Applications and Research Efforts Using FUN3D in HPC Environments

Eric J. Nielsen and Dana P. Hammond
FUN3D Development Team
NASA Langley Research Center

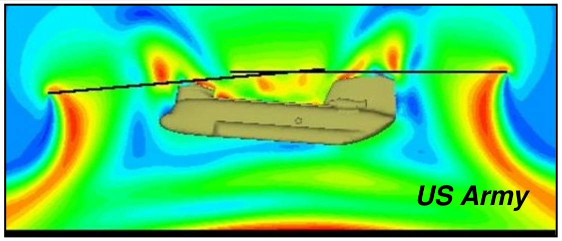
HPC User Forum
September 14, 2010

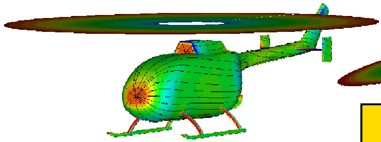
http://fun3d.larc.nasa.gov

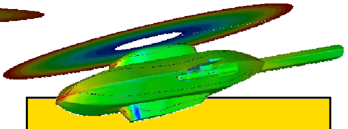
FUN3D Core Capabilities

http://fun3d.larc.nasa.gov

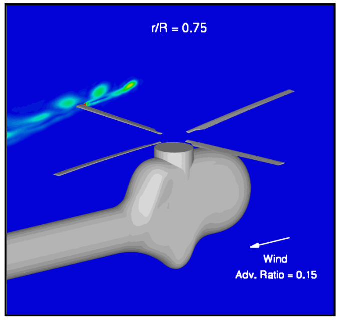
- Established as a research code in late 1980's; now supports numerous internal and external efforts across the speed range
- Solves 2D/3D steady and unsteady Euler and RANS equations on node-based mixed element grids for compressible and incompressible flows
- General dynamic mesh capability: any combination 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 and performance testing
- Capabilities fully integrated, very responsive support team, online documentation / training videos / tutorials

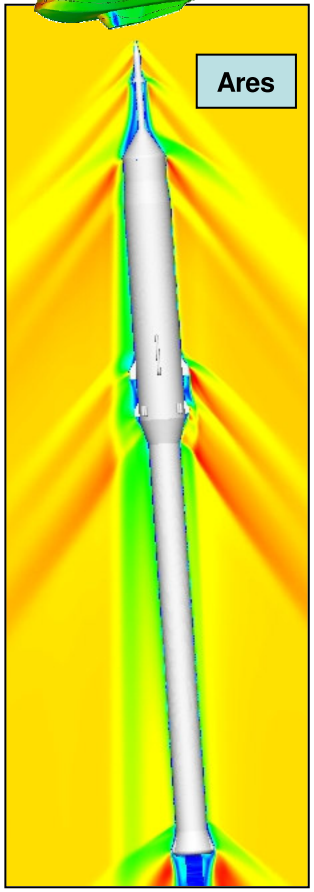




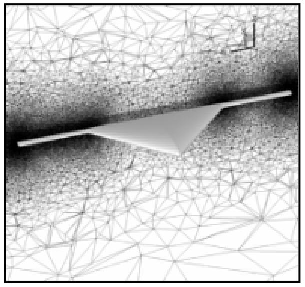


Rotorcraft

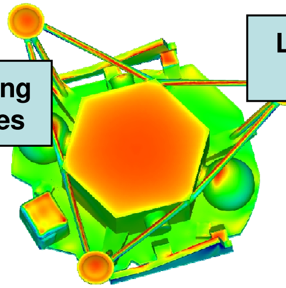




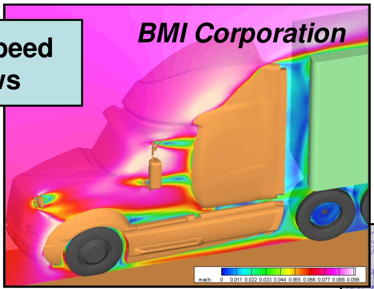
Ares

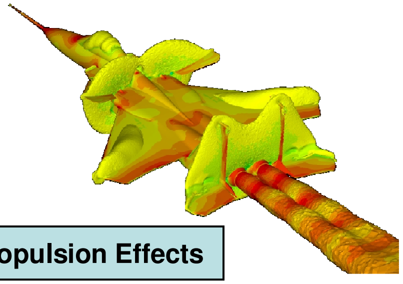


Morphing Vehicles

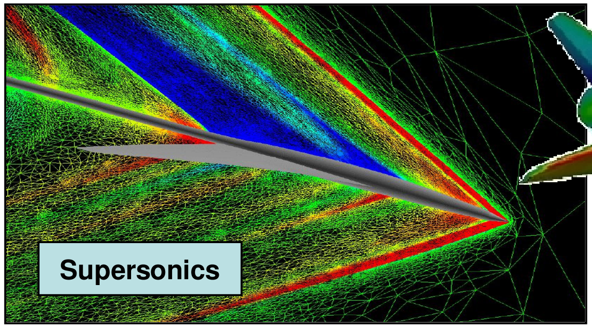


Low-Speed Flows

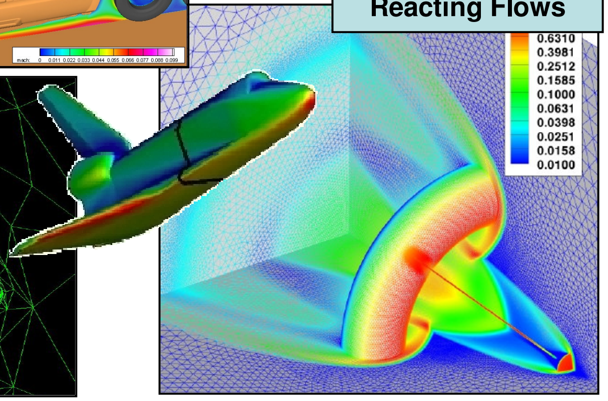




Propulsion Effects



Supersonics



Reacting Flows

User Base

Internal

- Widely used within NASA for projects across the speed range
  - Both engineering and research applications
- Users routinely running on several thousand cores

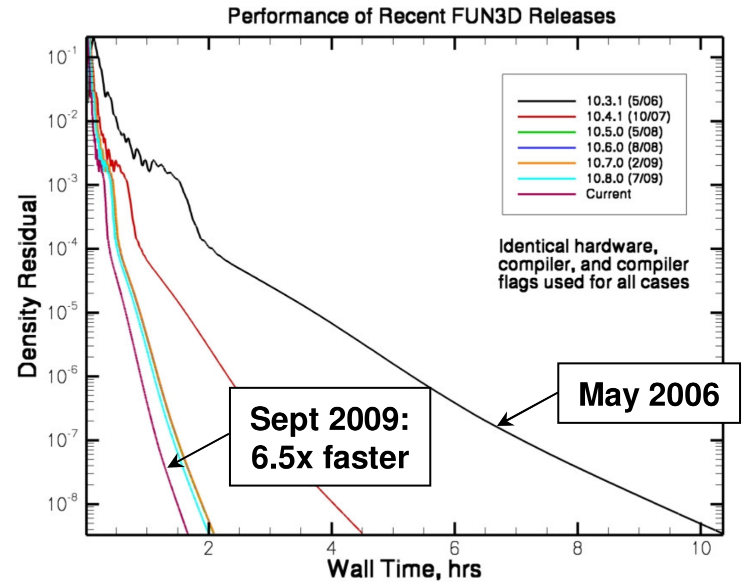
External

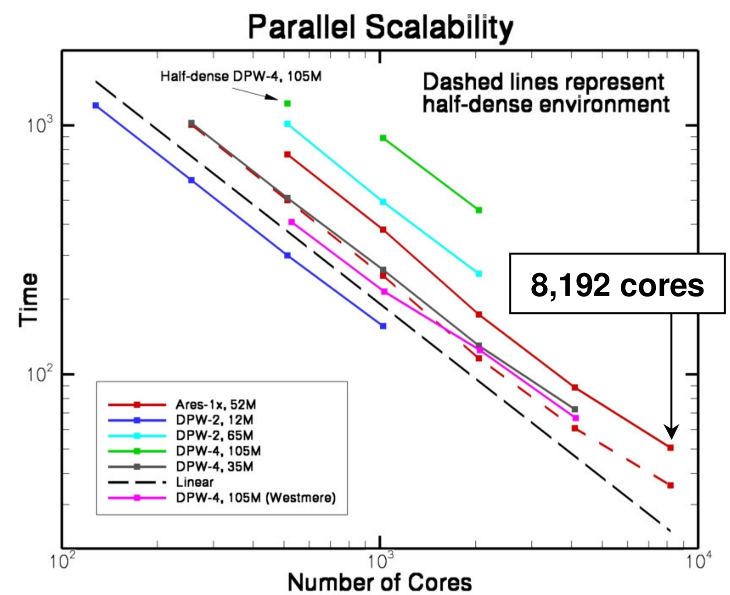
- Distributed to hundreds of individuals and organizations across academia, industry, DoD, and OGA
  - Export-controlled; source code freely available to any US person
  - Average about 100 distributions / year
  - Wide range of uses including aerospace, automotive, HPC, etc.
    - Some problems are as complex, if not more so, than NASA's
- Wide range of hardware being used
  - From RC enthusiasts on single workstation to groups generating matrices of hundreds of solutions on thousands of HPC cores
  - Majority of external users have $O(100)$ cores available to run on; this is steadily rising

Development team fields numerous questions per day; frequent feedback influences many development efforts

HPC Aspects of FUN3D

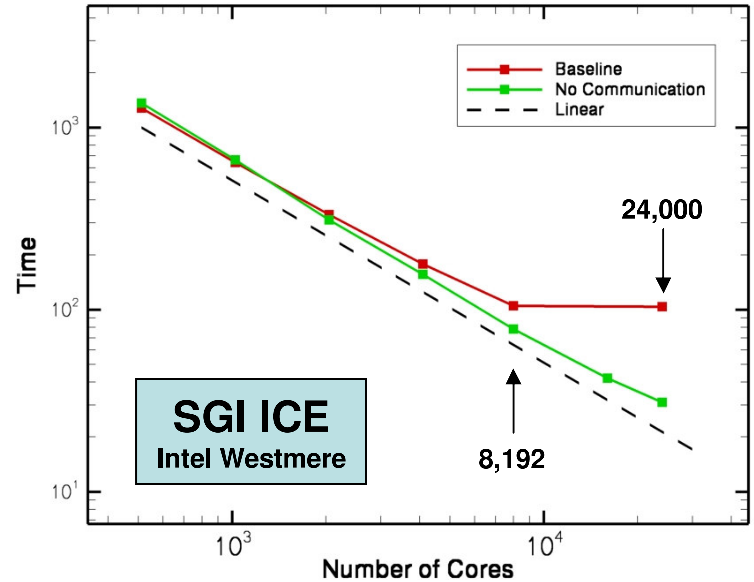
- FUN3D/PETSc effort won 1999 Gordon Bell Prize
- Recently completed multi-year optimization effort
- End-to-end parallelization nominally complete
  - Preprocessing: grid with 105M points / 600M elements previously required 2 weeks and 800 GB of shared memory; now done in 5 minutes using 1,024 cores with distributed memory
  - Exporting of solution visualization data
- Parallel I/O using MPI-IO layer or native implementation based on Fortran 2003 stream I/O
- Some components recently ported to multi-CPU/GPU environment
  - Performance limited by unstructured data and need for communication
  - Need access to large-scale hardware for further work
- Some experience with hierarchical partitioning strategies; need to revisit with higher cores/node
- Solver scales well to ~10,000 cores on SGI ICE
  - Evaluating/improving at higher core counts now
  - Memory scaling successfully tested through 32,768 cores
  - Tough to routinely get more than ~4,000 cores with current NASA scheduling paradigm
  - Working with ORNL using jaguar; very open to other collaborations

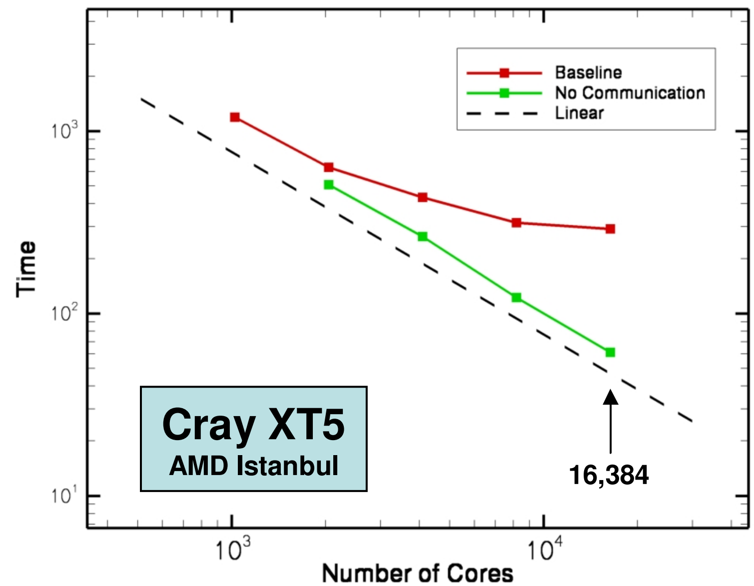




Scaling at Higher Core Counts

- Preliminary results shown for SGI ICE (NASA's pleiades) and Cray XT5 (ORNL's jaguar)
- Red curves indicate stock solver; green curves indicate communication disabled
- Results are for 105 million grid points: only ~6,400 local grid points at 16,384 cores
- Load balancing at high core counts is challenging for ParMetis
  – Robustness issues calling from > 4,000 cores
  – Yields higher quality partitions calling from far fewer cores (32 used here)
  – Results with no communication indicate reasonable work-balance
  – Communication is the question: could be partitioning problem (many edge-cuts and/or poor topology) and/or MPT tuning
- Poor scaling on XT5 has been observed in previous cases
  – Very sensitive to MPT tuning; needs to be investigated further
- Attempted up to 48,000 cores; selecting tuning parameters for MPI implementation is the limitation
- Collective communication issues are likely in play
- Plan to examine other communication strategies
- Examined hyperthreading; little perceived benefit
- Have also studied scaling of massively parallel I/O





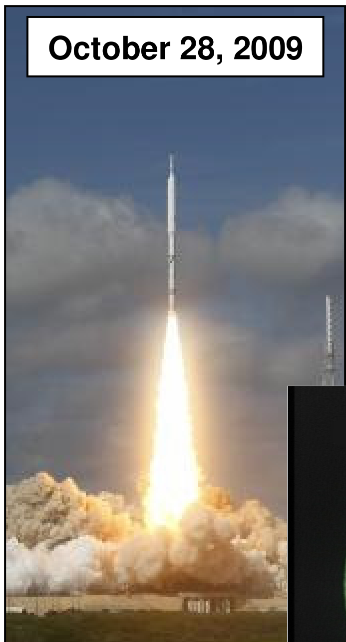
NASA Applications

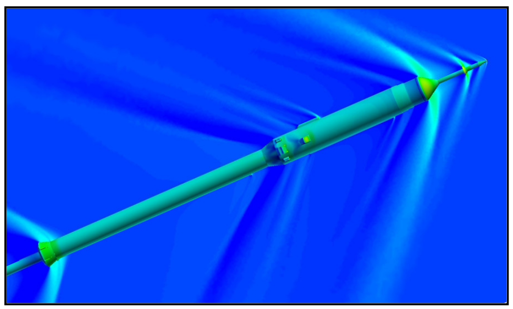
Ares 1-X Launch Vehicle

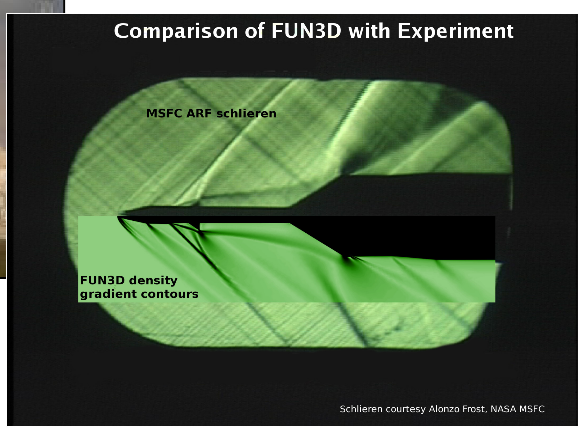
- FUN3D used for full-vehicle ascent aero and unsteady aeroelastic analysis, ground wind loads
- Millions of core-hours used over 2-3 years

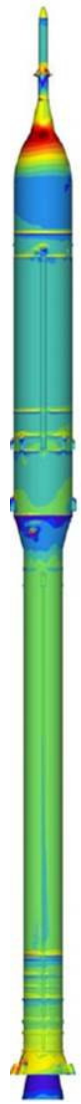
Supersonic Retropropulsion

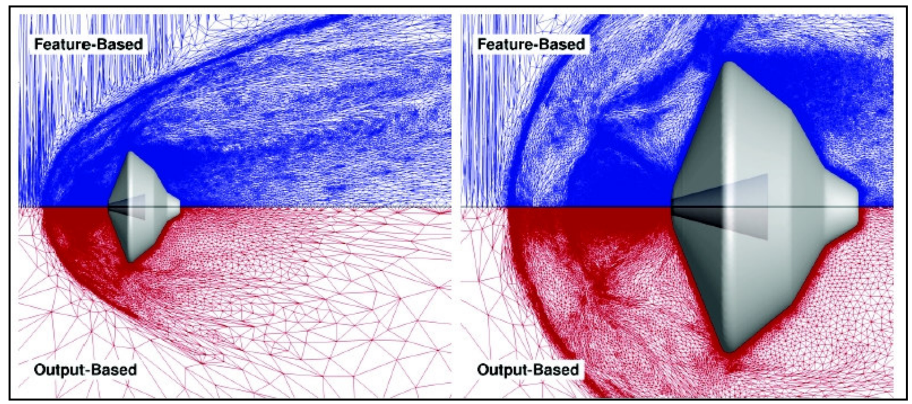
- Decelerator option being explored for entry at Mars with heavy payloads
- Hundreds of solutions on thousands of processors supporting ongoing experiments

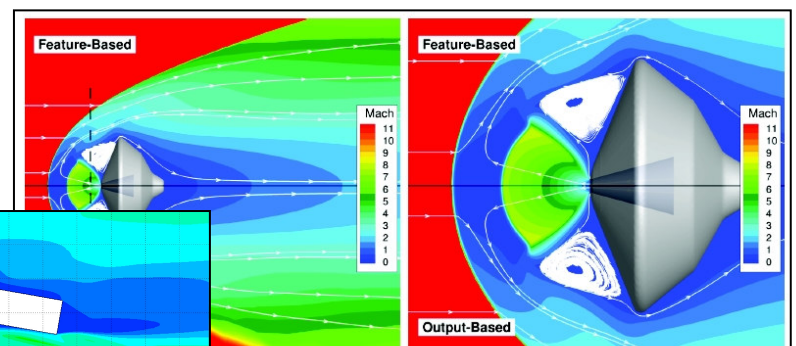


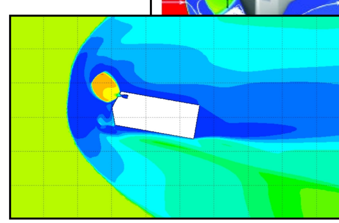








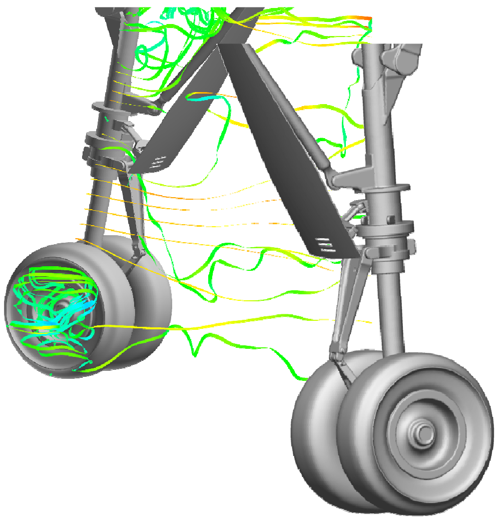




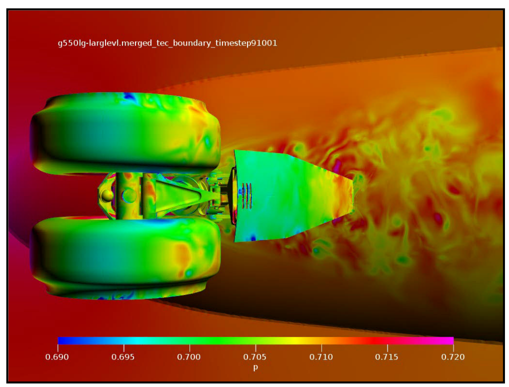
NASA Applications

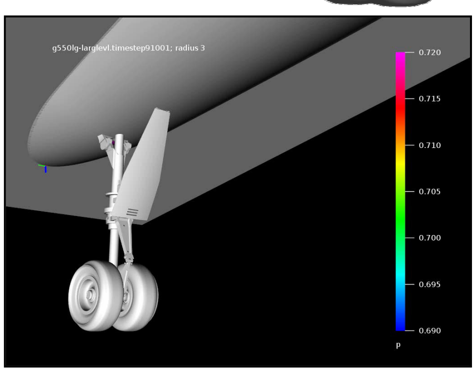
Airframe Noise: Landing Gear

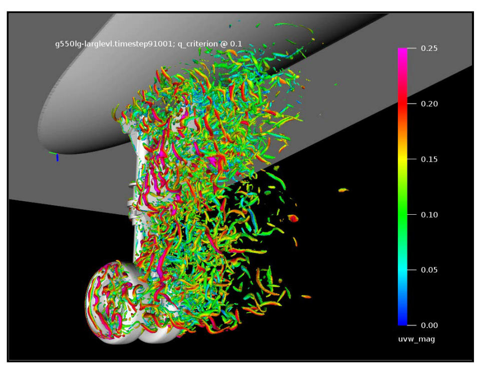
- One simulation takes ~2 months on 1,024 cores
- Exploring effects of discretization, turbulence modeling, and grid refinement



Courtesy NASA/Gulfstream Partnership Effort on Airframe Noise Research

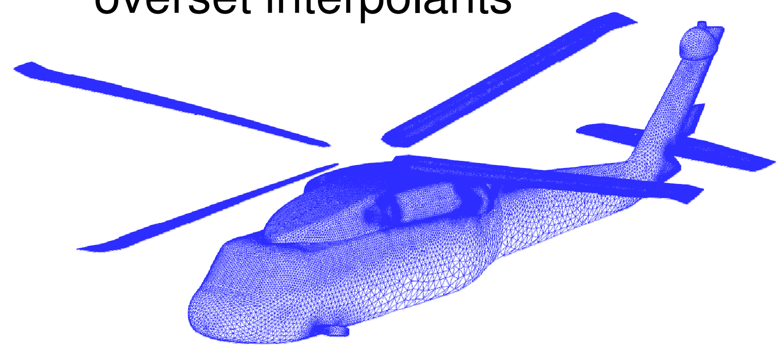


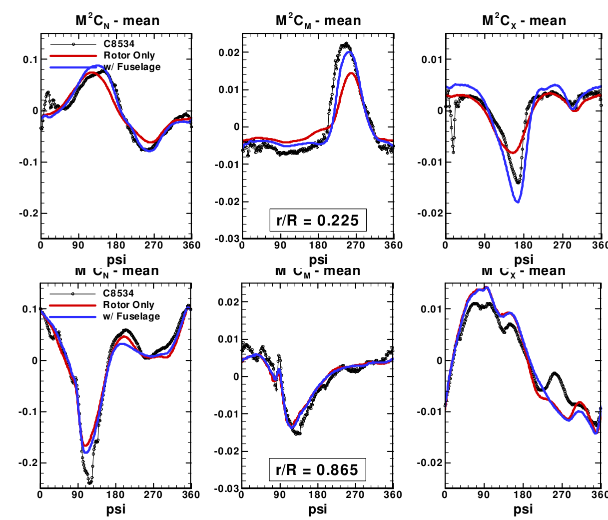




Rotorcraft

- Fully-coupled & trimmed aeroelastic simulations
- Exercises rigidly-moving, deforming, and overset grids
- Scaling limited by computation of overset interpolants

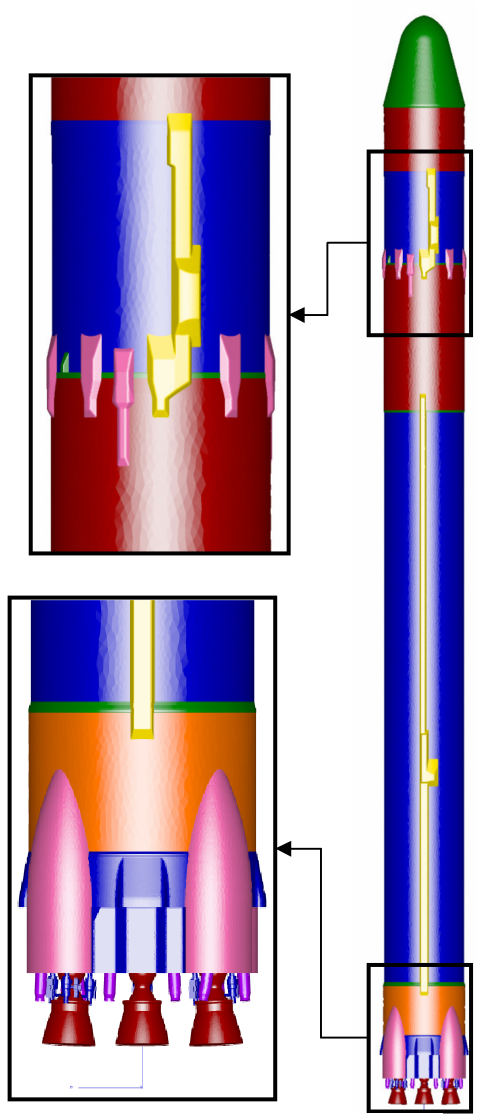




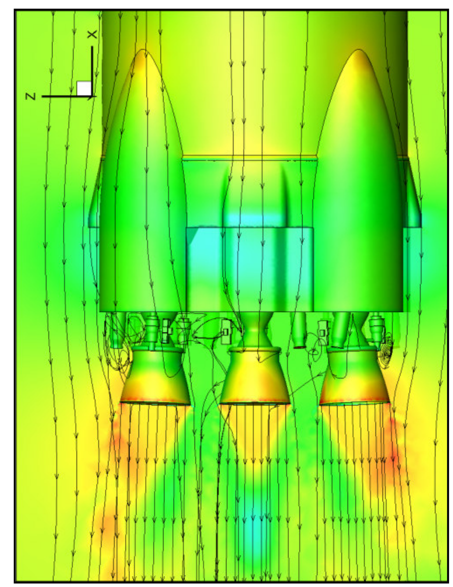
Industry: Vehicles at SpaceX

- Using FUN3D as primary CFD tool for:
  - Falcon 1 ascent aero
  - Falcon 9 ascent aero
  - Lower speed Dragon reentry aero
- Full, detailed vehicle models, including up to 18 plumes
- Performing hundreds of simulations per vehicle across the flight envelope
- CFD predictions agree very well with all flight and wind tunnel data
- Typically use several hundred processors on a local cluster system

Images and Information Courtesy of SpaceX

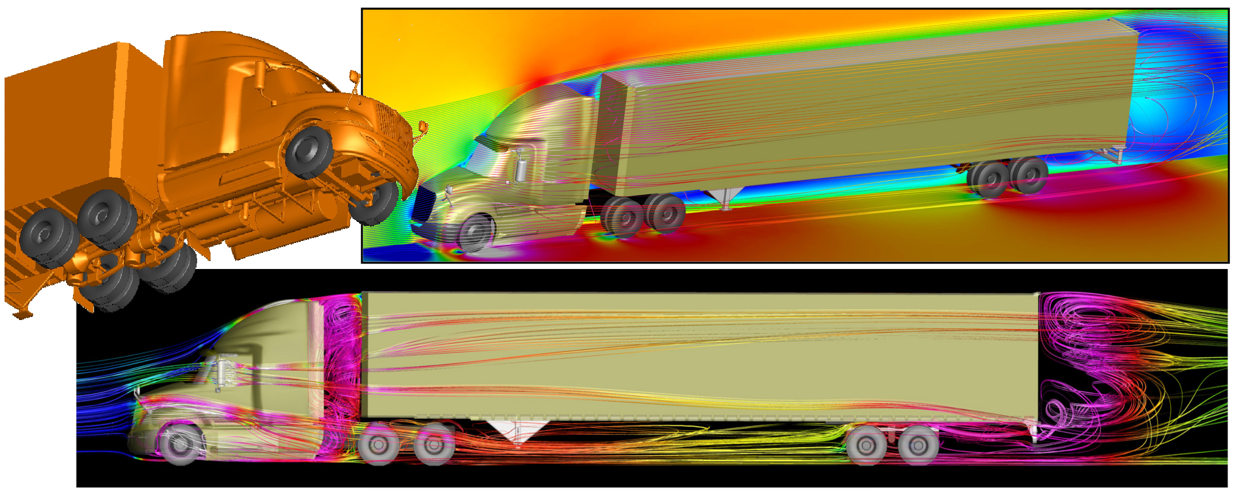


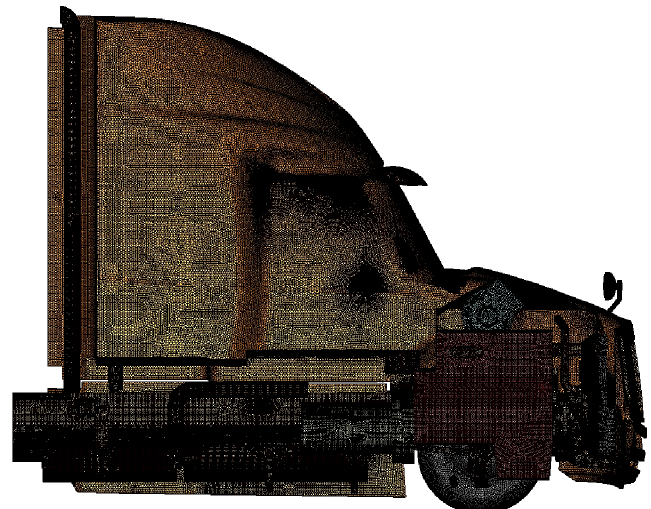
Falcon 9
First Launch
June 4, 2010



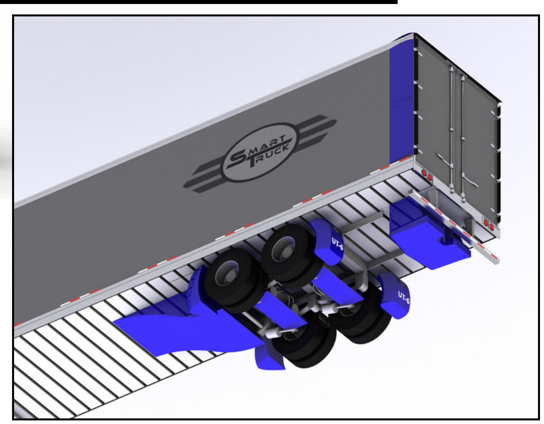
Industry: Tractor-Trailers at BMI Corporation

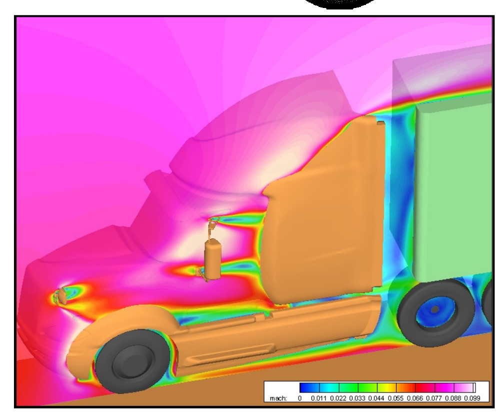
- Working to improve gas mileage of 18-wheeler configurations
- Actual trucks road-tested at shuttle landing facility at NASA Kennedy
- Current UnderTray system increases MPG by >11%
- FUN3D simulations performed at ORNL using thousands of cores
- Goal is to entirely redesign a "SmartTruck" that would cut drag by >50%

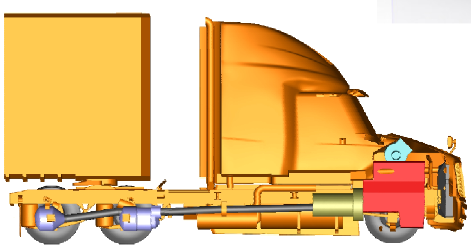


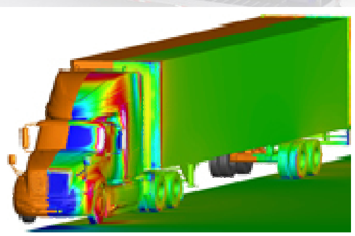


Images and Information Courtesy of BMI Corporation









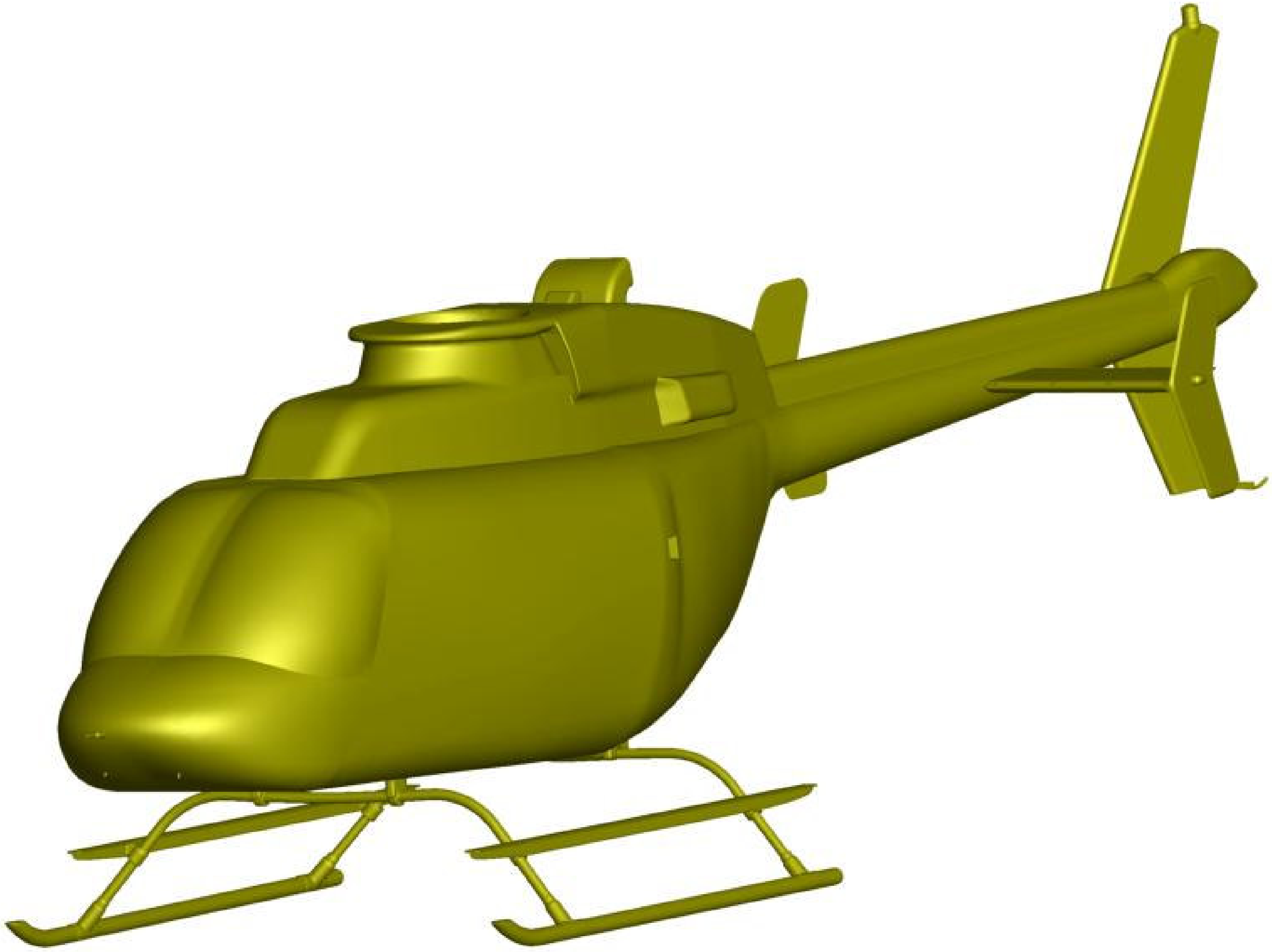
DoD: Rotorcraft at the US Army

- Examples provided by Tin-Chee Wong of the Aviation Engineering Directorate at Redstone Arsenal, AL

Aerodynamic Analysis Matrix

- CFD computations for each configuration
  - Angle of attack (α) sweep:
    - 19 cases per three yaw angles (β) - β: -3°, 0°, 3°; -30°< α< 30°
  - β sweep:
    - 13 cases for α=-2°; -12°< β< 12°
    - Each configuration: 39 cases + 1 trim case = 40 cases
- Total of 160 cases were computed
  - DoD Super Computing Resource Center (DSRC)
    - SGI Altix ICE computer with 10752 cores at ARL
  - Each case takes 2 hrs of wall-clock time with 256 cores
  - Equivalent of 512 hrs with a single processor
  - 81920 hrs were used for four configurations.
- Aerodynamic forces and moment are normalized to a free stream dynamic pressure, q; in form of L/q, M/q

Bell-407 CFD Model

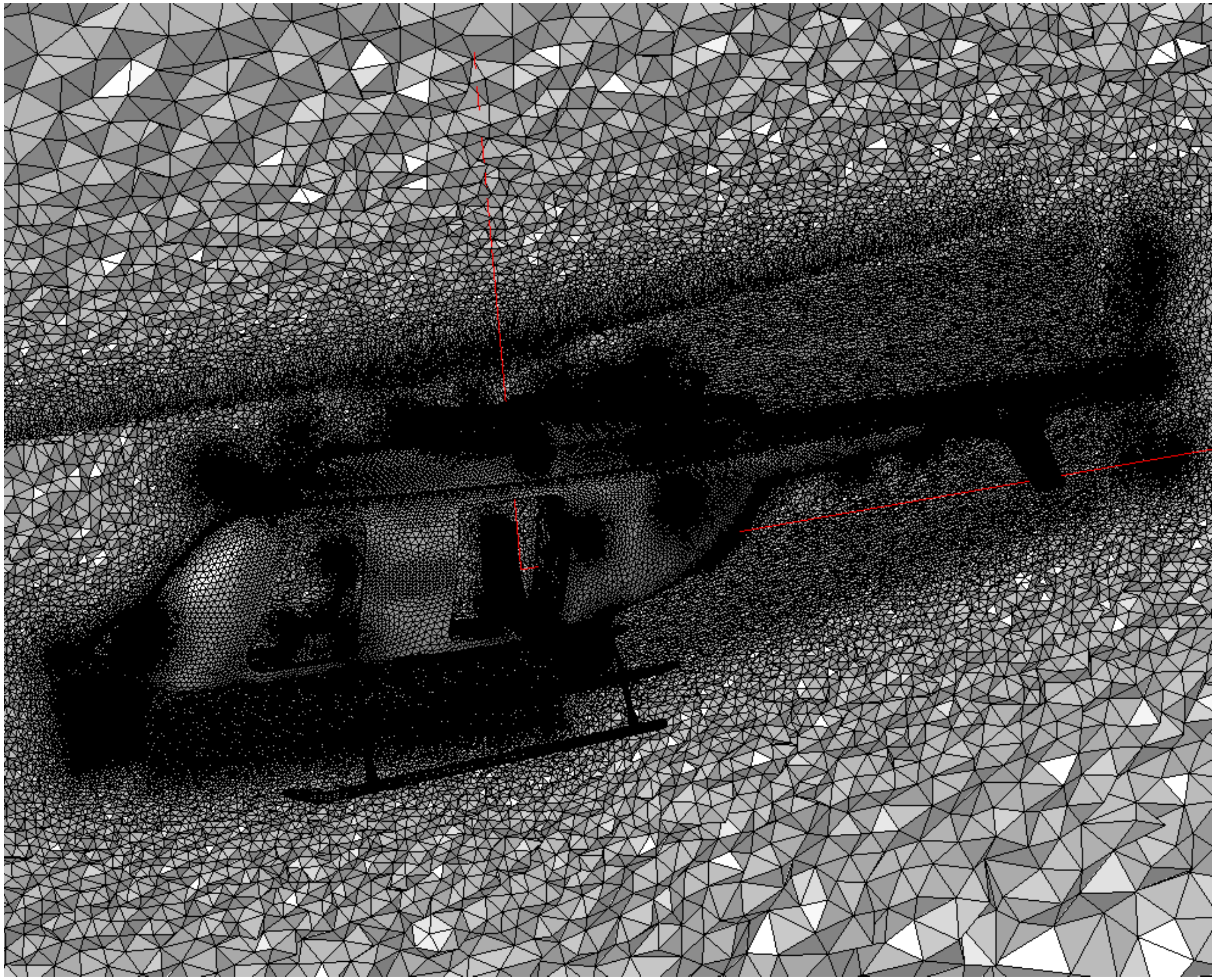


IA-407 ISR Configuration

IA-407 ISR Door Off Configuration

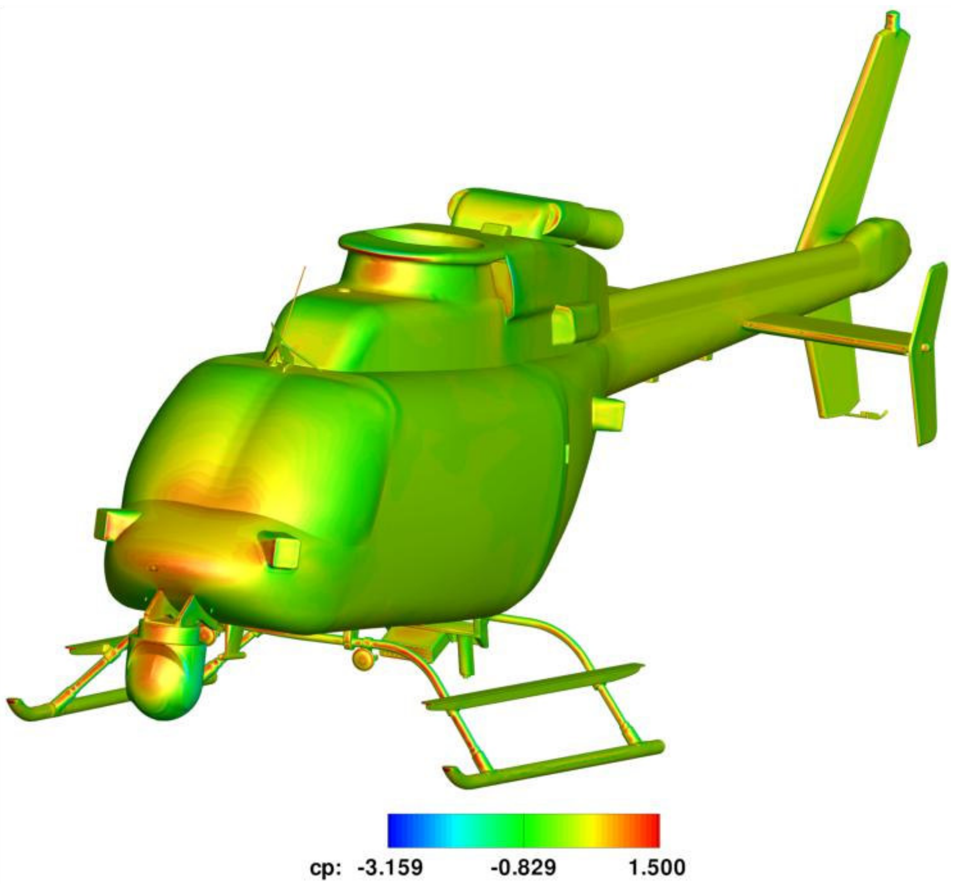
IA-407 ARMED Configuration

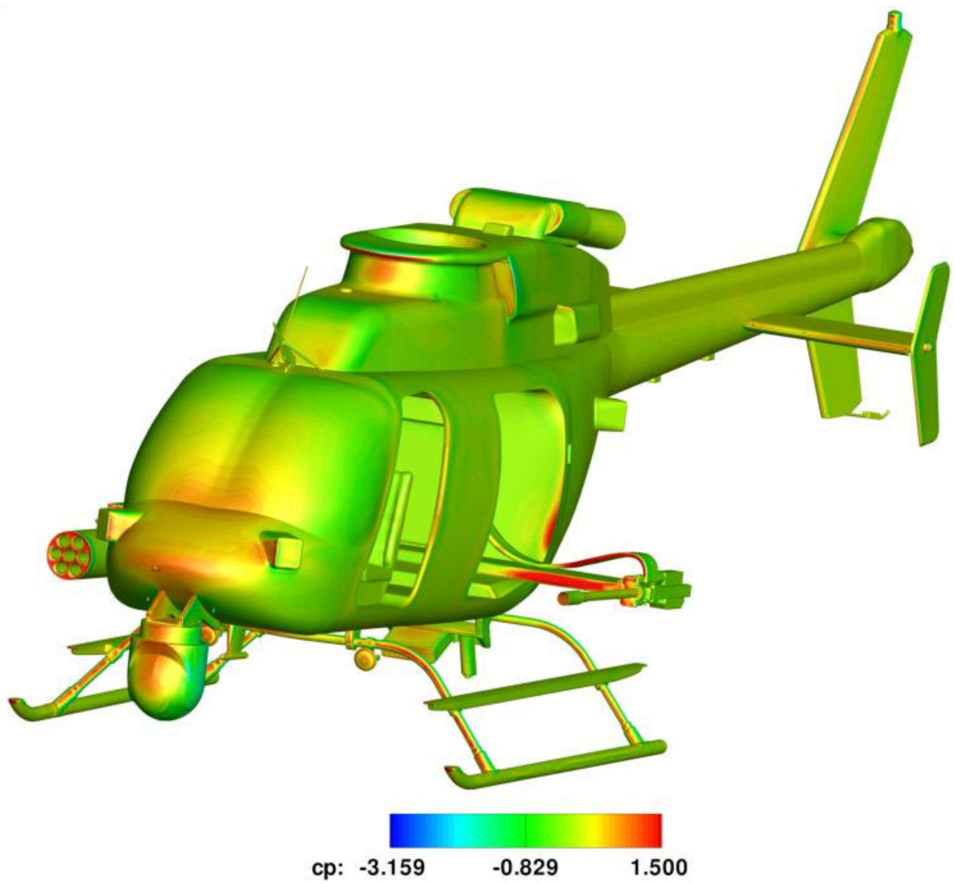
CFD Meshes for Armed Configuration



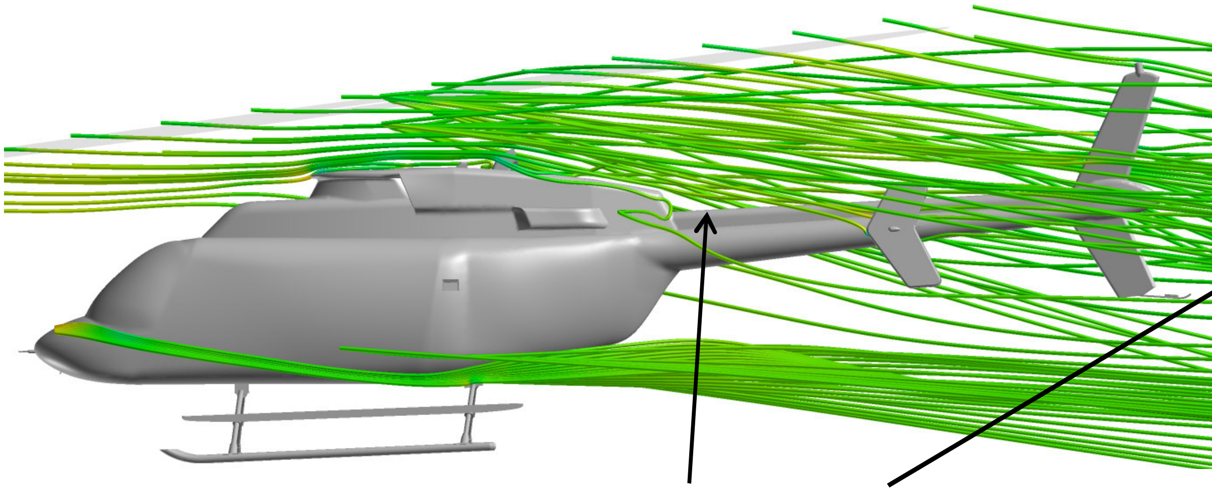
17 Million grid points

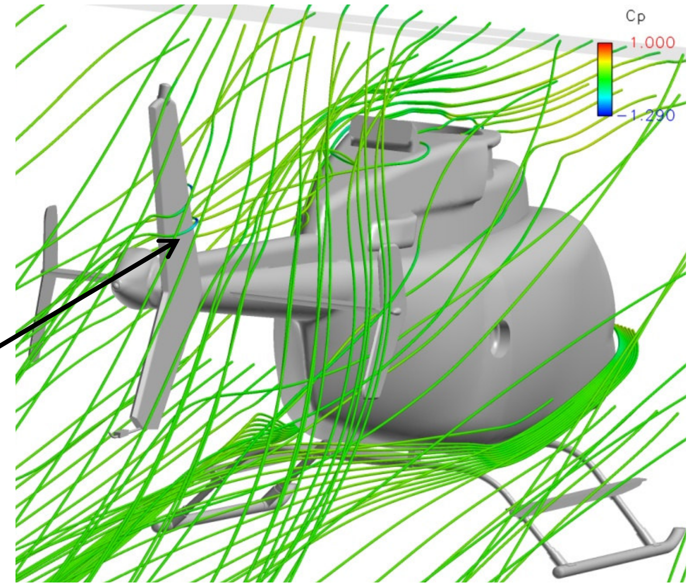
Surface Pressure of ISR and ARMED Configurations



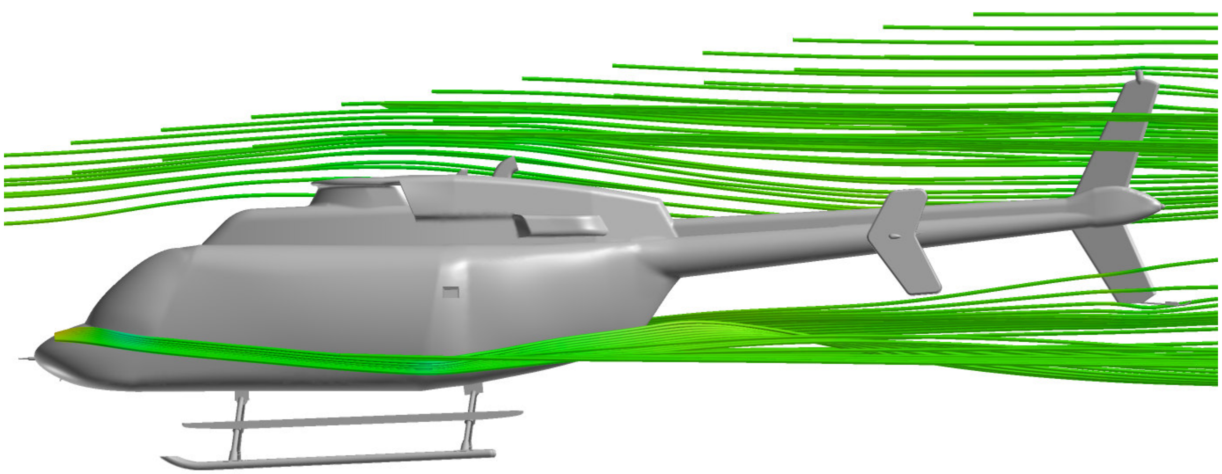


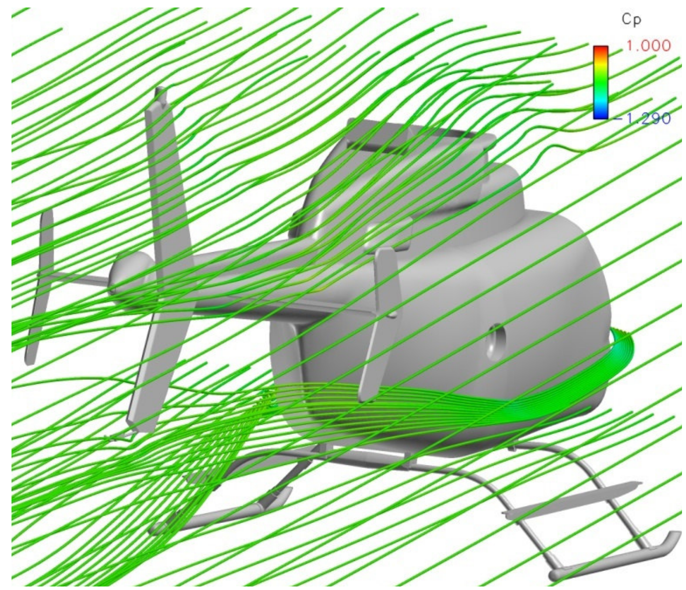
Bell-407 with and without Rotor at Forward Flight of 70 Knots





Downwash Effect





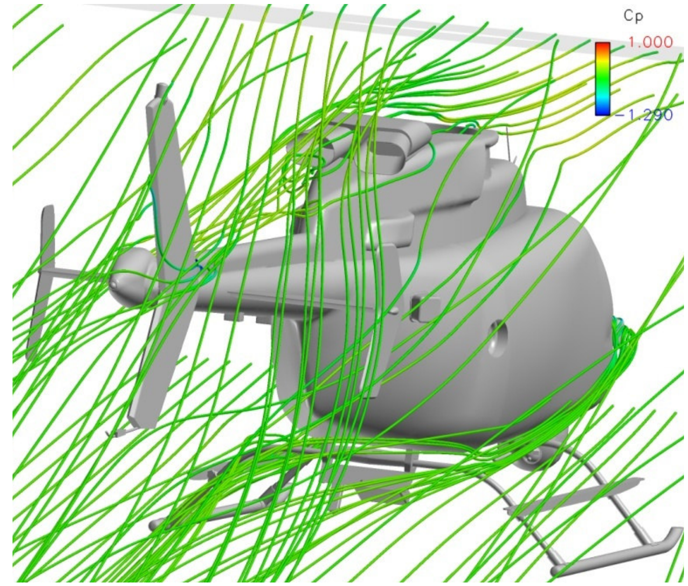
No Rotor

RDECOM

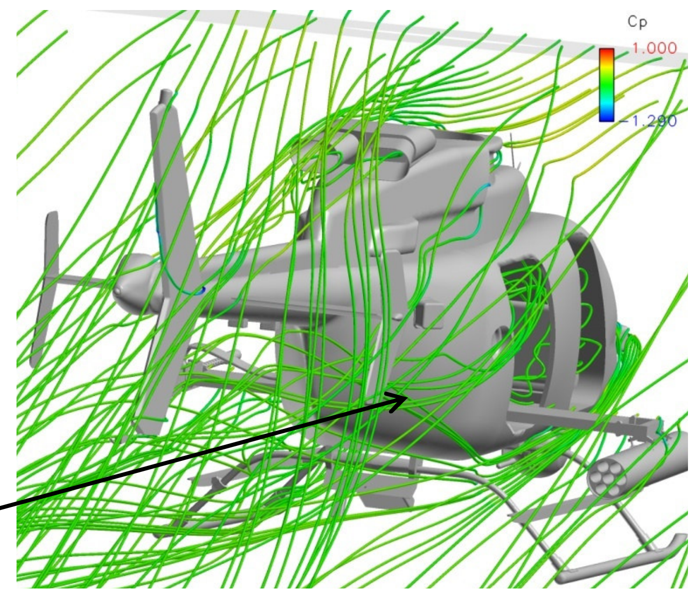
IA-407 ISR and ARMED Configurations at FWD Flight of 70 Knots

AMRDEC
STRENGTH THROUGH TECHNOLOGY

ISR



ARMED

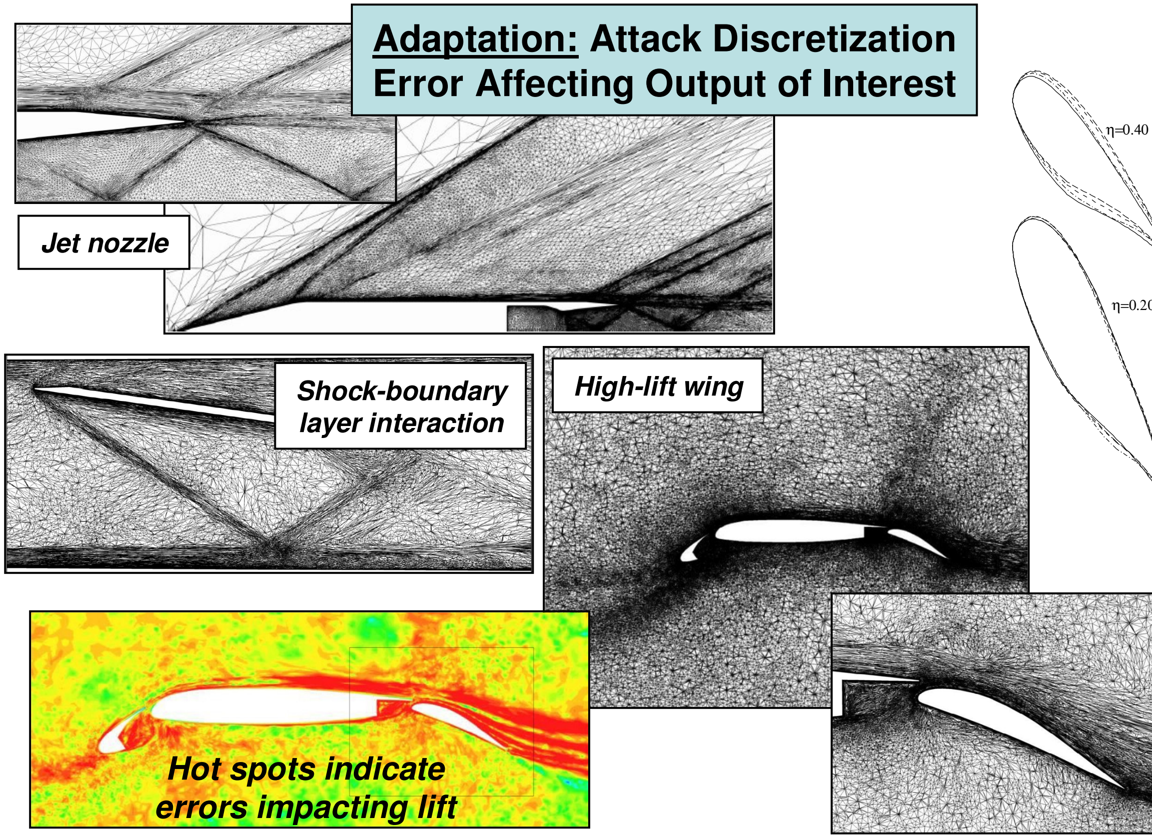


Interference effects from door off

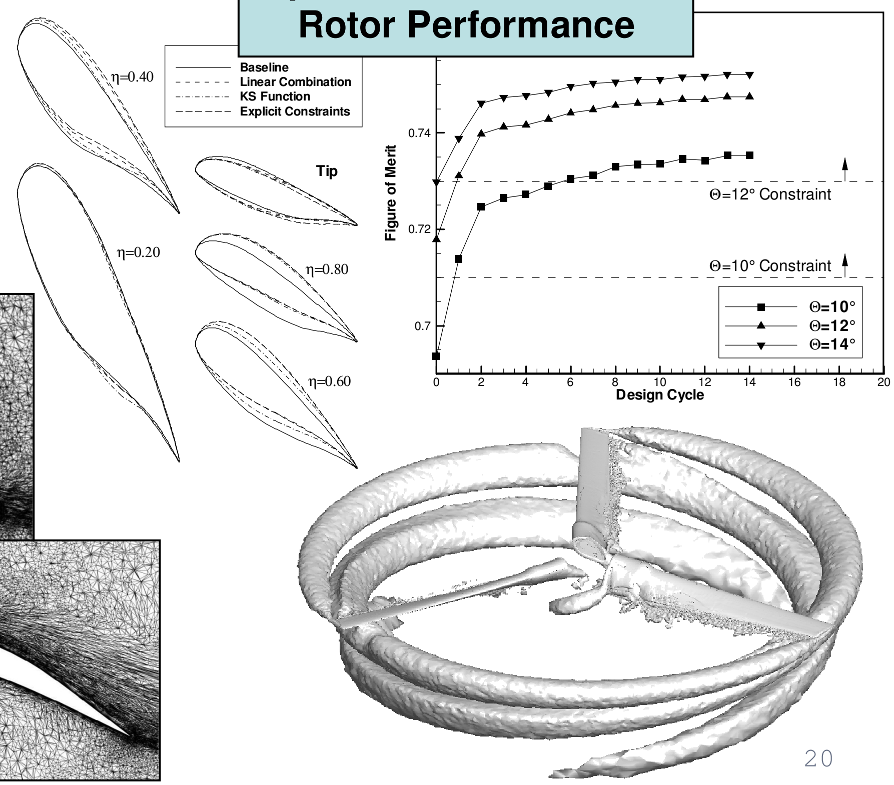
What Are We Working On?
Adjoint-Based Schemes for Steady Flows

- Adjoint-based schemes use system of auxiliary equations similar to the governing equations
  - Solution represents sensitivity of an output function to a source term
- Unlike traditional feature-based approaches, adjoints provide mathematically-rigorous error estimation and mesh adaptation
- Adjoints also enable rigorous sensitivity analyses for 1000’s of variables at cost of one solution
  - Formal design optimization becomes quite tractable
- FUN3D team has worked extensively in these areas for 15+ years: at forefront of this technology
  - Design optimization and mesh adaptation on thousands of cores steadily becoming routine

Adaptation: Attack Discretization Error Affecting Output of Interest



Optimization: Maximize Rotor Performance

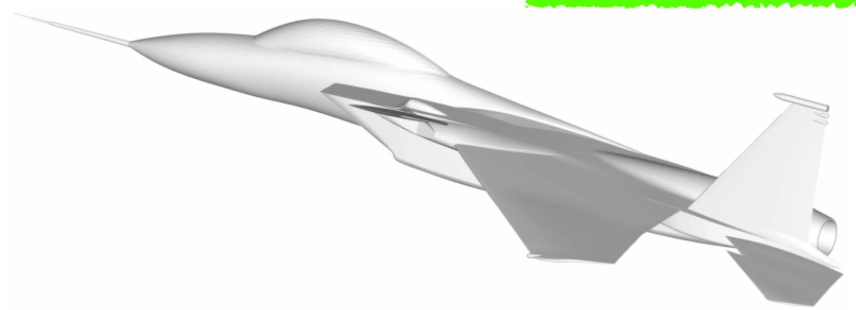


Adjoint-Based Schemes for Unsteady Flows

- Enables formal design and adaptation for any general unsteady phenomena
  - Flow control devices
  - Aeroelastic problems
  - Maneuvering flight/6-DOF
  - Specified motion
  - Biologically-inspired flapping wings
- Practical limitations are the biggest challenges
  - *Entire* unsteady solution must be available – not just last few planes
    - Dynamic simulation with 50 million grid points, 5,000 time steps, and a one-equation turbulence model requires ~24 TB of disk space
    - Frequent, intense parallel I/O
  - The cost of a single unsteady simulation can be considerable
    - Despite algorithmic efficiencies, the cost of formal adjoint-based design optimization for general unsteady flows can still be ***tremendous***

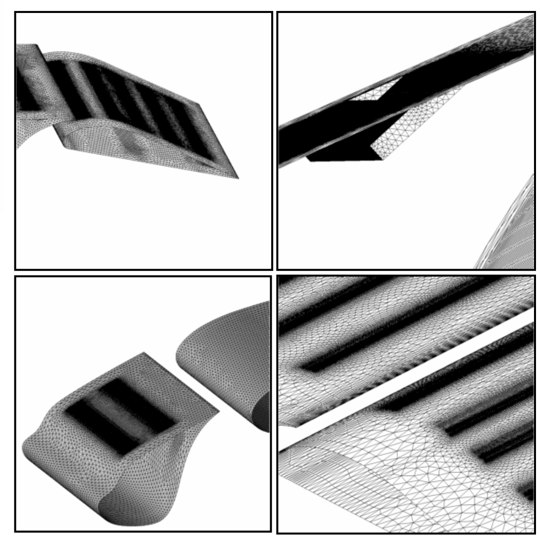
Massive **HPC resources are critical**

Fighter Jet with Propulsion System and Simulated Aeroelastic Effects



Tilt Rotor Design

Integrated Design of Active Flow Control



Issues, Needs, and Future Work

- More HPC scientists embedded full-time with development efforts
- Higher-level languages, programming paradigms
- Simplifying development for coming generations of (heterogeneous) hardware
- Partitioning, MPT “tuning” at high core counts
- Improving collective communications
- Very frequent massively parallel I/O
- Fault-tolerant algorithms (system or application level)
- Tools for performance optimization

- Resolving ever-smaller temporal and spatial scales
- Aeroacoustics
- Is mesh generation keeping pace?
  - Looking to massively parallel adaptive methods to get us to where we need to be
- Uncertainty quantification
- Real-time simulations – “digital flight”
- Multidisciplinary analysis and optimization

Very interested in new collaborations!