Recent Applications and Algorithm Development Using FUN3D

Eric J. Nielsen
Computational AeroSciences Branch
NASA Langley Research Center

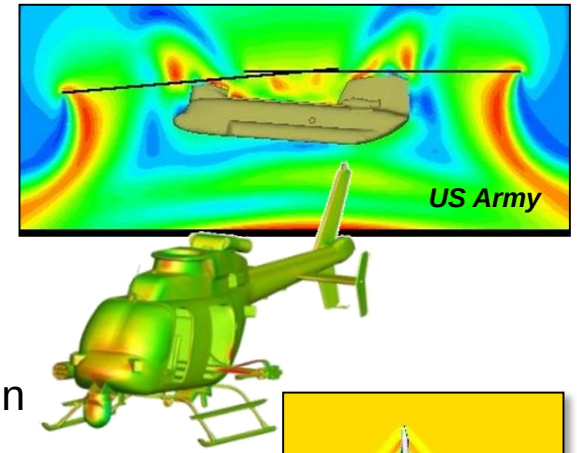
VT AOE ’94, ‘98

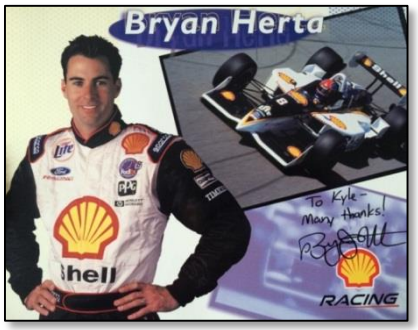
Langley Research Center

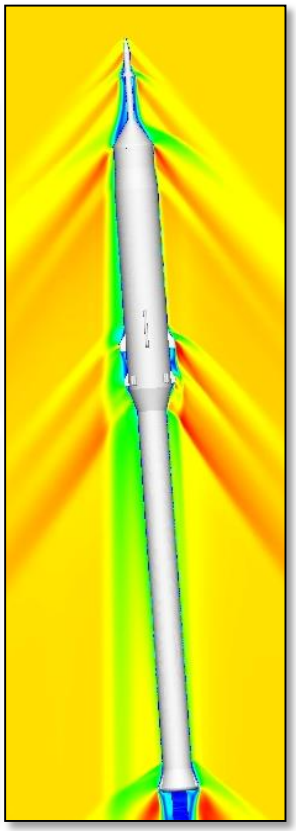
FUN3D Core Capabilities

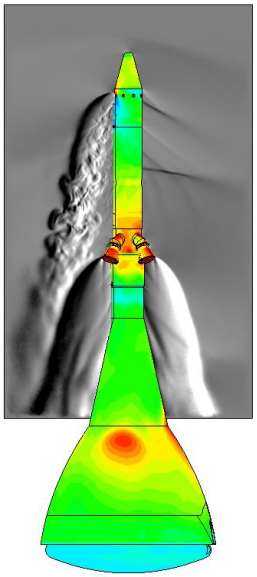
http://fun3d.larc.nasa.gov

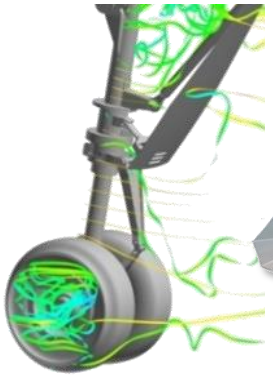
- Established as a research code in late 1980's; now supports numerous internal and external efforts across the speed range
- Solves 2D/3D steady and unsteady Euler and RANS equations on node-based mixed element grids for compressible and incompressible flows
- General dynamic mesh capability: any combination of rigid / overset / morphing grids, including 6-DOF effects
- Aeroelastic modeling using mode shapes, full FEM, CC, etc.
- Constrained / multipoint adjoint-based design and mesh adaptation
- Distributed development team using agile/extreme software practices including 24/7 regression, performance testing
- Capabilities fully integrated, online documentation, training videos, tutorials

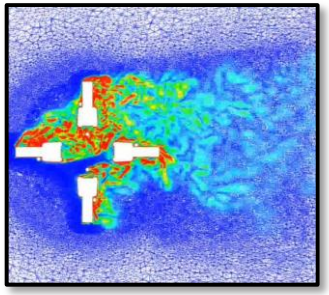


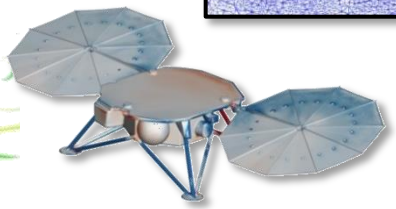


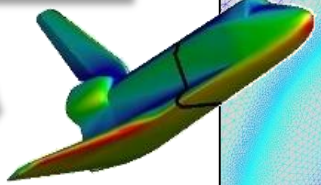


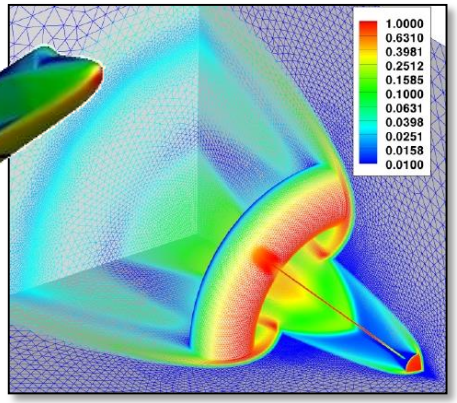




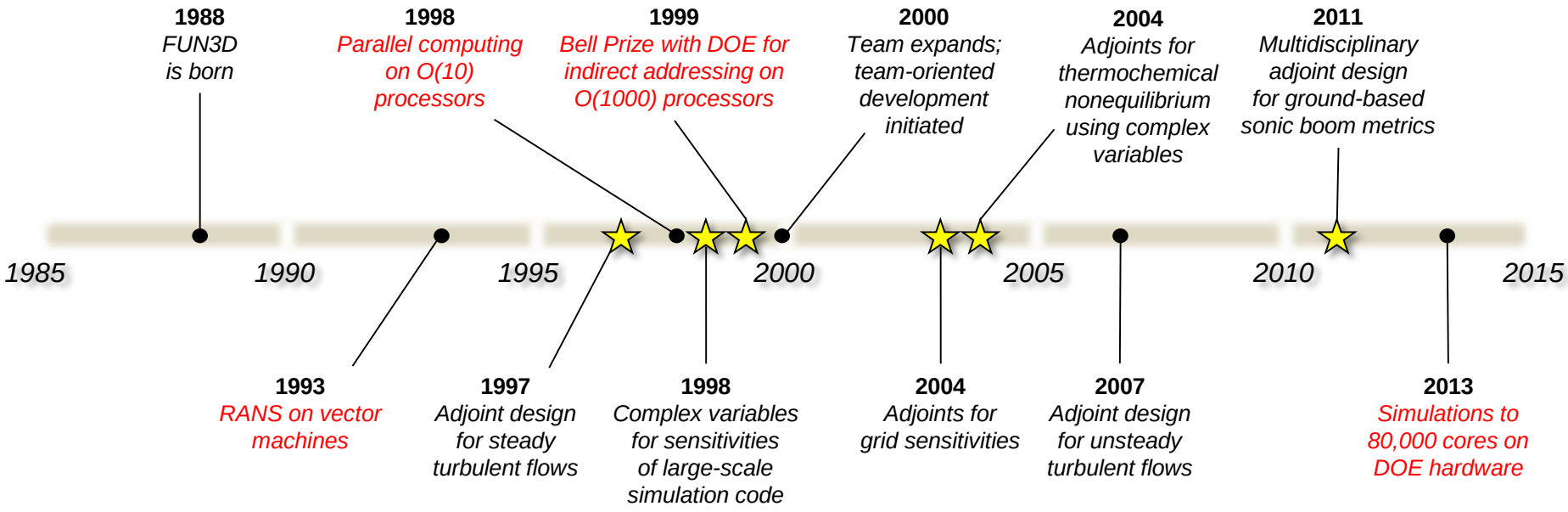








Towards Physics-Based Design



HPC-related milestones in red

★ Internationally-unique capability

FUN3D User Base

- Widely used across multiple industries
  - Aerospace, automotive, computing…
- Other government agencies
  - DoD, DoE, FAA…
- Also individuals, small businesses, research groups
- Has been used by hundreds of universities
  - Over 60 students have interned with the team or conducted thesis work using FUN3D: 20 MS and PhD theses generated
  - Graduates with hands-on FUN3D experience are highly sought after

Team-Oriented Development

Team-oriented strategy is critical

- Broad expertise necessary for broad success
- High *truck* number

Pair programming boosts *truck* number, improves code quality

Weekly scrums

- Very quick summary of “Did, Do, In-the-Way”
- Scrum master notes impediments
- No “Death by PowerPoint”
- Management may attend but cannot talk

Automated testing of repository commits

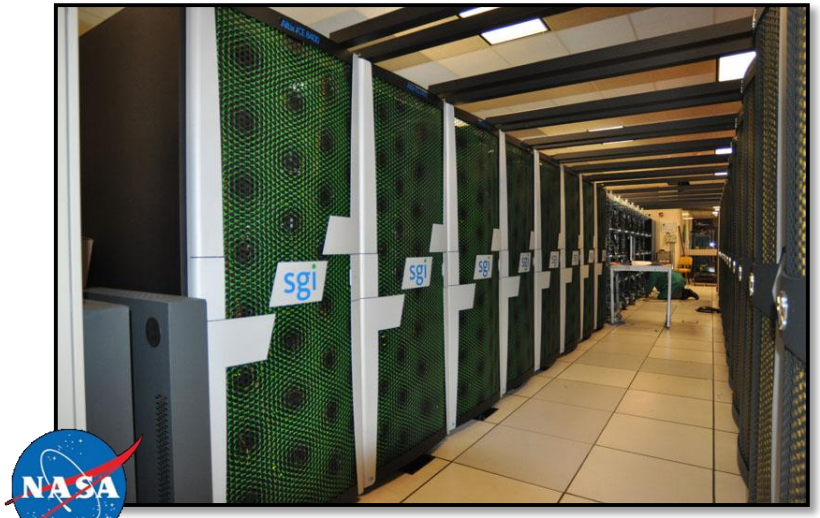
- Every commit screened for adherence to coding standard
- Each commit triggers a series of regression tests
- Successful gauntlet means FUN3D is safe to ship

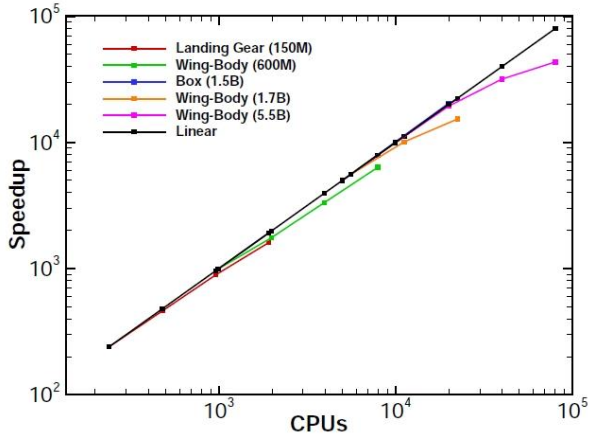
Jennifer Abras
Natalia Alexandrov
Kyle Anderson
Bala Balakumar
Bob Bartels
Karen Bibb
Bob Biedron
Alejandro Campos
Jan-Renee Carlson
Mark Carpenter
Joe Derlaga
Boris Diskin
Austen Duffy
Peter Gnoffo
Dana Hammond
Clara Helm
Bill Jones
Bil Kleb
Beth Lee-Rausch
Eric Lynch
Steve Massey
Eric Nielsen
Hiro Nishikawa
Dave O’Brien
Mike Park
Sriram Rallabhandi
Shatra Reehal
Chris Rumsey
Rajiv Shenoy
Marilyn Smith
Jim Thomas
Veer Vatsa
David Venditti
Jeff White
Bill Wood
Kan Yang

FUN3D and High Performance Computing

FUN3D is used on a broad range of HPC installations around the country

OLCF
Oak Ridge Leadership Computing Facility





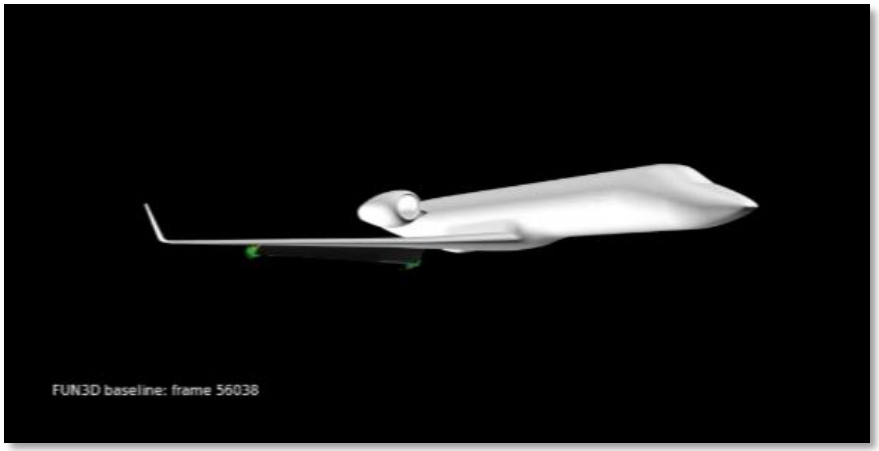
Scaled to 80,000 cores on DoE's Cray XK7 'Titan' using grids containing billions of elements

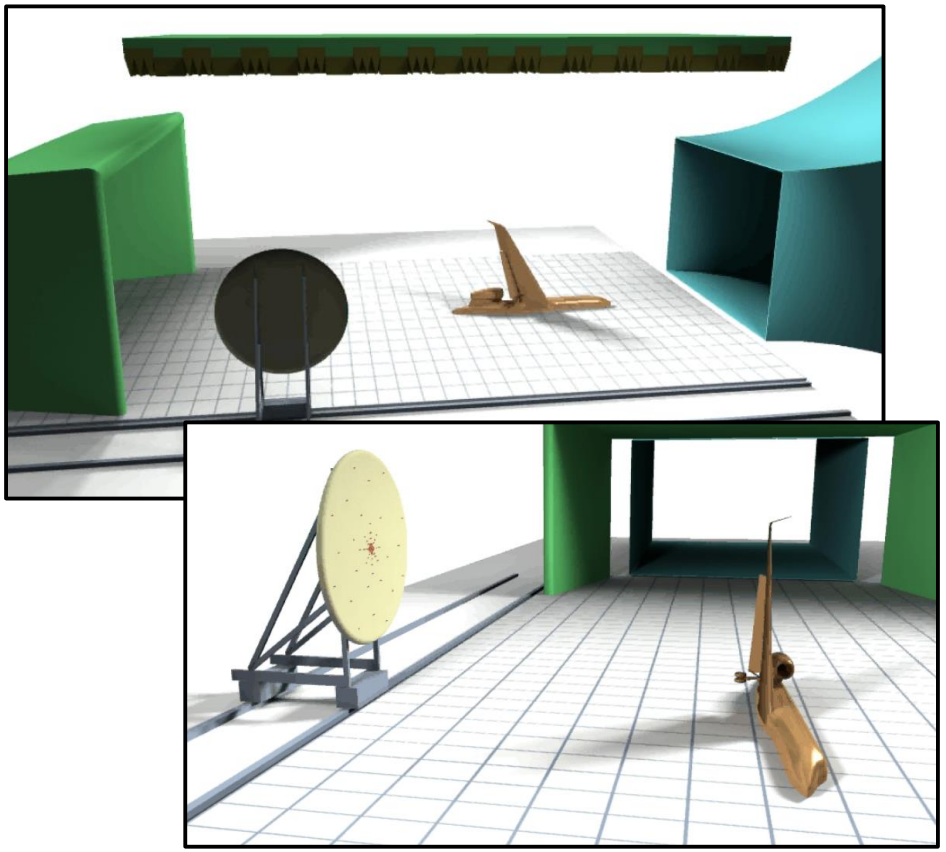
Awarded the prestigious Gordon Bell Prize in a past collaboration with Argonne National Lab

Motivation for HPC

Airframe Noise Challenge Problem

Images Courtesy NASA/Gulfstream Partnership Effort on Airframe Noise Research





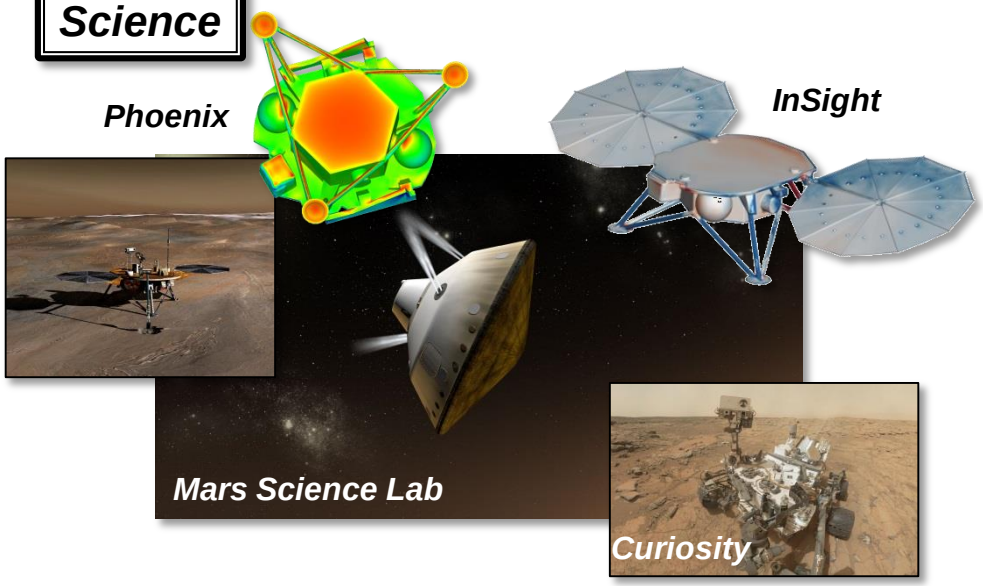
Semi-span Aircraft in High-Lift Configuration

- Current simulations use ~1B elements
- Computation takes ~4 months on 3000 cores; represents ~0.3 seconds of real time
- Would like to run at least 100x larger spatial mesh, much finer temporal mesh, with much longer physical durations (exp avgs over ~15 seconds)
- Single solution would require 6,750x more CPU time (2,250 years wall-clock) in current environment
- Numerous solutions ultimately needed for parametric studies, open/closed tunnel configs, etc.

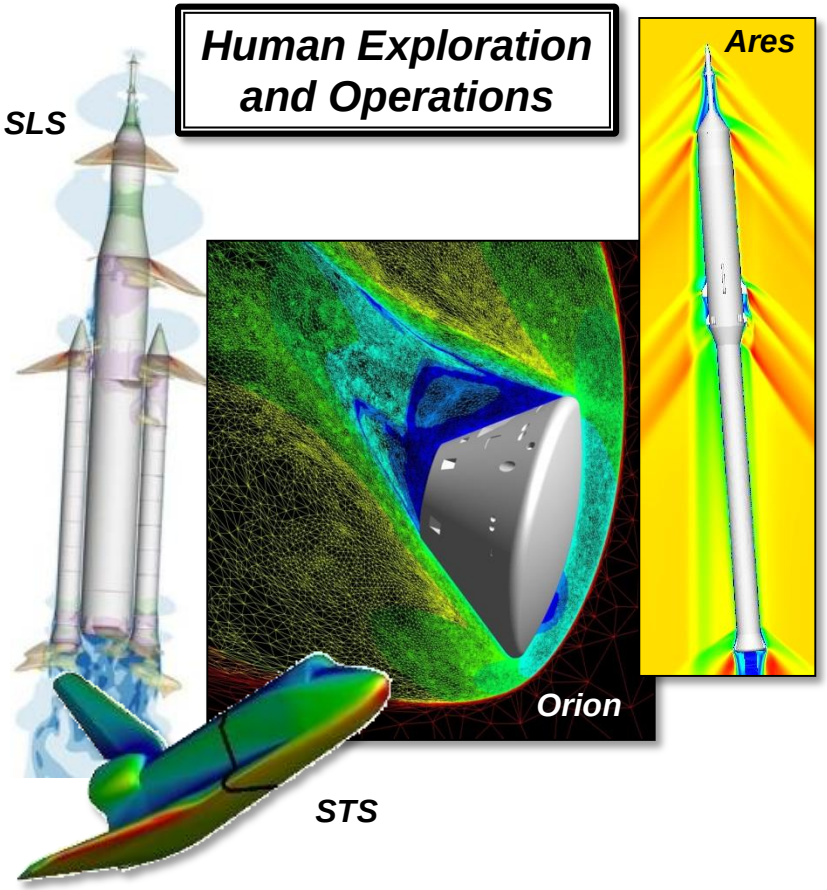
Across NASA Missions

FUN3D supplies critical physics-based aerodynamics for a broad range of applications across all Mission Directorates

Science



Human Exploration and Operations



"The FUN3D team has developed a capability that continues to find new and unique applications of significant importance to the agency."

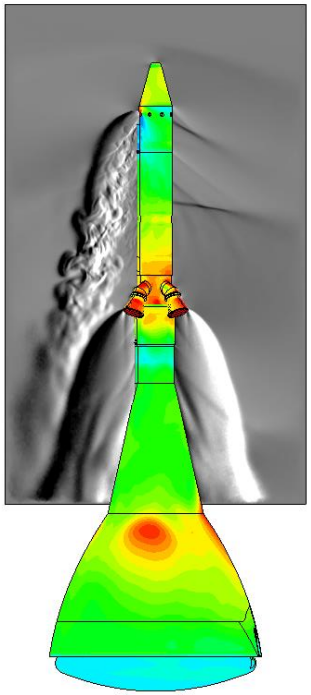
- Dave Schuster
NASA Technical Fellow for Aerosciences
NASA Engineering and Safety Center

Langley Research Center

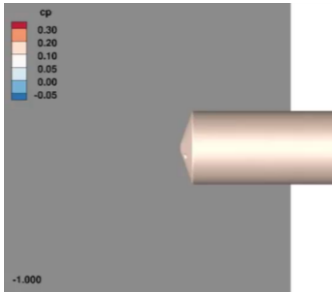
Across NASA Missions

Space Technology

Launch Abort

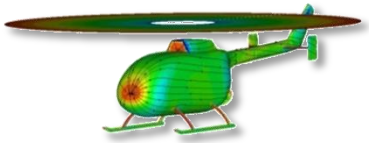


Supersonic Retropropulsion

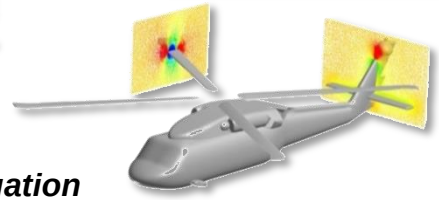


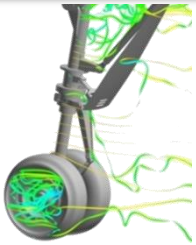
IRVE

Aeronautics Research

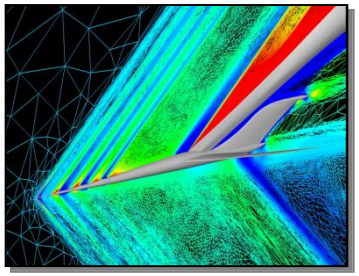


Rotorcraft

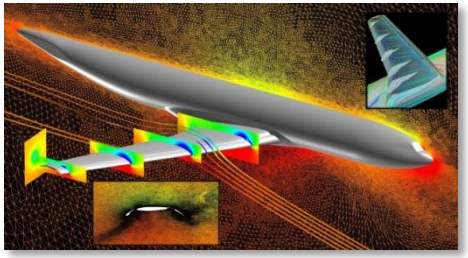


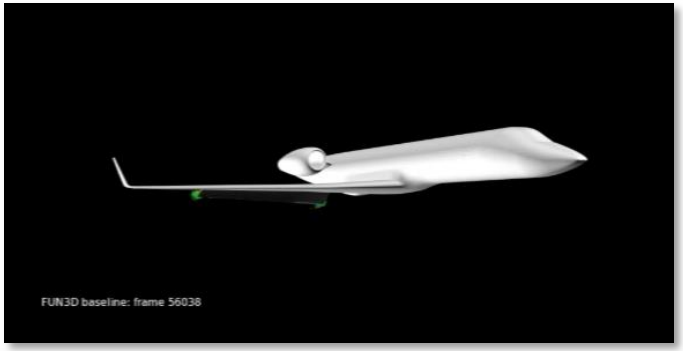


Sonic Boom Mitigation



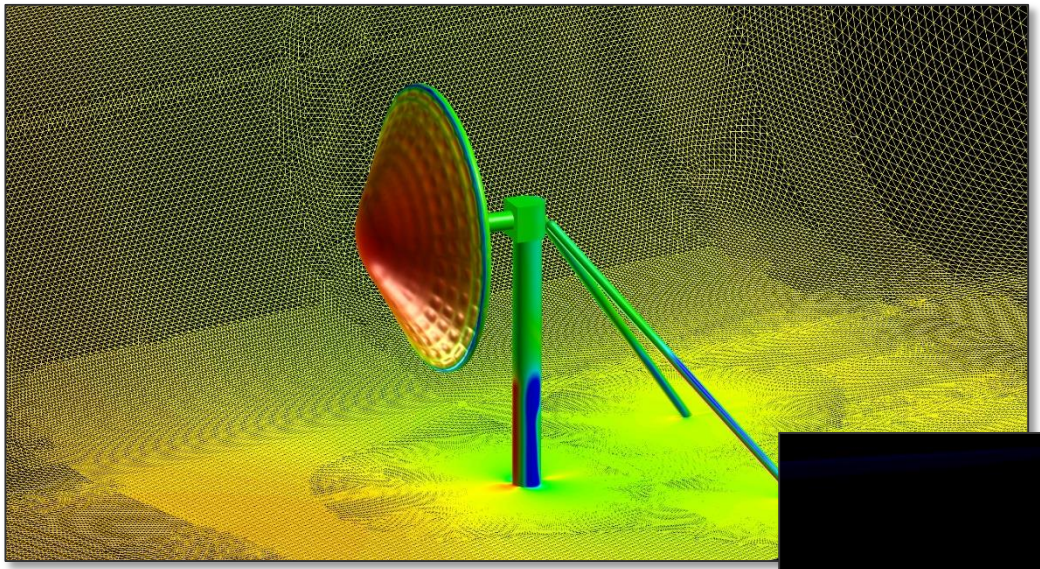
Airframe Noise





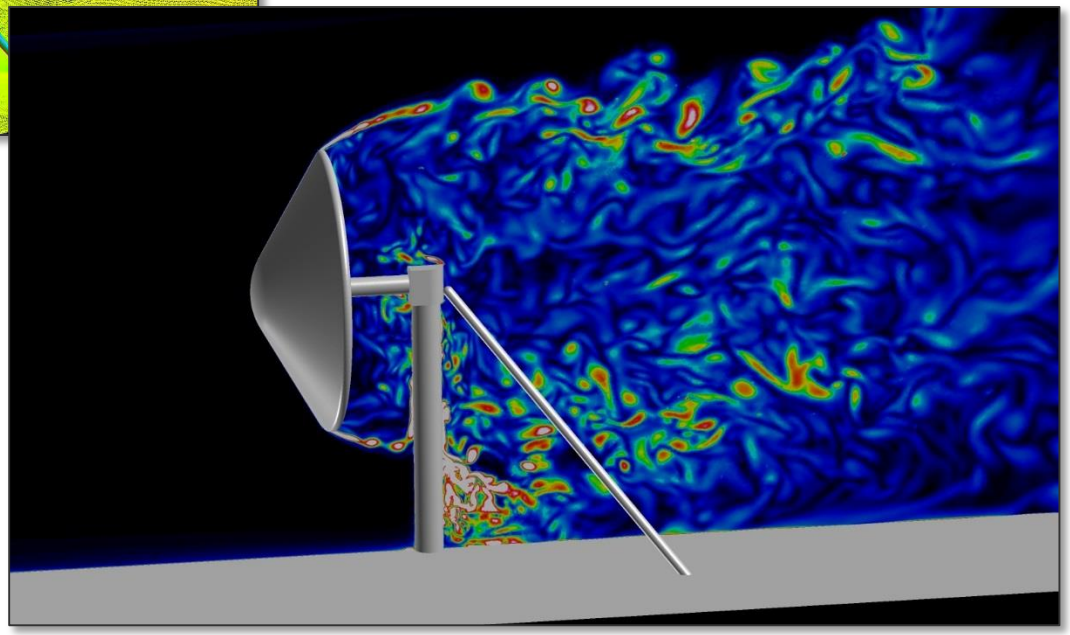
Due to increased demand from NASA scientists and engineers, FUN3D simulations now account for the single largest block of supercomputing cycles at the Agency: 12% of NAS, or approximately 200 million hours per year

Hypersonic Inflatable Aerodynamic Decelerators

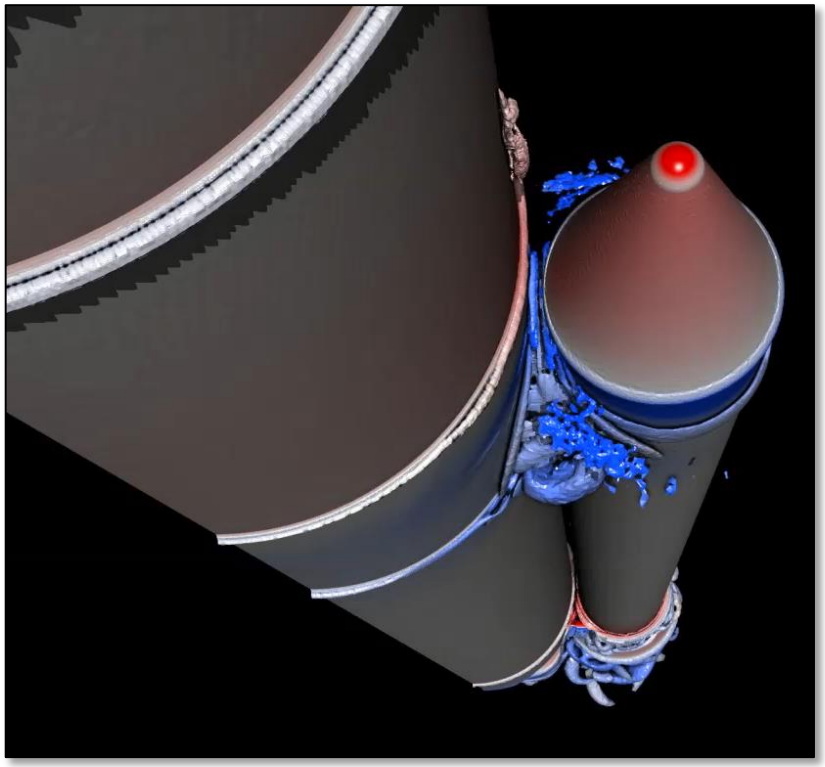


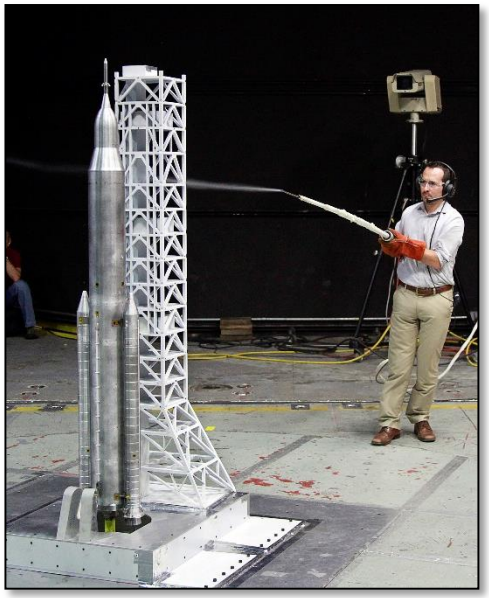
HIAD devices offer a lightweight option for decelerating very large masses from hypersonic speeds

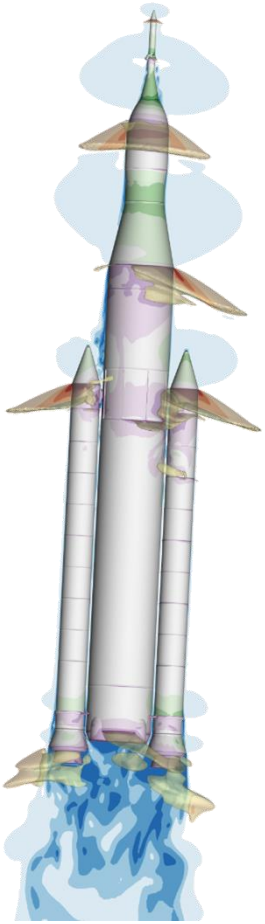
Aeroelastic computations are used to predict performance characteristics



Designing the Space Launch System



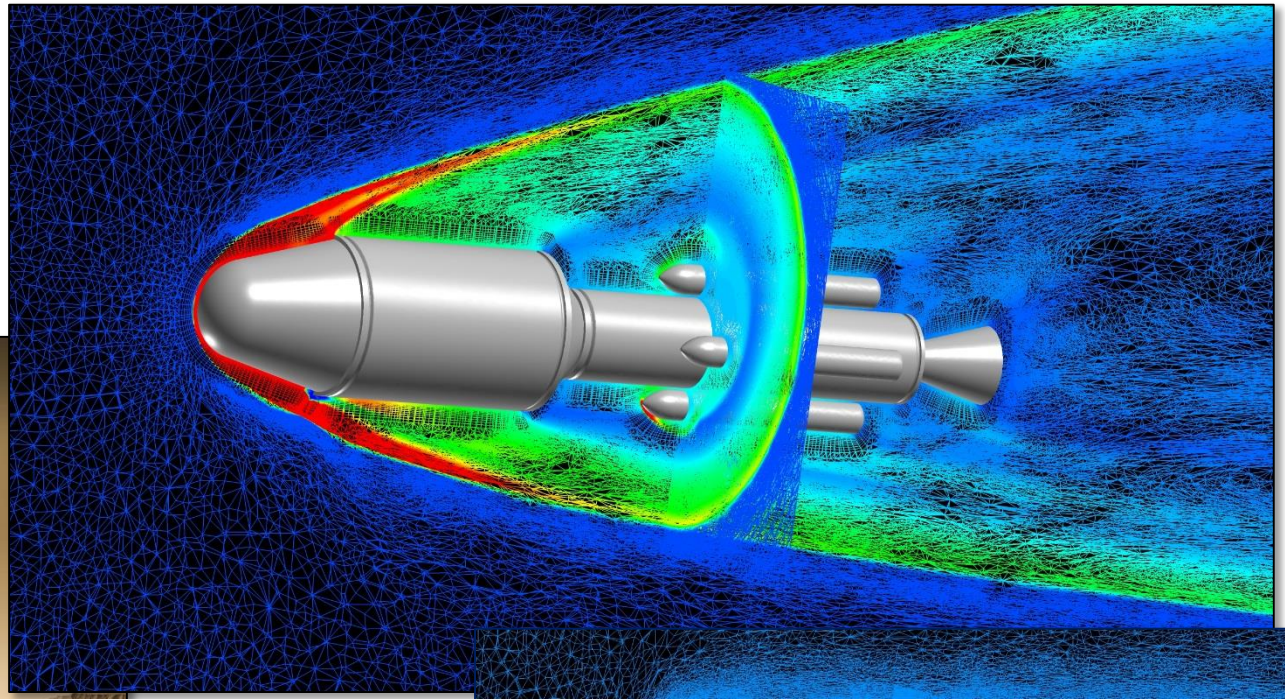


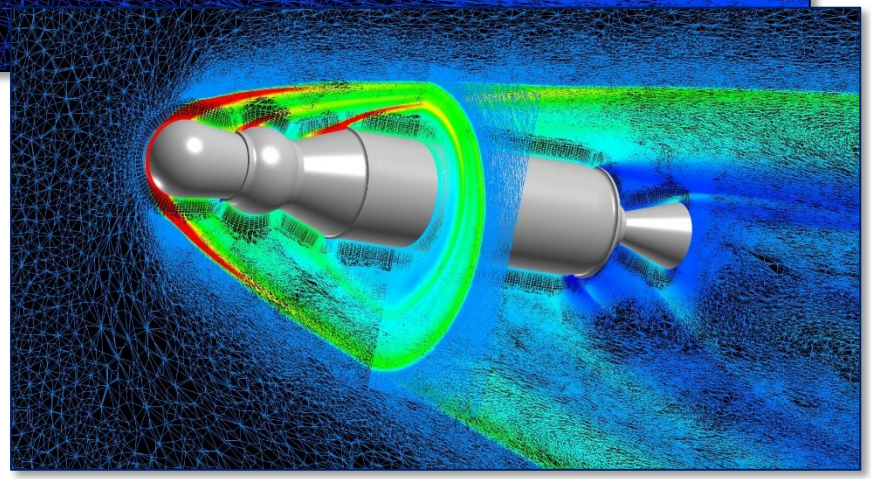


Thousands of simulations are being performed to develop NASA's newest launch vehicle

Mars Ascent Vehicle

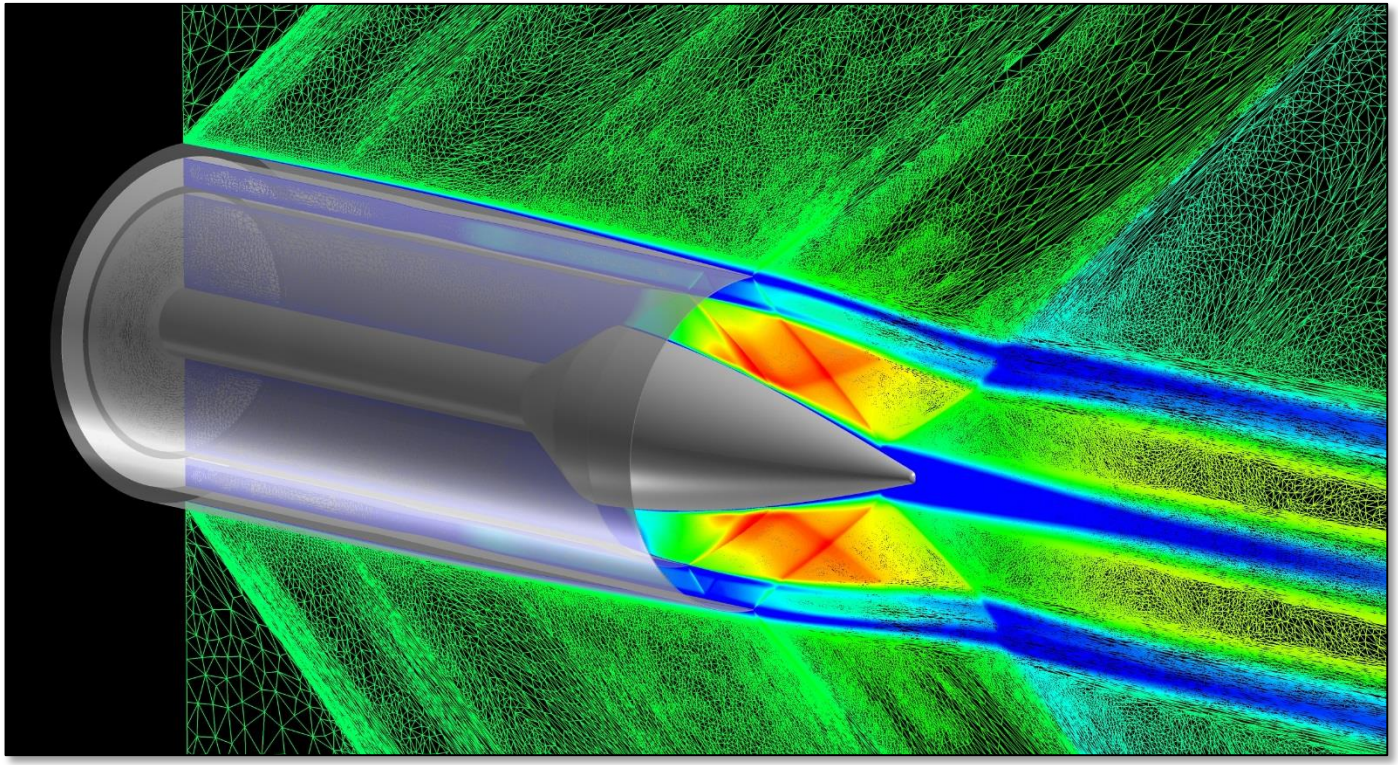
A vehicle like this will be used to return a sample from the Martian surface





Courtesy
Ashley Korzun

Supersonic Nozzle Design

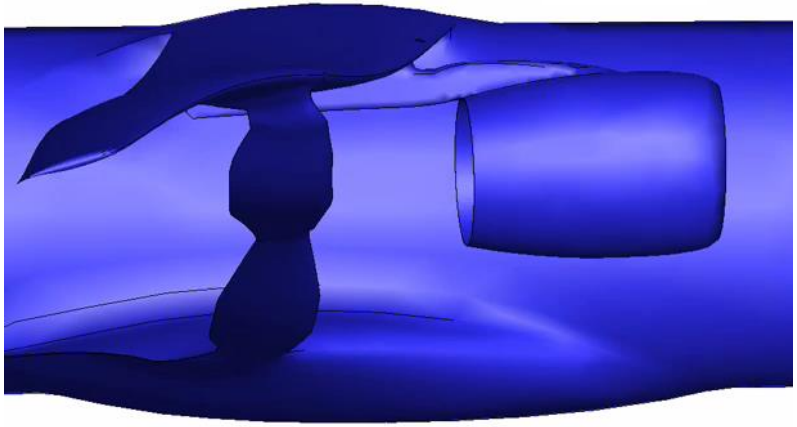


Courtesy
Chris Heath

FUN3D's unique adjoint-based design capability is being used to mitigate sonic boom effects

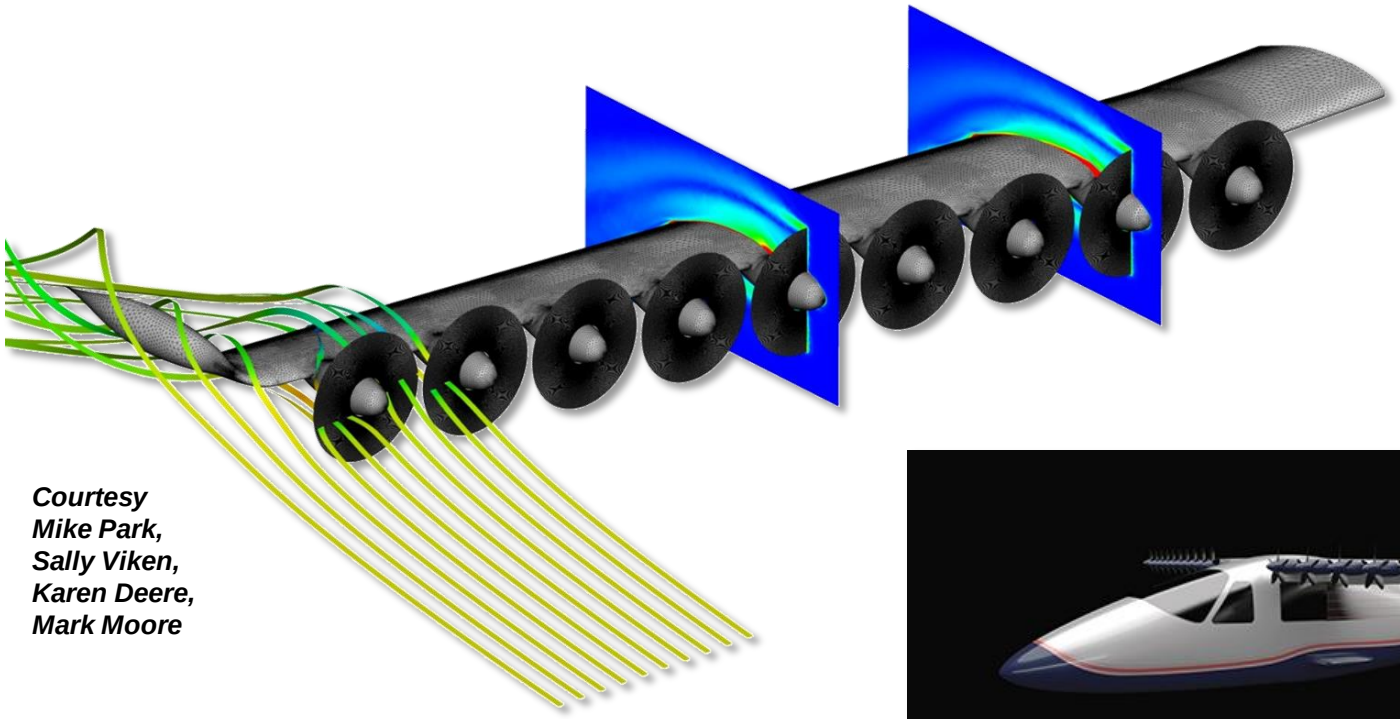
SUGAR Truss-Braced Wing

Courtesy
Bob Bartels



Aeroelastic Analysis of the Boeing SUGAR Truss-Braced Wing Concept

Distributed Electric Propulsion

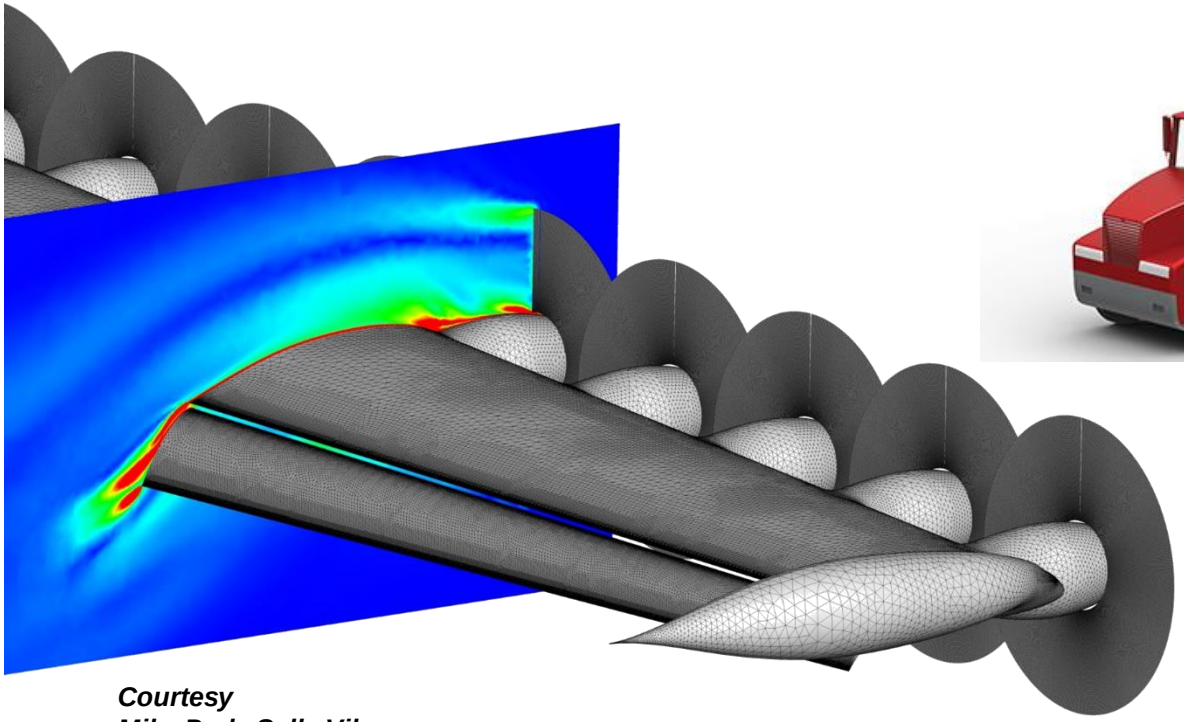


Courtesy
Mike Park,
Sally Viken,
Karen Deere,
Mark Moore

DEP offers vastly improved efficiencies for regional vehicles

Distributed Electric Propulsion

NASA will test a truck-mounted concept in November



Courtesy
Mike Park, Sally Viken,
Karen Deere, Mark Moore

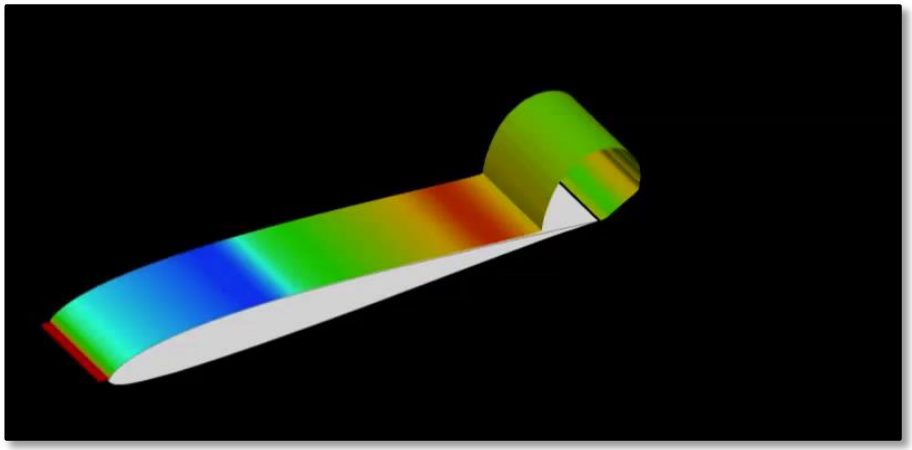
FUN3D and High-Order Methods

Raising the Bar for CFD

- High-order methods have failed to penetrate production aerodynamics due to robustness issues
- Entropy-stable approaches offer mathematical guarantees on stability
  - If implemented correctly, the code will never “blow up”
  - Nature is stable, so our discretizations should be also
- No dealiasing, artificial dissipation, or filtering necessary
- Being developed as a library for FUN3D

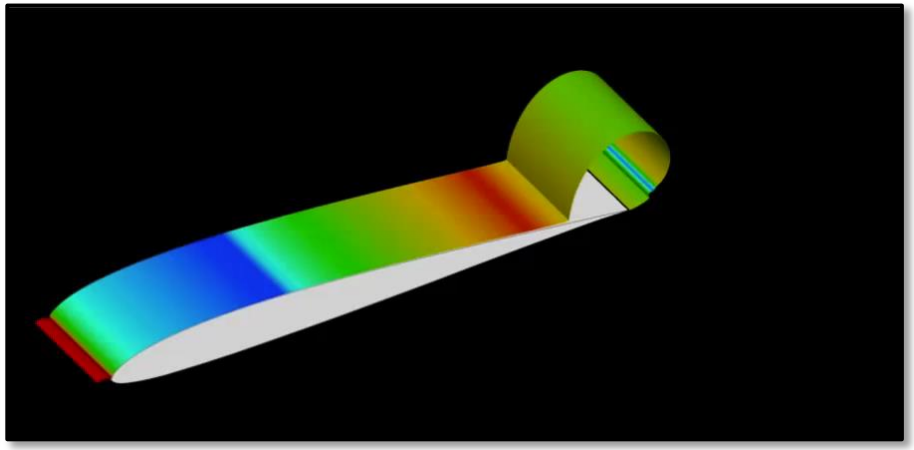
Standard Second-Order Finite Volume Scheme

- 137,060 DOFs
- Shows mild spanwise variations very late



Entropy-Stable Scheme p = 5 (sixth order)

- 29,066,688 DOFs
- Good qualitative resolution of expected flow physics



Across the Aerospace Industry

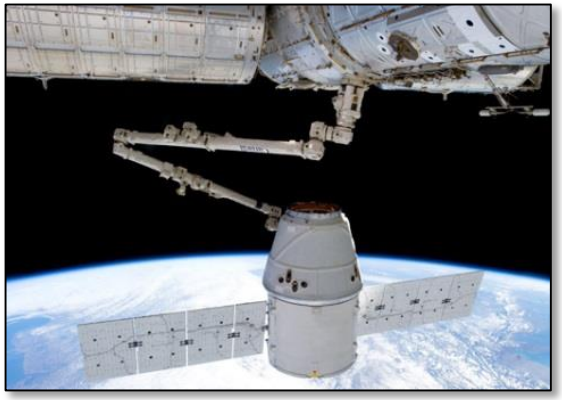
SPACEX

First private company to achieve orbit
and dock with the International Space Station
Primary aerodynamics tool: FUN3D

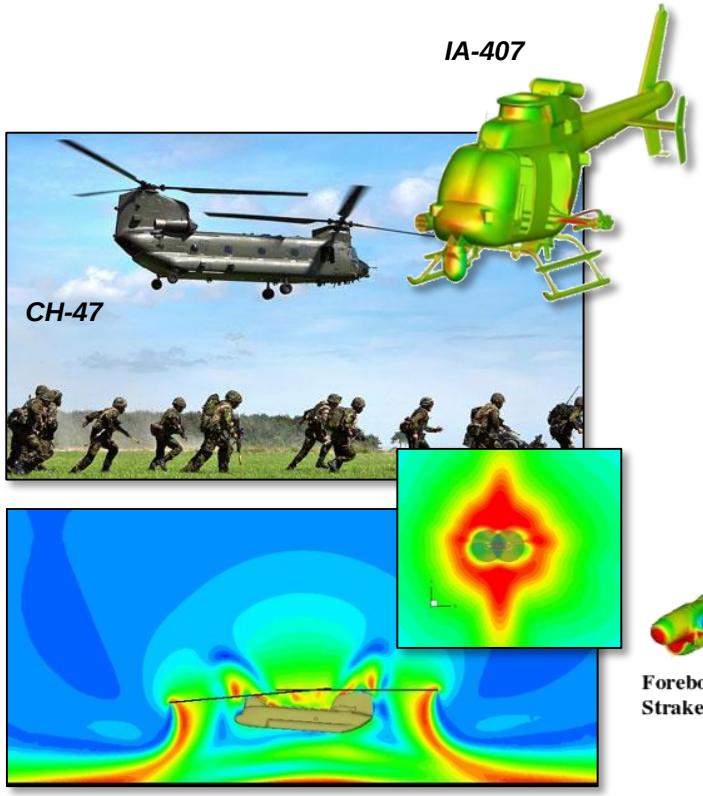
- FUN3D used for extensive analysis of Falcon 1 and Falcon 9 rockets, Dragon spacecraft
- Team consults frequently and provides new features and capabilities as requested

"The FUN3D software suite and development team have enabled SpaceX to rapidly design, build, and successfully fly a new generation of rockets and spacecraft."

- Justin Richeson
Manager, SpaceX Aerodynamics

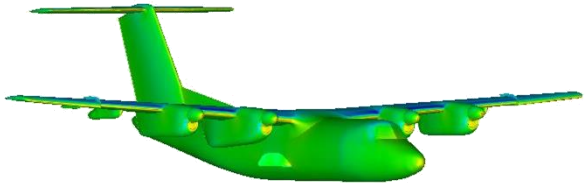


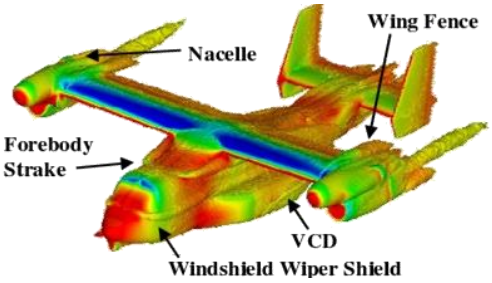
At the Department of Defense

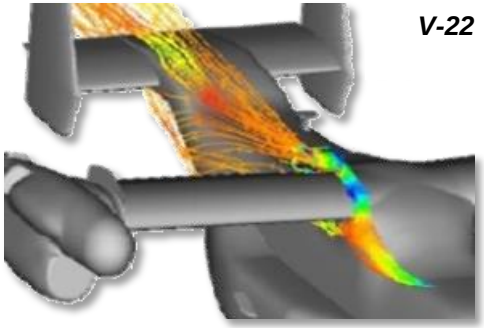


AMRDEC at Redstone Arsenal

- **Troop safety:** airworthiness qualification
- **Dramatic cost savings:** fewer tunnel & flight tests
- Intense demand for timely results on massive computing systems
- Decade of use in direct support of the US warfighter







Air Force Research Lab

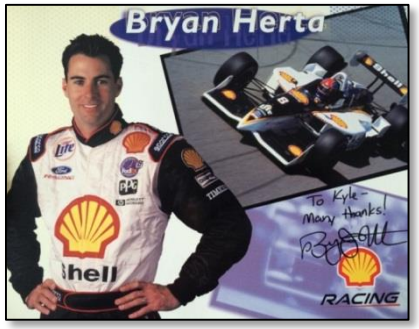
- Funded on-site training workshop (20 students)

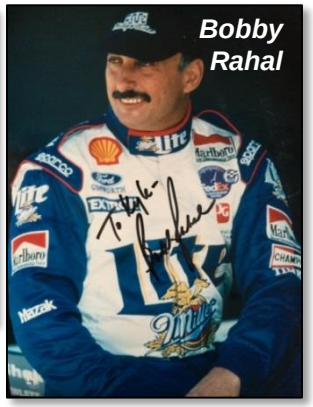
NAVAIR at Patuxent River

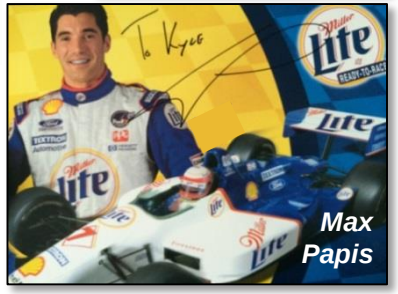
- Hosted on-site training workshop
- Hired two recent Georgia Tech PhD grads
  - FUN3D development for theses

Outside Aerospace

- Ford Motor Sports used FUN3D for improving rear wings for road races and the Indy 500
- Newman/Haas Team: Michael Andretti, Christian Fittipaldi
- Team Rahal: Bobby Rahal, Bryan Herta, and Max Papis
- Real-time designs tested and installed on cars within two weeks

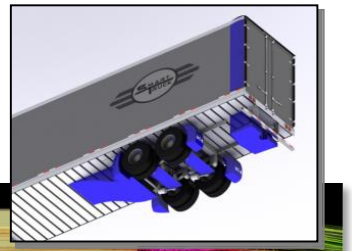


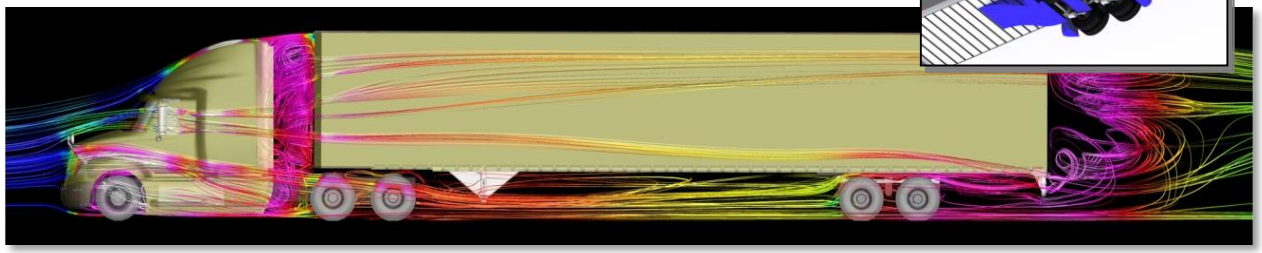




- Used FUN3D on world's largest supercomputer to perform the most advanced truck simulations ever attempted
- Developed add-on kits that improve fuel mileage by as much as 11%
- Spin-off company has sold over 40,000 units
- Covered by print and TV news

"FUN3D is a national asset."
- Mike Henderson,
BMI Founder





The Steady Adjoint Equations

Notation and Governing Equations

We wish to perform rigorous adaptation and design optimization based on the steady-state Euler/Navier-Stokes equations, ***without requiring any a priori knowledge of the problem***:

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}(\mathbf{D}, \mathbf{Q}, \mathbf{X}) = 0$$

$\mathbf{R}$ = Spatial residual

$\mathbf{Q}$ = Dependent variables

$\mathbf{D}$ = Design variables

$\mathbf{X}$ = Computational grid

- Incompressible through hypersonic flows
- May include turbulence models and various physical models from perfect gas through thermochemical nonequilibrium

The Steady Adjoint Equations

Combine cost function with Lagrange multipliers Λ:

$$L(\mathbf{D},\mathbf{Q},\mathbf{X},\Lambda_f,\Lambda_g) = \underbrace{f(\mathbf{D},\mathbf{Q},\mathbf{X})}_{\text{Cost Function}} + \Lambda_f^T \underbrace{\mathbf{R}(\mathbf{D},\mathbf{Q},\mathbf{X})}_{\text{Flowfield Equations}} + \Lambda_g^T \underbrace{(\mathbf{K}\mathbf{X} - \mathbf{X}_{surf})}_{\text{Mesh Movement Equations}}$$

f = Cost function (lift/drag/boom/etc)

Λ_f = Flowfield adjoint variable

$\mathbf{K}$ = Mesh movement elasticity matrix

Λ_g = Grid adjoint variable

Differentiate with respect to $\mathbf{D}$:

$$\frac{dL}{d\mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right]^T \Lambda_f + \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{D}}\right]^T \left\{\frac{\partial f}{\partial \mathbf{Q}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \Lambda_f\right\}$$

$$+ \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \left\{\frac{\partial f}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T \Lambda_f + \Lambda_g^T \mathbf{K}\right\} - \Lambda_g^T \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]_{surf}$$

$$\rightarrow \underbrace{\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T}_{\text{Governing Eqns}} \Lambda_f = \underbrace{-\frac{\partial f}{\partial \mathbf{Q}}}_{\text{Engineering Output}}$$

This adjoint equation for the flowfield has powerful implications for:

- Error estimation & mesh adaptation
- Sensitivity analysis

The Steady Adjoint Equations

Error Estimation and Mesh Adaptation

It is apparent that:

$$\Lambda_f \equiv \frac{\partial f}{\partial \mathbf{R}} \longrightarrow$$

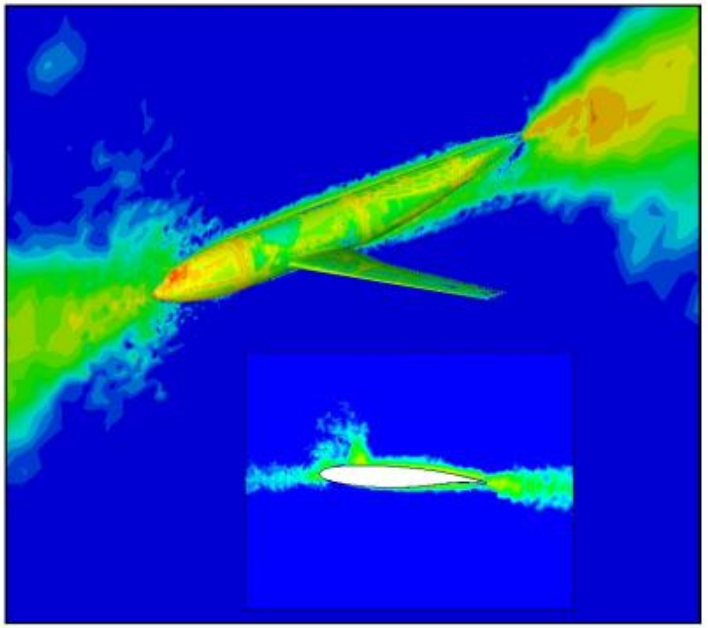
Direct relationship between local equation error and the output we are interested in

- These relationships can be used to get error estimates on f
- Also used to compute a scalar field explicitly relating local point spacing requirements to output accuracy for a user-specified error tolerance
- Often yields non-intuitive insight into gridding requirements
- Relies on underlying mathematics to adapt, rather than heuristics such as solution gradients

User no longer required to be a CFD expert to get the right answer

Transonic Wing-Body:

"Where do I need to put grid points to get 10 drag counts of accuracy?"



Blue=Sufficient Resolution
Red=Under-Resolved

The Steady Adjoint Equations

Sensitivity Analysis

Examine the remaining terms in the linearization:

$$\frac{dL}{d\mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right]^T \boldsymbol{\Lambda}_f + \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \left\{\frac{\partial f}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T \boldsymbol{\Lambda}_f + \boldsymbol{\Lambda}_g^T \mathbf{K}\right\} - \boldsymbol{\Lambda}_g^T \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]_{surf}$$

$$\rightarrow \quad \mathbf{K}^T \boldsymbol{\Lambda}_g = -\left\{\frac{\partial f}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T \boldsymbol{\Lambda}_f\right\}$$ Discrete adjoint equation for mesh movement

$$\rightarrow \quad \frac{dL}{d\mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + \boldsymbol{\Lambda}_f^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}} - \boldsymbol{\Lambda}_g^T \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]_{surf}$$ Sensitivity equation

Function Evaluation	**Sensitivity Evaluation**
1. Compute surface mesh at current D	3. Solve flowfield adjoint equations
2. Solve mesh movement equations	2. Solve mesh adjoint equations
3. Solve flowfield equations	1. Matrix-vector product over surface

Analysis Cost = Sensitivity Analysis Cost

Even for *1000's* of design variables

Deriving the Unsteady Adjoints

$$L(\mathbf{D},\mathbf{Q},\mathbf{X},\Lambda,\Lambda_g) = f\Delta t + \sum_{n=1}^{N}[\Lambda_g^n]^T\mathbf{G}^n\Delta t$$
$$+ \sum_{n=1}^{N}\Bigg\{[\mathbf{C}_s^n \circ \Lambda_s^n]^T\Bigg[a\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t}\circ\mathbf{V}_s^n$$
$$+ c\frac{\mathbf{I}_s^n\mathbf{Q}^{n-2} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t}\circ(\mathbf{I}_s^n\mathbf{V}^{n-2})$$
$$+ d\frac{\mathbf{I}_s^n\mathbf{Q}^{n-3} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t}\circ(\mathbf{I}_s^n\mathbf{V}^{n-3})\Bigg]$$
$$+ [\Lambda_s^n]^T[\mathbf{R}^n + ((\mathbf{I}_s^n\mathbf{Q}^{n-1})\circ\mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n)\circ\mathbf{R}_{\mathrm{GCL}}^n]$$
$$+ [\Lambda_f^n]^T[\mathbf{A}^n\mathbf{Q}^n] + [\Lambda_h^n]^T[\mathbf{P}^n\mathbf{Q}^n]\Bigg\}\Delta t$$
$$+ (f^0 + [\Lambda_g^0]^T\mathbf{G}^0 + [\Lambda^0]^T\mathbf{R}^{\mathrm{in}})\Delta t$$

- Flow field and grid adjoint equations derived for the time-dependent Navier-Stokes equations on arbitrary combinations of static/rigidly moving/deforming overset grids undergoing parent-child motion
- The following terms are included in the Lagrangian
  - Objective function
  - Grid terms
  - Higher-order temporal terms
  - Fluxes
  - Geometric Conservation Law term
  - Overset interpolation terms
  - Initial conditions
- Implemented by hand and verified using complex variables

∘ *is the Hadamard vector multiplication operator; see*

Nielsen, E.J. and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," AIAA Journal, Vol. 51, No. 6, June 2013.

The Unsteady Adjoint Equations

$$\mathbf{C}_s^n \circ \left[a \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + c \frac{\mathbf{I}_s^n \mathbf{Q}^{n-2} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n \mathbf{V}^{n-2} + d \frac{\mathbf{I}_s^n \mathbf{Q}^{n-3} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n \mathbf{V}^{n-3} \right] + \mathbf{R}^n + ((\mathbf{I}_s^n \mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta \tilde{\mathbf{C}}_s^n) \circ \mathbf{R}_{\mathrm{GCL}}^n = 0 \quad \text{(A1)}$$

Proceeding as before, the Lagrangian can be written as

$$L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \boldsymbol{\Lambda}, \boldsymbol{\Lambda}_g) = f \Delta t + \sum_{n=1}^{N} [\mathbf{A}_g^n]^T \mathbf{G}^n \Delta t + \sum_{n=1}^{N} \left\{ [\mathbf{C}_s^n \circ \boldsymbol{\Lambda}_s^n]^T \left[a \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + c \frac{\mathbf{I}_s^n \mathbf{Q}^{n-2} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n \mathbf{V}^{n-2}) + d \frac{\mathbf{I}_s^n \mathbf{Q}^{n-3} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n \mathbf{V}^{n-3}) \right] + [\mathbf{A}_s^n]^T [\mathbf{R}^n + ((\mathbf{I}_s^n \mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta \tilde{\mathbf{C}}_s^n) \circ \mathbf{R}_{\mathrm{GCL}}^n] + [\mathbf{A}_f^n]^T [\mathbf{A}^n \mathbf{Q}^n] + [\mathbf{A}_h^n]^T [\mathbf{P}^n \mathbf{Q}^n] \right\} \Delta t + (f^0 + [\mathbf{A}_g^0]^T \mathbf{G}^0 + [\mathbf{A}^0]^T \mathbf{R}^{\mathrm{in}}) \Delta t \quad \text{(A2)}$$

On time levels 1 and 2, the time derivatives are assumed to be discretized with the BDF1 and BDF2 schemes, respectively. Taking into account the dependencies on data at time levels $n-2$ and $n-3$, the adjoint equations are obtained as follows:

S:

$$\frac{a}{\Delta t} \mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \boldsymbol{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_s^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_s^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_s^n} \right]^T - \mathbf{I}_s^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

F:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_f^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_f^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_f^n} \right]^T - \mathbf{I}_f^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

H:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_h^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_h^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_h^n} \right]^T - \mathbf{I}_h^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

for $3 \le n \le N$ (A3)

S:

$$\frac{3}{2\Delta t} \mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \boldsymbol{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_s^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_s^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_s^n} \right]^T - \mathbf{I}_s^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

F:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_f^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_f^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_f^n} \right]^T - \mathbf{I}_f^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

H:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_h^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_h^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_h^n} \right]^T - \mathbf{I}_h^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{a}{\Delta t} \mathbf{V}_s^{n+1} - \frac{c}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} - \frac{d}{\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \boldsymbol{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \boldsymbol{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\}$$

for $n = 2$ (A4)

S:

$$\frac{1}{\Delta t} \mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \boldsymbol{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_s^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_s^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_s^n} \right]^T - \mathbf{I}_s^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{3}{2\Delta t} \mathbf{V}_s^{n+1} - \frac{1}{2\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \right. \right.$$
$$\left. + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \circ \right. \right.$$
$$\left. \left. + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \right. \right. \right\}$$

F:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n} \right]^T \boldsymbol{\Lambda}_s^n + [\mathbf{A}_f^n]^T \boldsymbol{\Lambda}_f^n + [\mathbf{P}_f^n]^T \boldsymbol{\Lambda}_h^n = - \left[\frac{\partial f}{\partial \mathbf{Q}_f^n} \right]^T - \mathbf{I}_f^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{1}{2\Delta t} \mathbf{I}_s^{n+1} \mathbf{V}^{n-1} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \right. \right.$$
$$\left. + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t} \mathbf{I}_s^{n+2} \mathbf{V}^n \right) \right. \right.$$
$$\left. \left. + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t} \mathbf{I}_s^{n+3} \mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \boldsymbol{\Lambda}_s^{n+3} \right] \right\} \right.$$

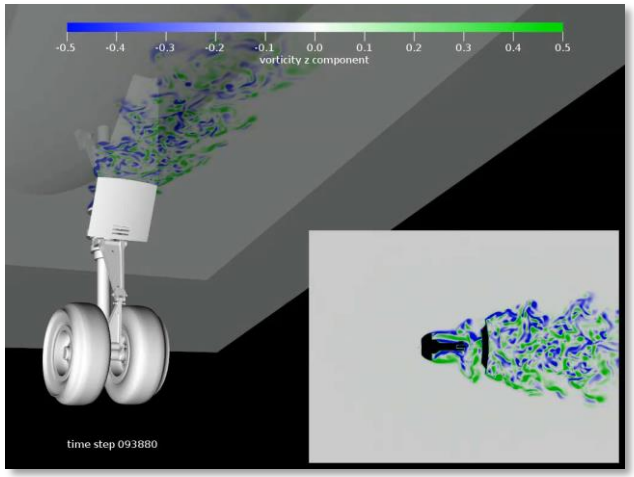
- The time-dependent adjoint equations are considerably more complex than the Navier-Stokes equations
- Total FUN3D implementation consists of nearly 1 million lines of code
- Tremendous amount of software infrastructure required

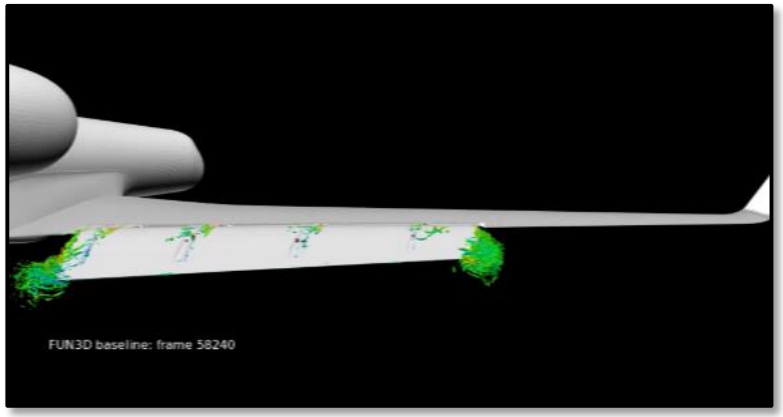
Page 1 of 4 of the adjoint equations derived and implemented in:

Nielsen, E.J., and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," AIAA Journal, Vol. 51, No. 6, June 2013.

Challenges for Unsteady Problems

- Extensive linearization and infrastructure effort, particularly for dynamic and overset grids
- Sheer cost – every simulation is now a time-dependent run
  - For steady flows, terms could be computed once and stored for efficiency
  - Unsteady flows require these linearizations to be recomputed at every time step
- Need for entire forward solution
  - Brute force it: Store to disk (big data)
  - Recompute it: Store periodically, recompute intermediate steps as needed (checkpointing)
  - Approximate it: Store periodically, interpolate intermediate steps as needed
- Chaotic flows





Challenges for Unsteady Problems

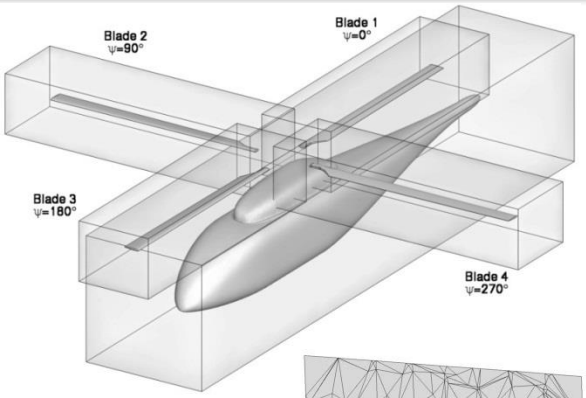
Big Data

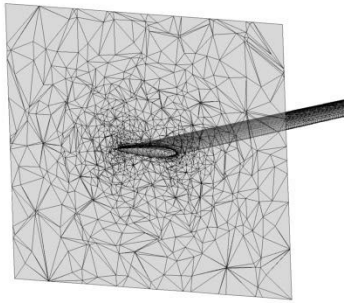
- The amount of data adds up fast – consider an example:
  - 50,000,000 grid points and 10,000 physical time steps
  - Using a 1-equation turbulence model (6 unknowns per grid point)
  - Dynamic grids (3 additional unknowns per grid point)
    - → 50,000,000 x 10,000 x (6+3) x 8 bytes = 36 Terabytes
- This amount of data has not been prohibitively large for our resources, but it is a lot (and we need to go much bigger)
- So far, the challenge has been efficiently getting the data to/from the disk at every single time step
- Implemented sophisticated parallel asynchronous I/O mechanisms
  - Data is brought in for the next time step during current time step

Verification of Implementation

Problem Definition

- Fully turbulent flow: M_∞=0.1, α=2º, *Re*=1M, μ=0.12
- Composite grid consists of six component grids
- All verification cases run on 360 cores





Component	Topology	Motion	Motion Paradigm	Ancestry
Domain	Hex (Cartesian)	Inertial	Static	Great-grandparent
Fuselage	Prz/pyr/tet	Rotation, translation	Rigid	Grandparent
Blades	Tet	Azimuthal rotation	Rigid	Parent
Blades	-	1º vertical oscillatory rotation about hub	Deforming	Child
Total Composite Grid	**1,033,243 nodes 3,190,160 elements Hex/prz/pyr/tet**	**-**	**Deforming**	**Four generations**

Verification of Implementation

Compressible Results Shown; Incompressible Also Available

Adjoint Result Complex Variable Result (ε=1x10^{-50})

All equation sets converged to machine precision

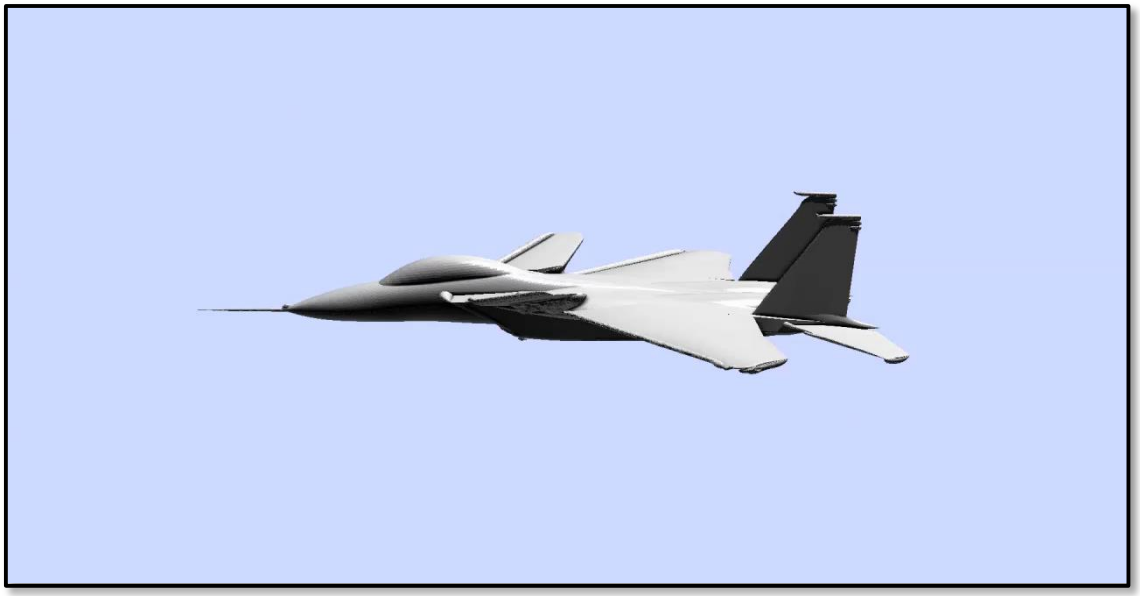
	$\partial C_L / \partial \mathbf{D}$ After 5 Physical Time Steps			
Design Variable	**BDF1**	**BDF2**	**BDF2opt**	**BDF3**
Angle of Attack	0.032387388401060 0.032387388401060	0.032390834852470 0.032390834852468	0.032382969025224 0.032382969025223	0.032374960728472 0.032374960728471
Rot Rate Blade 1	0.049010917009587 0.049010917009599	0.049303058989982 0.049303058989996	0.049392787479850 0.049392787479863	0.049505103043920 0.049505103043932
Shape Blade 2	-0.004741396075215 -0.004741396075140	-0.005822463933444 -0.005822463933378	-0.005891431208194 -0.005891431208081	-0.006004976330078 -0.006004976329965
Flap Freq Blade 3	-0.117898939551988 -0.117898939551986	-0.117819415724222 -0.117819415724217	-0.117766926835991 -0.117766926835985	-0.117703857525237 -0.117703857525232
Rot Rate Fuselage	0.069017024693610 0.069017024693502	0.064234646041659 0.064234646041451	0.064468559766846 0.064468559764283	0.064688175664501 0.064688175664242
Trans Rate Fuselage	-0.002337944913071 -0.002337944913072	-0.002888267191799 -0.002888267191802	-0.002909479741304 -0.002909479741305	-0.002940703514842 -0.002940703514857
Shape Fuselage	-0.000035249806854 -0.000035249806854	-0.000039222298162 -0.000039222298162	-0.000039485944155 -0.000039485944155	-0.000039831885096 -0.000039831885096

Examples

Forward / Reverse Solutions for F-15

Forward Solution

- Transonic turbulent flow over modified F-15 configuration
- Propulsion effects included as well as simulated aeroelastic deformations of canard/wing/h-tail
- Objective is lift-to-drag ratio



Reverse Solution

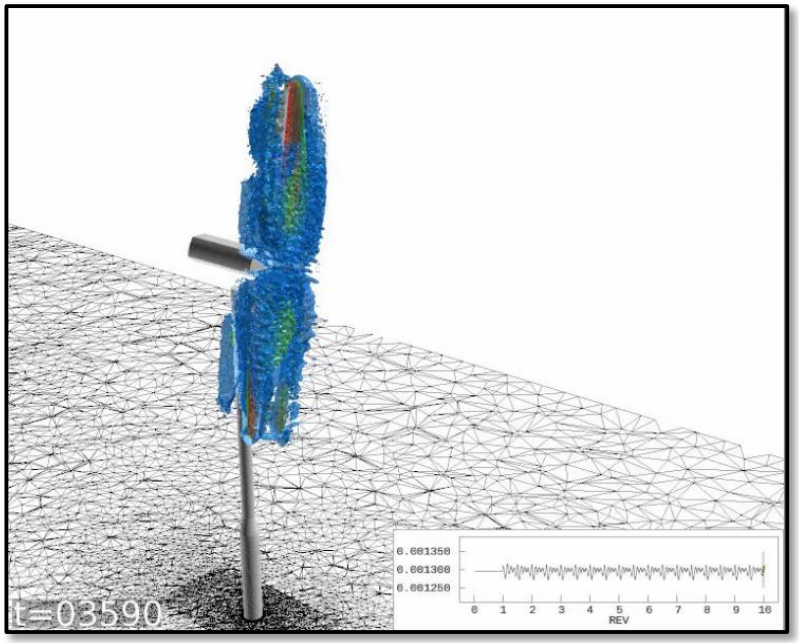
Examples

Forward / Reverse Solutions for Wind Turbine

Forward Solution

- Incompressible turbulent flow over NREL Phase VI wind turbine
- Overset grids used to model rotating blade system
- Objective function is based on the torque

Reverse Solution

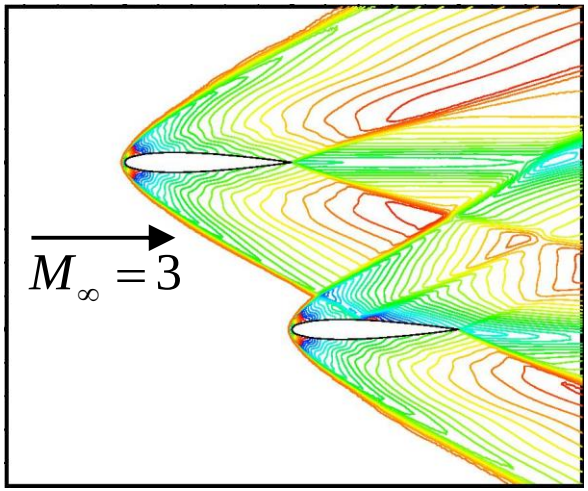


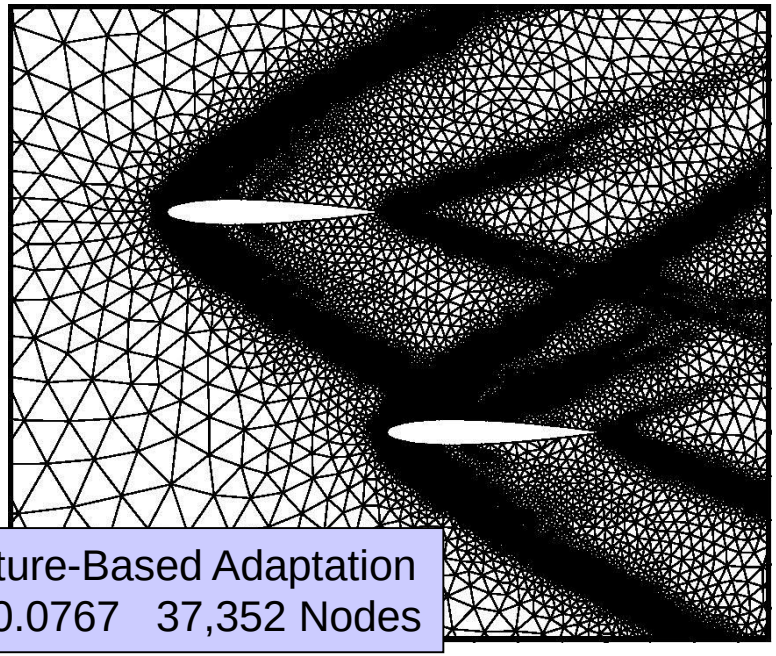
Adjoint-Based Mesh Adaptation

Collaboration with Venditti / Darmofal of MIT

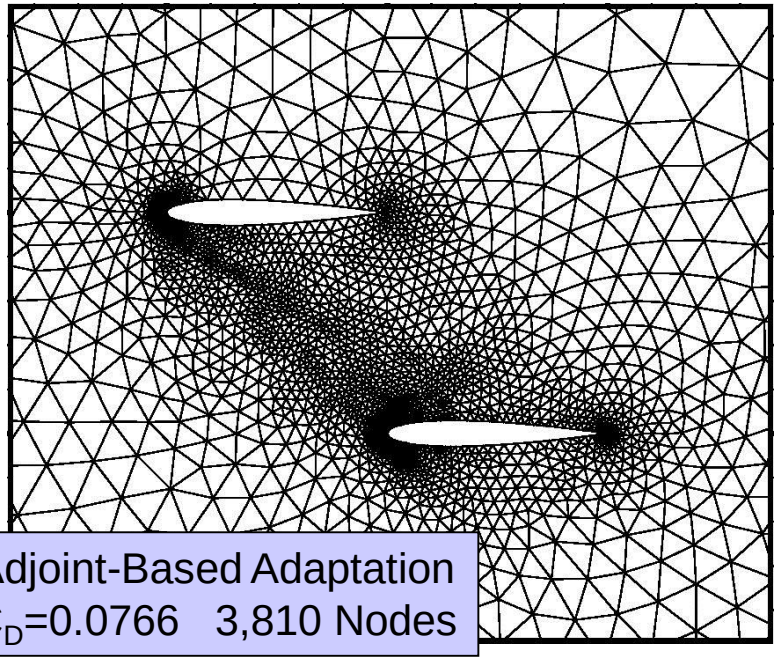
- **Objective:** Adapt grid to compute drag on lower airfoil as accurately as possible

- **Result of adjoint-based adaptation:**
  - Uniformly-resolved shocks are not required
  - Drag is computed accurately with a 90% smaller grid



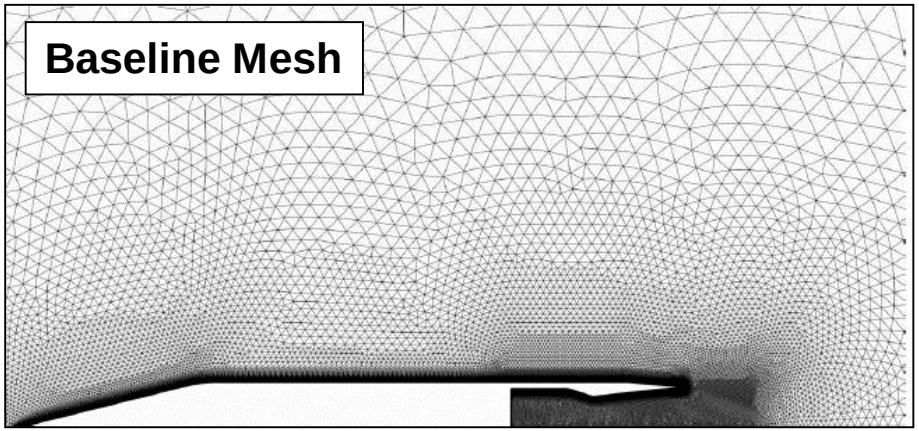


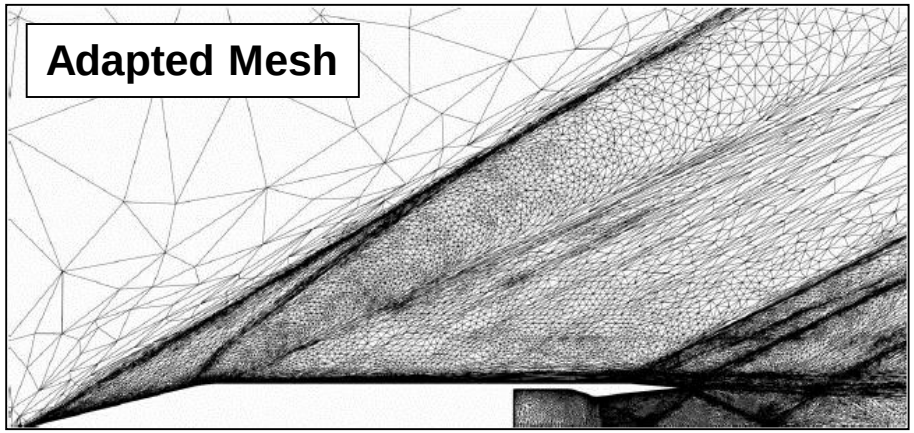
Feature-Based Adaptation
C_D=0.0767 37,352 Nodes

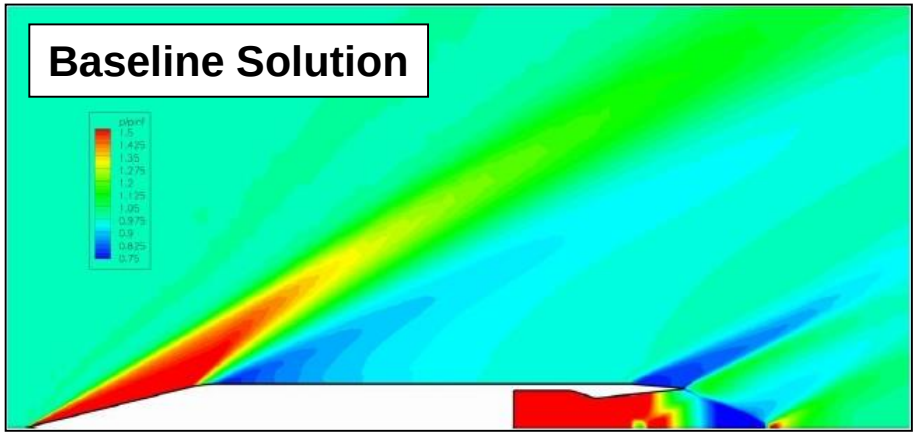


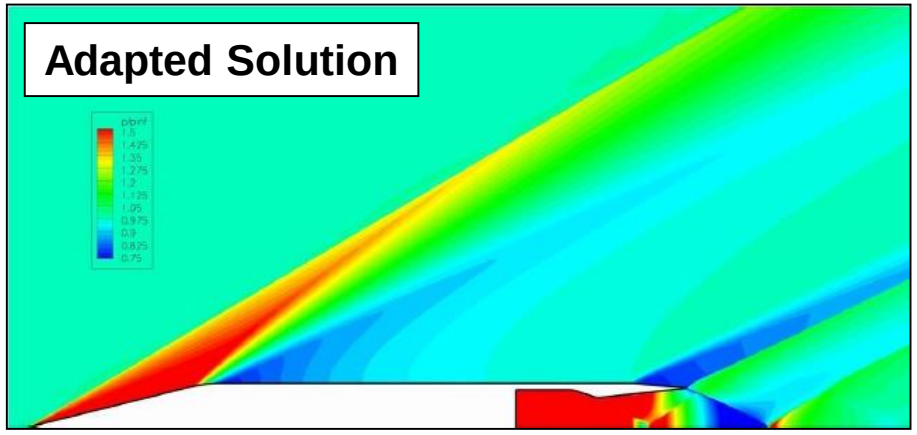
Adjoint-Based Adaptation
C_D=0.0766 3,810 Nodes

Mesh Adaptation for Jet Plume



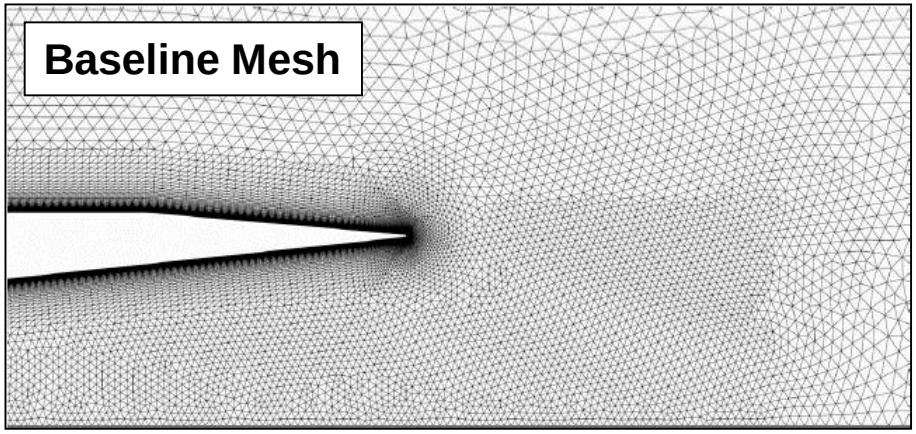


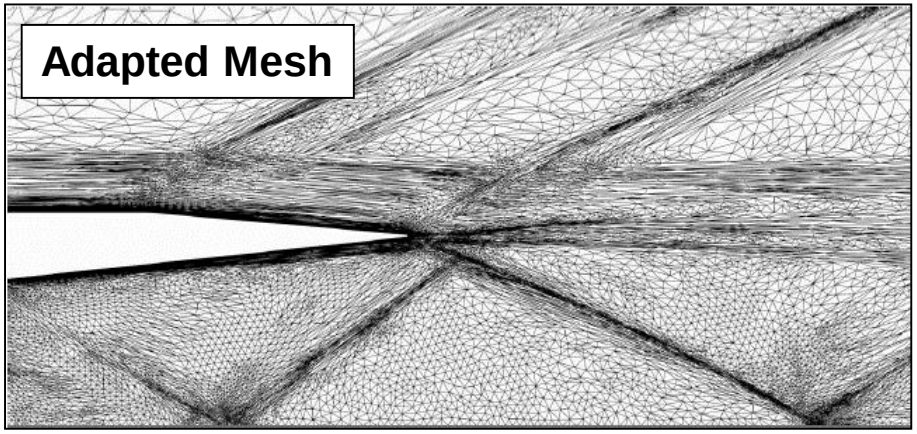


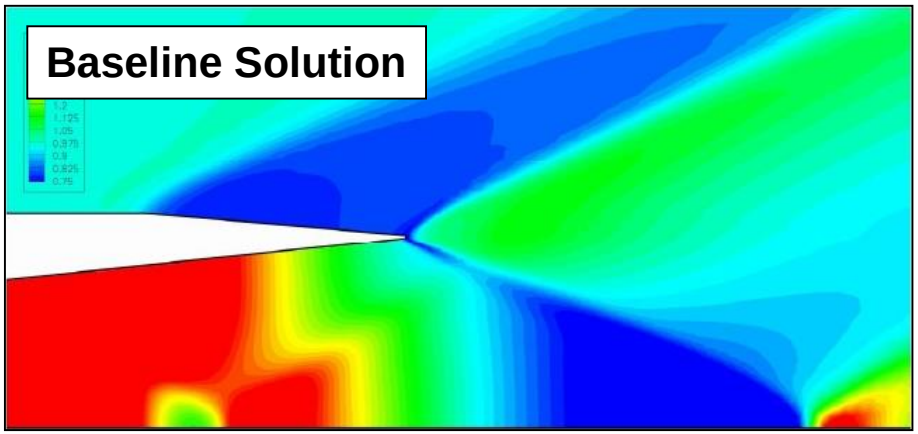


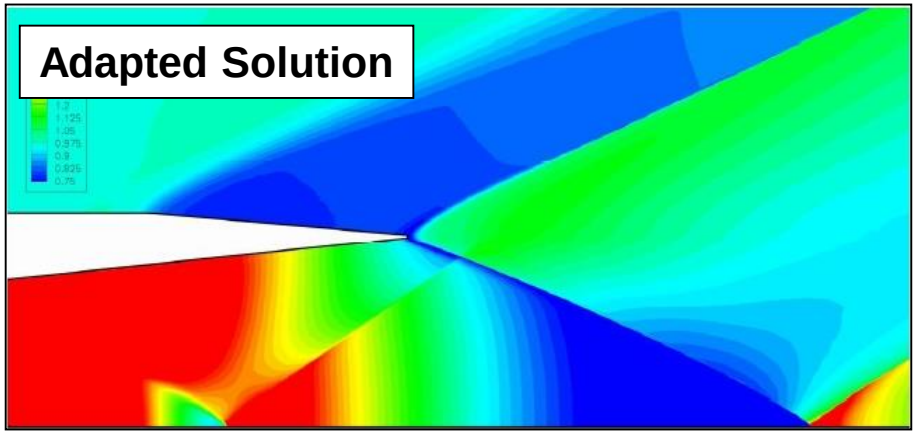
- Quarter of axisymmetric domain modeled; M_∞=2.2, Re_D=1.86M
- Adjoint objective function is integrated pressure signal at 1D distance
- Mesh adapted from 1.3M nodes to 2.9M nodes

Mesh Adaptation for Jet Plume

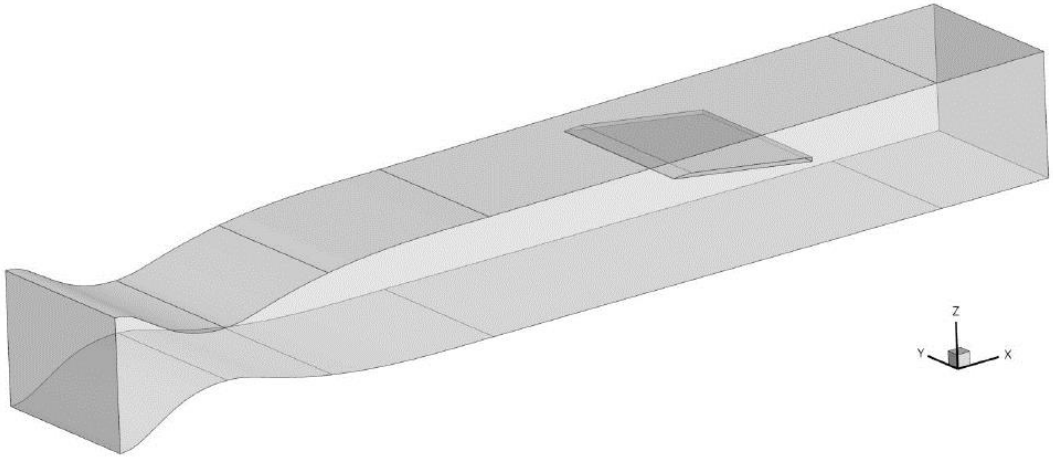


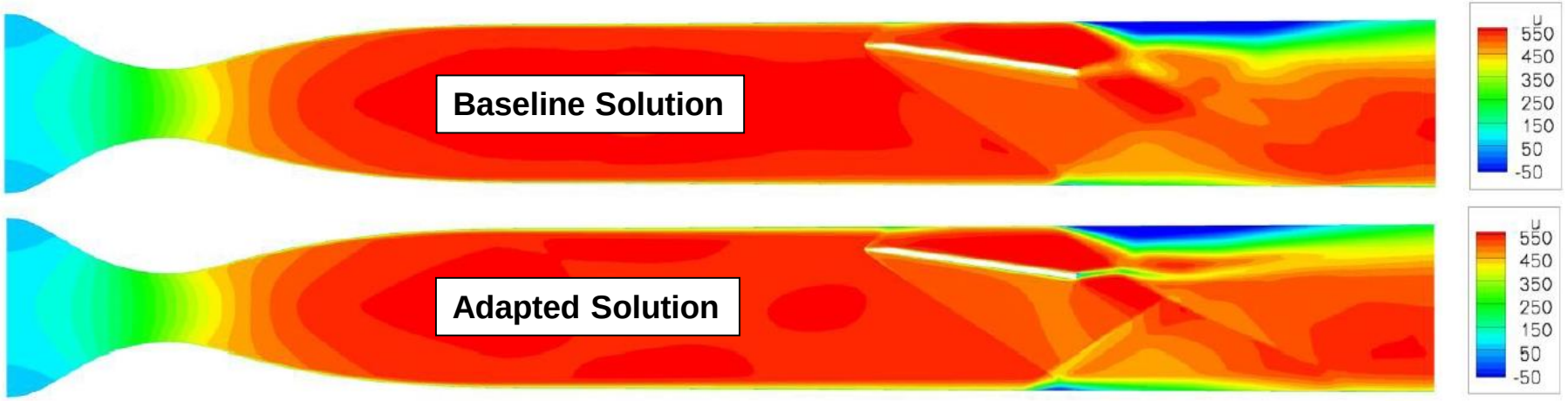






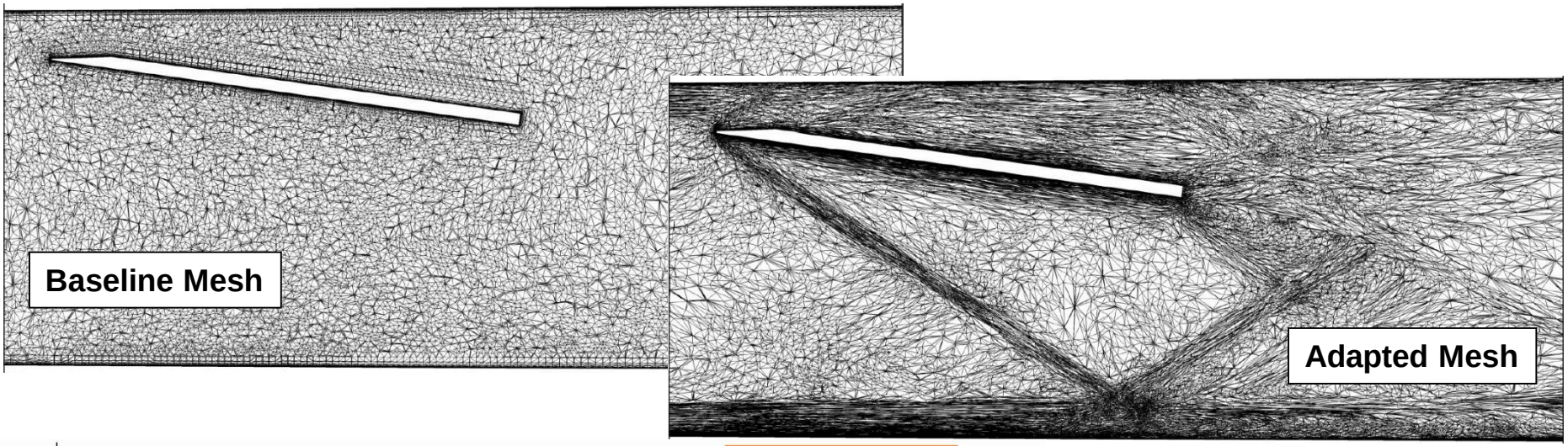
Mesh Adaptation for SBLI

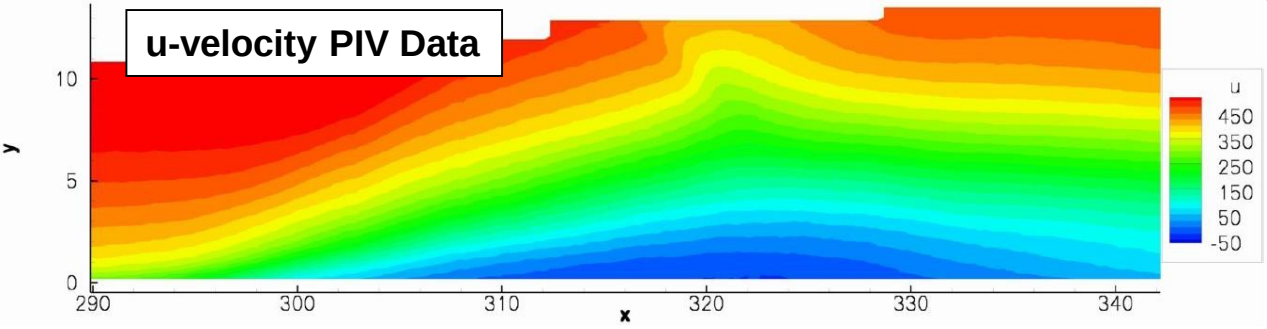


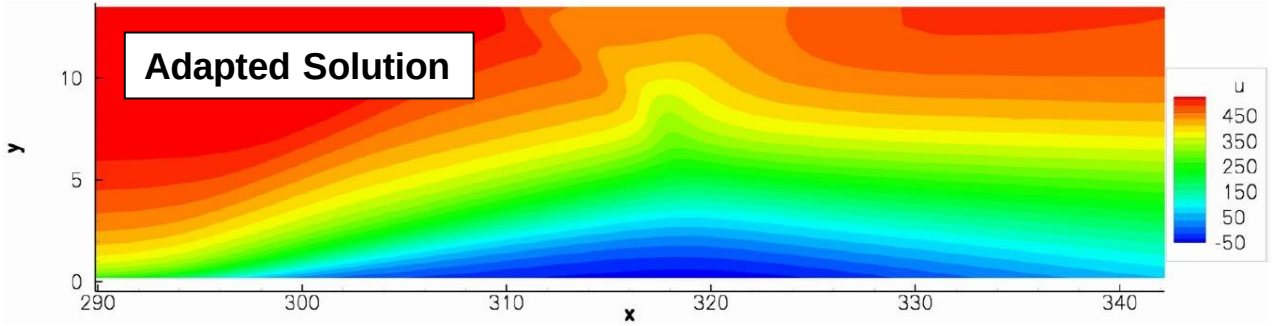


- Part of SBLI workshop at 2010 AIAA Orlando ASM conference
- M_∞=2.25, Re=5683/cm
- Adjoint objective function is drag on lower wall
- Mesh adapted from 0.7M nodes to 1.3M nodes

Mesh Adaptation for SBLI





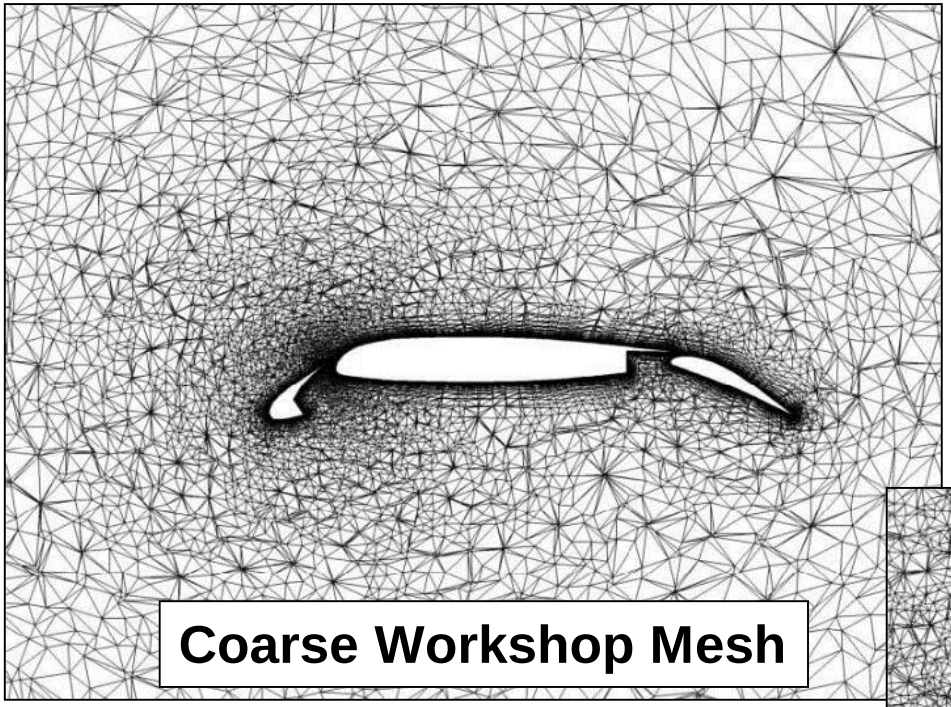


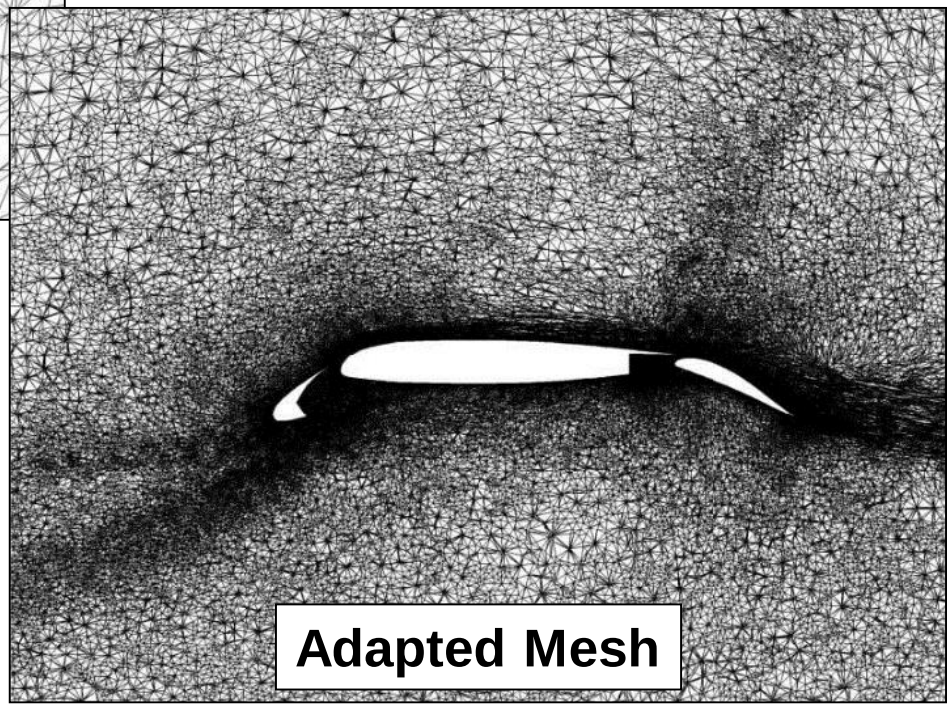
Mesh Adaptation for HLPW-1

Held at 2010 AIAA Summer Meeting in Chicago

Mesh Adaptation for HLPW-1

Held at 2010 AIAA Summer Meeting in Chicago

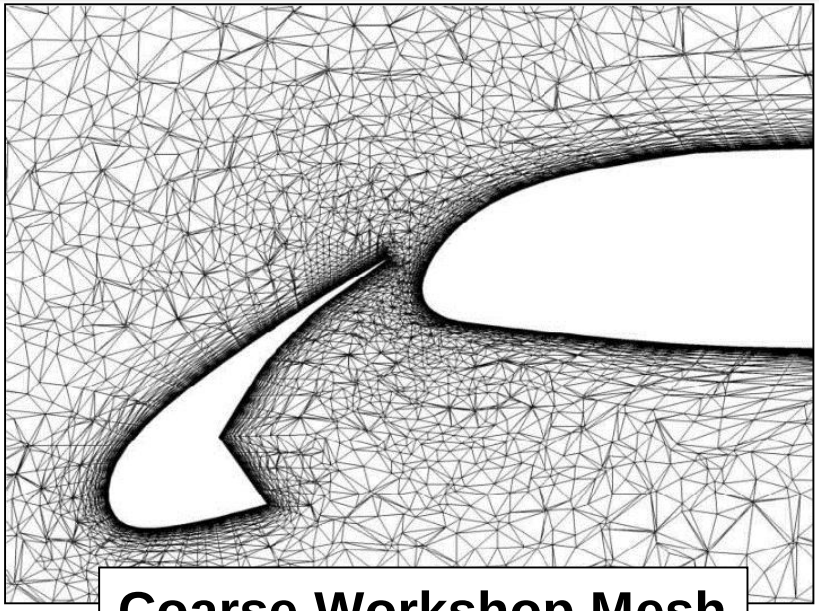




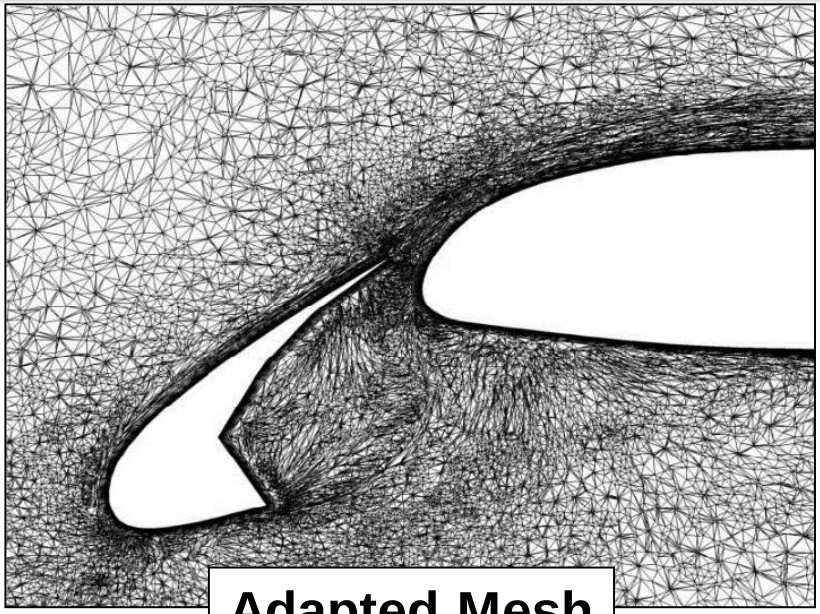
Langley Research Center

Mesh Adaptation for HLPW-1

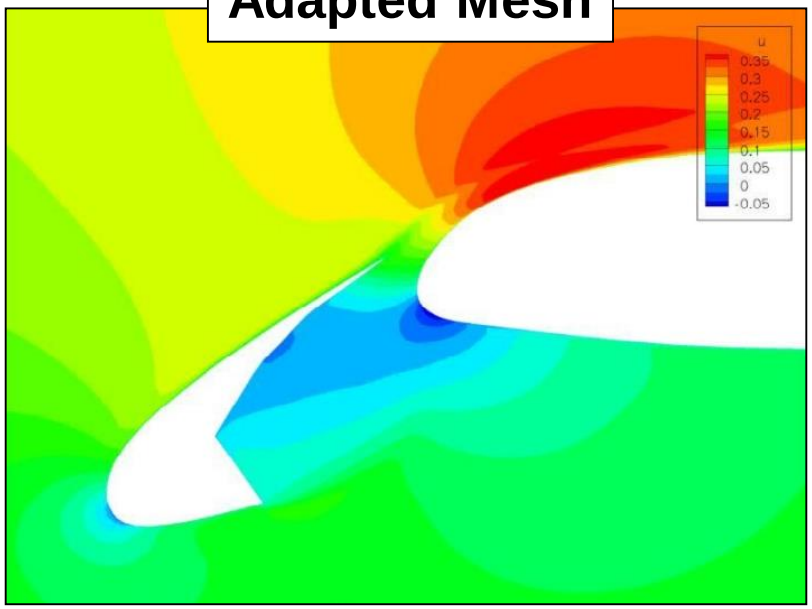
Held at 2010 AIAA Summer Meeting in Chicago



Coarse Workshop Mesh

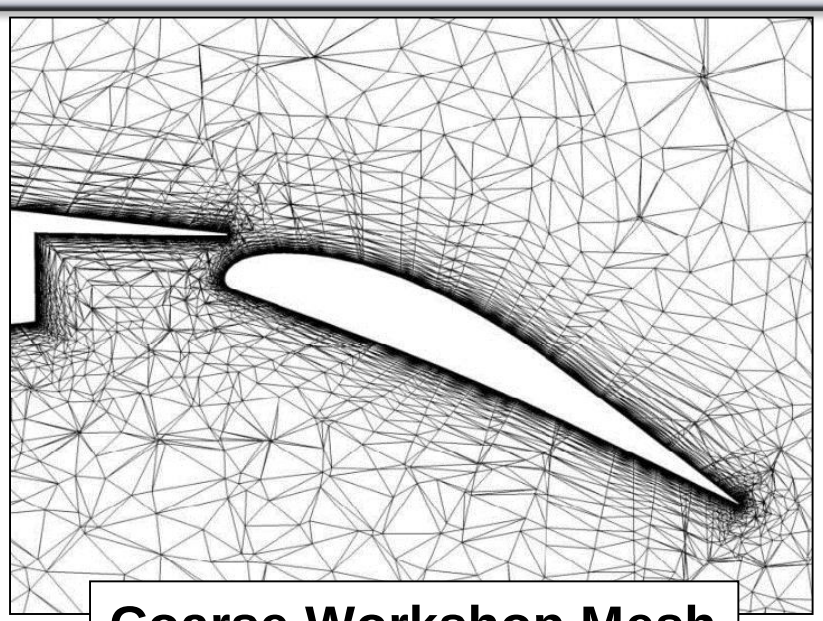


Adapted Mesh

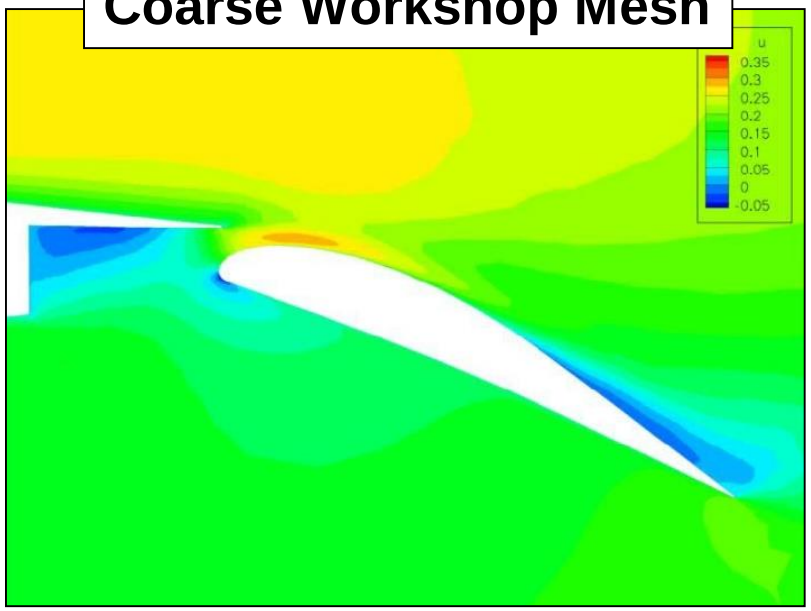


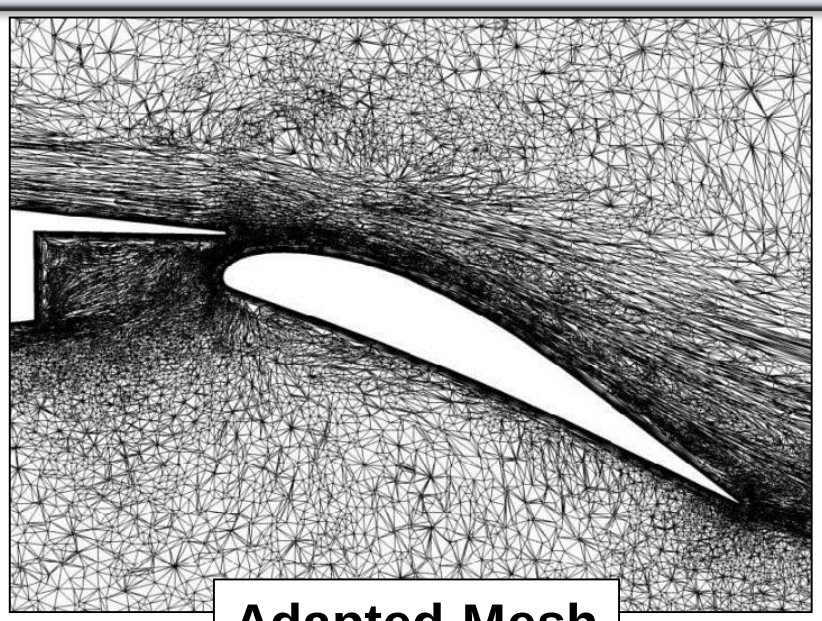
Mesh Adaptation for HLPW-1

Held at 2010 AIAA Summer Meeting in Chicago

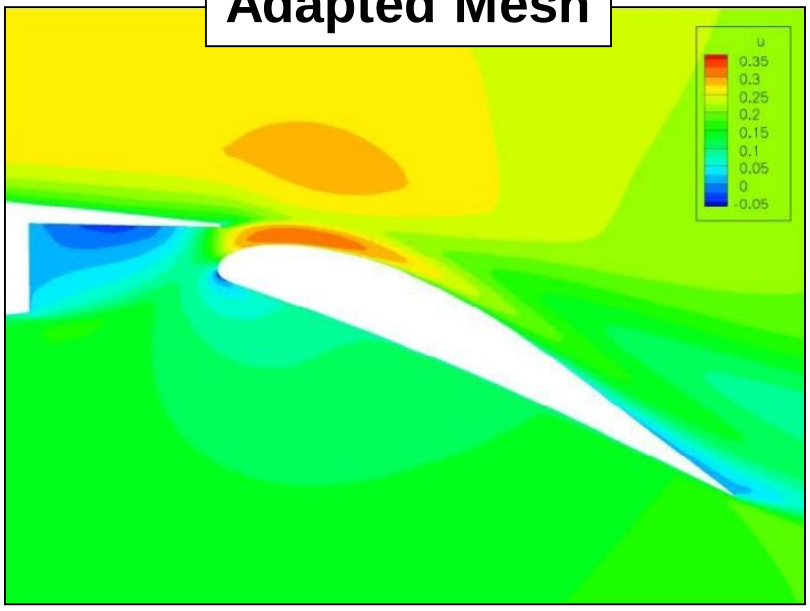


Coarse Workshop Mesh



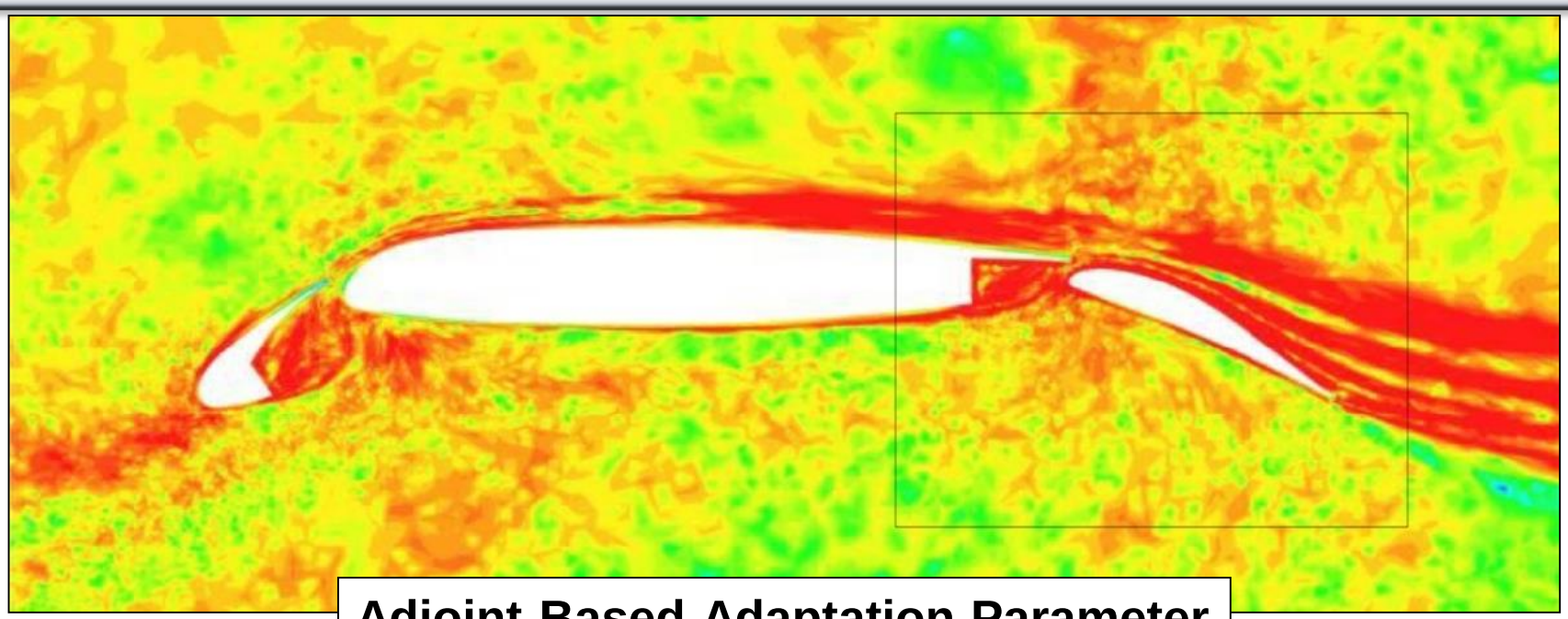


Adapted Mesh

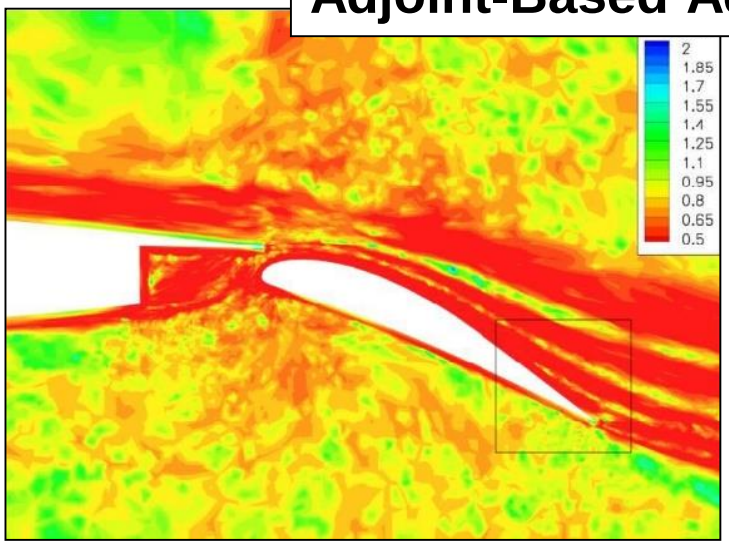


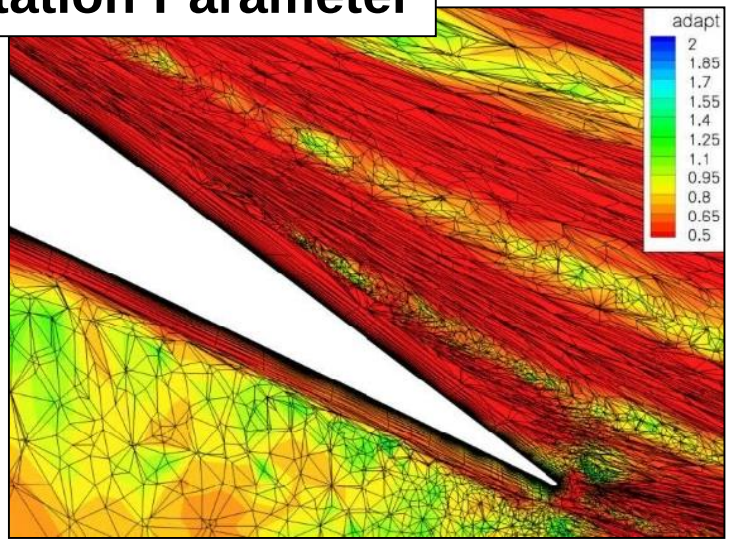
Mesh Adaptation for HLPW-1

Held at 2010 AIAA Summer Meeting in Chicago



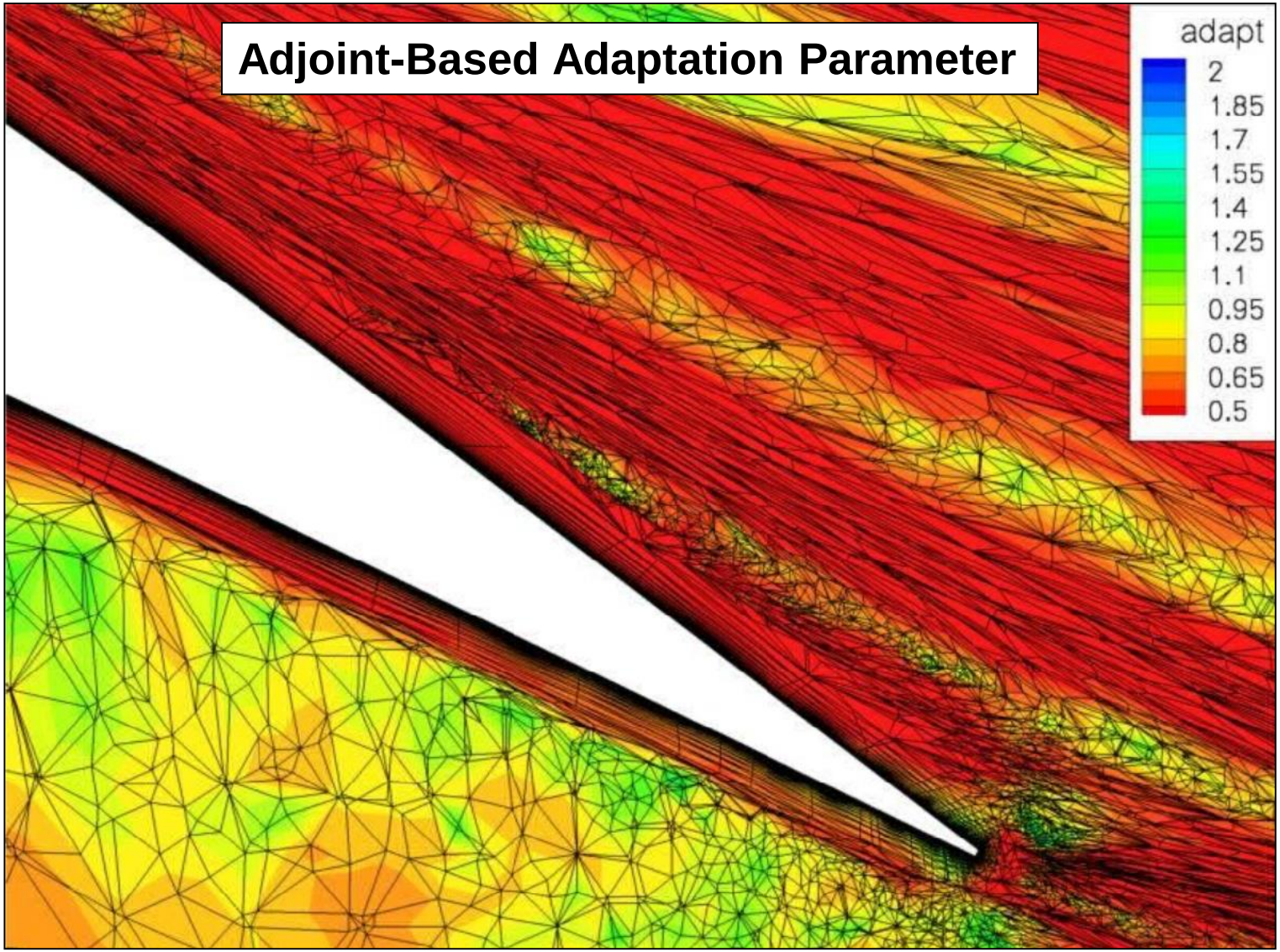
Adjoint-Based Adaptation Parameter





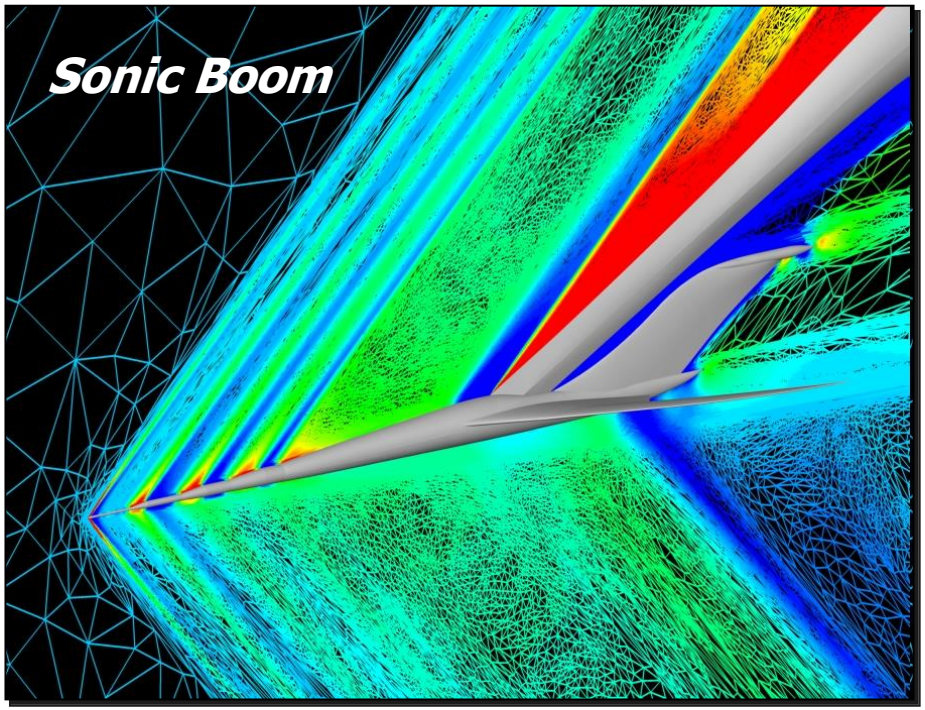
Mesh Adaptation for HLPW-1

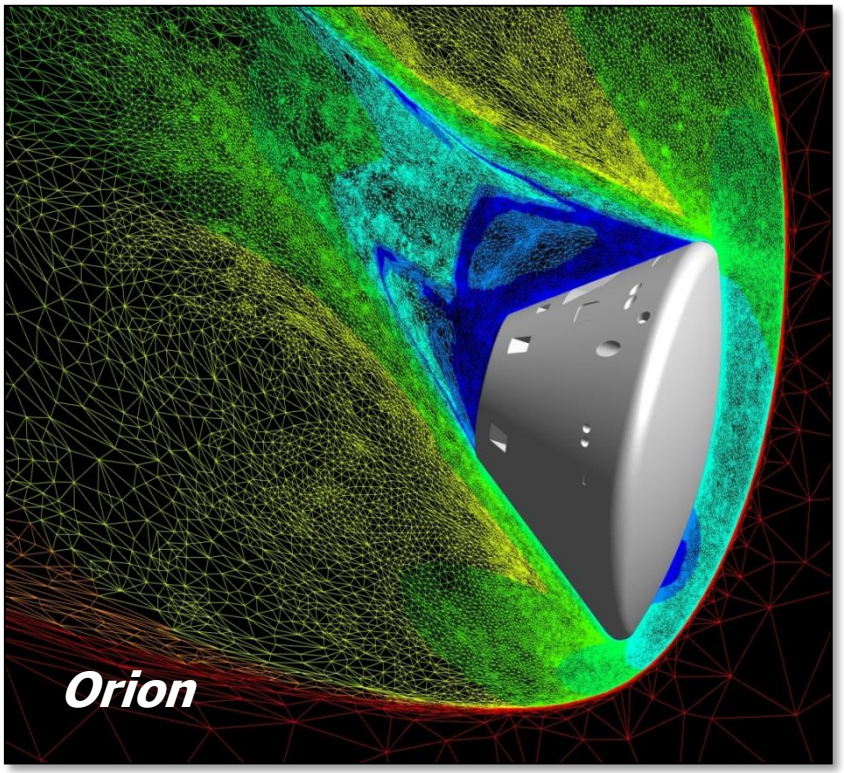
Held at 2010 AIAA Summer Meeting in Chicago

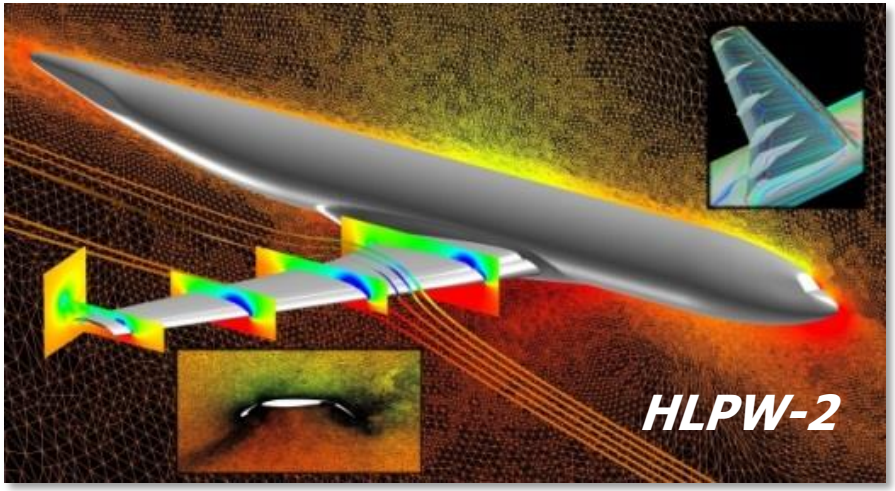


Adjoint-Based Mesh Adaptation

Other Applications

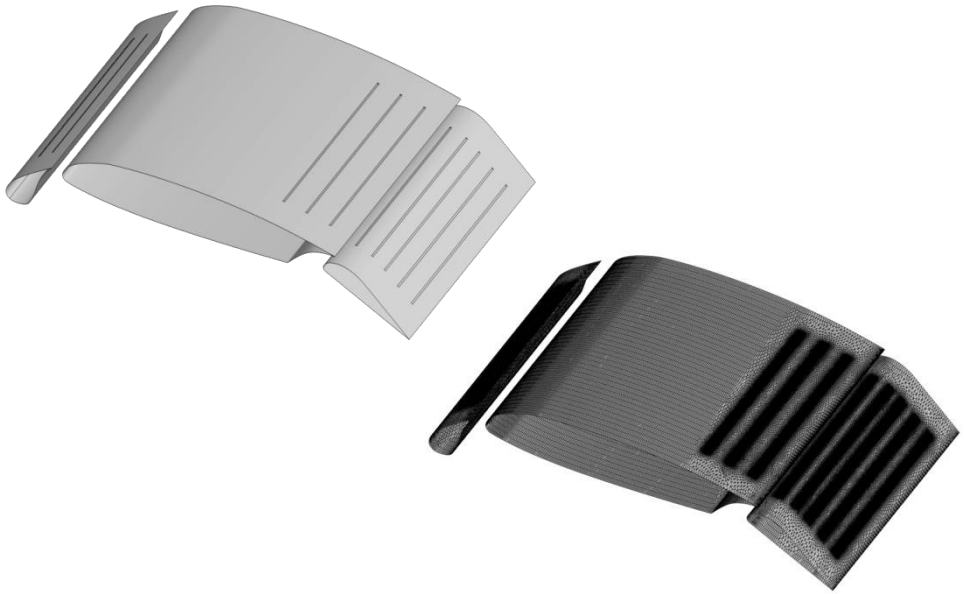




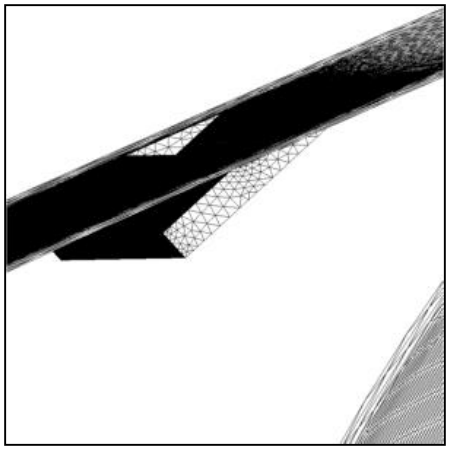


Design of AFC for High-Lift

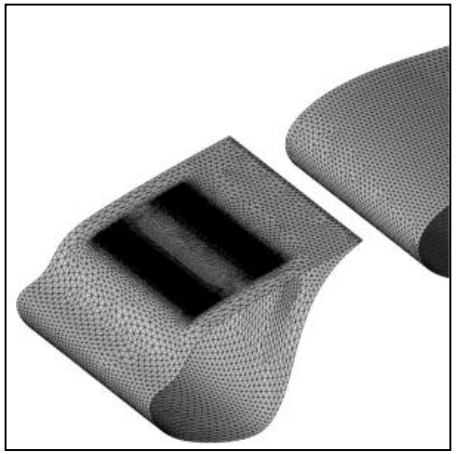
- Objective: Maximize lift using all available parameters
- Design variables include
  - External wing shape
  - Jet blowing parameters
  - Jet incidence and location
  - Relative location of slat/main/flap
- Designs performed using 2,048 cores for ~5 days per run
- Mean value of lift increased by 27%

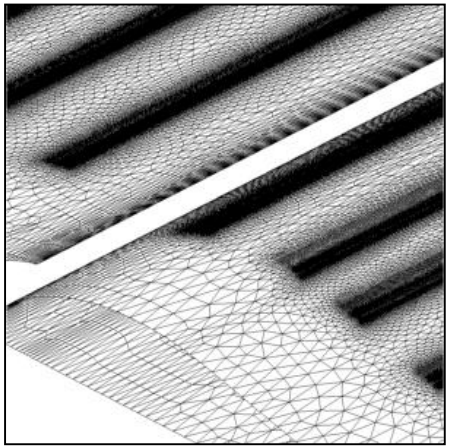


Jet Incidence

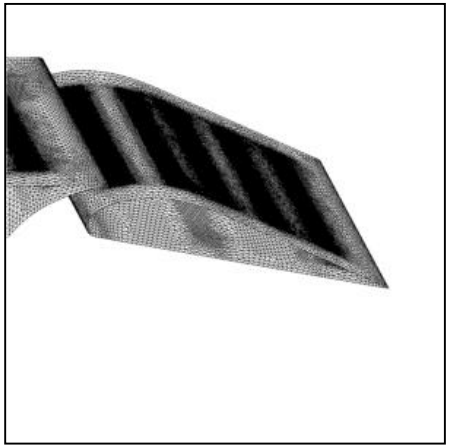


Shape Deformation



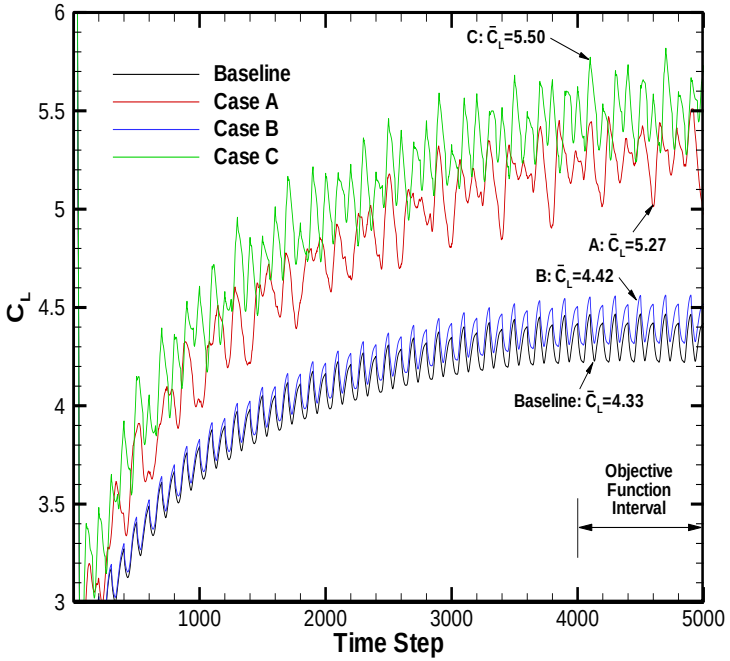


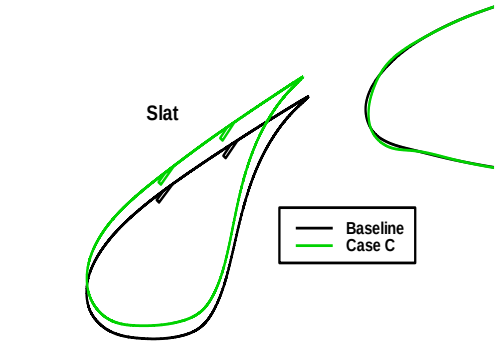
Jet Sliding

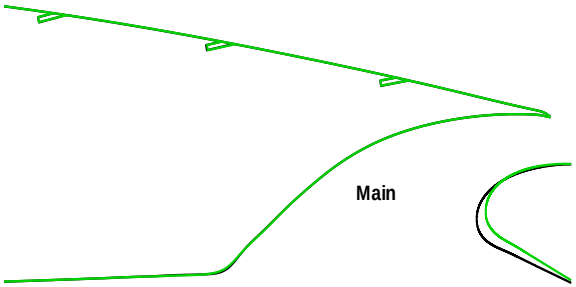


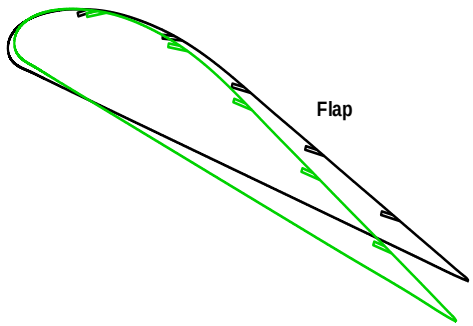
Relative Translation And Rotation

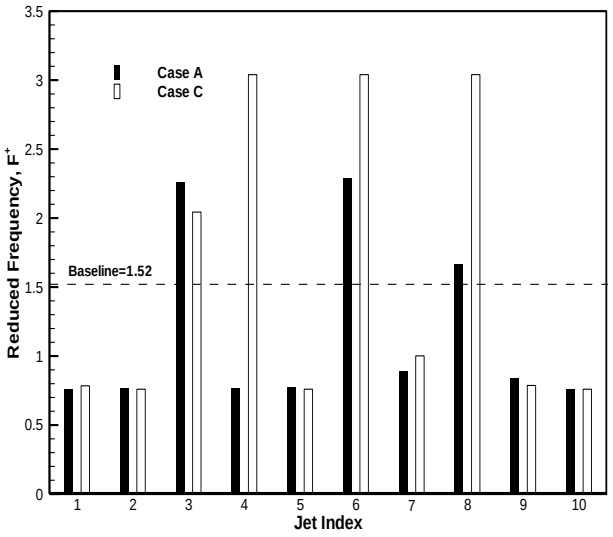
Design of AFC for High-Lift

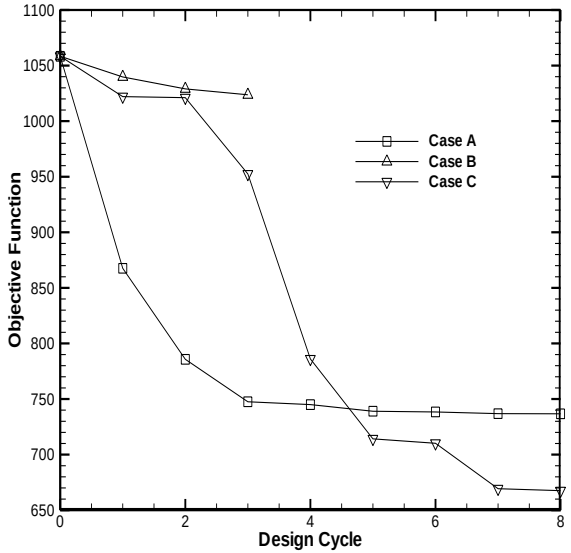






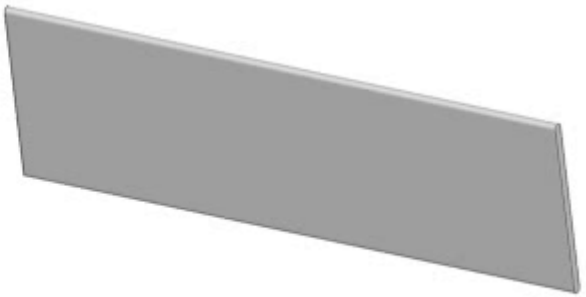






Biologically-Inspired Flapping Wing

Overview



- Simple wing geometry with kinematic motion based on Hawkmoth insect
  - Screen represents plane of symmetry
- Composite mesh totals 8,355,344 nodes / 49,088,120 tetrahedra
- Wing operates at 26 Hz in quiescent conditions with Re=1,280
- Governing equations: incompressible laminar N-S
- Kinematics consist of ±60º sweeping and ±45º feathering motions
- Net result of motion is a thrust force in the upward direction
- BDF2opt scheme run for 5 periods with 50 subiterations and 250 steps/period

Mesh, problem statement courtesy John Moore (MIT/AFRL)

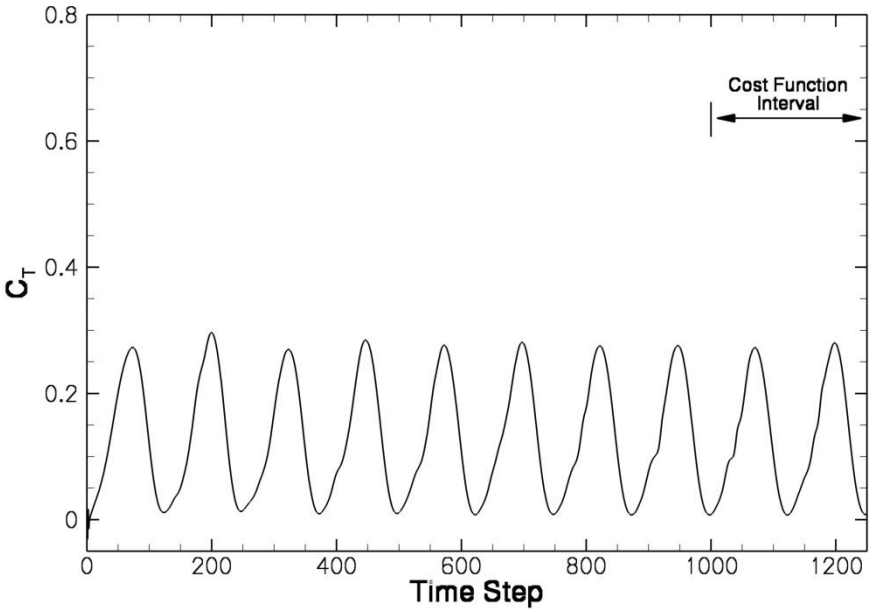
Biologically-Inspired Flapping Wing

Problem Definition

- Motion transform matrix $\mathbf{T}^n$ specified via user-defined kinematics interface:

$$\theta_x^n = A_x\left[\cos(\omega_{1x}t)-1\right]+B_x\sin(\omega_{2x}t) \Rightarrow \mathbf{T}_x^n$$
$$\theta_y^n = A_y\left[\cos(\omega_{1y}t)-1\right]+B_y\sin(\omega_{2y}t) \Rightarrow \mathbf{T}_y^n$$
$$\theta_z^n = A_z\left[\cos(\omega_{1z}t)-1\right]+B_z\sin(\omega_{2z}t) \Rightarrow \mathbf{T}_z^n$$

$$\mathbf{T}^n = \mathbf{T}_z^n\mathbf{T}_y^n\mathbf{T}_x^n$$



- Thrust profile shows 2/cyc behavior

Goal is to maximize $\bar{C}_T$ over final period using two different objective functions:

$$f = \sum_{n=1,001}^{1,250}\left(C_T^n - 5.0\right)^2 \Delta t$$

Distribution function

$$f = \left[\left(\frac{1}{250}\sum_{n=1,001}^{1,250} C_T^n\right) - 5.0\right]^2 \Delta t$$

Time-average function

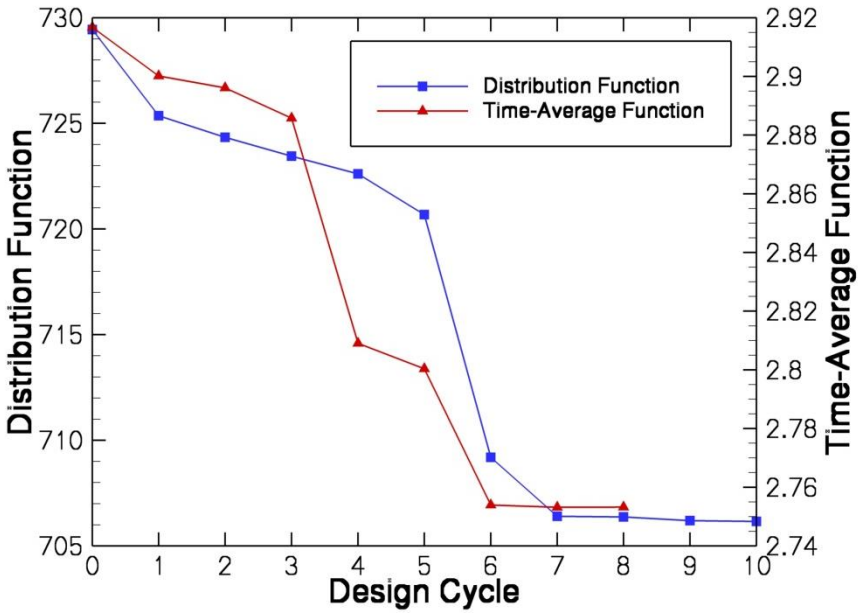
Design variables: 3 coords of rotation center, 12 kinematic parameters A, B, ω_1, ω_2

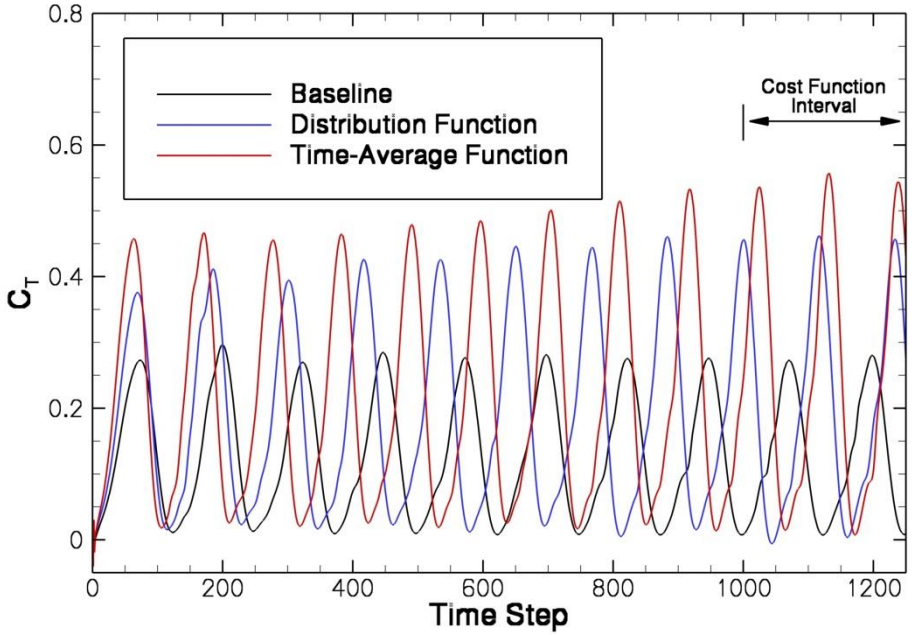
Biologically-Inspired Flapping Wing

Results

- Very moderate changes to all design variables
- Both designs now yield three peaks in cost interval
- Shape optimization using 88 parameters describing twist/shear/thickness/camber also attempted; opposing effects during sweeping negate improvements

	$\bar{C}_T$	Flow Solves (4 hrs)	Adjoint Solves (3 hrs)	Total Time
Baseline	0.127	-	-	-
Distribution Function	0.207	22	10	5 days (227,000 CPU hrs)
Time-average Function	0.265	25	8	5+ days (238,000 CPU hrs)

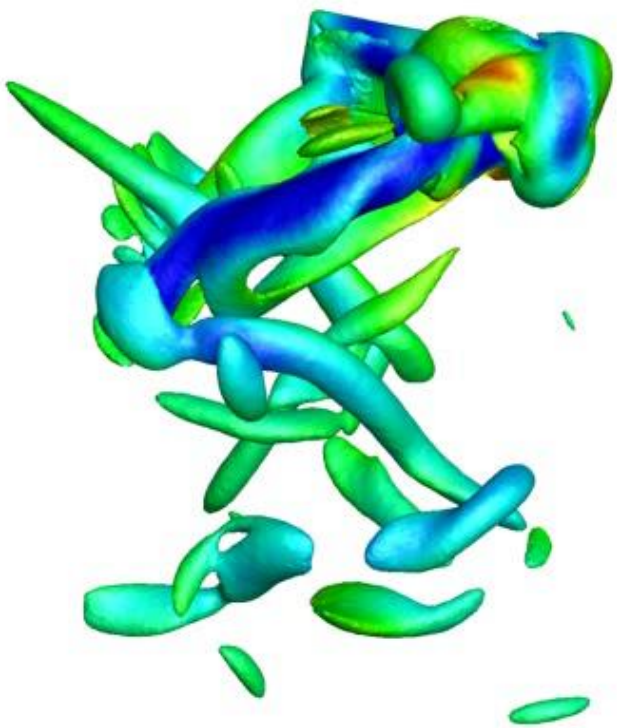




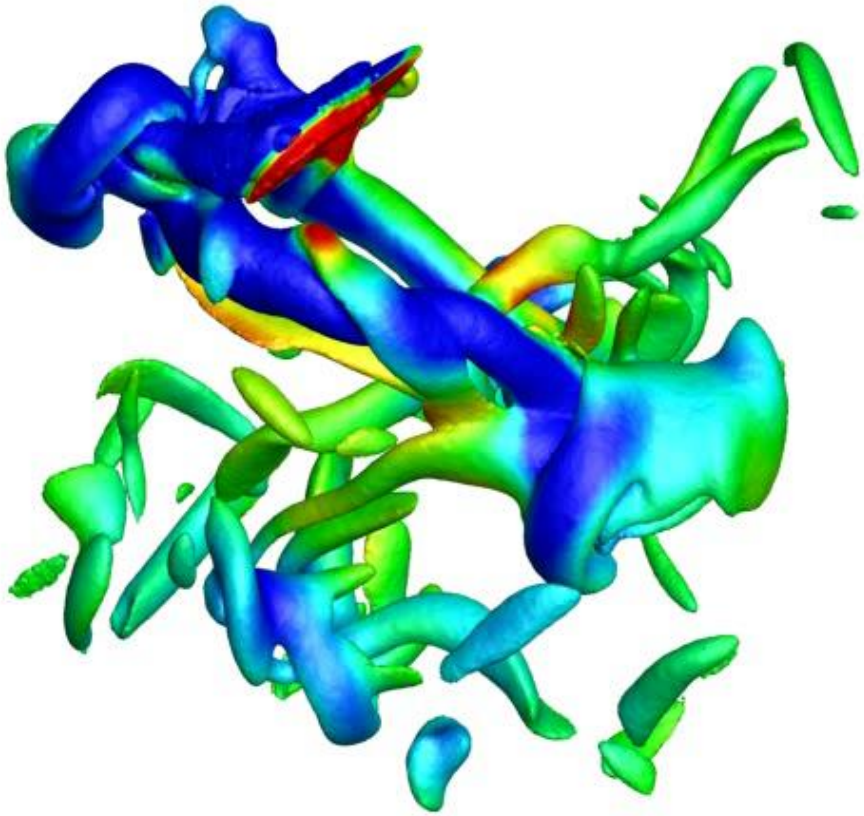
Biologically-Inspired Flapping Wing

Baseline and Optimal Flow Fields

Isosurfaces of Vorticity Colored by Pressure



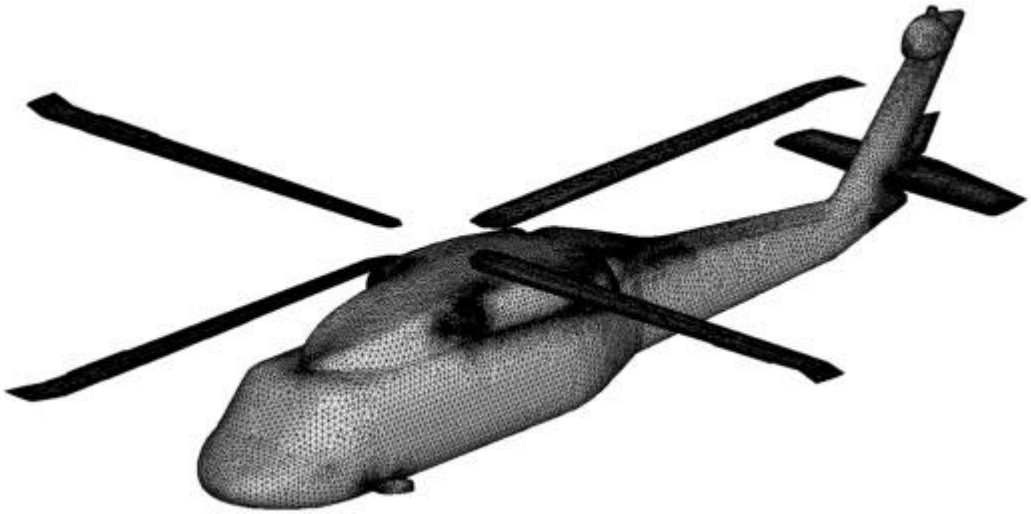
Baseline



Optimal
(Time-Average Function)

UH-60A Blackhawk Helicopter

Overview



- Composite grid consists of 9,262,941 nodes / 54,642,499 tetrahedra
- Compressible RANS: M_{tip}=0.64, Re_{tip}=7.3M, μ=0.37, α=0.0º
- Blade pitch has child motion governed by collective and cyclic control inputs:

$$\theta = \theta_c + \theta_{1c}\cos\psi + \theta_{1s}\sin\psi$$

Blade pitch — Collective — Lateral cyclic — Longitudinal cyclic

- Baseline value of all control inputs is zero

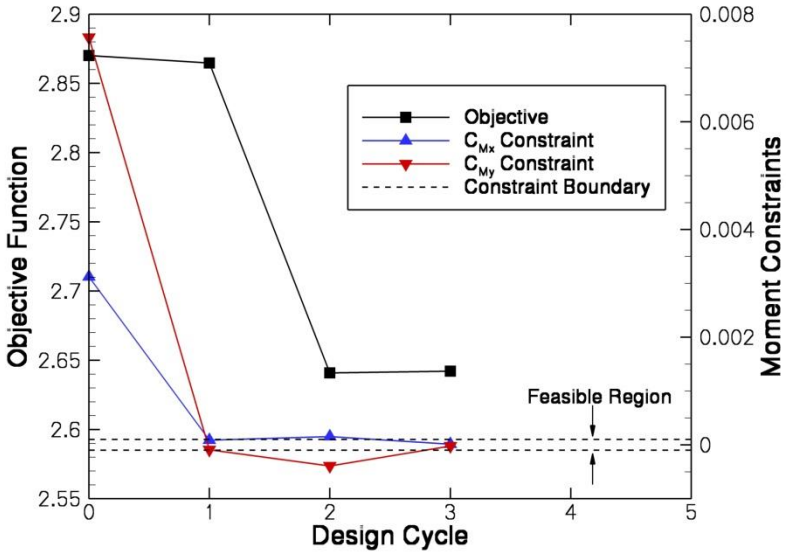
UH-60A Blackhawk Helicopter

Problem Definition and Results

- Objective is to maximize $\bar{C}_L$ while satisfying trim constraints over second rev:

$$\min f = \left[\left(\frac{1}{360}\sum_{n=361}^{720} C_L^n\right) - 2.0\right]^2 \Delta t \quad \text{such that} \quad g_1 = \frac{1}{360}\sum_{n=361}^{720} C_{M_x}^n \Delta t = 0 \qquad g_2 = \frac{1}{360}\sum_{n=361}^{720} C_{M_y}^n \Delta t = 0$$

- Separate adjoint solutions required for all three functions
- 67 design variables include 64 thickness and camber variables across the blade planform, plus collective and cyclic control inputs up to $\pm 7^o$

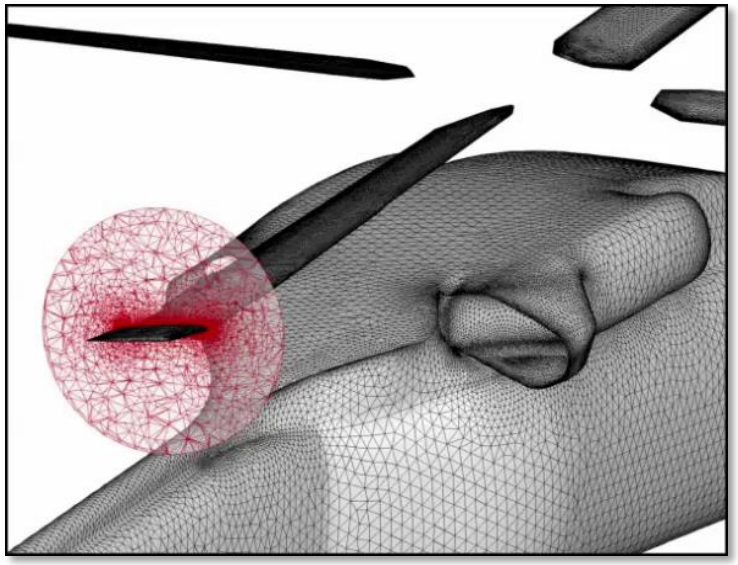


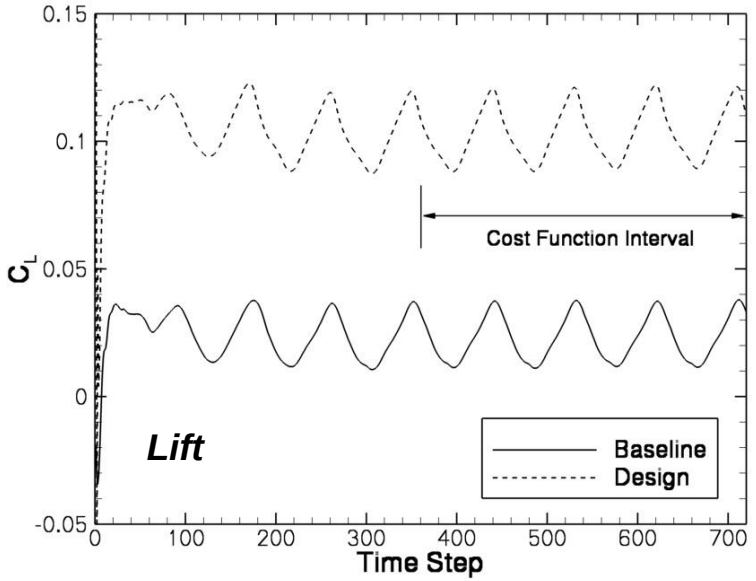
	$\bar{C}_L$	Flow Solves (2 hrs)	Adjoint Solves (3 hrs)	Total Time
Baseline	0.023	-	-	-
Design	0.103	4	4	0.8 days (38,400 CPU hrs)

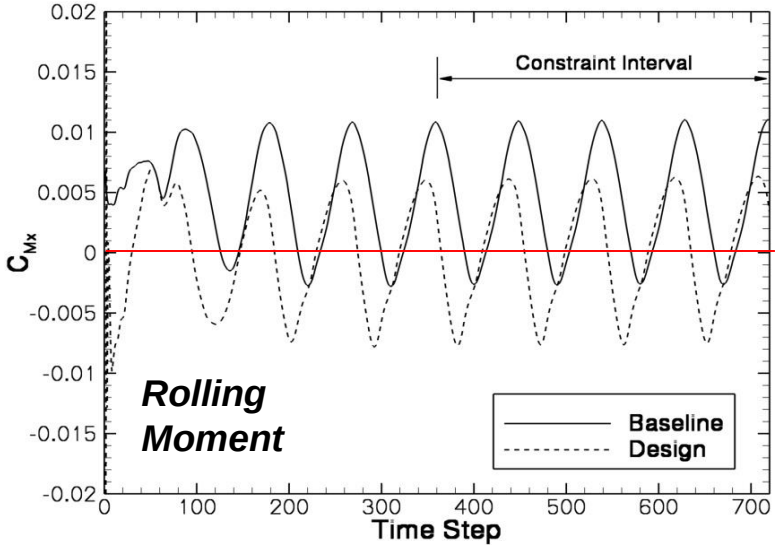
- Feasible region is quickly located
- Both moment constraints are satisfied within tolerance at the optimal solution
- Final controls: θ_c=6.71º, θ_{1c}=2.58º, θ_{1s}=-7.00º

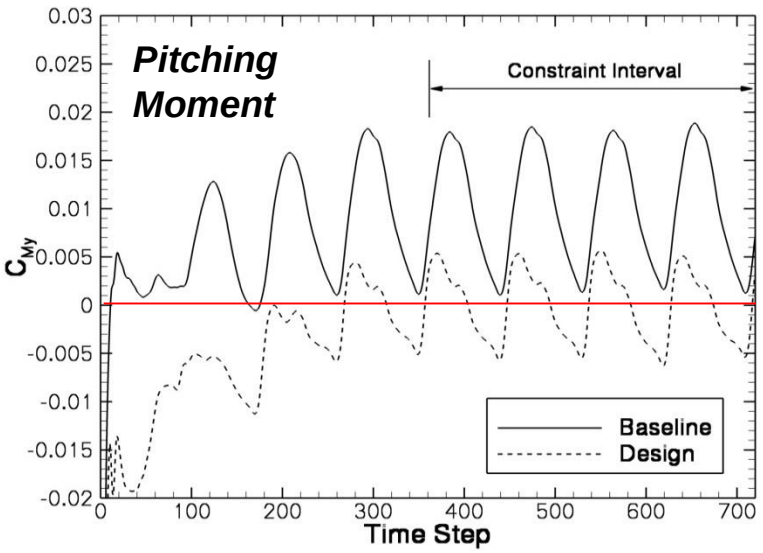
UH-60A Blackhawk Helicopter

Results



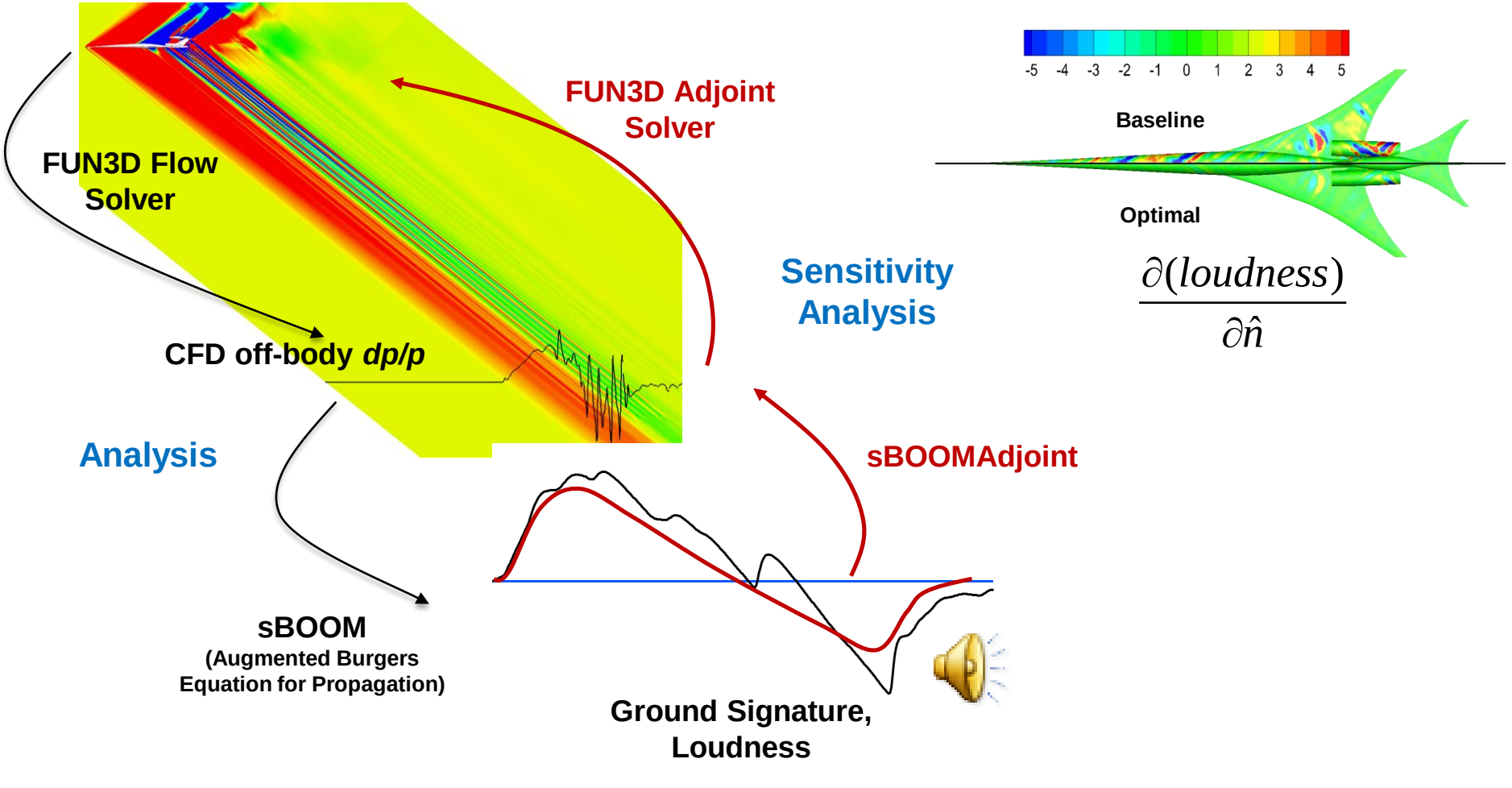






Multidisciplinary Design
Sonic Boom Mitigation

- Multidisciplinary discrete adjoint has been very successful for sonic boom mitigation - discrete derivatives of ground-based metrics with respect to OML
- Many other disciplines being considered / pursued



The Chaos Problem

Compute sensitivities of infinite time averages for chaotic flows (DES, HRLES, LES, etc)

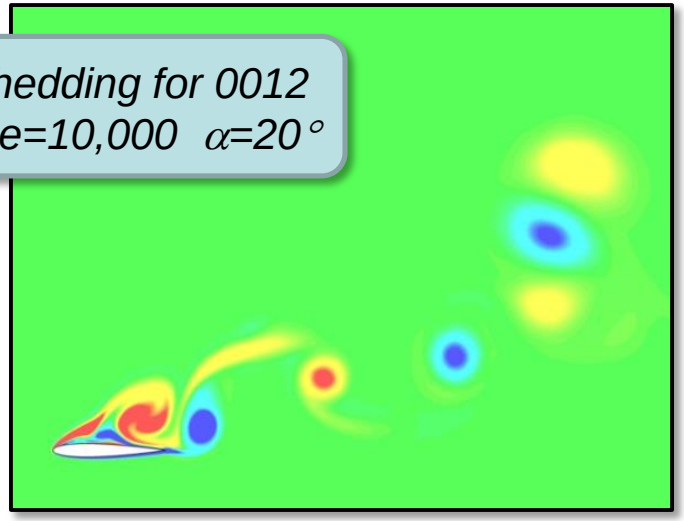
- Theory exists that states these sensitivities are well-defined and bounded

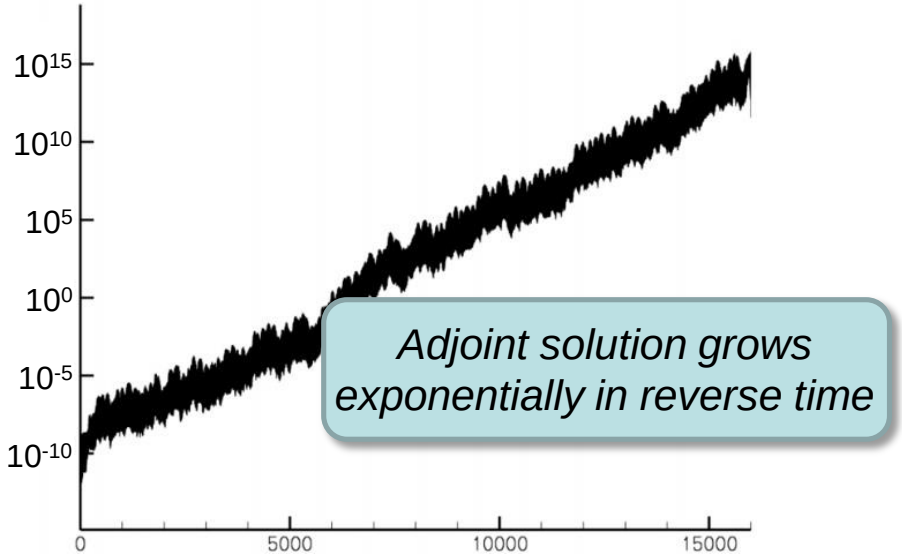
Why does conventional approach not work?

For chaotic flows:

- The finite time average approaches the infinite time average
- The sensitivity for a finite time average does not approach the sensitivity for the infinite time average

Chaotic shedding for 0012
M_∞=0.1 Re=10,000 α=20°





Adjoint solution grows exponentially in reverse time

Approach

- Least-Squares Shadowing (LSS) method proposed by Wang and Blonigan
  - Key assumption is ergodicity of the simulation: long time averages are essentially independent of the initial conditions
  - Also assumes existence of a shadowing trajectory
- The LSS formulation involves a linearly-constrained least squares optimization problem which results in a set of optimality equations
- Preliminary LSS exploration for fluids applications

Define the following quantities:

$\boldsymbol{Q}_i \equiv$ Vector of conserved variables at time level i

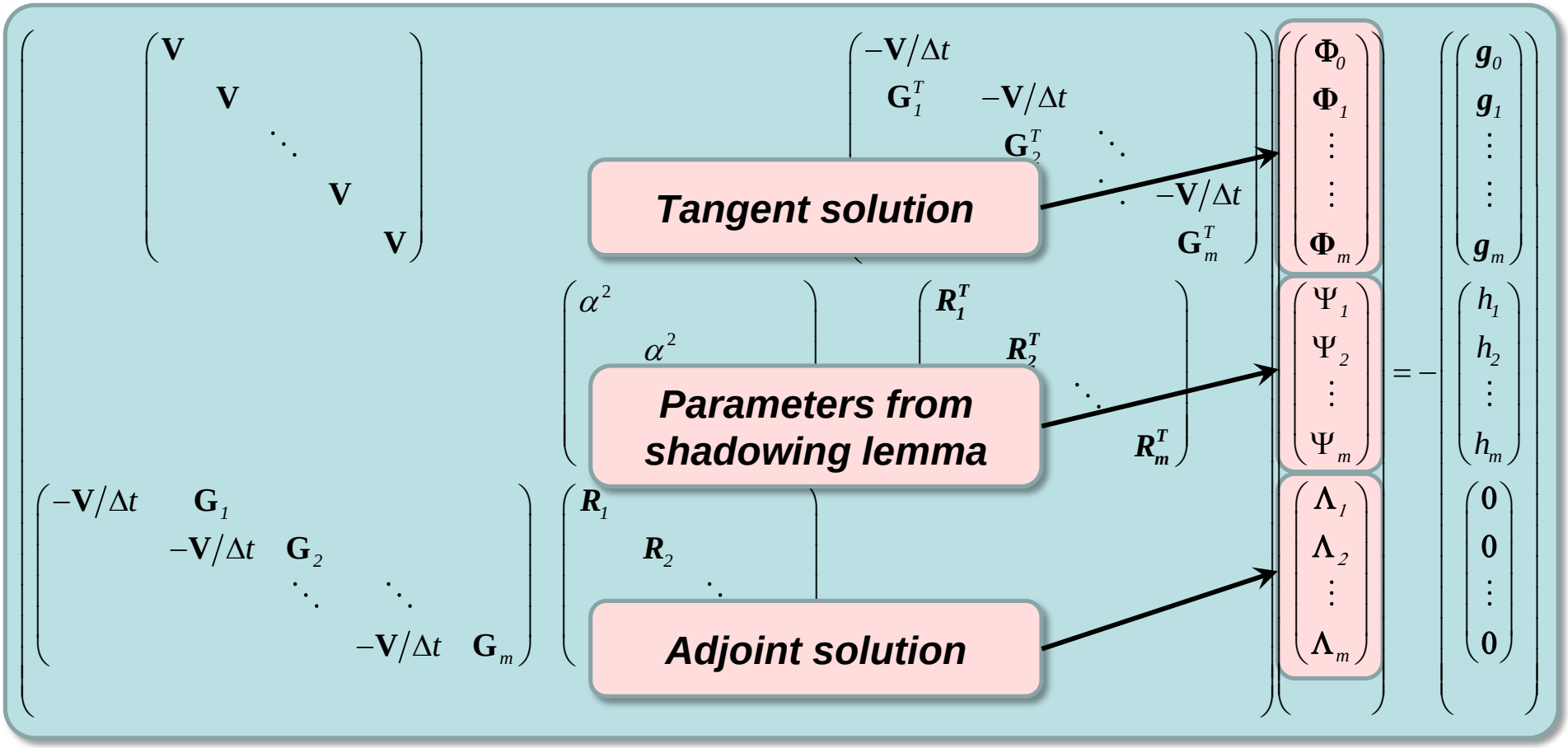
$\boldsymbol{R}_i \equiv$ Vector of spatial residuals at time level i

$\mathbf{V} \equiv$ Matrix of cell volumes

$t \equiv$ Time

$f_i \equiv$ Objective function at time level i

LSS System



$$\mathbf{G} = \mathbf{V}/\Delta t + \partial \boldsymbol{R}/\partial \boldsymbol{Q}$$

$$\boldsymbol{g} = \partial f/\partial \boldsymbol{Q}$$

$\boldsymbol{h}$ is related to time dilation

α is a regularization parameter

This is a globally coupled space-time problem, where each sub-row represents a time level

Reduced LSS System

- To determine sensitivities, we need the LSS adjoint solution
- Use a Schur complement approach to arrive at a reduced system for the LSS adjoint variables:

Writing the previous system as

$$\begin{pmatrix} \mathbf{V} & 0 & \mathbf{B}^T \\ 0 & \alpha^2\mathbf{I} & \mathbf{C}^T \\ \mathbf{B} & \mathbf{C} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Psi} \\ \boldsymbol{\Lambda} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{h} \\ \mathbf{0} \end{pmatrix}$$

The LSS adjoint solution can be determined from

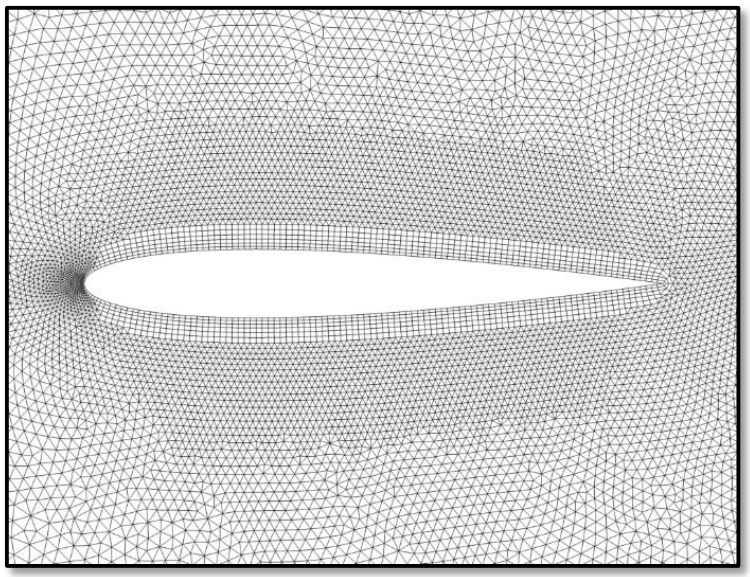
$$\left[\mathbf{B}\mathbf{V}^{-1}\mathbf{B}^T + \frac{1}{\alpha^2}\mathbf{C}\mathbf{C}^T \right] \boldsymbol{\Lambda} = -\mathbf{B}\mathbf{V}^{-1}\boldsymbol{g} - \frac{1}{\alpha^2}\mathbf{C}\boldsymbol{h}$$

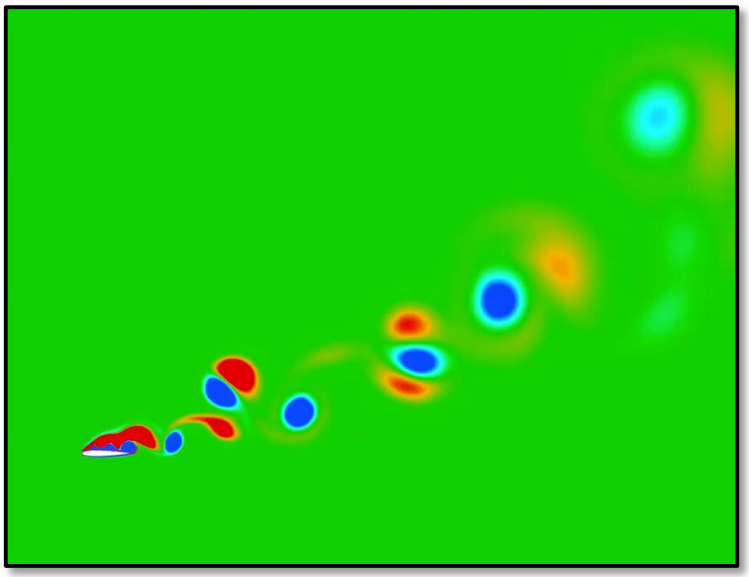
- This remains a globally coupled space-time problem
- $\mathbf{B}\mathbf{B}^T$ increases the fill of the matrix
- Furthermore, the system is dense due to $\mathbf{C}\mathbf{C}^T$ term

Problem Definition

Shedding NACA 0012

M_∞=0.1 Re=10,000 α=20°

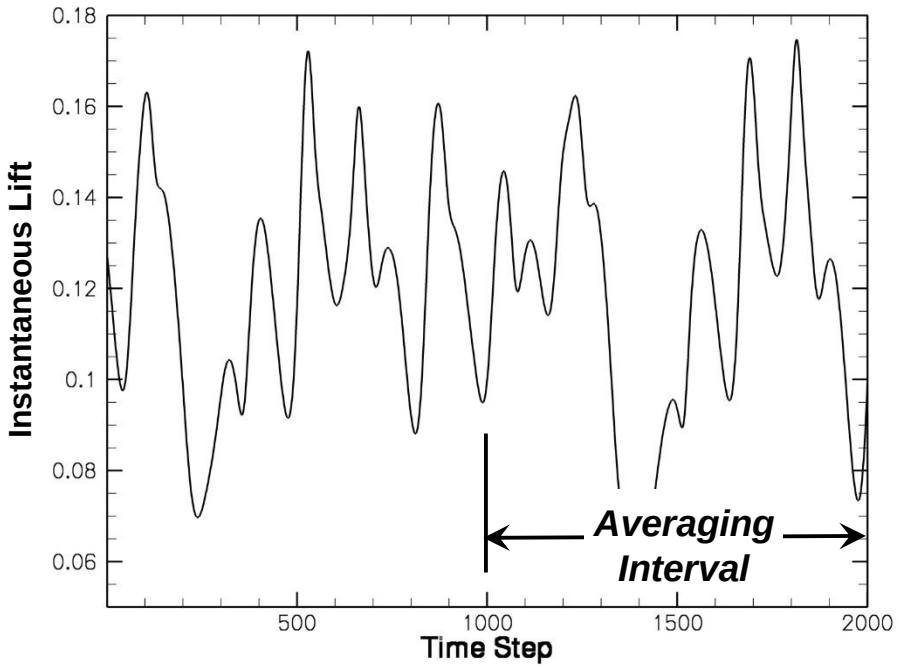




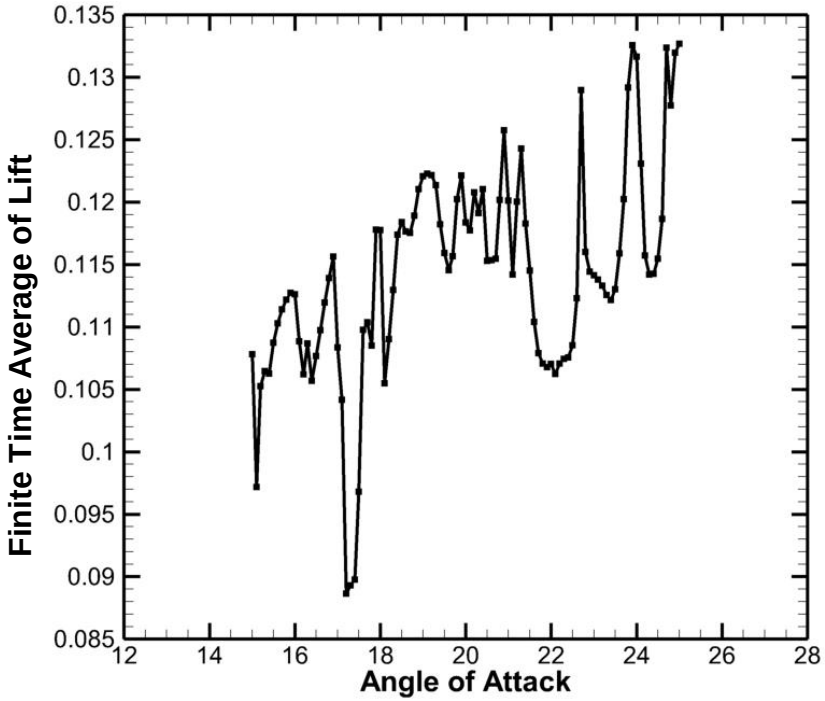
- Unstructured mesh consisting of 102,940 grid points with 100,139 prisms and 1,144 hexes in spanwise direction
- Relatively coarse wall spacing to alleviate stiffness in LSS system
- Laminar Navier-Stokes equations with second-order spatial discretization
- First-order backward differencing in time for LSS simplicity

Problem Definition

Instantaneous Lift vs Time



Finite Time Average of Lift vs Alpha



- To improve ergodicity, simulation started from chaotic initial solution (generated by 2,000 steps from free stream)
- Objective is to maximize time-averaged lift over final 1,000 time steps

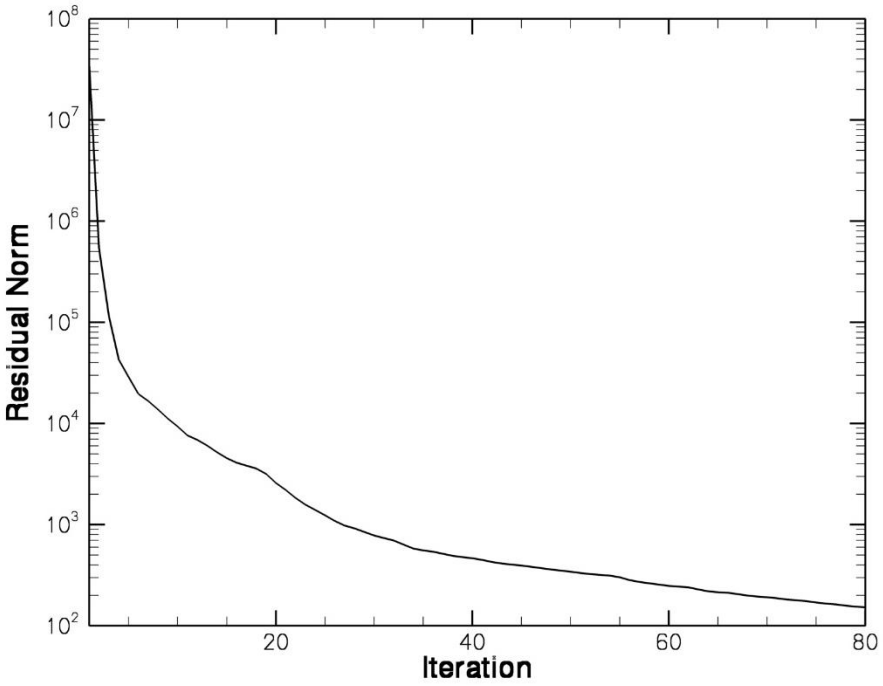
Approach

- Execute FUN3D flow/adjoint solvers to output data to disk for use in LSS: nonlinear residual vectors and Jacobians of residual and objective function
- For this tiny problem, the raw dataset is ~1.1 TB (in-core requirement much larger)
- Developed standalone LSS solver, where partitioning is performed in time with a single time plane per core
  - For simplicity, the spatial discretization fits on a single core
  - Spatial decompositions are possible but not considered here
- Global GMRES solver used with a local ILU(0) preconditioner for each time plane, with $\mathbf{C}\mathbf{C}^T$ term neglected in preconditioner
- Execution was constrained to a subset of the cores available on each 128 GB Haswell node to provide sufficient memory for solving the LSS adjoint system
- Checked discrete consistency of LSS implementation using complex variables
- This complex variable test does not provide the same rigor for LSS as for conventional adjoint implementations; additional verification approaches needed

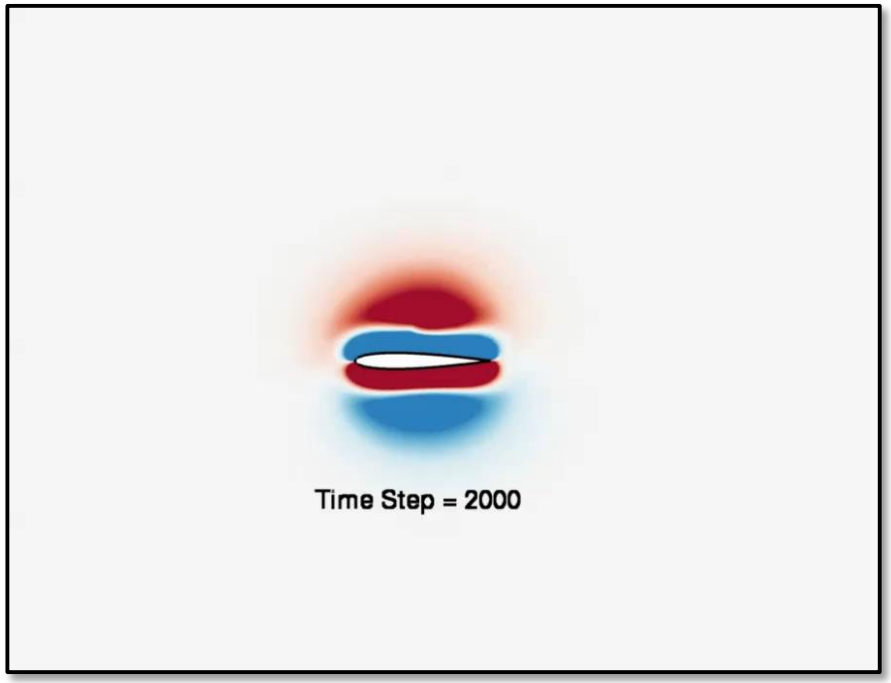
Solution of LSS Adjoint System

- After ~30 minutes for I/O, solution converges 5 orders of magnitude in ~30 mins on 2,000 cores
- Solution remains bounded
- Just tip of the iceberg – how will we extend to billions of mesh points with millions of time steps?

Convergence of LSS Adjoint System



LSS Adjoint Solution for Energy Equation



Thank You For Having Me!

- Always interested in new collaborations – need help in numerous areas
- Please do not hesitate to get in touch about opportunities at Langley

Eric.J.Nielsen@nasa.gov

http://fun3d.larc.nasa.gov