Rapid Aerodynamic Database Generation for the X-57 Maxwell Electrified Aircraft

Jared C. Duensing, Gaetan Kenway, Jeffrey A. Housman, James C. Jensen, Daniel Maldonado, Cetin C. Kiris

NASA Ames Research Center

Seung Y. Yoo, Michael Frederick, Trong Bui

NASA Armstrong Flight Research Center

AMS Seminar Series, October 13th, 2020

NASA Ames Research Center, Moffett Field, California

Outline

- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

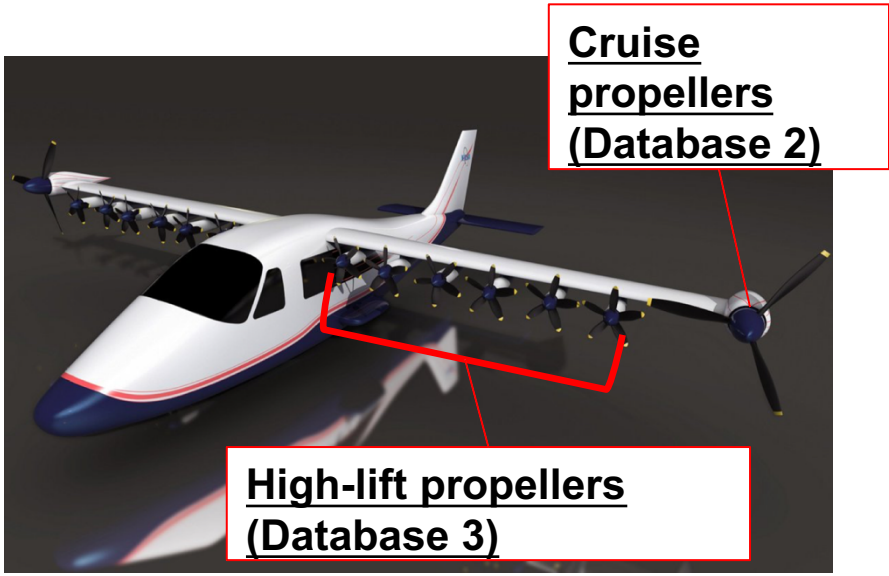
***Image source**: https://www.nasa.gov/press-release/nasa-electric-research-plane-gets-x-number-new-name*

X-57 Design Overview

Image source:
https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001218.pdf

X-57 CFD Analysis Objectives

- Establish best-practices to generate aerodynamic databases using the LAVA (Launch Ascent and Vehicle Aerodynamics) and Star-CCM+ flow solvers
- Apply best-practices to CFD databases which cover a variety of flight conditions and aircraft configurations
  - Database 1 (188 simulations): Power-off
  - Database 2 (233 simulations): Cruise power-on
  - Database 3 (2000+ simulations): High-lift power-on
- Post-process database results to assess aircraft performance and stability for the aircraft flight simulator
- Demonstrate advantages of simulating the X-57 with refactored LAVA



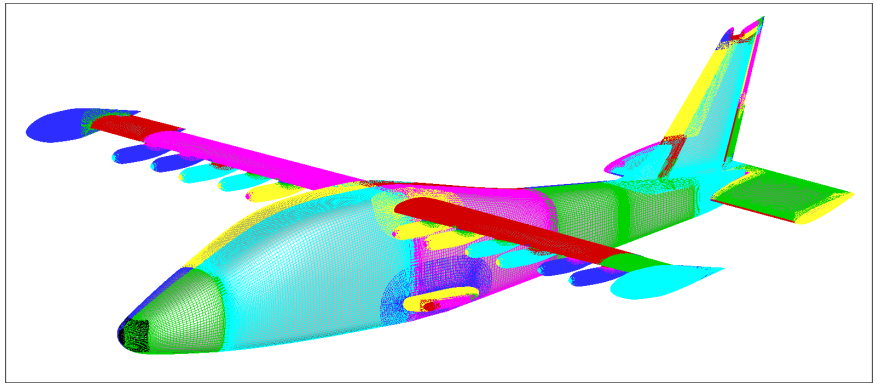
Outline

- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation[1]
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

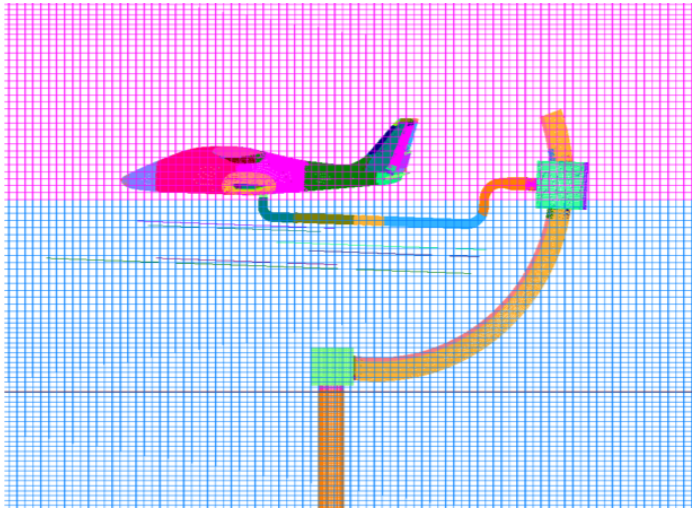
[1]Duensing, Jared C., et al. "Establishing Best Practices for X-57 Maxwell CFD Database Generation." *AIAA SciTech 2019 Forum*. 2019.

X-57 Database Preparation: Verification and Validation

1. Free-air mesh refinement study
2. Study effects of:
   a. Low Mach Number Preconditioning
   b. Turbulence Model
   c. Numerical Schemes
3. Wind tunnel validation studies



X-57 structured curvilinear grid



X-57 structured curvilinear grid in wind tunnel

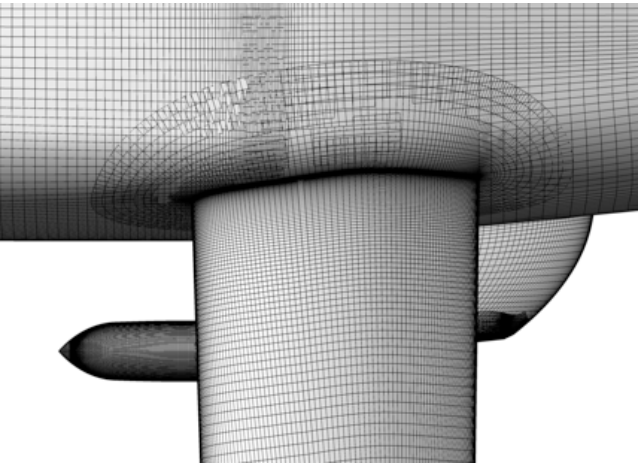
Mesh Refinement Study Statistics

Structured Curvilinear

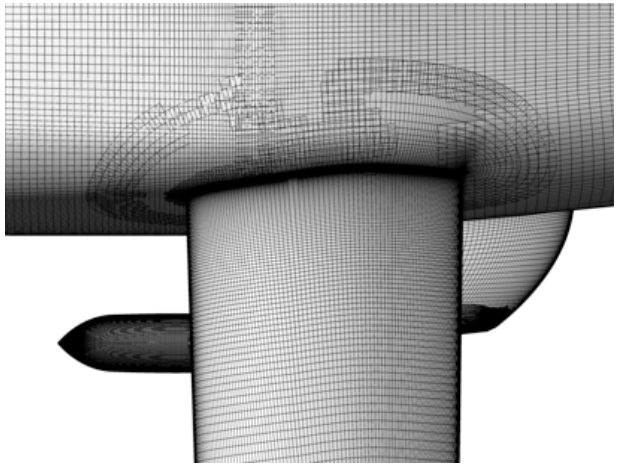
Mesh	Grid Points
Coarse	80.0 M
Medium	159 M
Fine	322 M

Unstructured Polyhedral

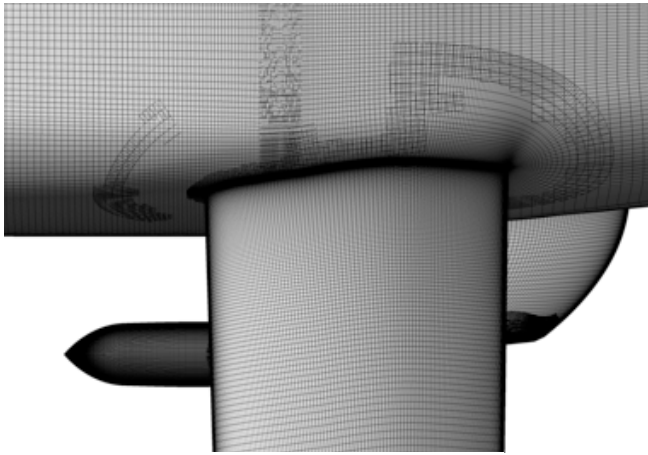
Mesh	Cell Count
Coarse	17.2 M
Medium	49.7 M
Fine	120 M



Structured Coarse



Structured Medium

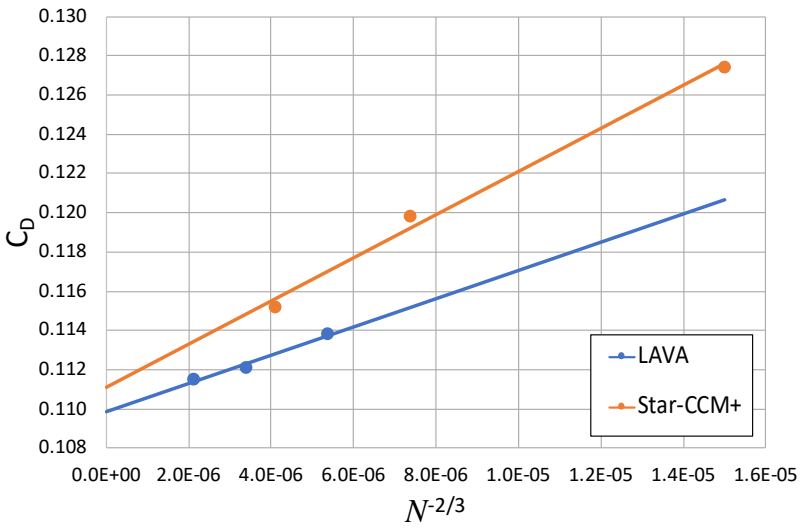


Structured Fine

Selecting Mesh Refinement Levels

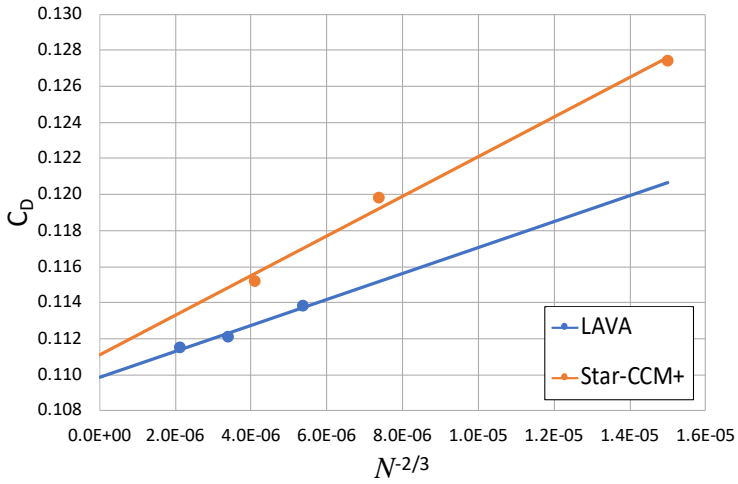
- ***Freestream Condition***: Altitude = 2,500 ft, Mach = 0.052, Re_{MAC} = 121,600, α = 6.0°, β = 5.0°
- Second-order asymptotic convergence observed for drag, with the extrapolated drag for an "infinitely fine" grid within 1.1% error relative to other solver

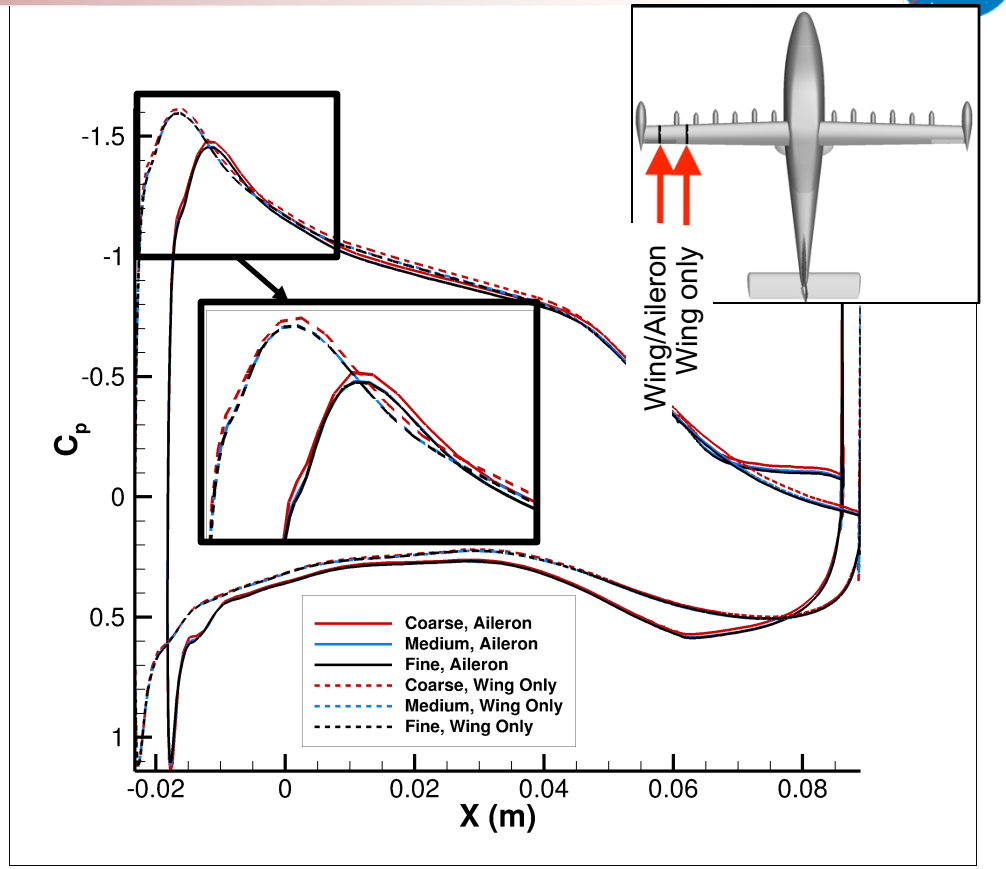
	LAVA Curvilinear		Star-CCM+	
Grid	***CL***	***CD***	***CL***	***CD***
Coarse	1.0072	0.1138	1.0867	0.1274
Medium	1.0070	0.1121	1.0599	0.1198
Fine	1.0070	0.1115	1.0122	0.1152
Experiment	1.068	0.099	1.068	0.099



Pressure Distribution at Selected Wing Location

- Second-order asymptotic convergence observed for drag, with the extrapolated drag for an "infinitely fine" grid within 1.1% error relative to other solver
- Asymptotic convergence behavior observed in pressure distributions





C_p distributions shown for the LAVA solutions

Selecting Appropriate Solver Settings for LAVA

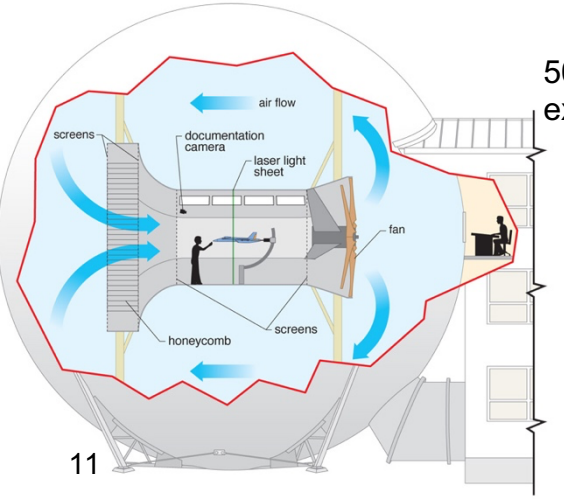
- Numerical discretization is a second-order convective flux scheme with Koren limiter
- "Medium" level grid was used to determine whether flow is best modeled fully turbulent or fully laminar (unless determined necessary, transitional flow would not be modeled)
- Effects of low-Mach preconditioning would be tested due to low speed flow
- Additional LAVA Unstructured simulations were also run to ensure consistency of solver settings within the LAVA framework

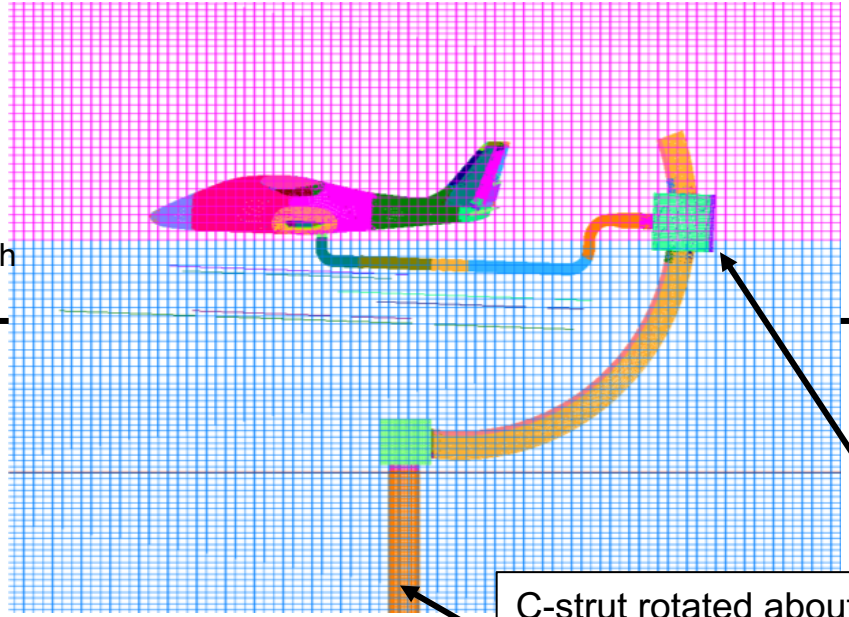
	LAVA Curvilinear		LAVA Unstructured	
Modeling Approach	***CL***	***CD***	***CL***	***CD***
Laminar without Preconditioning	0.579	0.151	1.016	0.171
Laminar with Preconditioning	0.585	0.146	0.664	0.160
SA Turbulence without Preconditioning	1.015	0.119	1.112	0.154
SA Turbulence with Preconditioning	1.007	0.114	1.066	0.125
Experiment	1.068	0.099	1.068	0.099

Wind Tunnel Validation

- Tunnel simulation designed to emulate blockage effects of wind tunnel hardware
- Test-section geometry extended 50 body lengths upstream and downstream (excludes inlet, diffuser, and surrounding recirculation chamber)

Image source:
https://researchdirectorate.larc.nasa.gov/12-foot-low-speed-tunnel-12-ft-lst/





50x body length extension

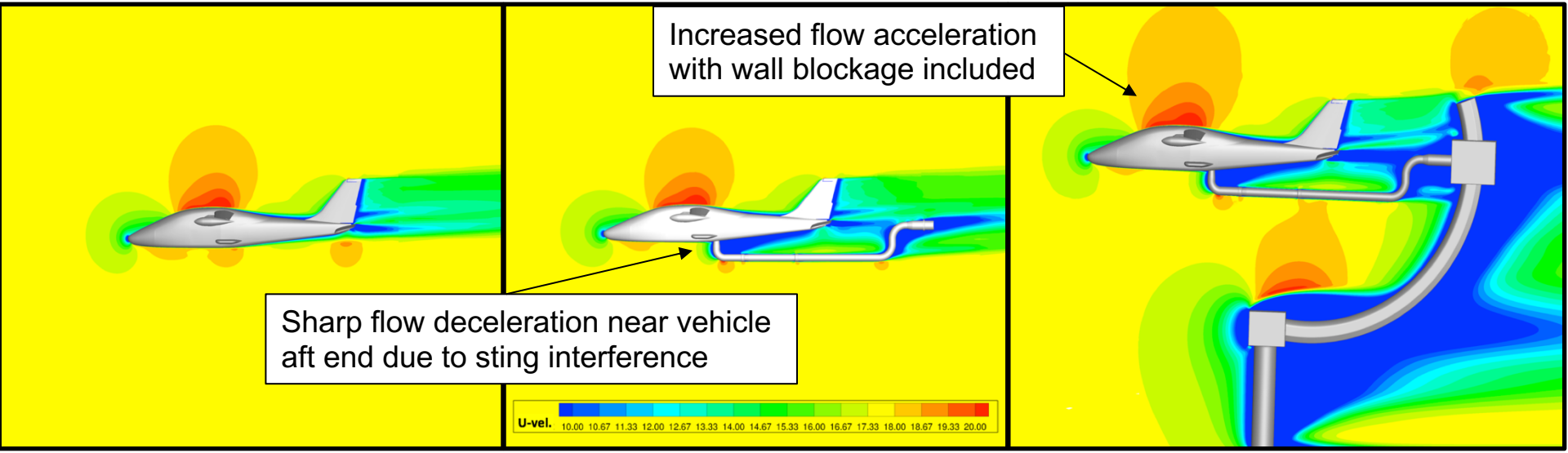
50x body length extension

Sting translated along C-strut to adjust angle of attack

C-strut rotated about a vertical axis to create sideslip

Validation Simulation Results (U-Velocity on Symmetry Plane)

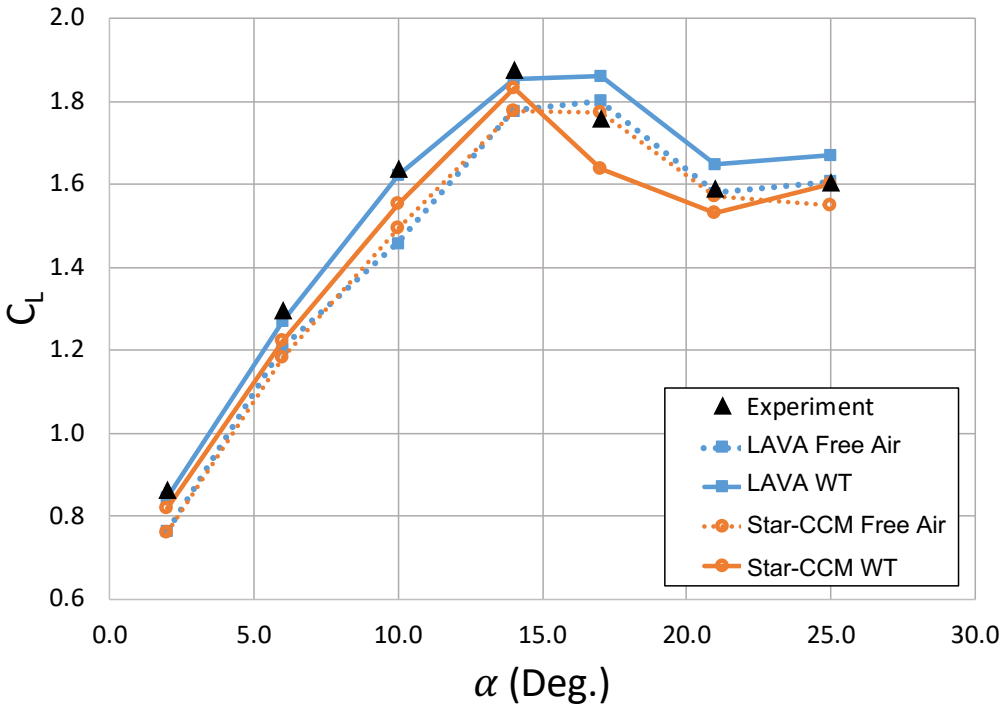
- Substantial qualitative differences in fluid dynamics resulting from sting, C-strut and wind tunnel walls
- Hardware locally impacts flow field while pressure effects also propagate upstream to test article location



Free air | **Free air + sting** | **Free air + sting + wind tunnel**

Angle of Attack Sweep Results

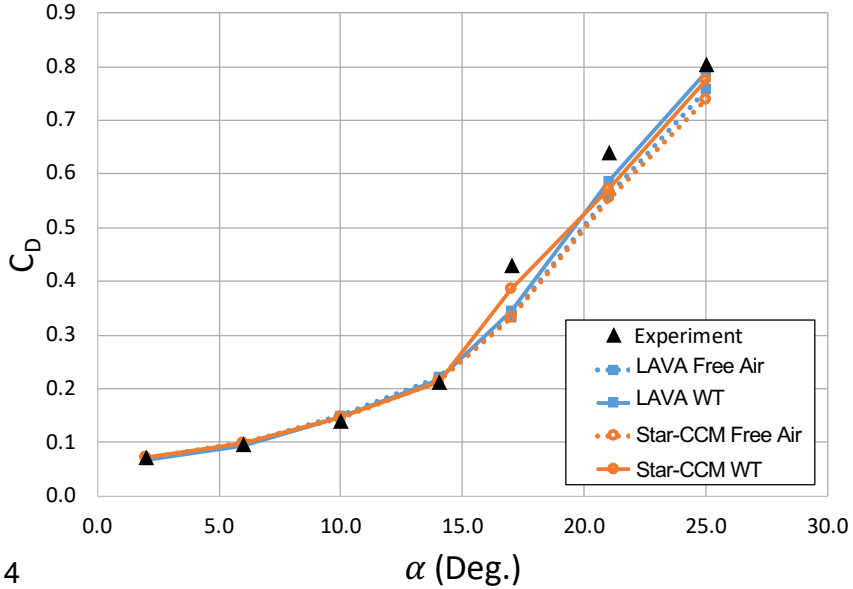
- Comparison of integrated loads at multiple angles of attack in free air and with wind tunnel hardware
- For both codes, incorporating wind tunnel effects to the CFD simulation improve lift predictions relative to experiment considerably across the linear regime of the C_L vs. α curve

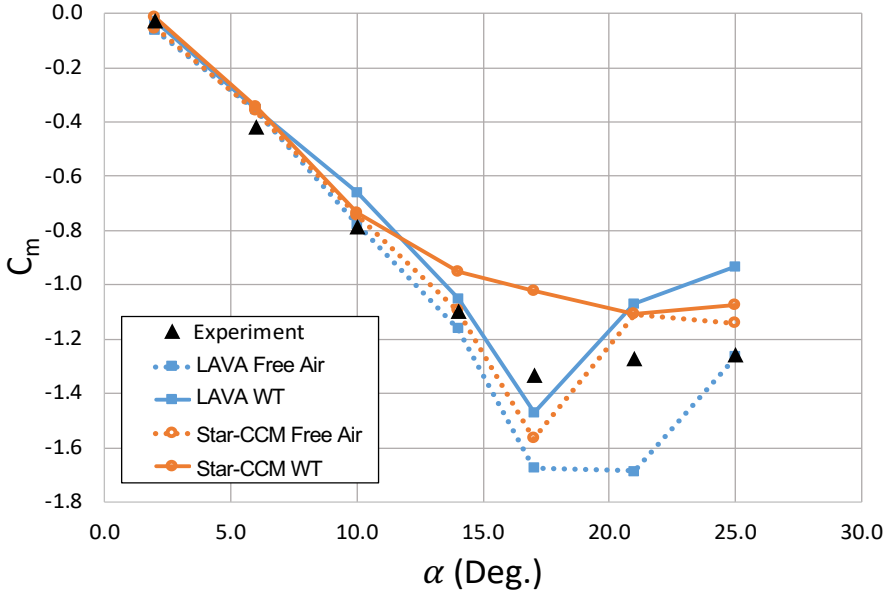


WT = Wind Tunnel Walls + Sting + C-strut

Angle of Attack Sweep Results

- Minor changes noted in drag and pitching moment in the pre-stall regime, but no significant change in error relative to experiment
- Larger scatter in results observed post-stall independent of modeling approach due to limitations of RANS models in the post-stall regime

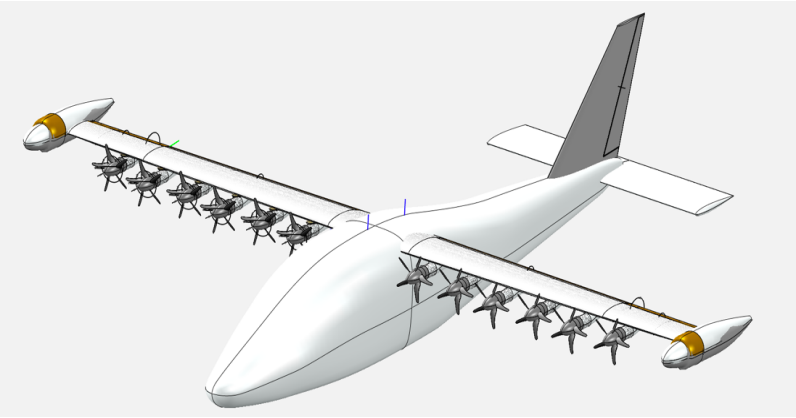




WT = Wind Tunnel Walls + Sting + C-strut

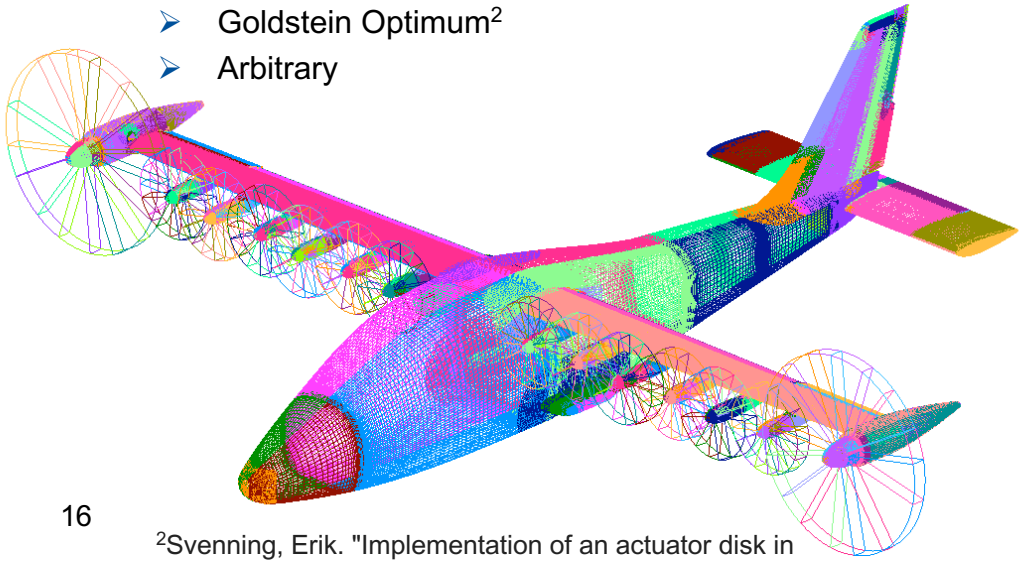
Power-On Simulation Preparation

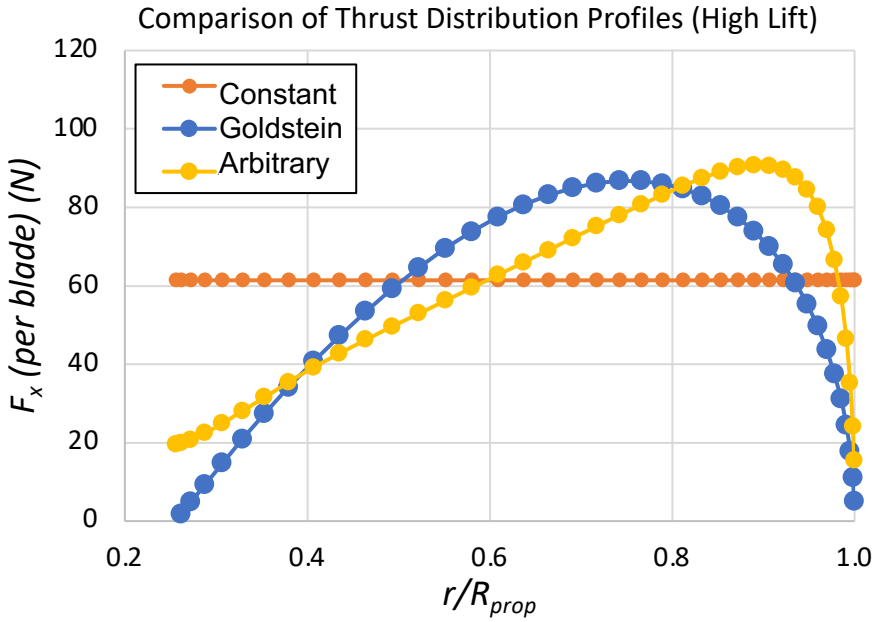
- Best practices from the power-off CFD database preparation were also applied to power-on database simulations
  - Mesh refinement studies
  - Numerical scheme/turbulence model determination
  - Code-to-code comparison
- Refinement level and numerical schemes/models identical to power-off simulations deemed adequate
- Additional studies need to be performed to determine best propulsion modeling approach



Propulsion Modeling Approach

- *Actuator zones* model the momentum imparted from the propeller to the surrounding flow field without the computational cost of simulating the moving blade
- Axial forces (thrust) and tangential forces (torque) as a function of propeller radius are needed to define the actuator zone model
- Radial thrust and torque distribution options studied with LAVA and Star-CCM+
  - Constant
  - Goldstein Optimum[2]
  - Arbitrary

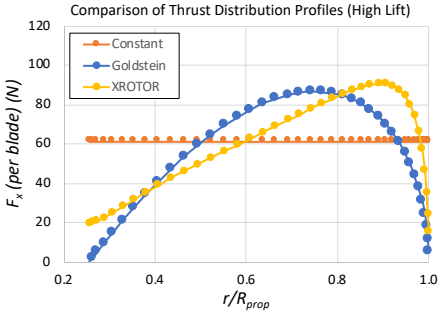




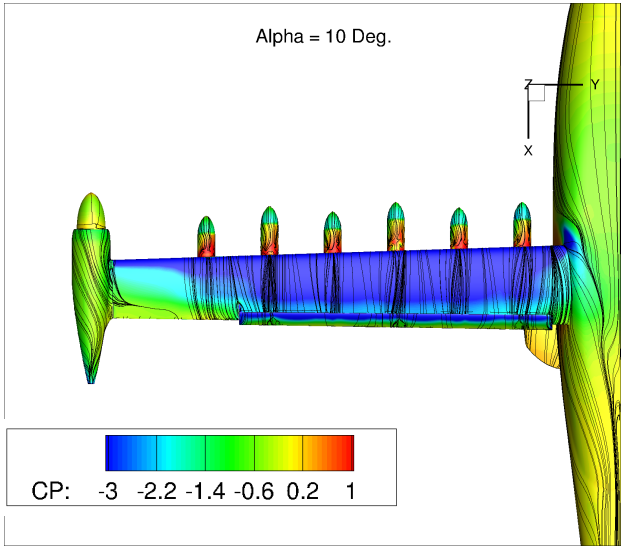
[2]Svenning, Erik. "Implementation of an actuator disk in OpenFOAM." *(Chalmers University of Technology, Sweden, 2010)* (2010).

Selecting Actuator Zone Distributions

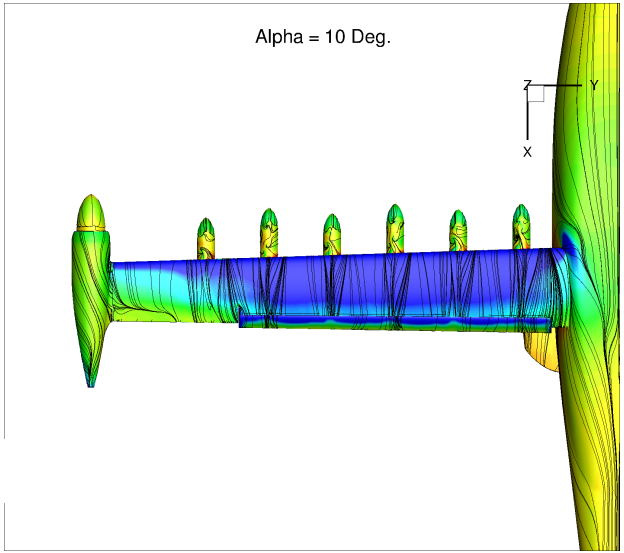
- For high-lift propulsion cases, XROTOR[2] data was available to define an arbitrary radial thrust and torque distribution
- Initial CFD simulations were performed using LAVA to understand impact of thrust and torque distributions on the solution (Altitude: 6000 ft., Re_{MAC} = 1,235,000, Mach = 0.098, α = 10°)



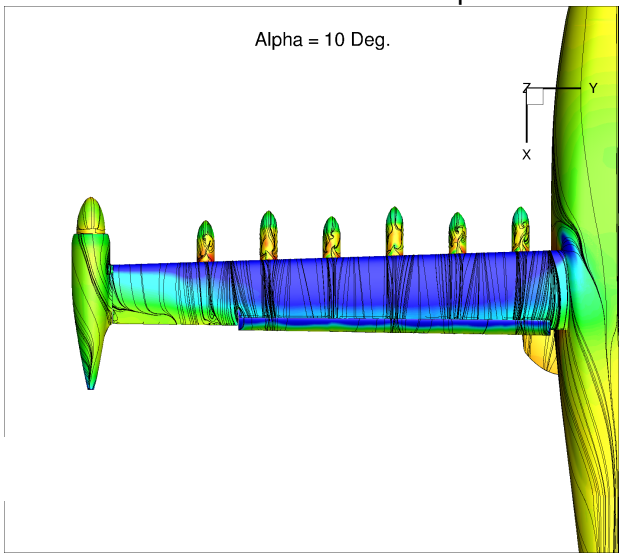
Constant Thrust and Torque



Goldstein Thrust and Torque



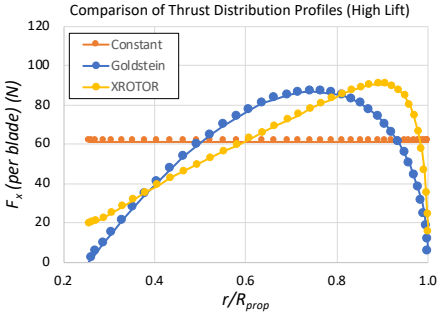
XROTOR Thrust and Torque



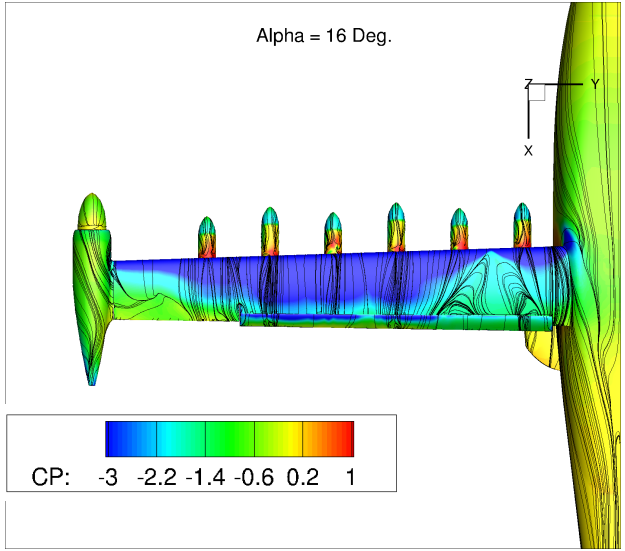
[2]Drela, M., and H. Youngren. "XROTOR: an interactive program for the design and analysis of ducted and free-tip propellers and windmills, 2011.[Software] Available at: http://web. mit. edu/drela."

Selecting Actuator Zone Distributions

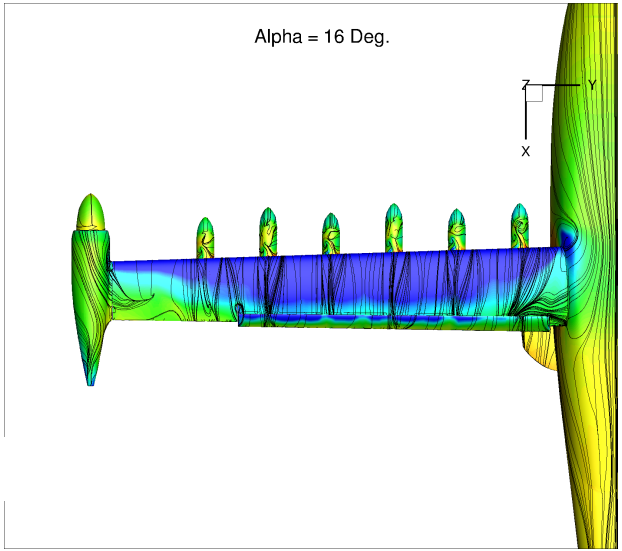
- Initial CFD simulations were performed using LAVA to understand impact of thrust and torque distributions on the solution
- Altitude: 6000 ft., Re_{MAC} = 1,235,000, Mach = 0.098, α = 16° shown below
- Separation behavior at high angle of attack highly dependent on thrust and torque distribution



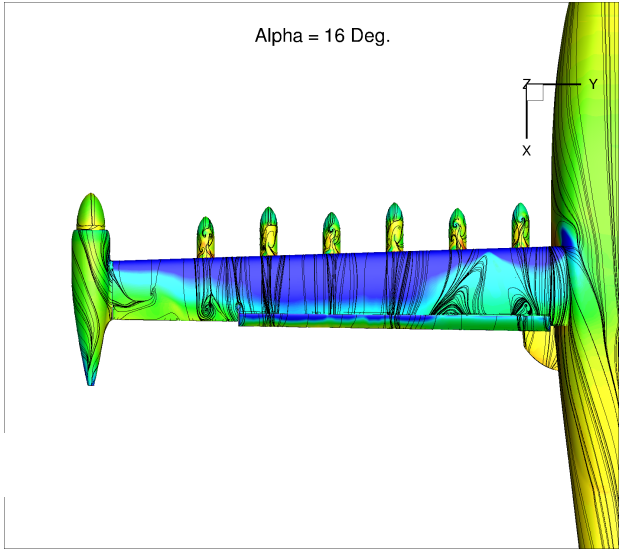
Constant Thrust and Torque



Goldstein Thrust and Torque

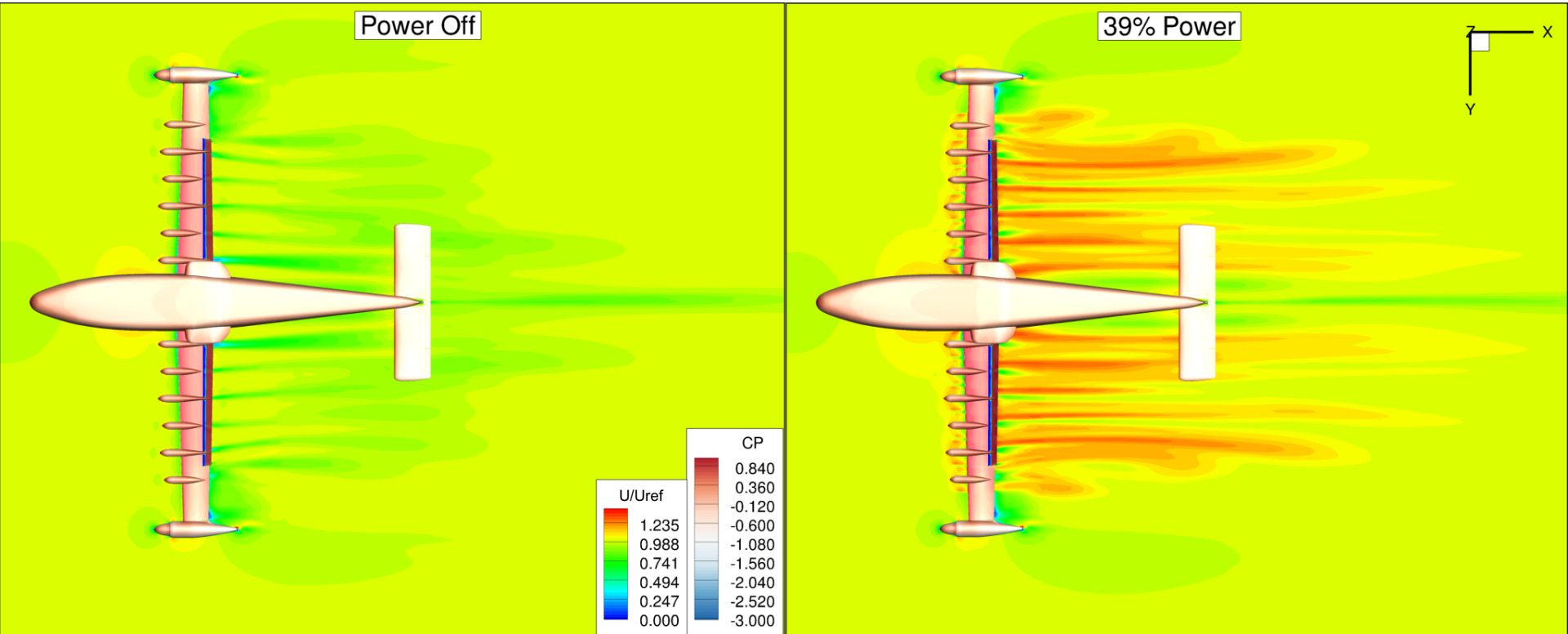


XROTOR Thrust and Torque



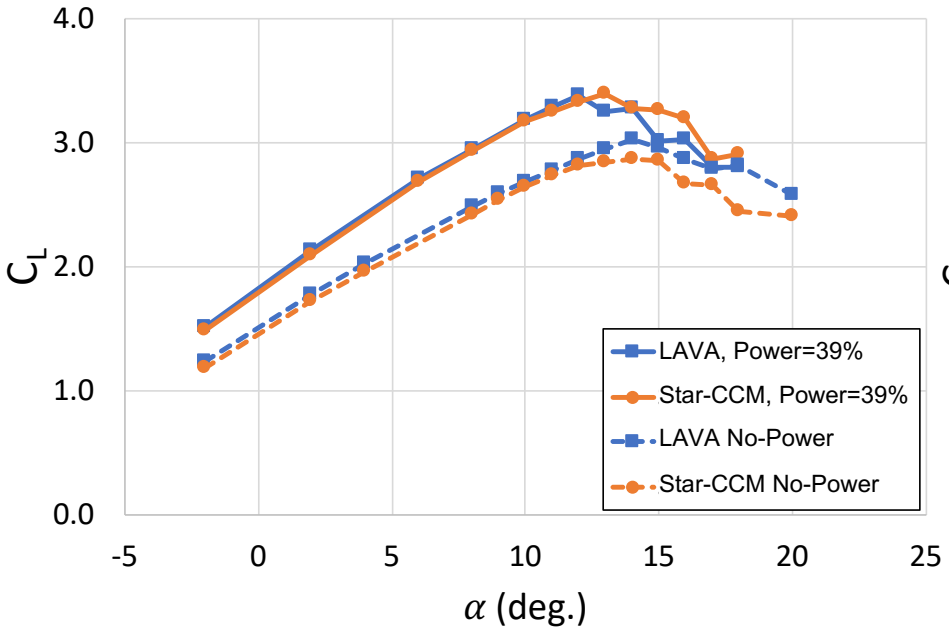
LAVA High-Lift Power-On Flow Visualizations

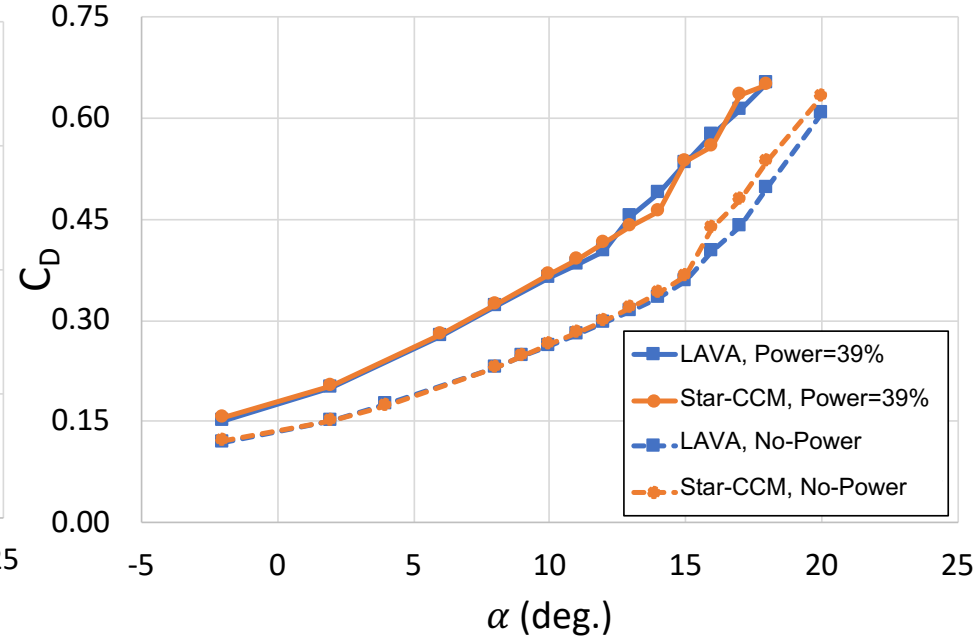
- Altitude: 2,500 ft, Mach = 0.119, Re_{MAC} = 1,682,000, Flaps = 30°
- Dimensionless streamwise velocity (U/U_{ref}) is shown on slice plane, pressure coefficient on aircraft surface



High-Lift Power-On Sample Database Results

- LAVA predicts power-on lift within 1.4% of Star-CCM pre-stall, and within 8.5% post-stall
- LAVA predicts power-on drag within 3.1% of Star-CCM pre-stall, and within 5.7% post-stall

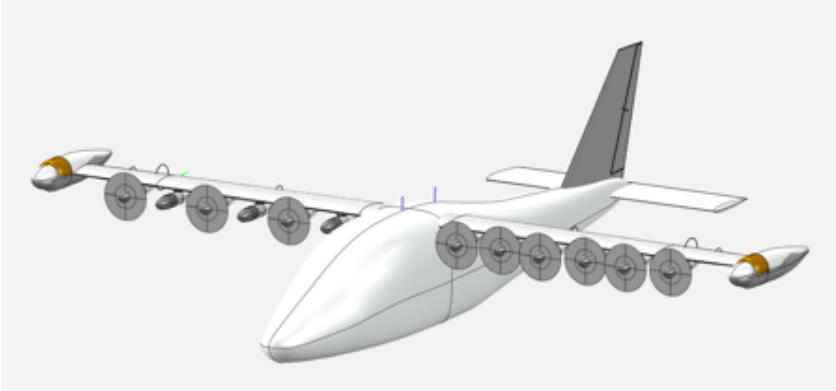


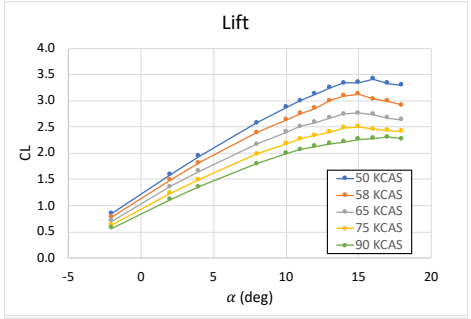


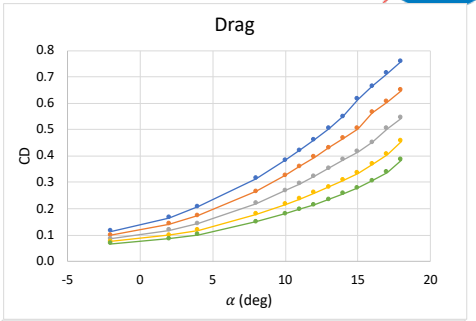
Condition: Altitude: 2,500 ft, Mach = 0.119, Re_{MAC} = 1,682,000, Flaps = 30°

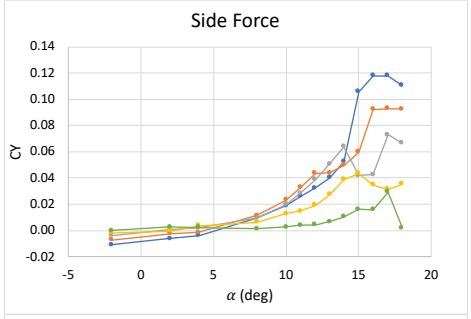
High-Lift Power-On Sample Database Results

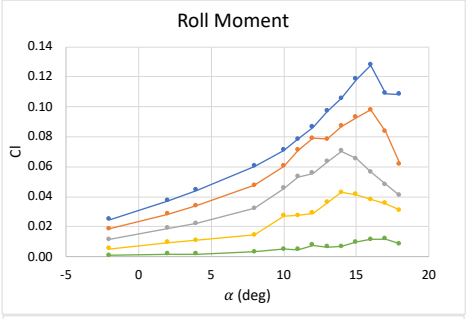
- Comparison curve families across the flight envelope shown for a sample power “failure” scenario
- Each curve represents a different high-lift power setting corresponding to a particular freestream condition

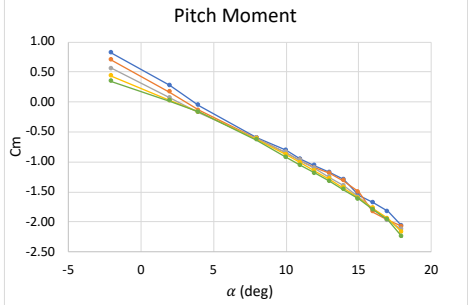


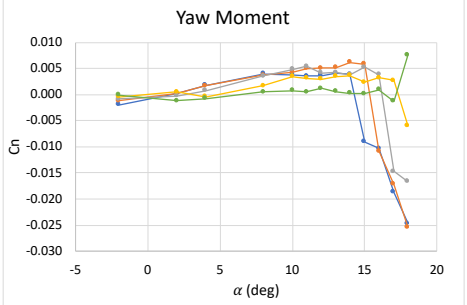










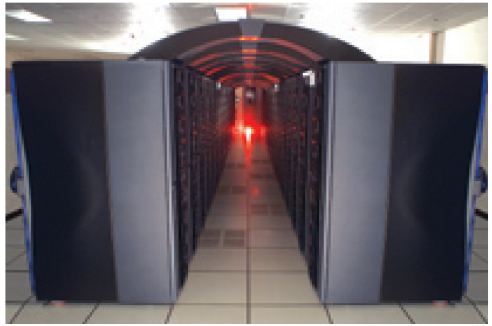


Outline

- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

LAVA Curvilinear Solver Refactoring

- Last major code restructuring took place at the turn of the century to support scalar processors
- Current computer architectures are vastly different:
  - Each node consists of 2 or more processors, each containing dozens of cores
- "Flat-MPI" approach: 1 MPI rank per processor



Chapman: 1024 Nodes SGI Origin 3800
(0.8 TFlops)

Aitken: 1152 40-core Nodes
(3690 TFlops)

LAVA Curvilinear Solver Refactoring

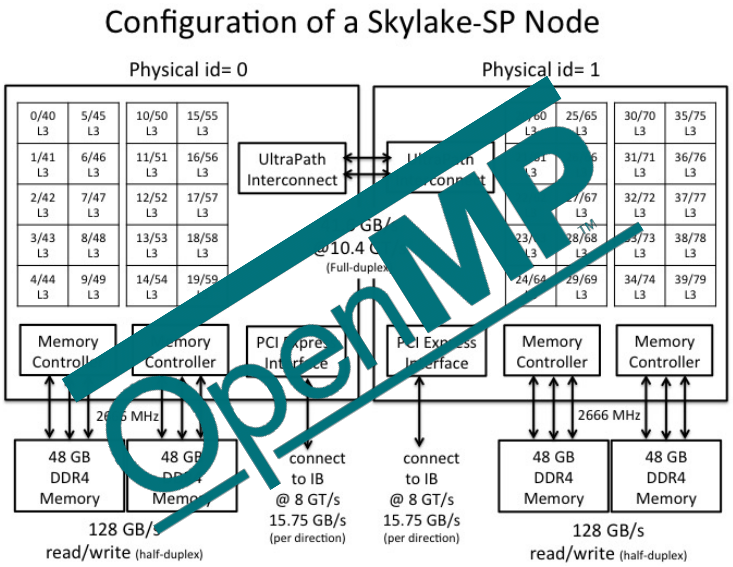
- Current refactoring aims to better match code structure to modern hardware:
  - Each “processor” is a shared memory cluster that can directly access all memory on the node
- Using one MPI rank per core is unnecessary and wasteful:
  - Redundant ”communication” between cores on node
  - Reduction in effectiveness of linear solver due to vastly increased number of subdomains
- Overall goal is to improve the computational efficiency of the flow solver while retaining the same discretizations

LAVA Curvilinear Solver Refactoring

- Code improvements fall into two main categories:
- Algorithmic Improvements
  - “Doing less work”
- “Computer Science” improvements:
  - “Doing the same amount of work faster”
- The order of magnitude speed-up in the refactored code is a combination of these effects

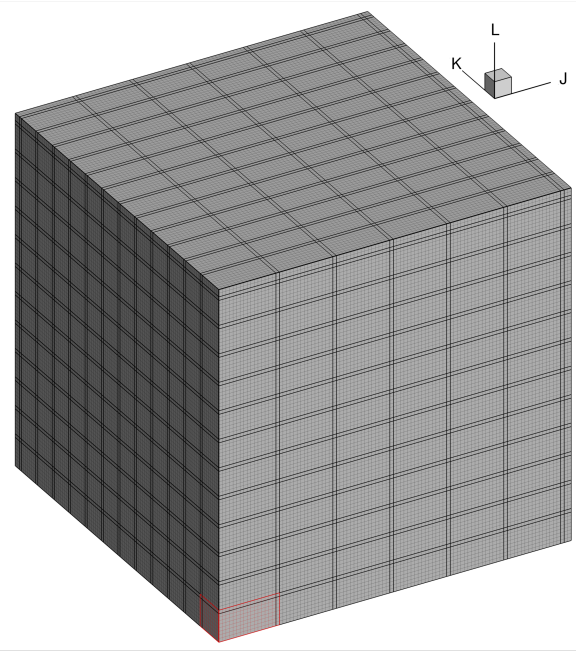
Refactored Architecture Solution: Hybrid MPI + OpenMP

- MPI + OpenMP is most common manifestation of “MPI+X”
- Ground-up hybrid implementation
- Much more challenging from a software engineering perspective
- All algorithms must be parallelized using both paradigms
- OpenMP 4.0 standard features make implementing many algorithms much easier



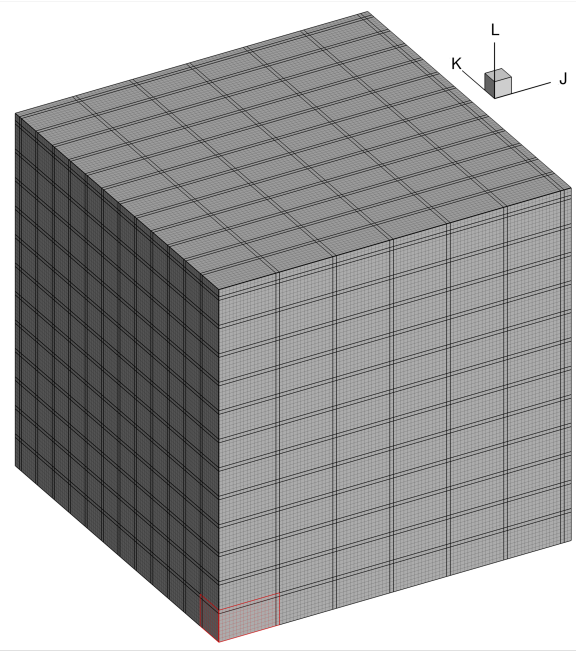
Refactored Architecture Implementation: Tiling

- OpenMP opens a new level of parallelism
- Use high-level OpenMP approach to limit number of parallel regions
- Tiles are (very) small chunks of a mesh block O(1000) nodes
- Tiles are sized such that all working variables for RHS evaluation fit in cache
- A single core will be responsible for hundreds of tiles

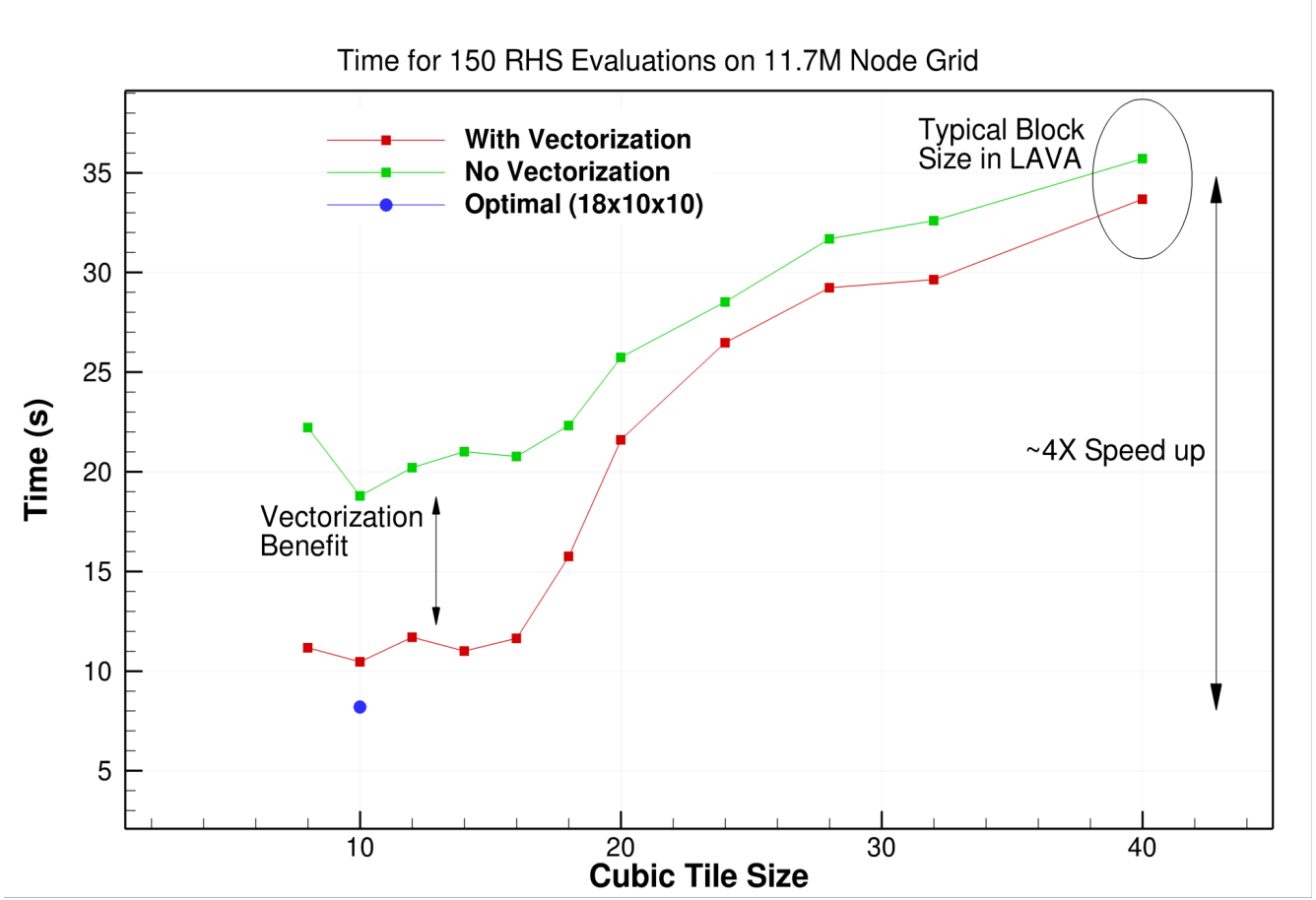


Tiling Advantages

- Tiling provides OpenMP-level of parallelism: Loops over tiles
- Dynamic load balancing for free
- MPI Load balancing vastly improved
  - Much easier to partition many small tiles than a small number of large blocks
- OpenMP eliminates the disadvantage of small domains: Most halo communication is performed using shared memory
- Easy to overlap communication and computation



Computer Science Improvement



Algorithmic Improvements

- Alternating line-Jacobi replaced with matrix-free based GMRES solution method
- Computer science improvements in RHS evaluation enable cost-competitive matrix-free matrix-vector product
- More accurate linear solutions enable CFL number up to O(1000)
- Non-linear iterations are more expensive, but vastly improved non-linear convergence

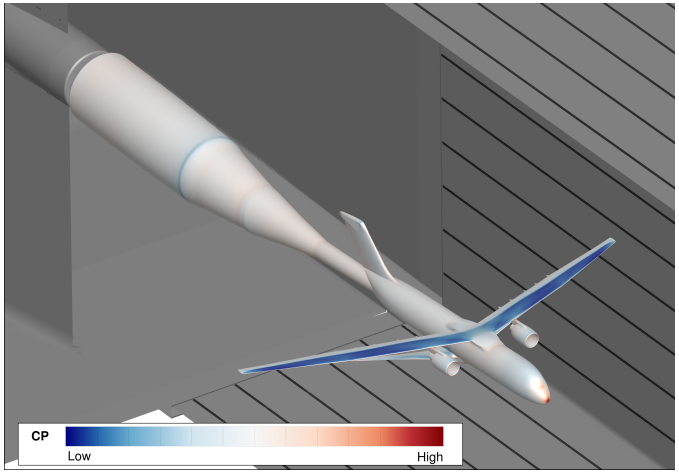
Boeing Transonic Truss Braced Wing (TTBW)

Background:

The Boeing TTBW project involves developing NASA CFD best practices for Transonic Truss-Braced wing concepts using experimental data for comparison

Simulations:

This data was simulated using both original and refactored LAVA. Not only was the resource cost significantly decreased using refactored LAVA, there was also a much tighter solution convergence



Mach 0.8 TTBW model simulated in-tunnel. The black lines represent where the baffled slots from the 11-ft wind tunnel test section are modeled with a porous wall BC in the final phase of these simulations.

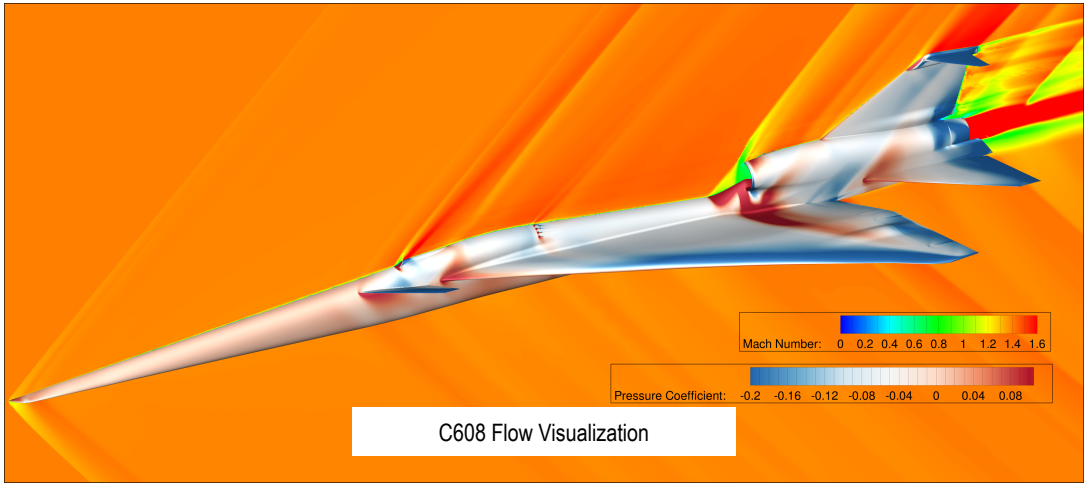
	Cores	Run Time	SBU	Resource Reduction	Mach Number Oscillation
Original	1100	72 Hours	3205	-	4e-3
Refactored	3000	1 Hours	119.59	26.88x	1e-4

Impact of Improvements within LAVA on X-59 QueSST

- The re-factored solver was run on an early version of the X-59 (the c608) and showed a speed-up of ~3-5x as part of the SBPW3
  - The production level runs using the latest X-59 geometry will be between the medium and fine grid levels for the workshop geometry
  - Have demonstrated the ability to recover nearly-identical solutions to the original version of the solver while getting the solution in 1.5 hours rather than 5.5 hours on a similar number of cores

Case	Grid Level	Old SBU	Re-factored SBU	Improvement
C608	Coarse	18.15	6.36	2.85x
C608	Medium	57.8	18.3	3.16x
C608	Fine	274.0	59.4	4.60x

	Old # CPUs	New #CPUs	Old Wall Time	New Wall Time
Medium	720	400	2.4	1.4
Fine	1500	1200	5.5	1.5



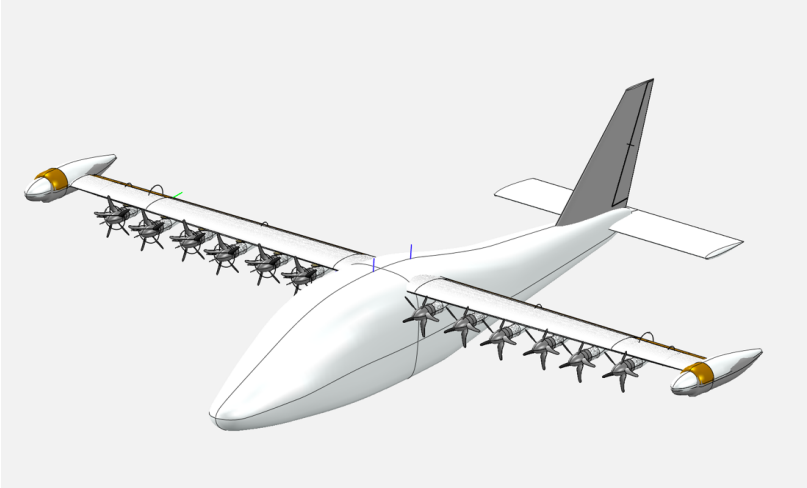
C608 Flow Visualization

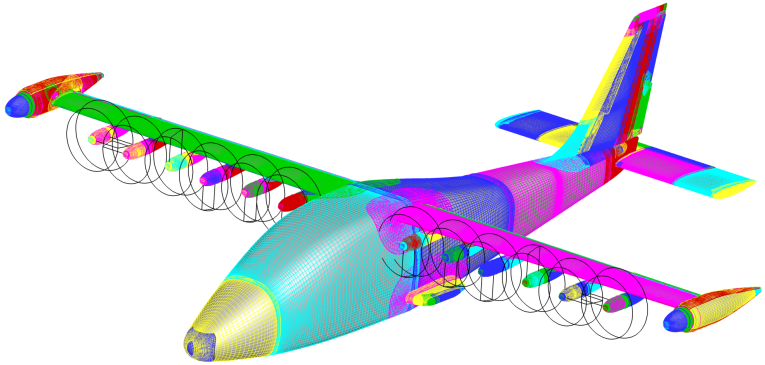
Outline

- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

Applying Refactored LAVA to X-57

- Best-practices maintained while simulating X-57 with the newly refactored LAVA Curvilinear
  - Identical grid systems
  - Consistent numerical scheme and turbulence modeling
  - Consistent propulsion modeling techniques
- Initial simulations performed to ensure consistency between the original and refactored LAVA Curvilinear solvers
- Tools for visualizing convergence history and computing integrated quantities for each database case were developed for efficient solution post-processing

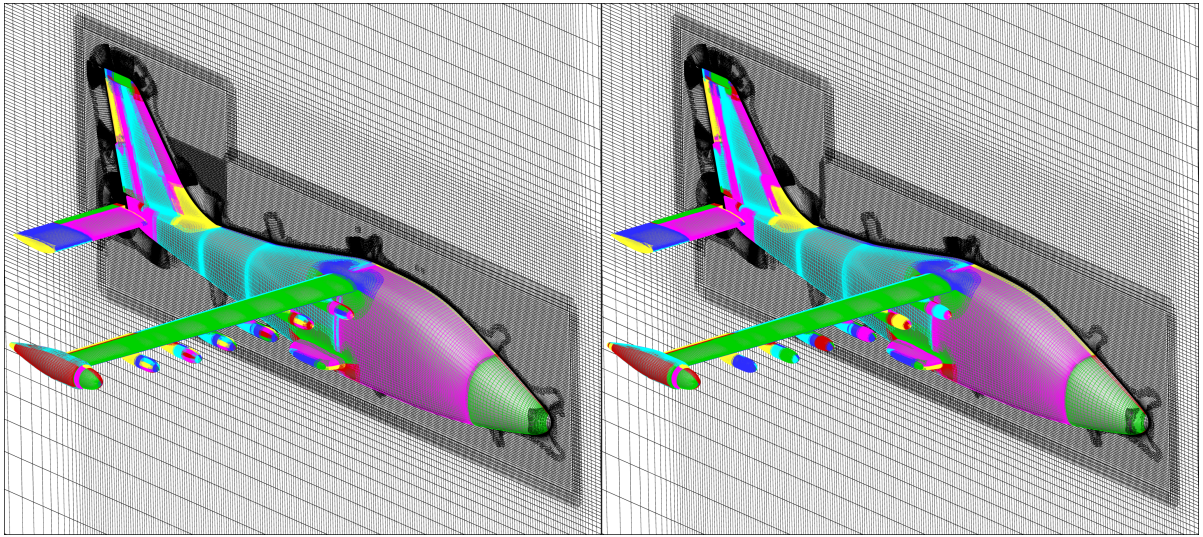




Improved Grid Connectivity Routine

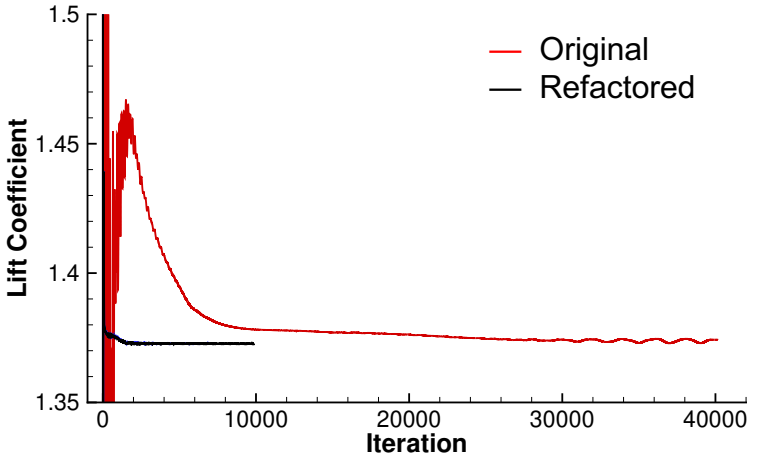
- Important step to optimize communication between overlapping mesh domains
- The implicit “hole-cutting” connectivity routine has been parallelized for more efficient pre-processing
  - Original Connectivity: ~6 hours
  - Refactored Connectivity: ~1.5 mins
- Explicit hole-cutting automation has also been implemented and will soon eliminate need for “X-ray” creation
- After initial mesh generation, refactored LAVA tool suite reduces pre-processing time by multiple orders of magnitude

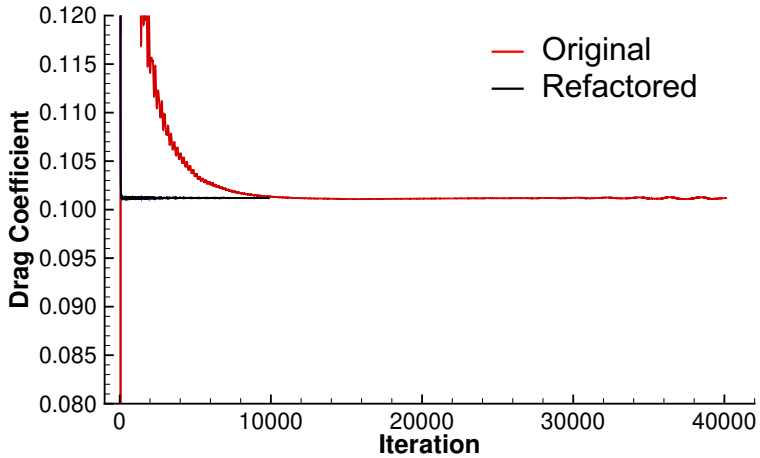
Original Connectivity ***Refactored Connectivity***



Power-Off Integrated Load Comparisons

- Initial simulations for power-off conditions run to verify both the *original* LAVA and *refactored* LAVA produce consistent results
- Much smaller standard deviations are observed in loads using the refactored solver versus the original solver
- Loads are consistent to the 3rd significant digit, while pitch moment is consistent to the 1st (2nd if rounded)



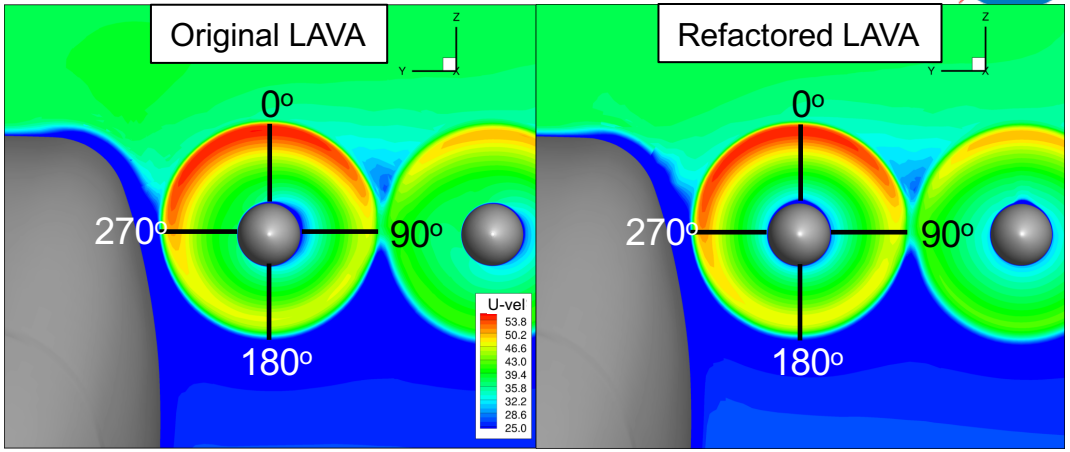


Version	Lift	Drag	Pitch
Original	1.37382977	0.10117798	-0.19443699
Refactored	1.37268912	0.10120113	-0.18943385

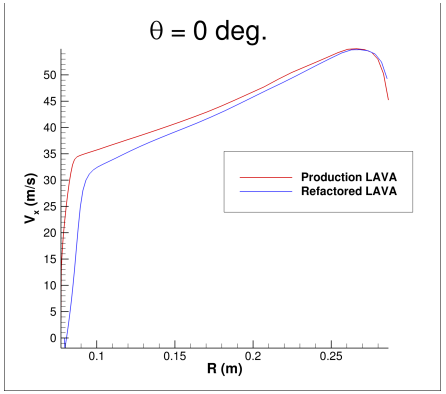
Condition: Altitude: 2,500 ft, Mach = 0.149, Re_{MAC} = 2.11 M, α = 4.0°, Flaps = 10°

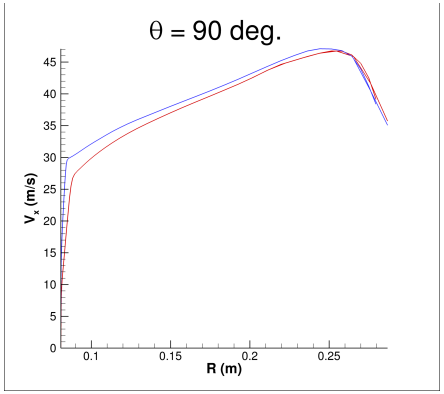
Actuator Zone Source Effect Comparison

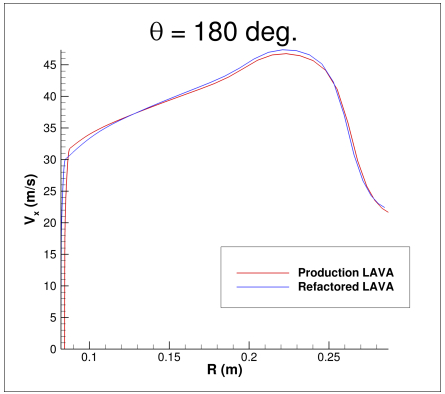
- Initial simulations with high-lift power performed to compare actuator zone effects
- Axial velocity line extractions taken immediately aft of the actuator zone region where the source term is applied

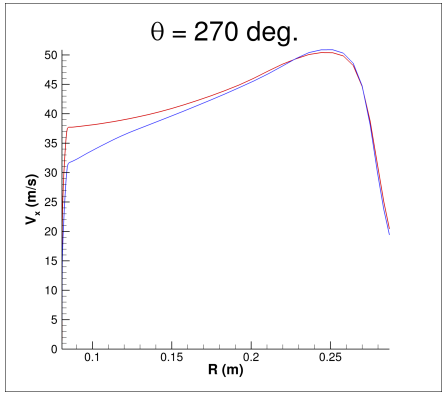


Condition: Altitude = 2,500 ft, Mach = 0.092, Re_{MAC} = 1.31 M, α = 4.0°, Thrust = 49.3 lbf, Torque = 16.0 lbf-ft



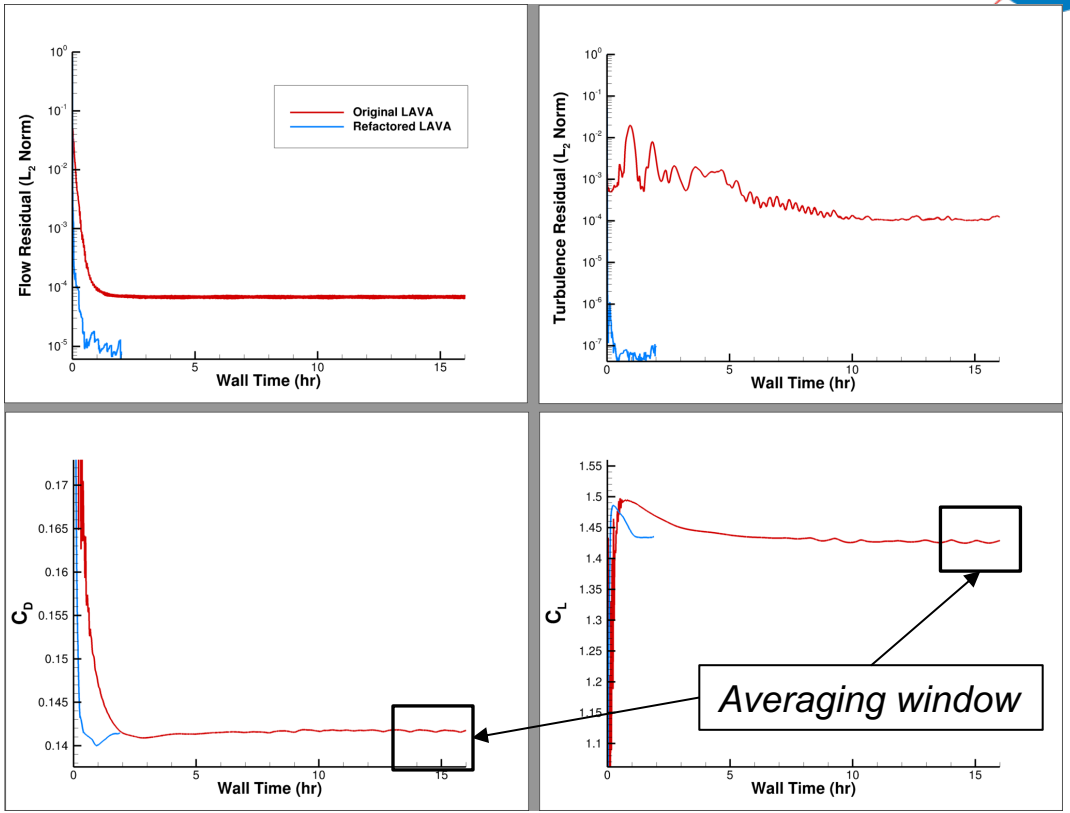






Power-On Convergence Comparisons

- Convergence of the flow equations and turbulence model equation is necessary for adequate load convergence
- Deeper convergence with the refactored code eliminates load "averaging" method for poorly converged cases
- Faster and deeper convergence results in loads consistent for both codes to 3 significant digits at a fraction of the time



Condition: Altitude = 2,500 ft, Mach = 0.119, Re_{MAC} = 1.69 M, α = -2.0°, β = 0°
Thrust = 22.0 lbf, Torque = 8.7 lbf-ft

Resource Usage Comparison

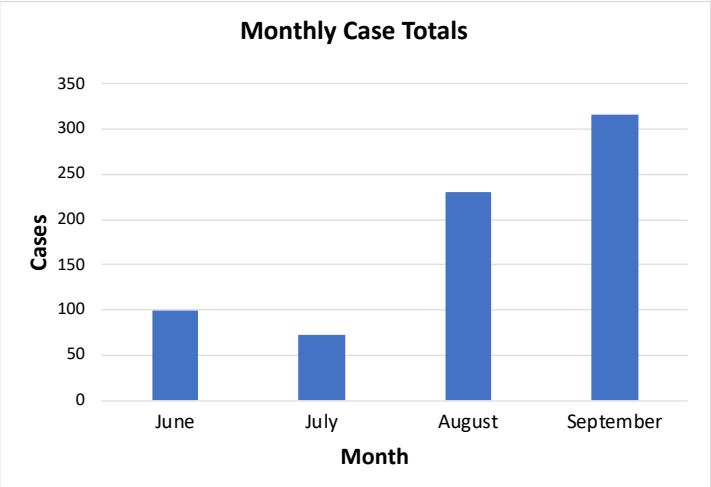
- A combined reduction in required CPUs and wall-time to perform the computation has been experienced using the refactored LAVA solver
- Flow condition and aircraft configuration both affect the timing required for steady convergence
- Wait time on Pleiades supercomputer is also substantially reduced, allowing many more cases to run per month

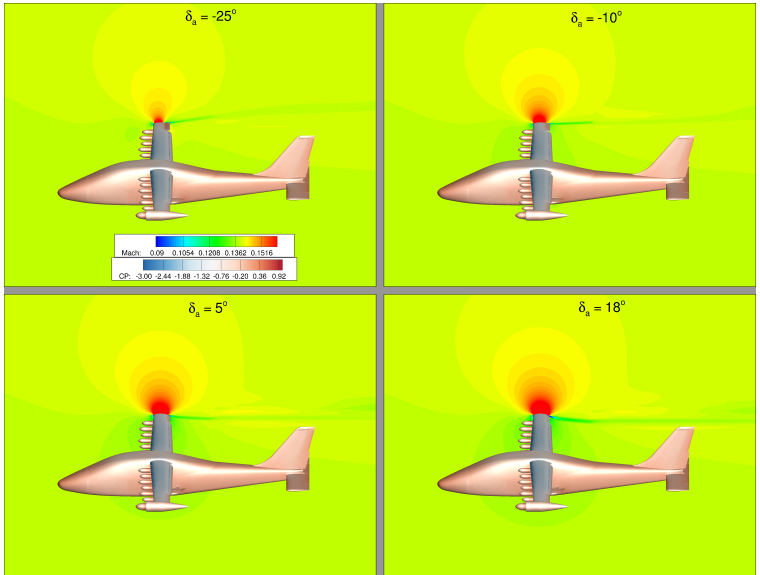
Examples of resource usage characteristic of X-57 database simulations

	CPU Usage (Ivy Bridge)		***Wall Time***		***Computational Savings***
Case Description	***Refactored***	***Original***	***Refactored***	***Original***	
75 KCAS, Flaps 30°, α = 8.0°, δ_a = -22°	800	1200	2 hrs	16 hrs	12X
75 KCAS, Flaps 30°, α = -2.0°, β = 0°	800	1200	2 hrs	16 hrs	12X
75 KCAS, Flaps 30°, α = 8.0°, β = 15°	800	1200	2 hrs	16 hrs	12X
58 KCAS, Flaps 30°, α = 2.0°, β = 20°	800	1200	4 hrs	24 hrs	9X

X-57 Steady RANS Database Summary

- Over 3,000 steady RANS simulations have been run in support of X-57 database generation
- Initial validation and verification throughout the database generation was crucial for ensuring solution accuracy
- Code-to-code comparison showed <5% differences in integrated quantities for pre-stall cases, with <10% observed for post-stall
- The refactored LAVA solver was applied to X-57 database generation, providing:
  - An order of magnitude (~9-12X) reduction in computational cost
  - Deeper steady-state convergence for all cases
  - Substantial increase in cases run per month





Outline

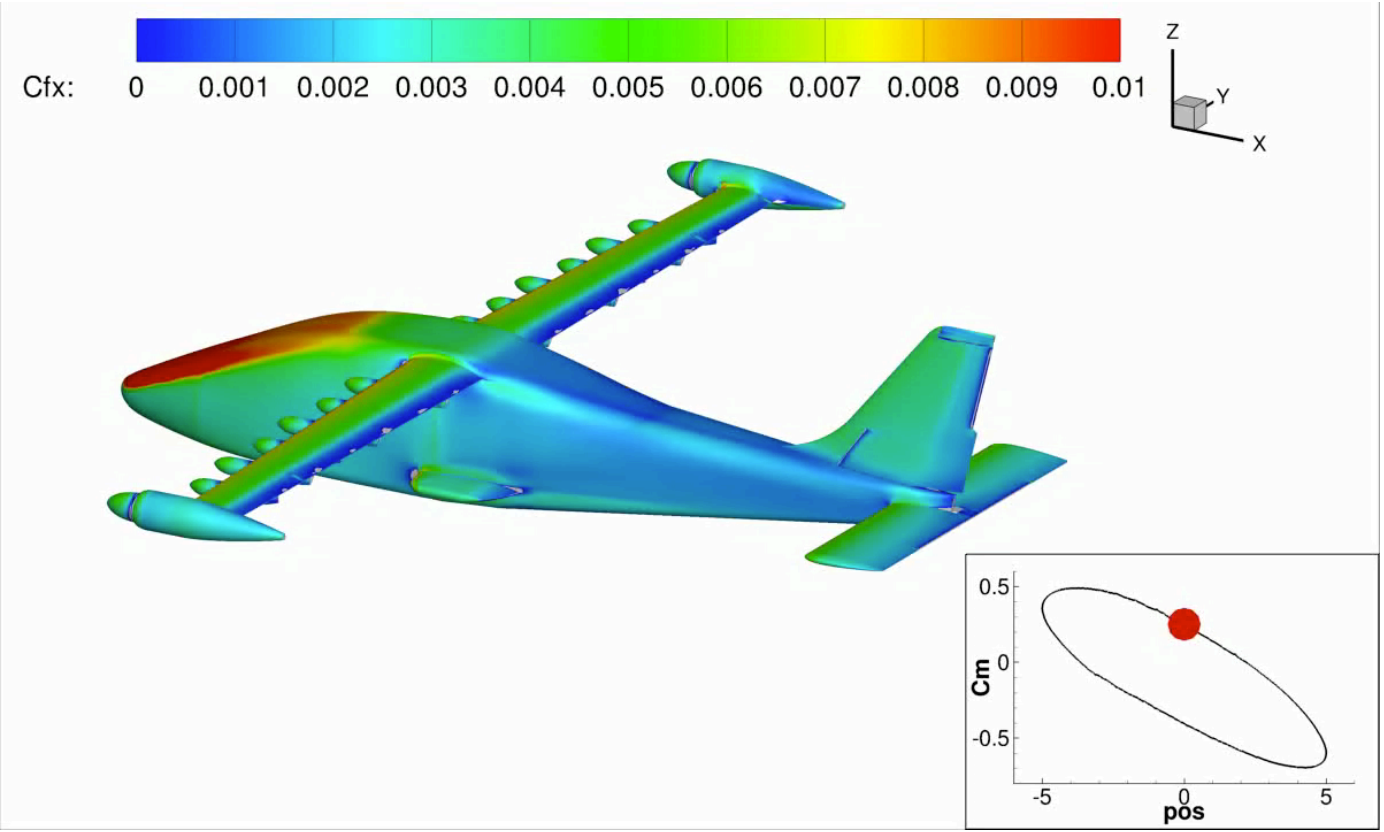
- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

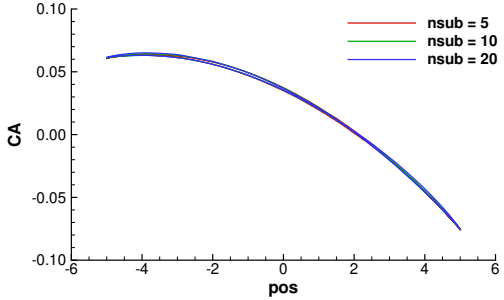
X-57 Unsteady Rate Derivative Analysis

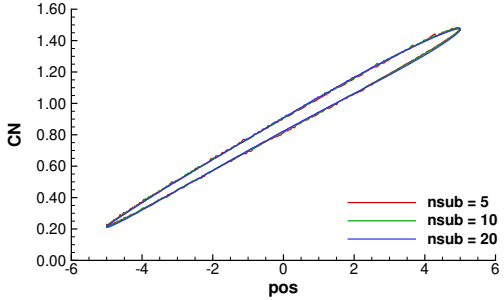
- URANS simulations of forced pitch, roll, and yaw oscillations were performed for the un-powered X-57 to construct a model for estimating damping coefficients
  - Second-order accurate in space and time
  - Time varying Geometric Conservation Law (GCL) preserving metric terms
  - SARC-QCR2000 turbulence model
  - Used best-practice mesh from steady-state grid convergence study
- Sensitivity studies were performed and code-to-code comparisons as well as WT comparisons were made
  - Sub-iteration convergence
  - Time-step (cycles per forcing frequency)

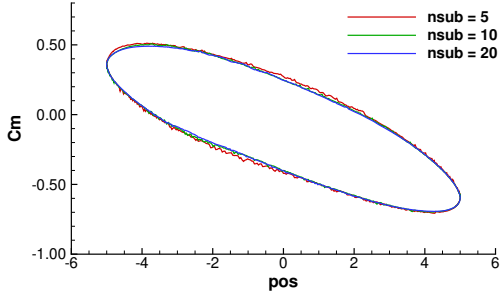
X-57 Unsteady Rate Derivative Analysis: Sub-iteration Sensitivity

Forced Pitch: Amplitude = 5°, Frequency = 0.82, 800 steps per cycle



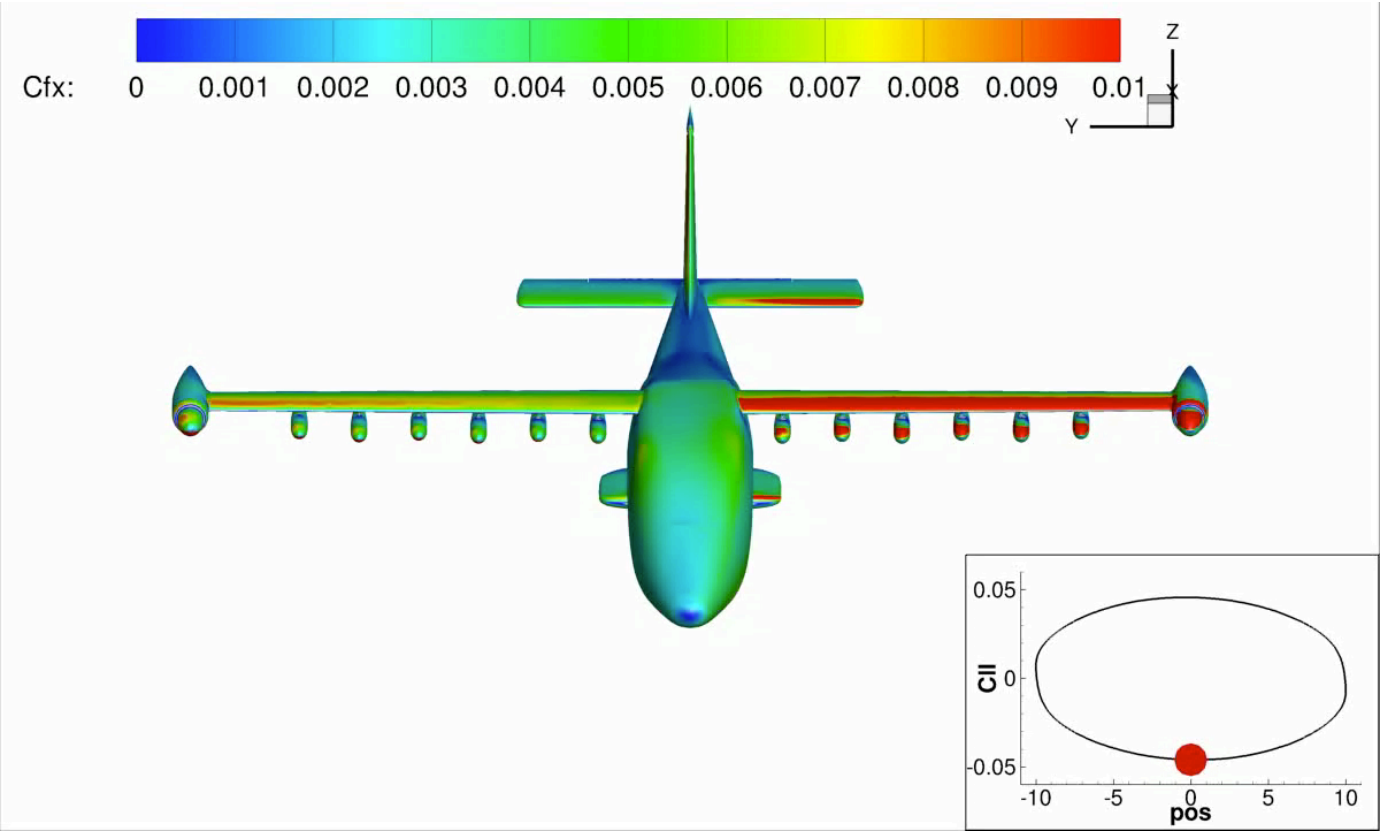


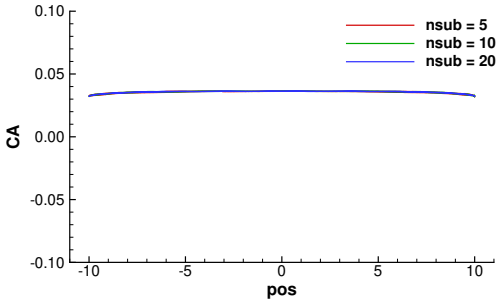


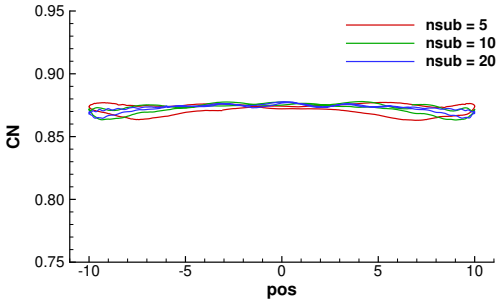


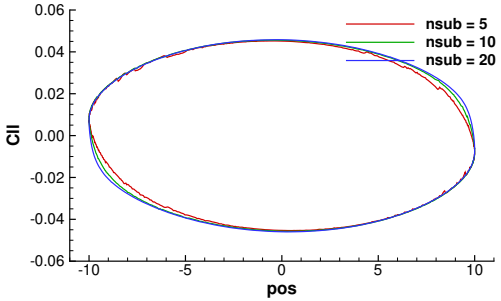
X-57 Unsteady Rate Derivative Analysis: Sub-iteration Sensitivity

Forced Roll: Amplitude = 10°, Frequency = 0.729, 800 steps per cycle



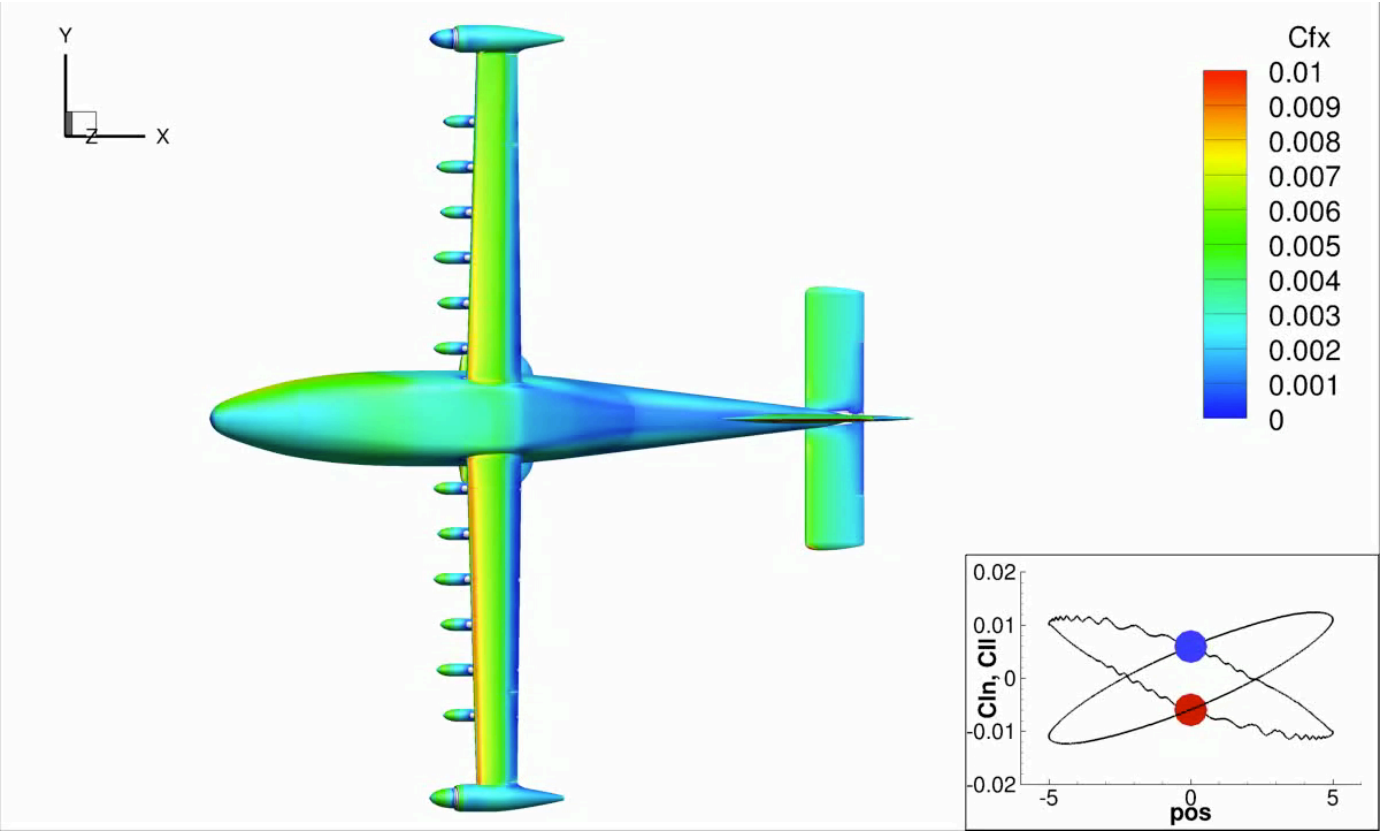


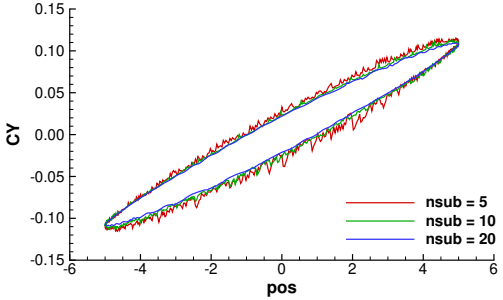


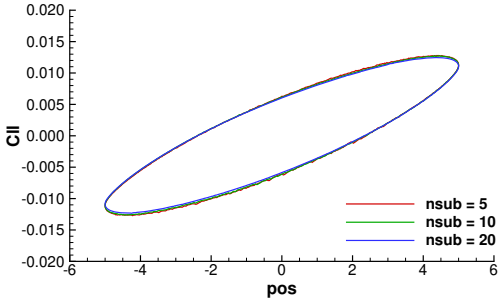


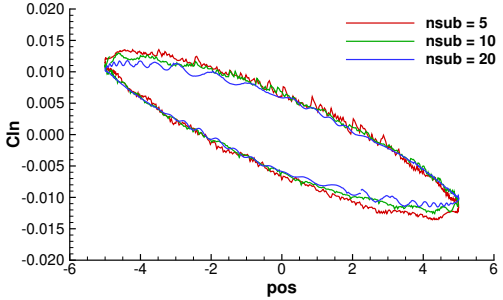
X-57 Unsteady Rate Derivative Analysis: Sub-iteration Sensitivity

Forced Yaw: Amplitude = 5°, Frequency = 0.365, 800 steps per cycle



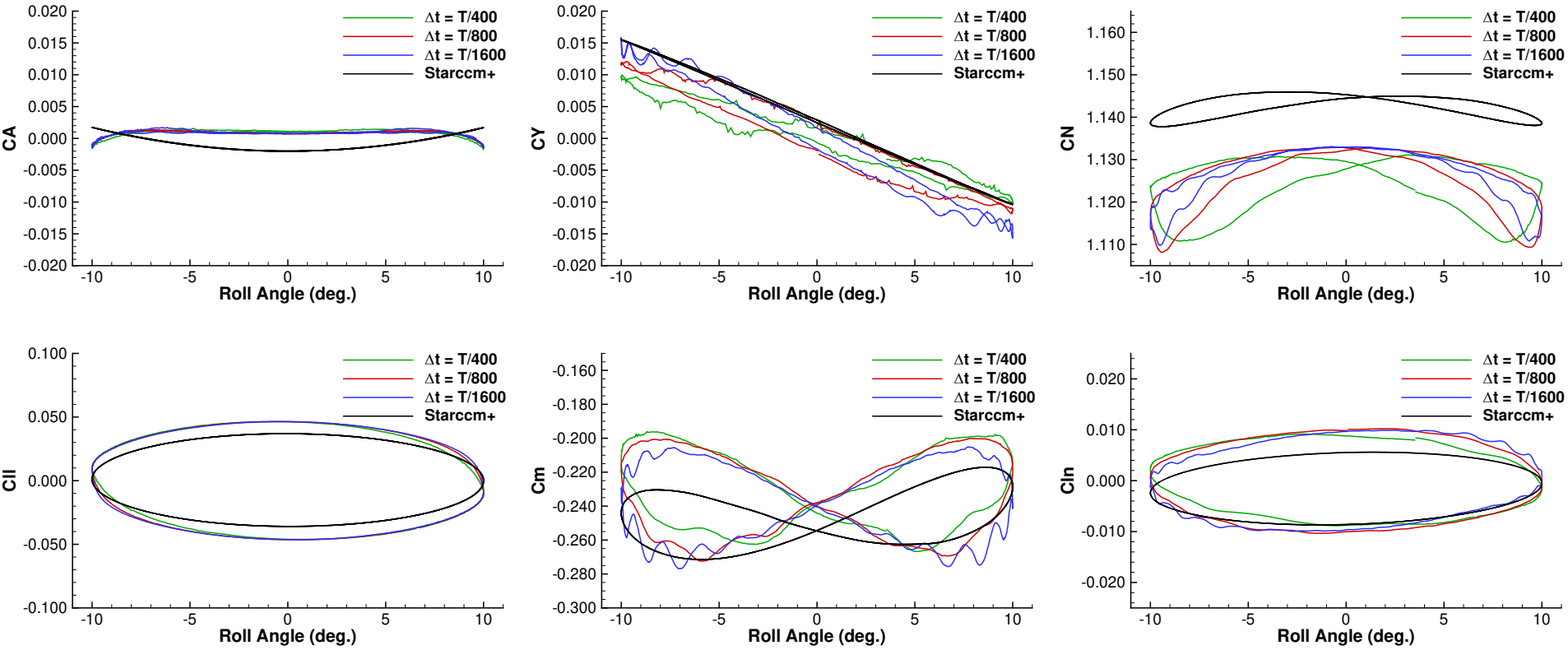






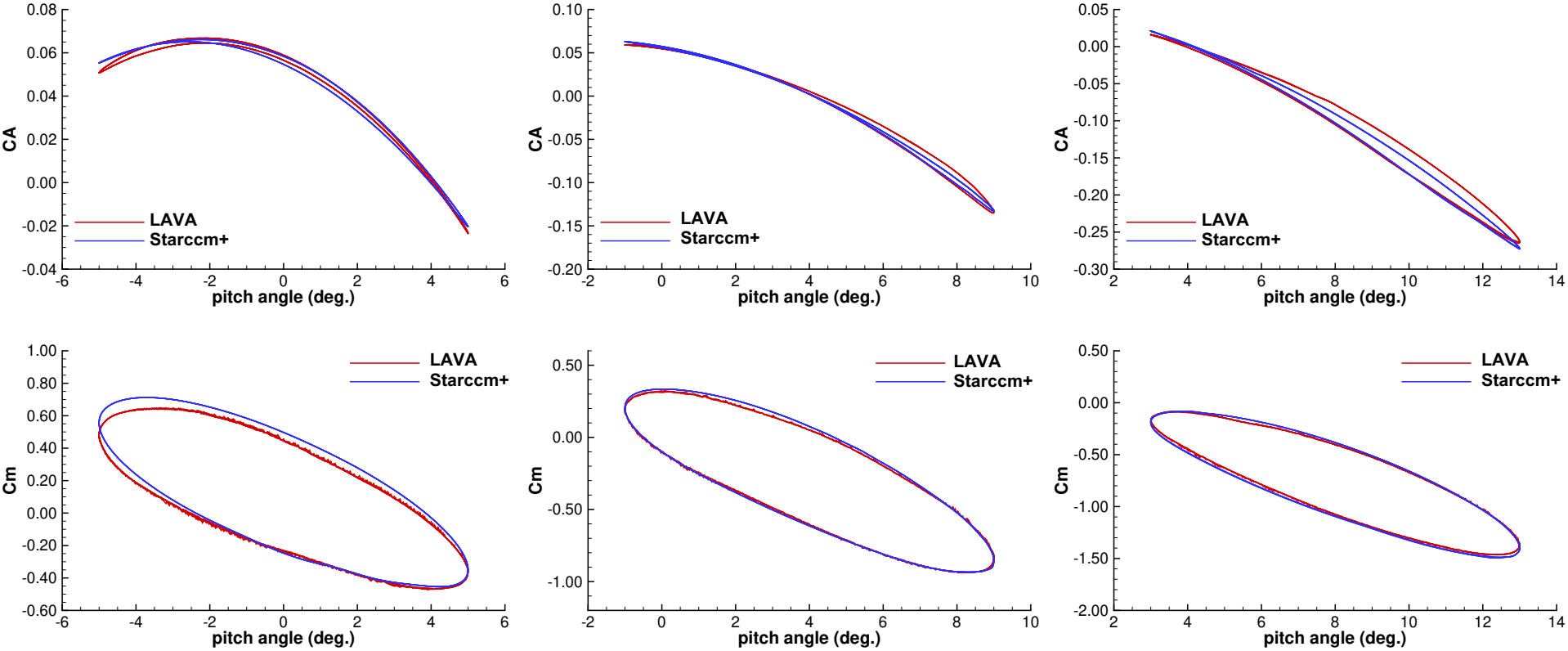
X-57 Unsteady Rate Derivative Analysis: Time Step Sensitivity

Forced Roll: Amplitude = 10°, Frequency = 0.729, AOA = 4°, nsub = 10



X-57 Unsteady Rate Derivative Analysis: Code-to-Code Comparison

Forced Pitch: Amplitude = 5°, Frequency = 0.82



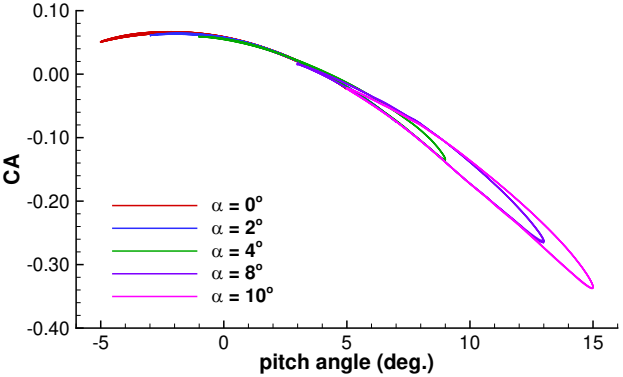
AOA = 0°

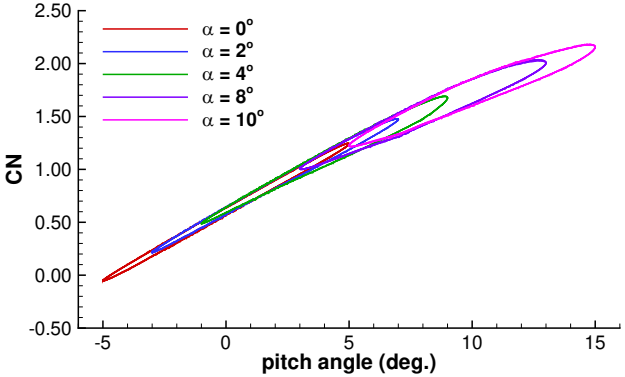
AOA = 4°

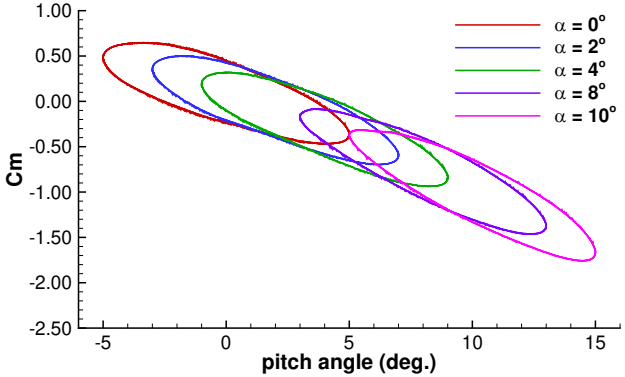
AOA = 8°

X-57 Unsteady Rate Derivative Analysis: AoA Sensitivity

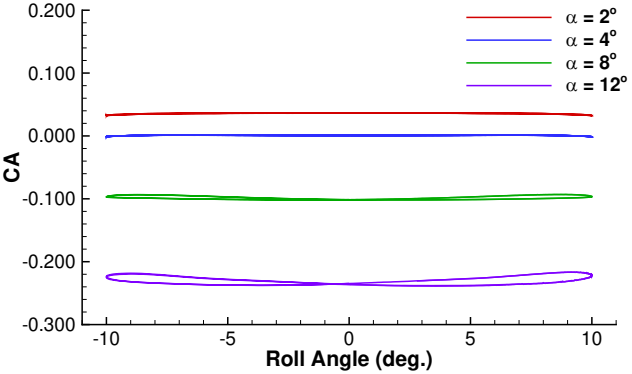
Forced Pitch: Amplitude = 5°, Frequency = 0.82

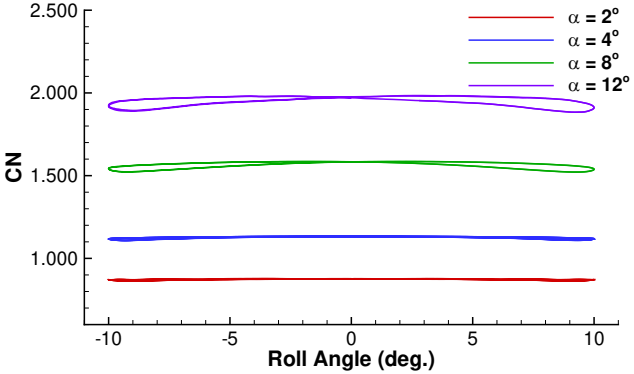


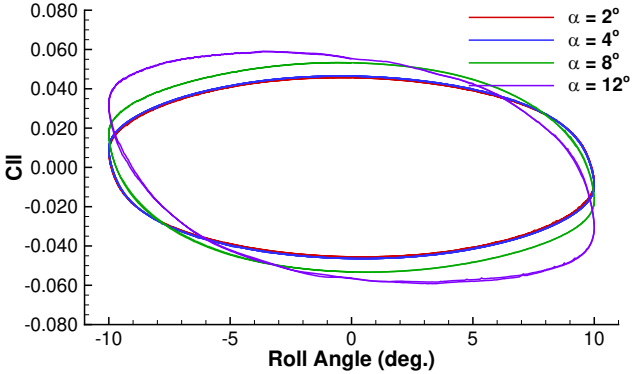




Forced Roll: Amplitude = 10°, Frequency = 0.729







X-57 Unsteady Rate Derivative Analysis

- 27 total URANS simulations of forced pitch, roll, and yaw oscillations were performed for the un-powered X-57 to construct a model for estimating damping coefficients
- Sensitivity studies were completed and code-to-code comparisons were made to ensure proper simulation methods are used
  - Sub-iteration convergence
  - Time-step (cycles per forcing frequency)
- Required wall-time for 6 periods using 800 time-steps per cycle
  - 0.75 days (nsub = 5), 1.5 days (nsub = 10), 3 days (nsub = 20)
- Results are currently being further post-processed to compute rate derivatives used in the final X-57 aerodynamic model

Outline

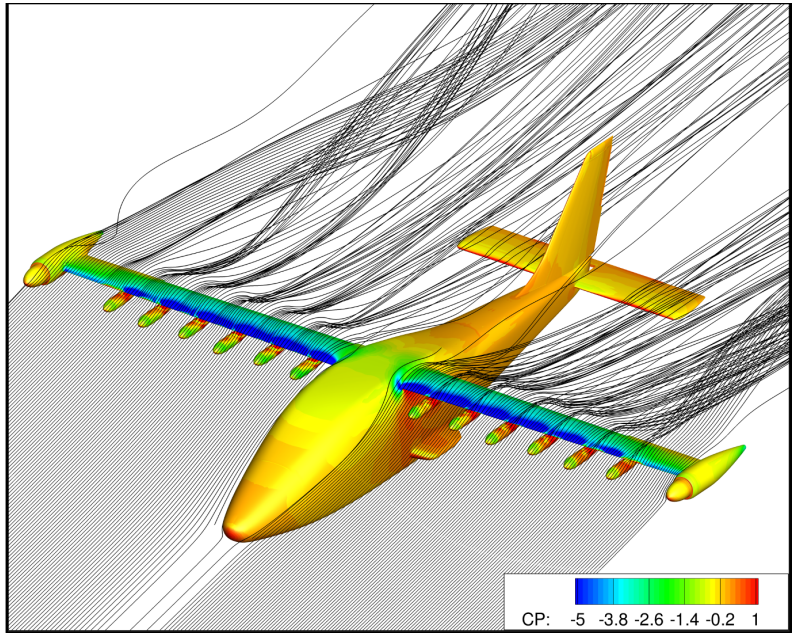
- Part I: Steady RANS X-57 Database Generation
  - Project Overview
  - CFD Verification and Validation
  - Database Results
  - Refactorization of the Launch Ascent and Vehicle Aerodynamics (LAVA) solver
  - Applying Refactored LAVA to X-57 Databases
- Part II: Unsteady RANS X-57 Database Generation
  - CFD Simulation Methodology
  - Database Results
- Summary

Summary

- Over 3,000 steady and unsteady RANS CFD simulations have been run in support of X-57 aerodynamic database generation
- Consistent predictions between the LAVA Curvilinear and Star-CCM+ solvers observed
- Order of magnitude reduction in computational resources required for X-57 database cases
  - LAVA Curvilinear Code Optimization and Refactoring
    - Demonstrated up to ~9-12x reduction in computing resources due to both deeper convergence per nonlinear iteration and faster time per iteration
    - Improved time to solution for a production level case from 16-24 hours to 2-4 hours
    - Wait time to run on Pleiades supercomputer drastically reduced due to lower resource requests
- *Large-scale database generation originally requiring a year can potentially be reduced to a few weeks*

Acknowledgments

- This project is funded by the Scalable Convergent Electric Propulsion Technology and Operations Research (SCEPTOR) program under the NASA Aeronautics Research Mission Directorate (ARMD)
- Refactored LAVA effort has been partially supported by Transformational Tools and Technologies Revolutionary Computational Aerosciences (T^3 RCA), Commercial Supersonic Technology (CST), and Advanced Air Transport Technology (AATT) projects
- Experimental data provided by David Cox at NASA Langley Research Center
- Computer resources provided by NASA Advanced Supercomputing (NAS) Pleiades facility



Questions?