iterations
- EXTEND = 500 additional iterations
- **PASS** based on:
    1. CORE
    2. LSRB
    3. RSRB



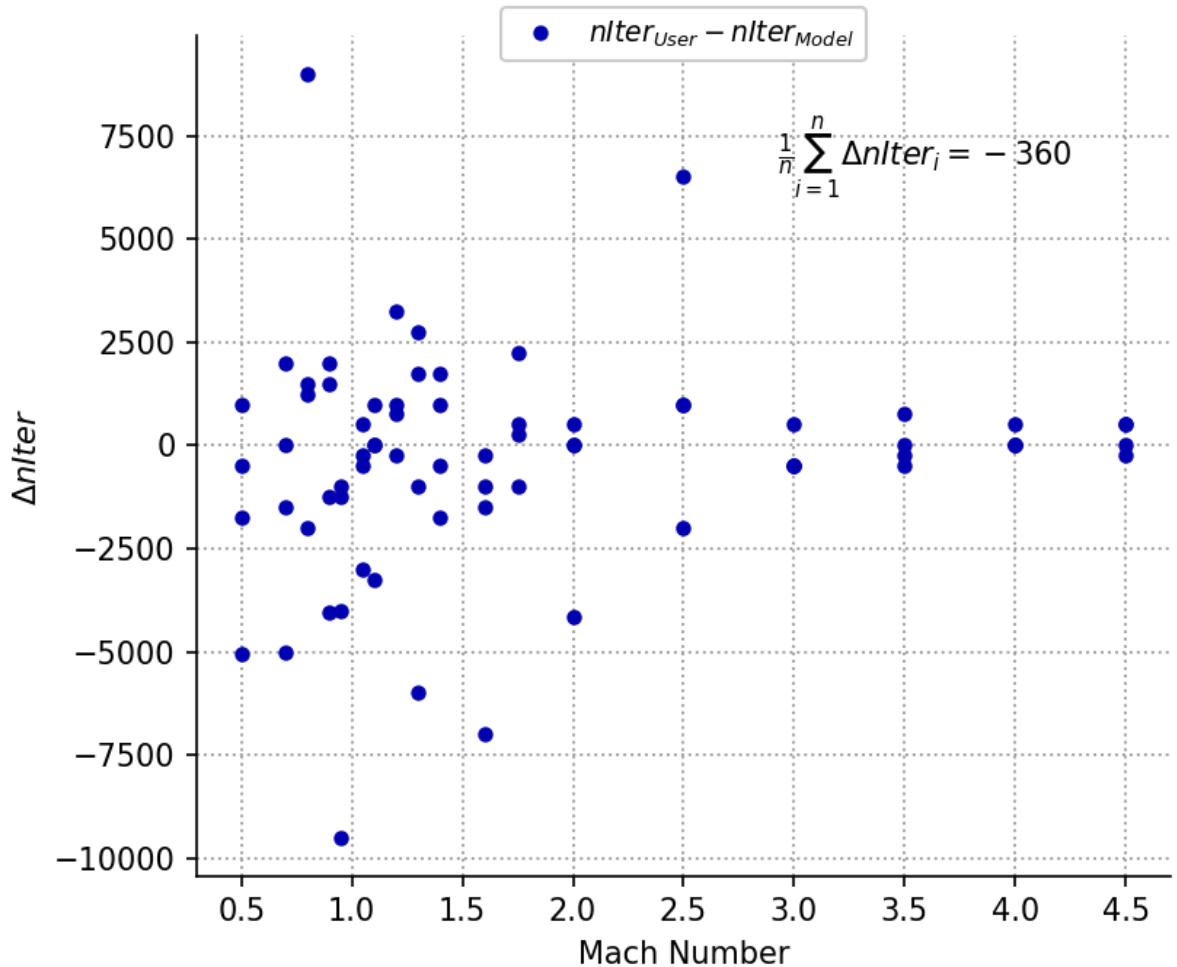
PASS Method	CORE	RSRB	LSRB	[%]
Unsteady Asym	42	42	45	60%
Unsteady DNN	7	6	4	8%
Steady Asym	20	22	21	29%
Steady DNN	3	2	2	3%

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

Model independently* re-ran 72 ascent cases for SLS Block 1B

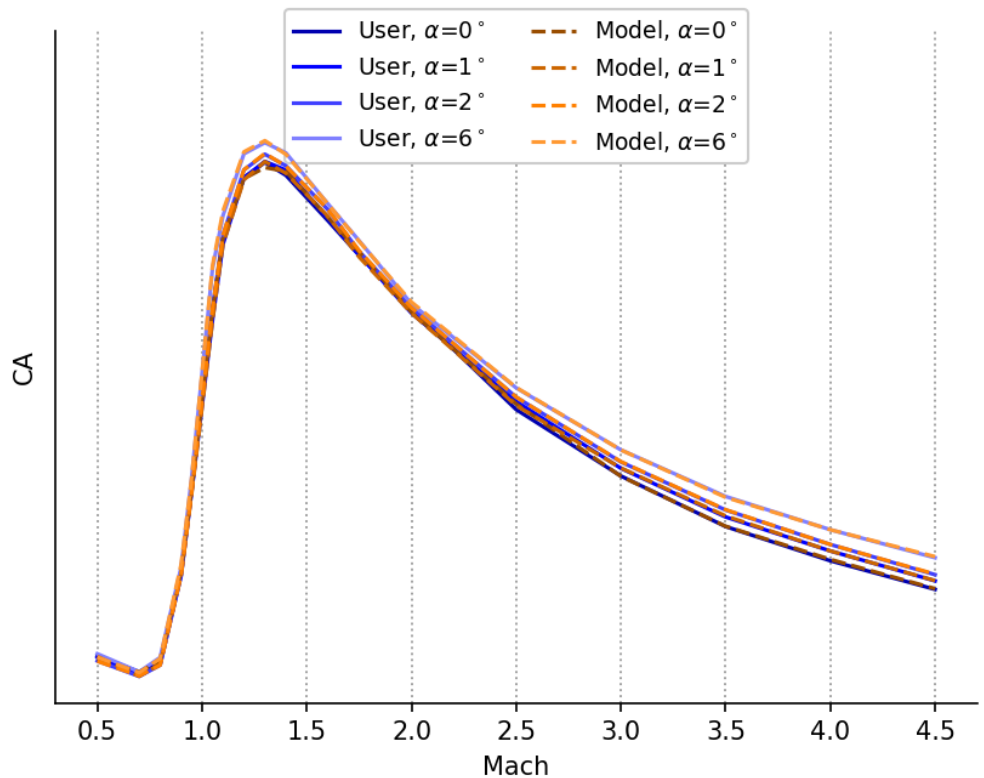
- Mach = [0.5 : 4.50]
- alpha = [0.0, 1.0, 2.0, 6.0]
- beta = 0.0
- sASYM = 0.001, uASYM = 0.01
- Default: 2500 Steady RANS iterations
- EXTEND = 500 additional iterations
- **PASS** based on:
    1. CORE
    2. LSRB
    3. RSRB

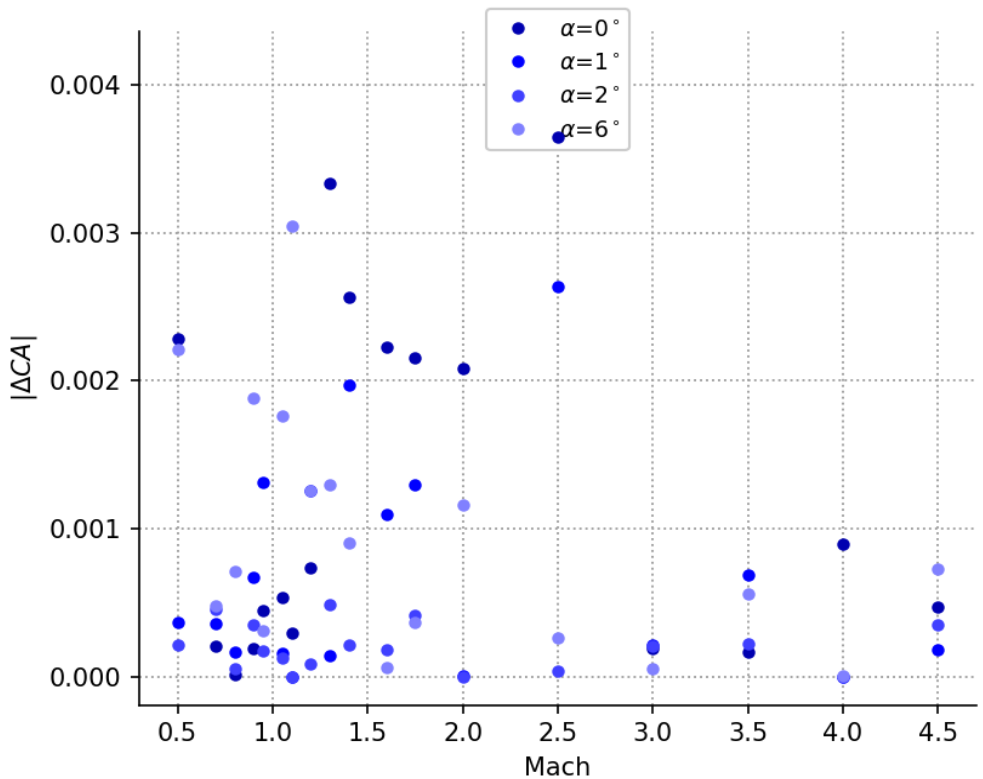


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

CORE - Axial Force Coefficient

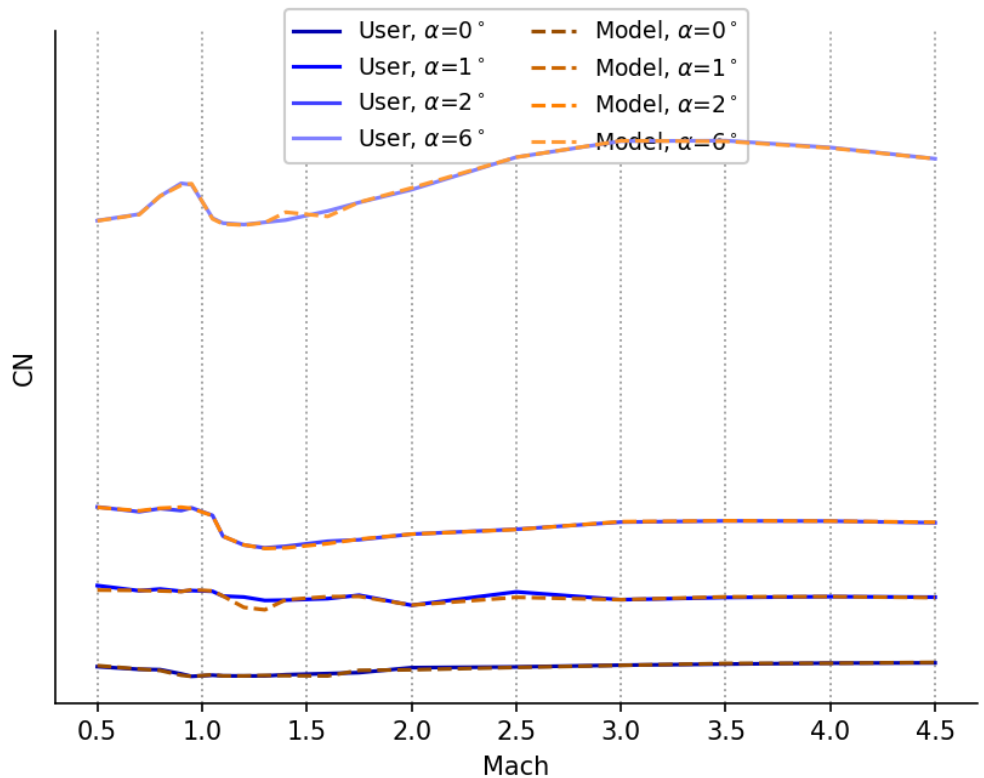


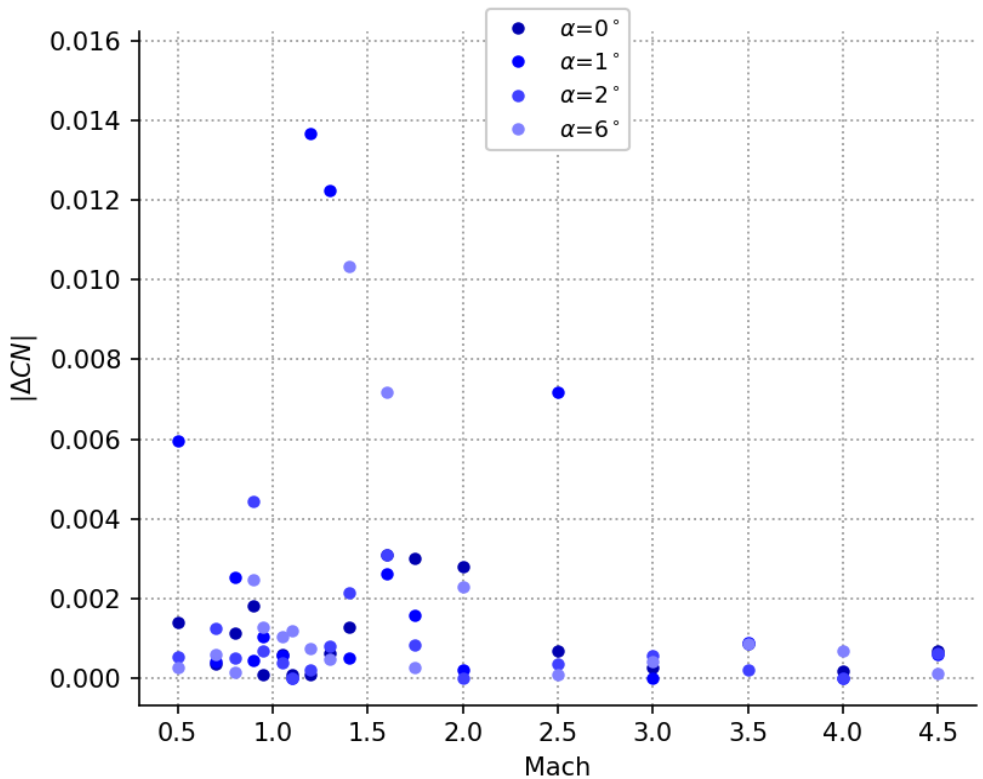


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

CORE - Normal Force Coefficient

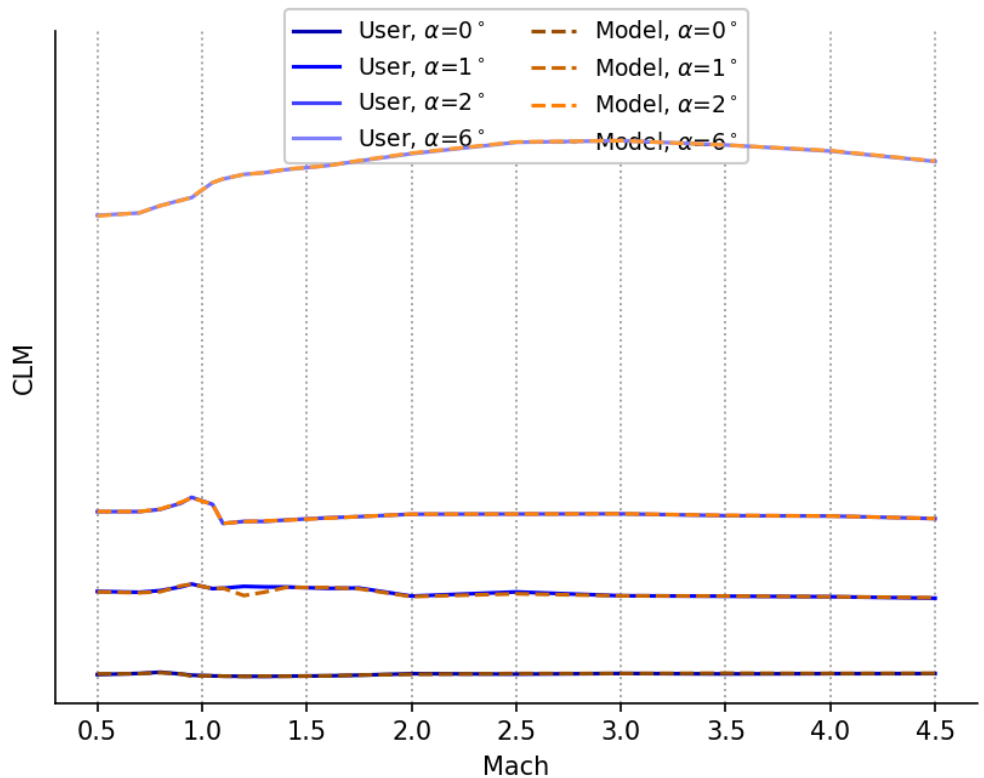


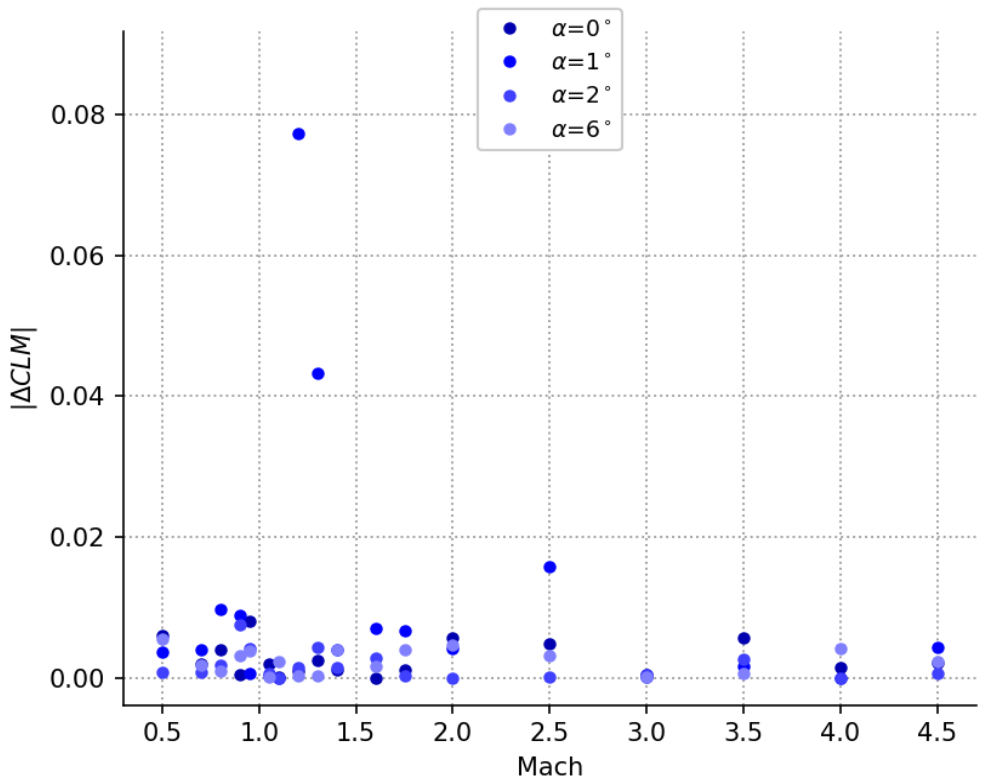


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

CORE - Pitching Moment Coefficient

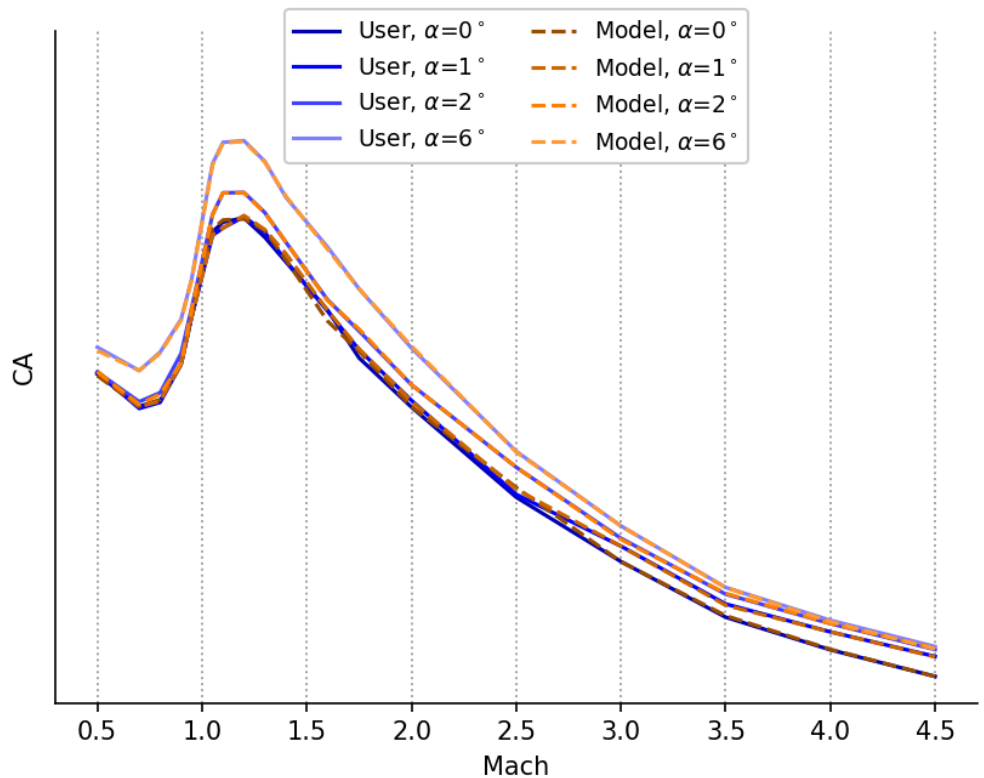


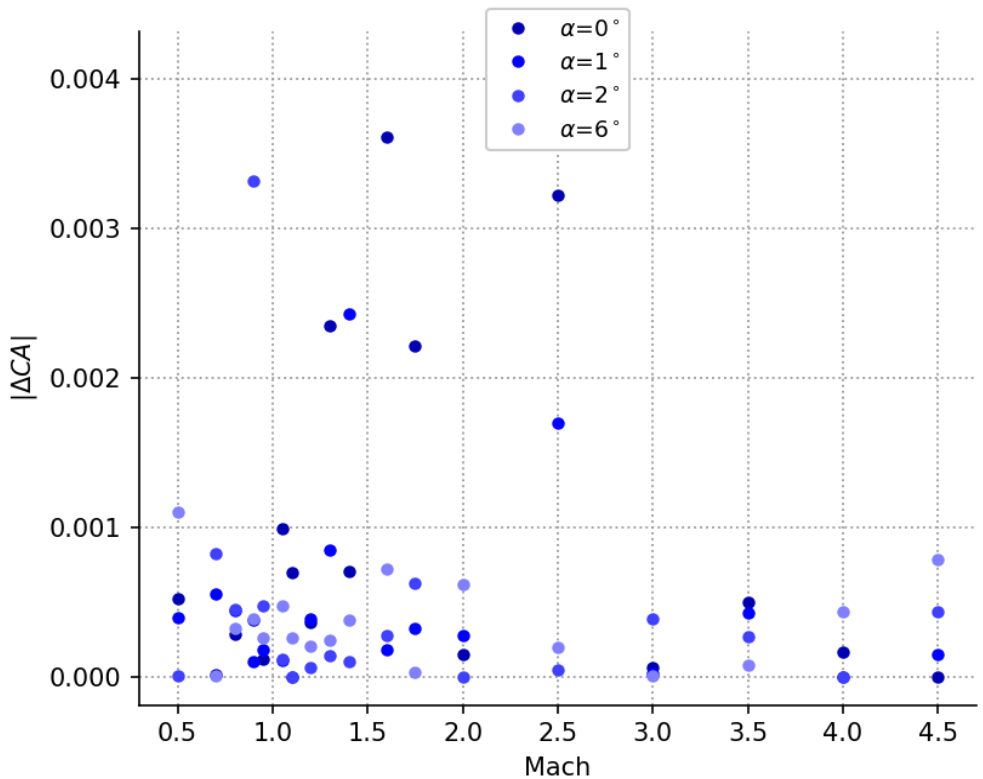


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

RSRB - Axial Force Coefficient

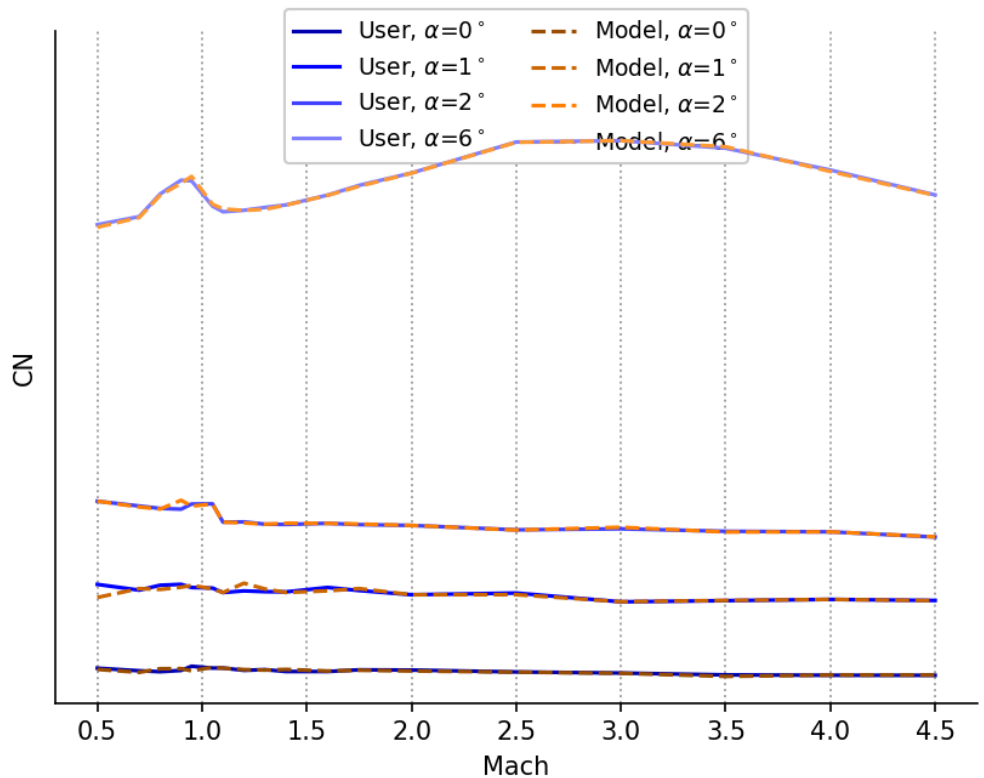


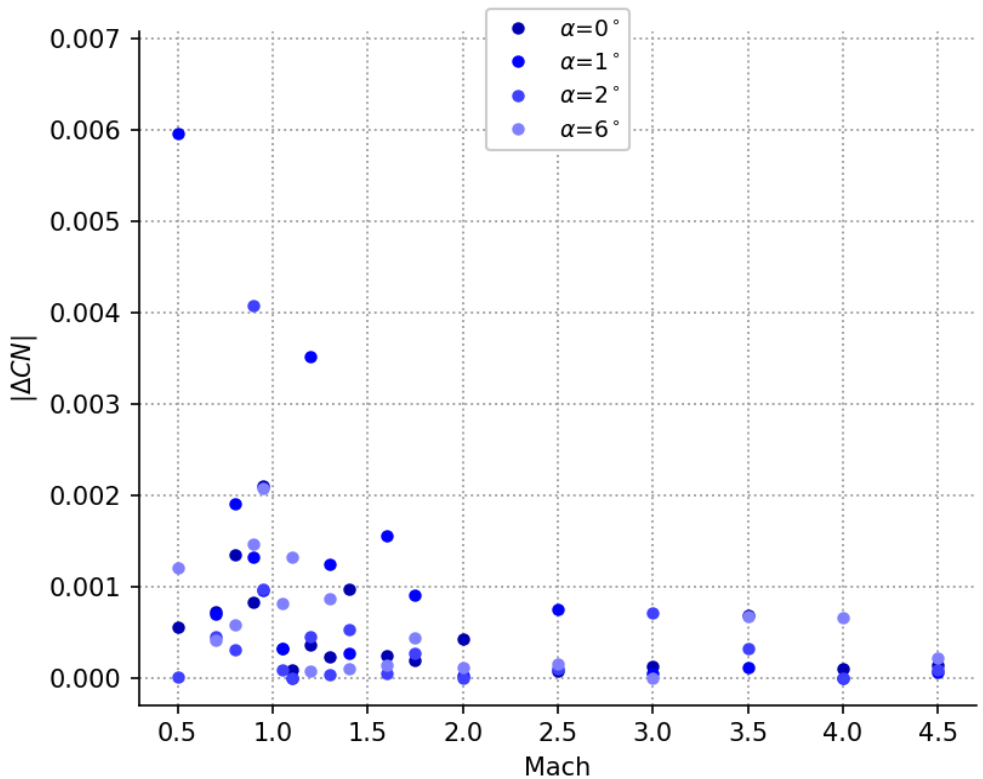


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

RSRB - Normal Force Coefficient

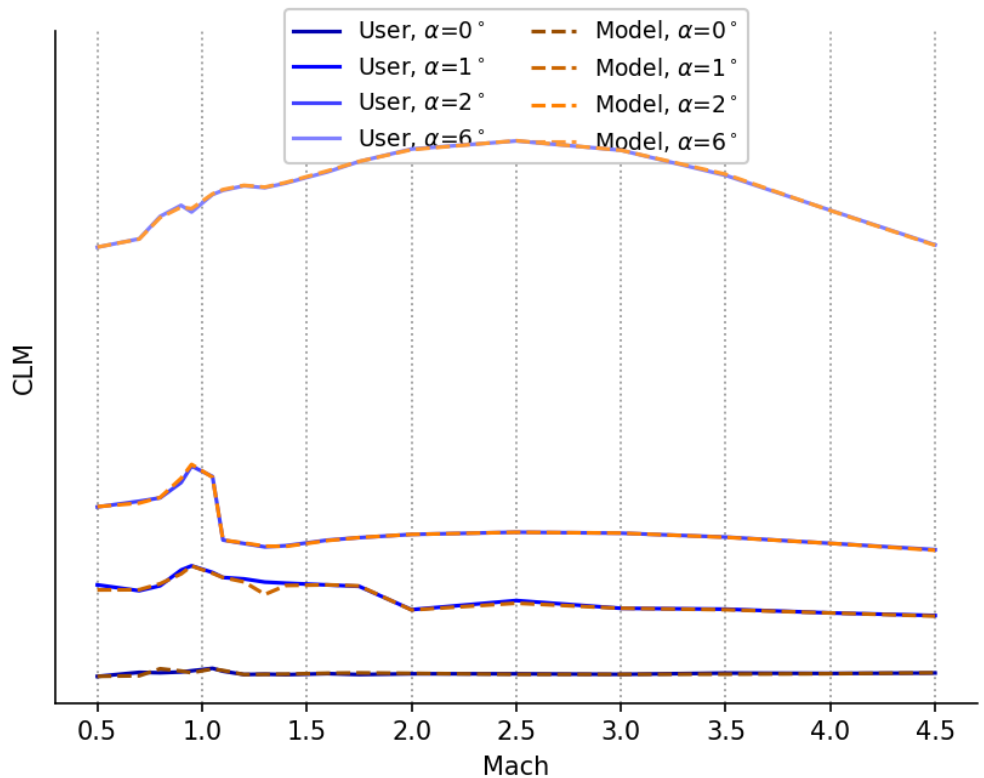


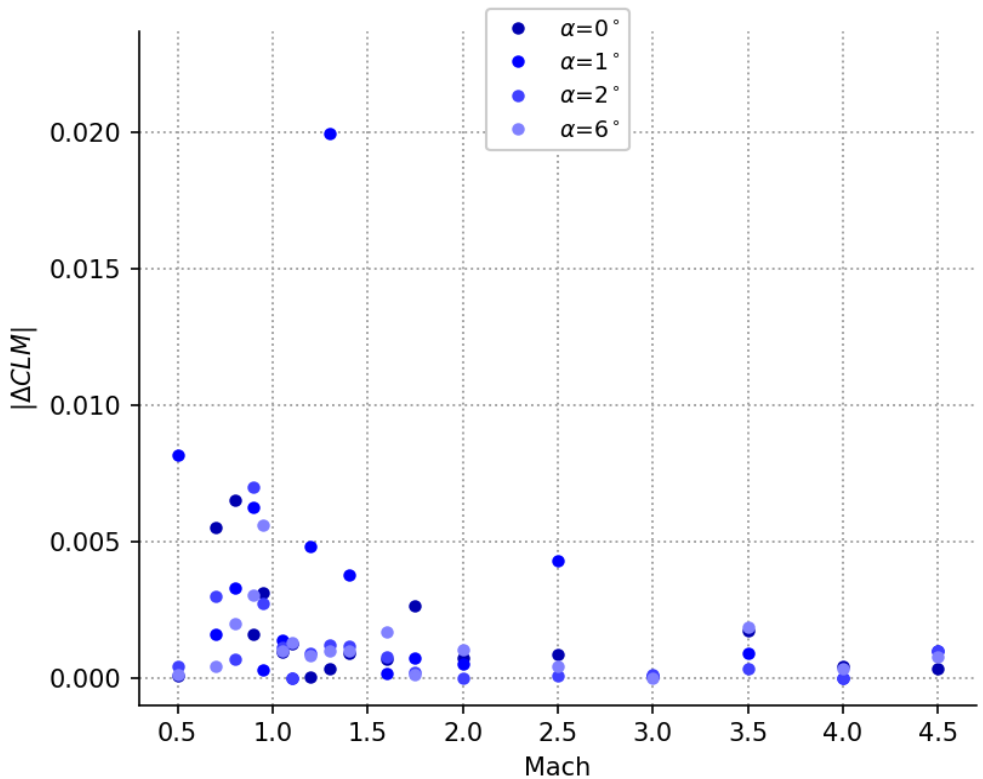


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

RSRB - Pitching Moment Coefficient

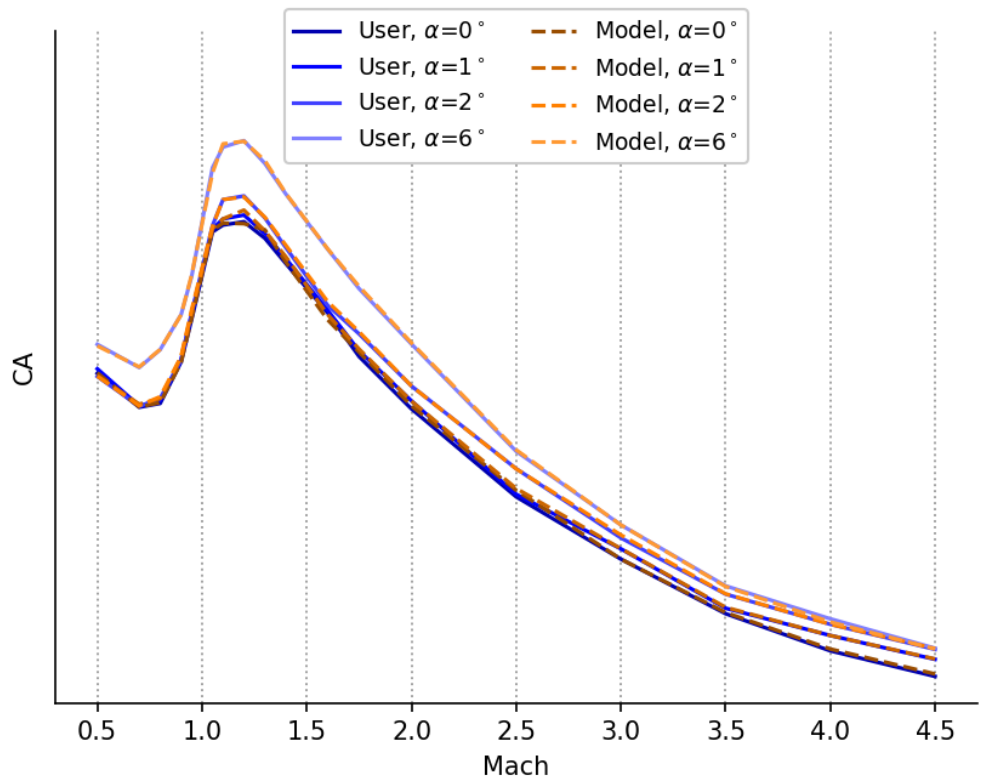


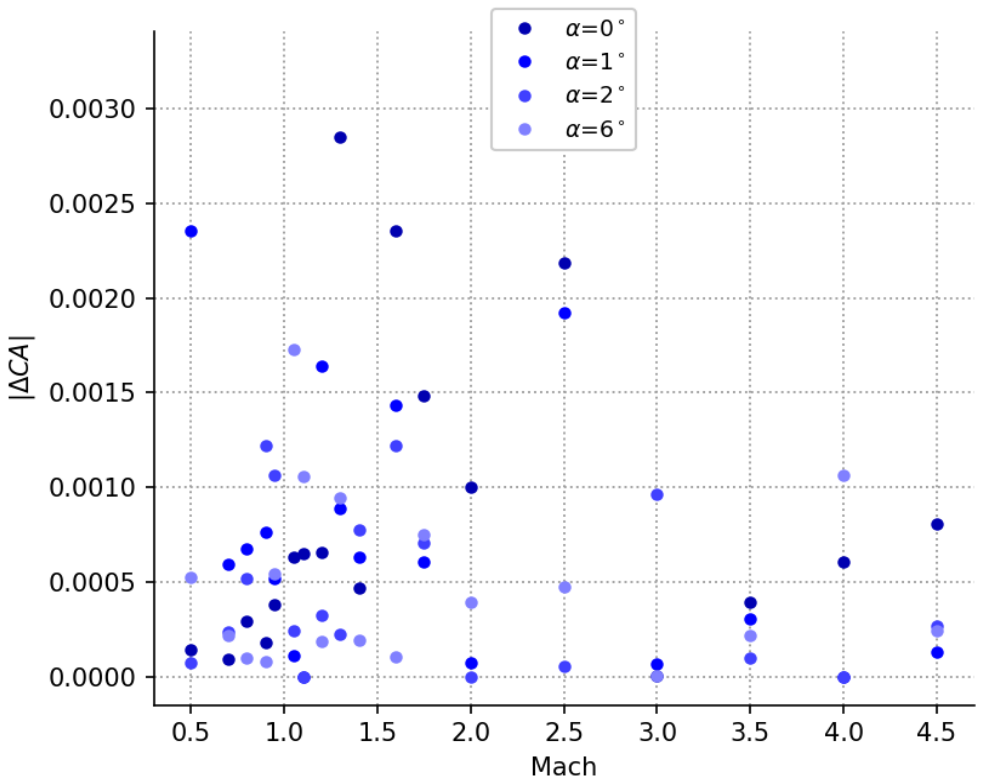


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

LSRB - Axial Force Coefficient

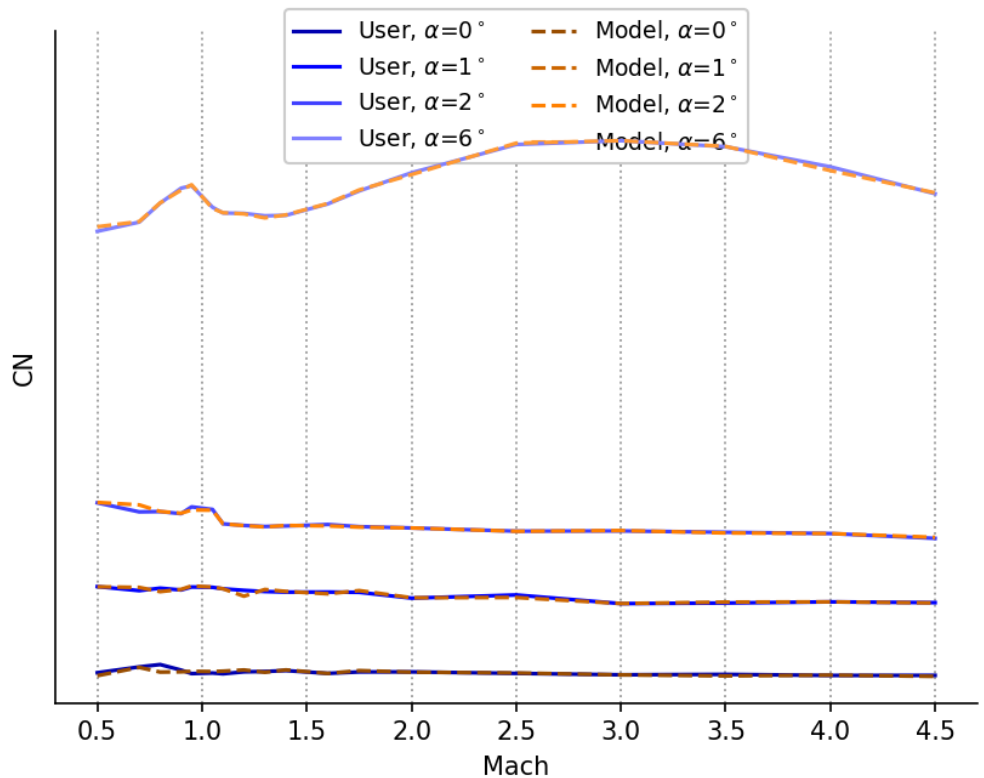


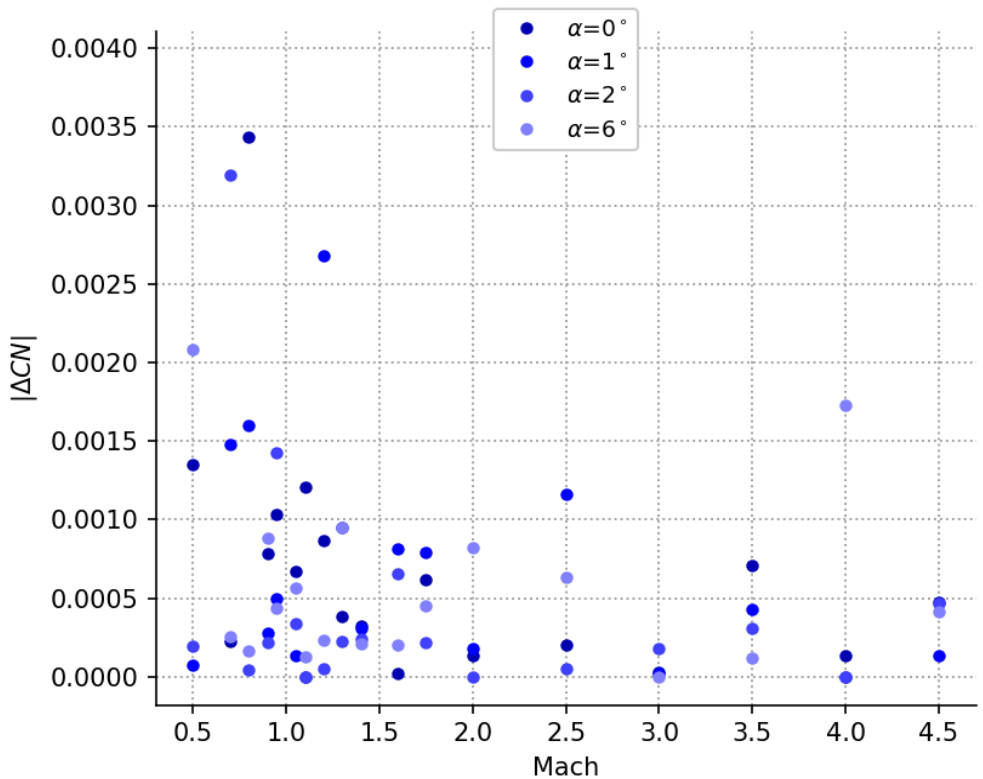


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

LSRB - Normal Force Coefficient

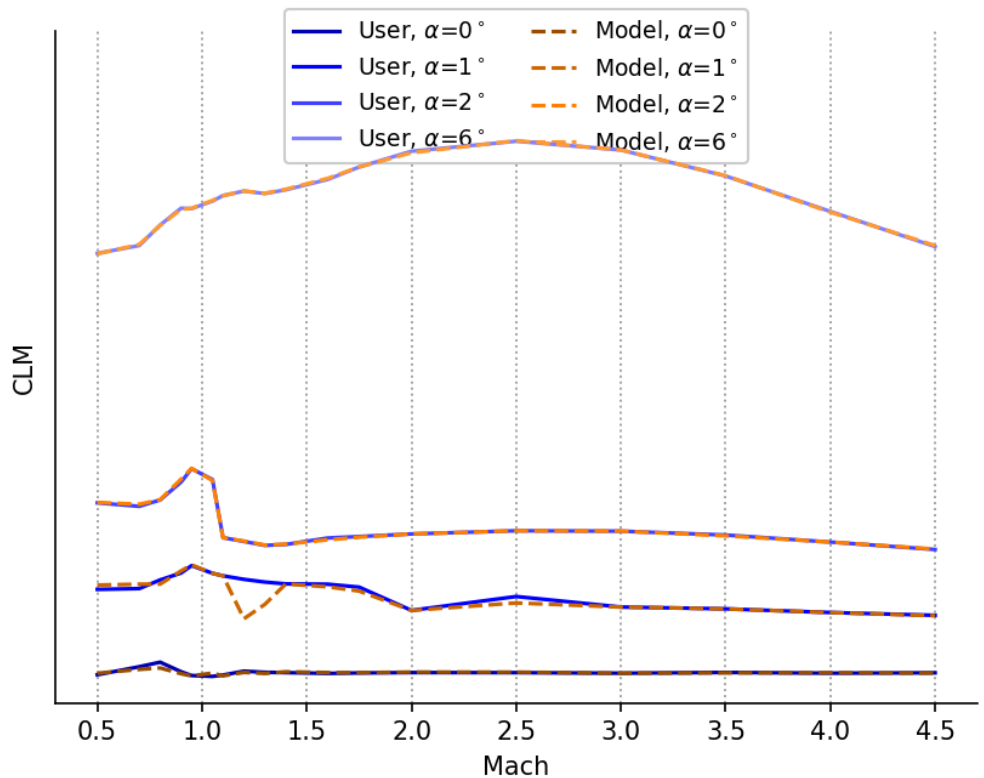


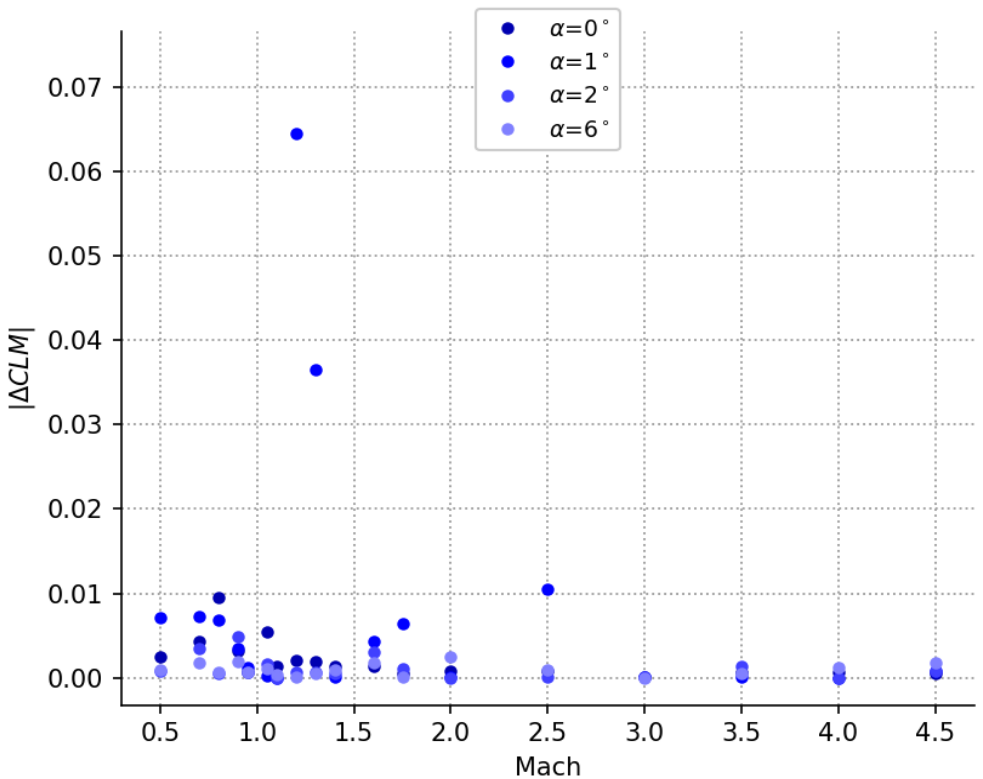


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Validation: User vs Model

LSRB - Pitching Moment Coefficient

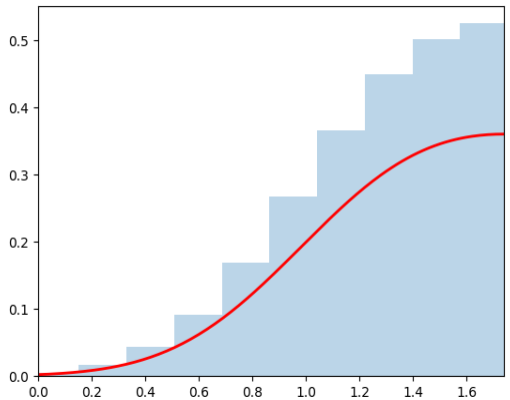


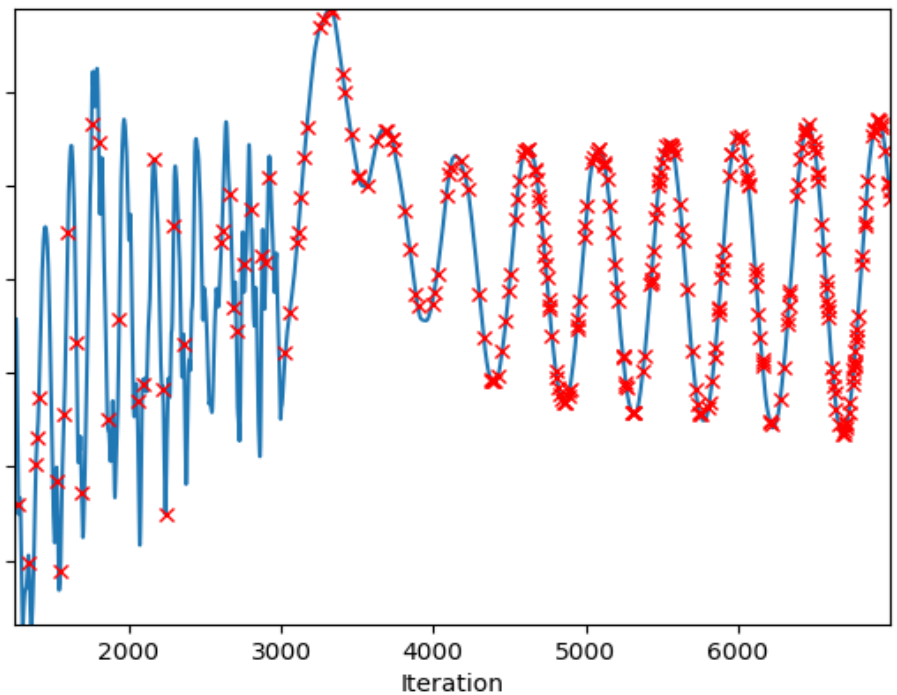


This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Current/Future Work

1. More generalized model
    - Probability sampling of iterative history
    - Feature selection studies
    - Support variety of feature variables
2. Support Additional Solvers
    - Supported: Cart3D, FUN3D, OVERFLOW
    - Next: Kestrel, LociCHEM





This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Summary

1. Machine learning techniques were shown to **reliably classify** complex iterative histories and be **more capable** than traditional convergence criteria.

2. A python module dubbed **PYCNIC** is in development to offer a convergence detection tool and machine learning pipeline for multiple solvers.

- AIAA SciTech2024 Paper:
  Joshua F. Diaz, Derek J. Dalle and Periklis E. Papadopoulos. “Deep Neural Network Based Convergence Classification for Computational Fluid Dynamics,” AIAA 2024-2515. AIAA SCITECH 2024 Forum. January 2024.
- CAPE: https://github.com/nasa/cape

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Acknowledgements

This work was funded by the NASA Space Launch System (SLS) Program in the Exploration Systems Development Mission Directorate (ESDMD). Computational resources were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

Ames SLS CFD Team

- Stuart Rogers, Derek Dalle, Jamie Meeroff, Guy Schauerhamer, Aaron Burkhead, Josh Diaz

CFD software highlighted in this work

- FUN3D (Langley)

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.



This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.