Discrete Adjoint-Based Design for Unsteady Turbulent Flows On Dynamic Overset Mixed-Element Grids

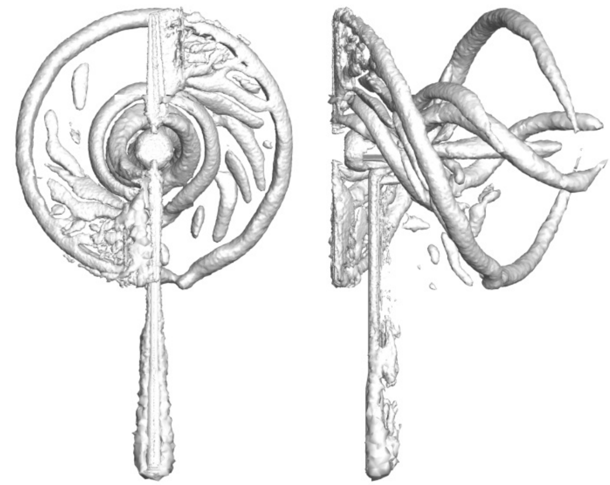
Eric J. Nielsen
Computational AeroSciences Branch
NASA Langley Research Center

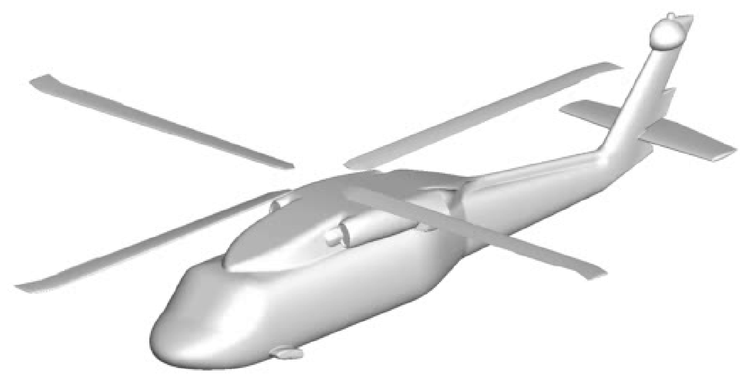
Boris Diskin
National Institute of Aerospace

http://fun3d.larc.nasa.gov

Motivation

- Adjoint methods provide very efficient and discretely consistent sensitivity analysis for PDE’s
- Presented general URANS design capability for dynamic grids in 2009
- Extension to dynamic overset grids opens up many new applications for optimization, especially those involving large relative motions
    - Rotorcraft
    - Store/stage-separation problems
    - Wind energy devices
    - Biologically-inspired configurations
    - Turbomachinery
- Many other exciting opportunities
    - Error estimation
    - Rigorous mesh adaptation
    - Uncertainty quantification





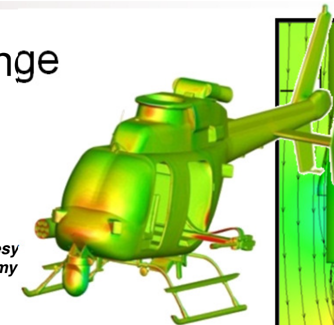
<u>Goal:</u> Adjoint-based design for compressible/incompressible URANS on dynamic overset grids, amenable to massively parallel HPC environments

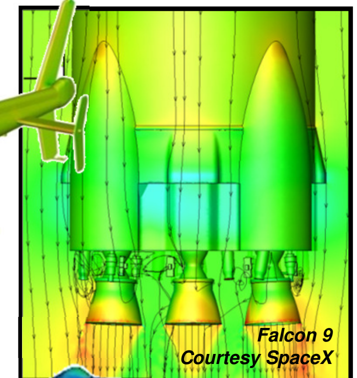
FUN3D Overview

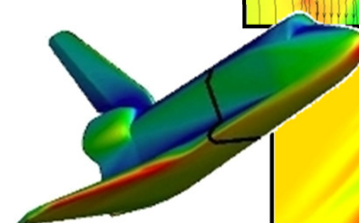
Effort Initiated in Late 1980's

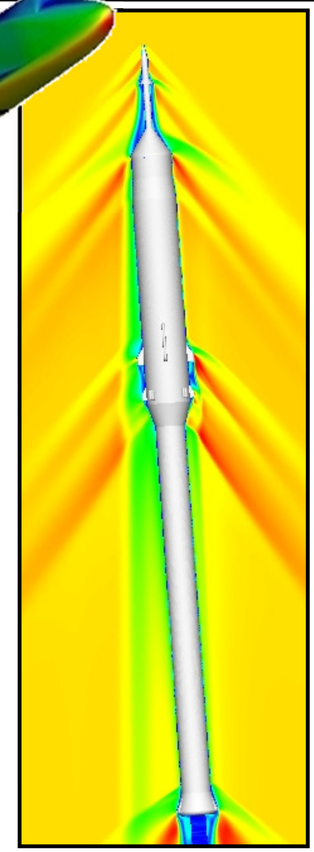
- Supports numerous internal/external efforts across speed range
- Solves 2D/3D steady/unsteady RANS equations on dynamic overset mixed-element grids using node-based finite-volume
- Discretely consistent adjoint formulation for design optimization, error estimation, and mesh adaptation
- Automated complex variable formulation for forward differentiation
- Scalable end-to-end paradigm demonstrated to ~50,000 cores using as many as 2 billion points / 12 billion elements
- Produces visualization data concurrently: surfaces, volumes, slices, schlierens, isosurfaces, etc; DOE VisIt also integrated
- Testing framework performs 24/7 regression testing, monitors accuracy and performance

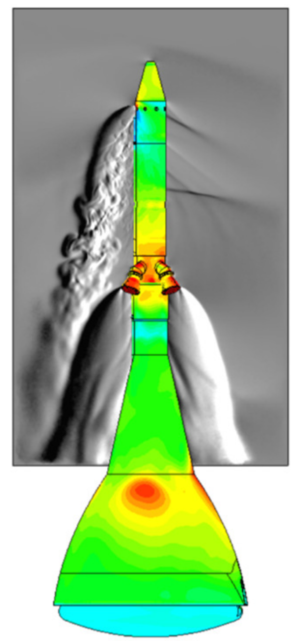
Courtesy US Army

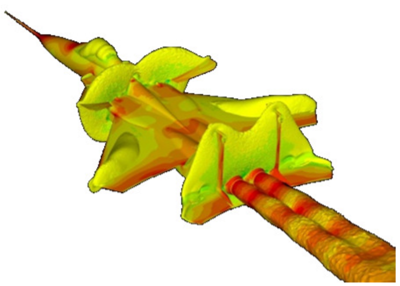


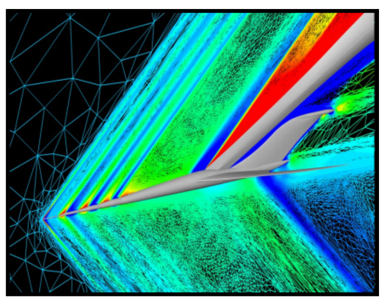


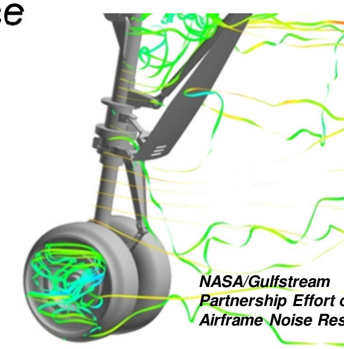


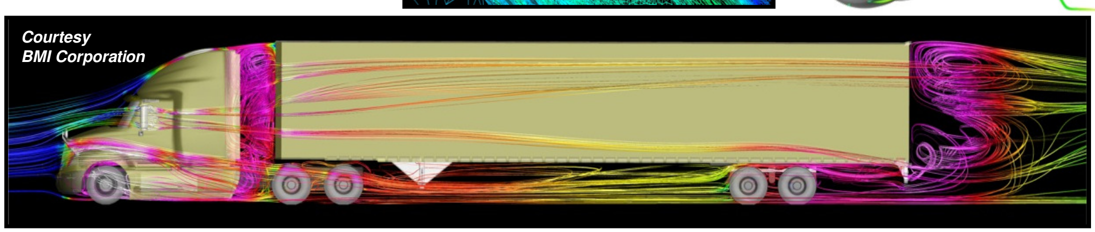












General Approach

Based on the governing equations for the flow field and component grids, the discrete adjoint equations are derived and implemented

Governing equations for flow field

- ***Solve*** points use the URANS equations
- ***Fringe*** points use interpolants to compute data from values on another grid
- ***Hole*** and ***orphan*** points use pseudo-Laplacian operators to average surrounding data
- Nature of each point may change in time

Governing equations for component grids

- May specify rigid motion according to 4x4 transform matrices
- May specify deformations according to linear elasticity relations
- May specify both rigid motions *and* elastic deformation
- General parent-child motions may also be specified

Parent-child motion using rigid and deforming mesh mechanics

Full details of formulation available in AIAA-2012-0554

Background / Status of Adjoint Capabilities

Presented basic approach on tetrahedral grid systems in January 2012;
This talk covers the extension to general mixed-element grids

Euler Fluxes

- Extension to full UMUSCL reconstruction

Viscous Fluxes

- Extension to include edge-based augmentation for Green-Gauss gradients

Turbulence Model

- Similar modifications for convection, diffusion, source terms
- Linearization of distance function on quads

Boundary Conditions

- Extension to quads

Dynamic Mesh Infrastructure

- Linearization of face speeds and geometric conservation law on other element types

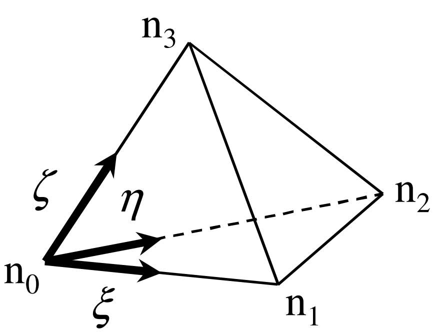
Overset Mesh Infrastructure

- Sensitivities for general nonlinear interpolation formulas

Nonlinear Interpolations

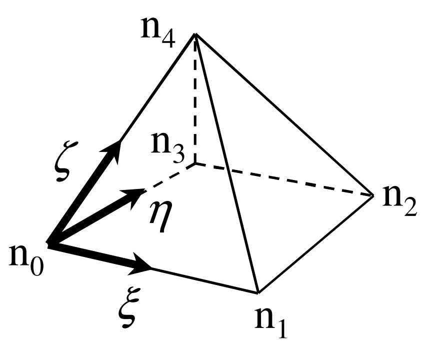
Based on DiRTlib Approach of Noack

Tetrahedron



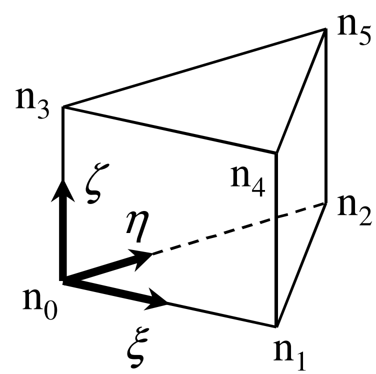
$$R = (1-\xi-\eta-\zeta)r_0 + \xi r_1 + \eta r_2 + \zeta r_3$$

Pyramid



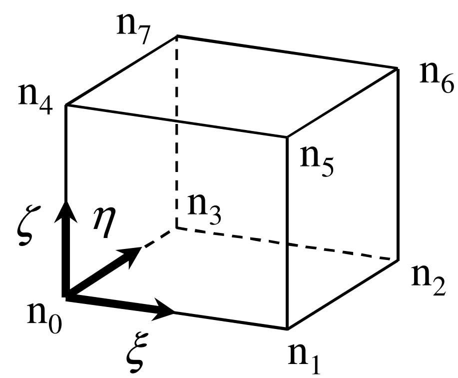
$$R = [(1-\xi)(1-\eta)r_0 + \xi(1-\eta)r_1 + \xi\eta r_2 + \eta(1-\xi)r_3](1-\zeta) + \zeta r_4$$

Prism



$$R = (1-\xi-\eta-\zeta+\xi\zeta+\eta\zeta)r_0 + \xi(1-\zeta)r_1 + \eta(1-\zeta)r_2 + \zeta(1-\xi-\eta)r_3 + \xi\zeta r_4 + \eta\zeta r_5$$

Hexahedron



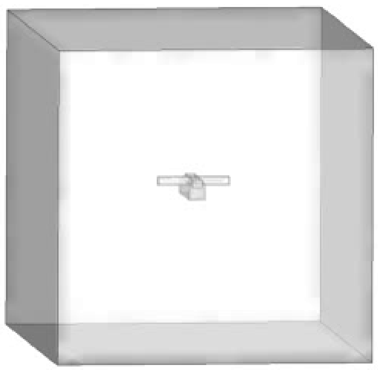
$$\begin{aligned} R = &(1-\xi)(1-\eta)(1-\zeta)r_0 \\ &+\xi(1-\eta)(1-\zeta)r_1 \\ &+\xi\eta(1-\zeta)r_2 \\ &+(1-\xi)\eta(1-\zeta)r_3 \\ &+(1-\xi)(1-\eta)\zeta r_4 \\ &+\xi(1-\eta)\zeta r_5 + \xi\eta\zeta r_6 \\ &+(1-\xi)\eta\zeta r_7 \end{aligned}$$

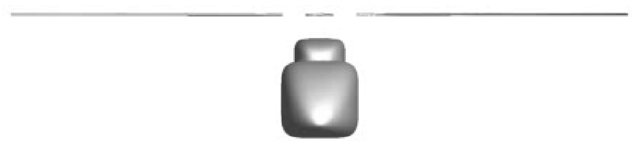
- Evaluating these expressions at receptor location and donor element vertices yields a nonlinear set of equations for (ξ,η,ζ)
- System solved using Newton’s method
- Sensitivities $\partial(\xi,\eta,\zeta)/\ \partial(x,y,z)$ required for adjoint obtained by differentiating this procedure

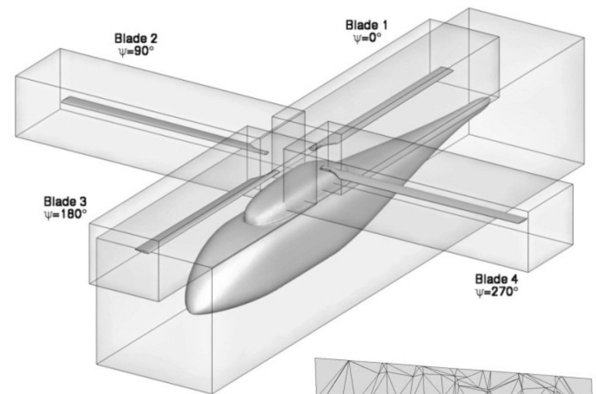
Verification of Implementation

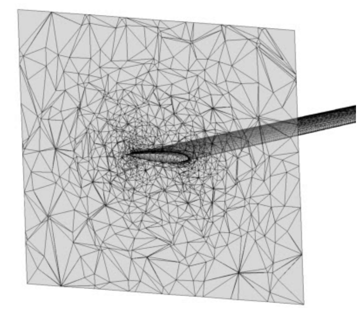
Problem Definition

- 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 ($\varepsilon = 1\times10^{-50}$)

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

Large-Scale Test Cases

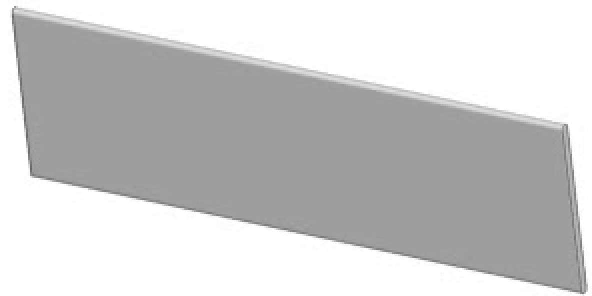
- Focus here is on performance of design methodology, not flow physics
- Results use tetrahedral grids; mixed-element cases to be performed shortly
- All cases shown run on SGI ICE system with parallel file system
  - 160 dual-socket hex-core nodes run fully dense – total of 1,920 cores
- Solution of unsteady adjoint problem requires I/O of flow solution *at every time step*
  - Extensive optimization of I/O paradigm in previous work
    - Relies on parallel I/O of individual unformatted direct-access files
    - Asynchronous read/writes to mask I/O behind computations
  - Each flow field solution shown here consists of ~1 TB of data

Implementation handles deforming bodies; however, such cases are typically driven by other coupled disciplines (e.g., structures).

Current formulation does not address coupling, so all cases shown are forced motions

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 $\pm$60º sweeping and $\pm$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

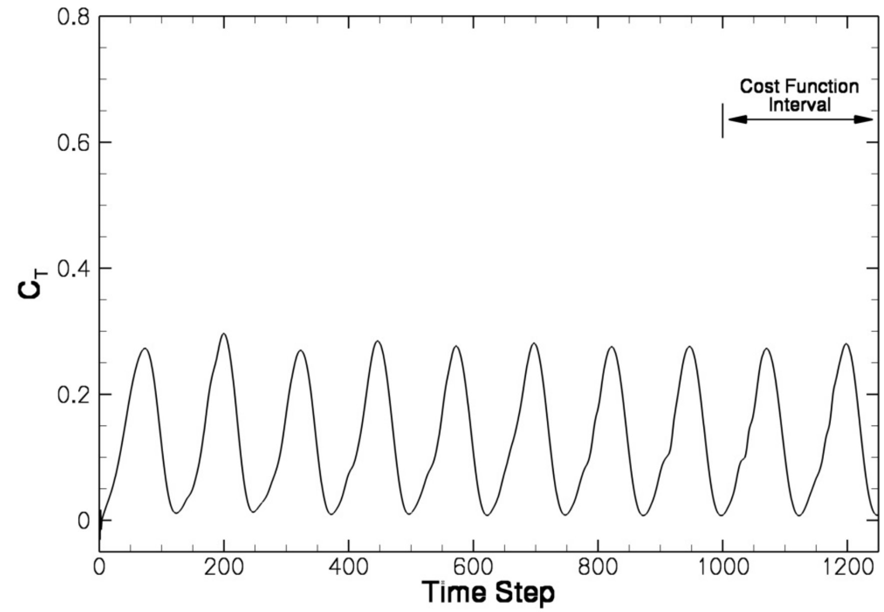
Problem Definition

- 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

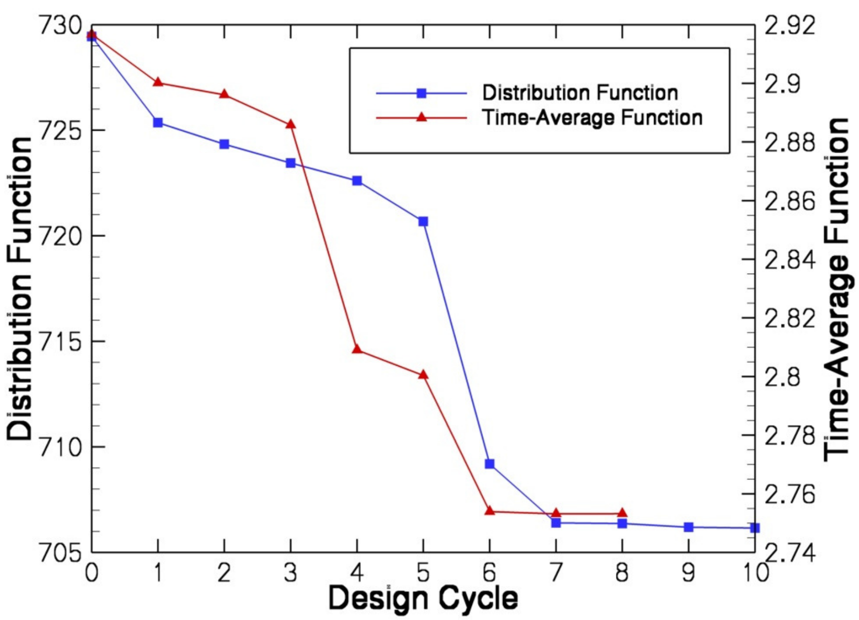
Design variables: 3 coords of rotation center, 12 kinematic parameters A, B, ω_1, ω_2

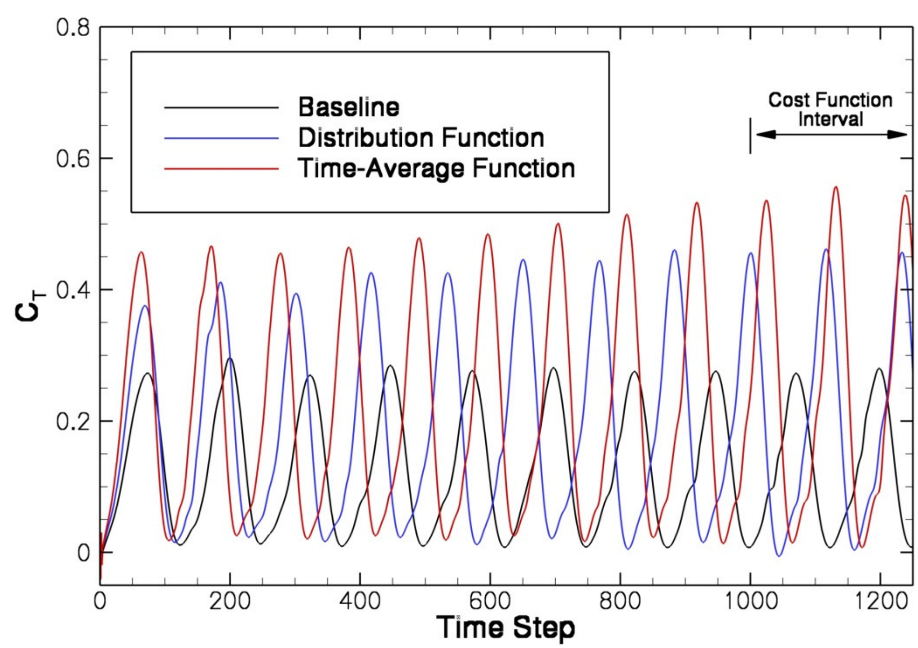
Biologically-Inspired Flapping Wing

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

	$\overline{C}_T$	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)

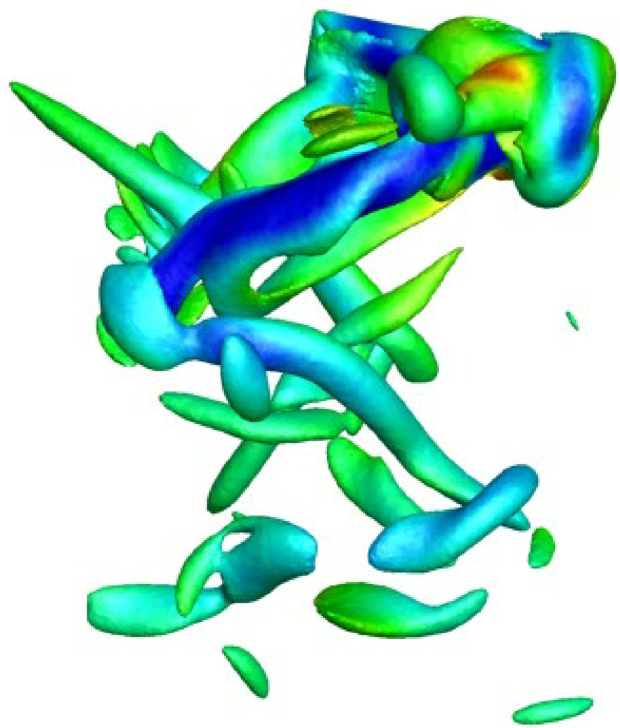




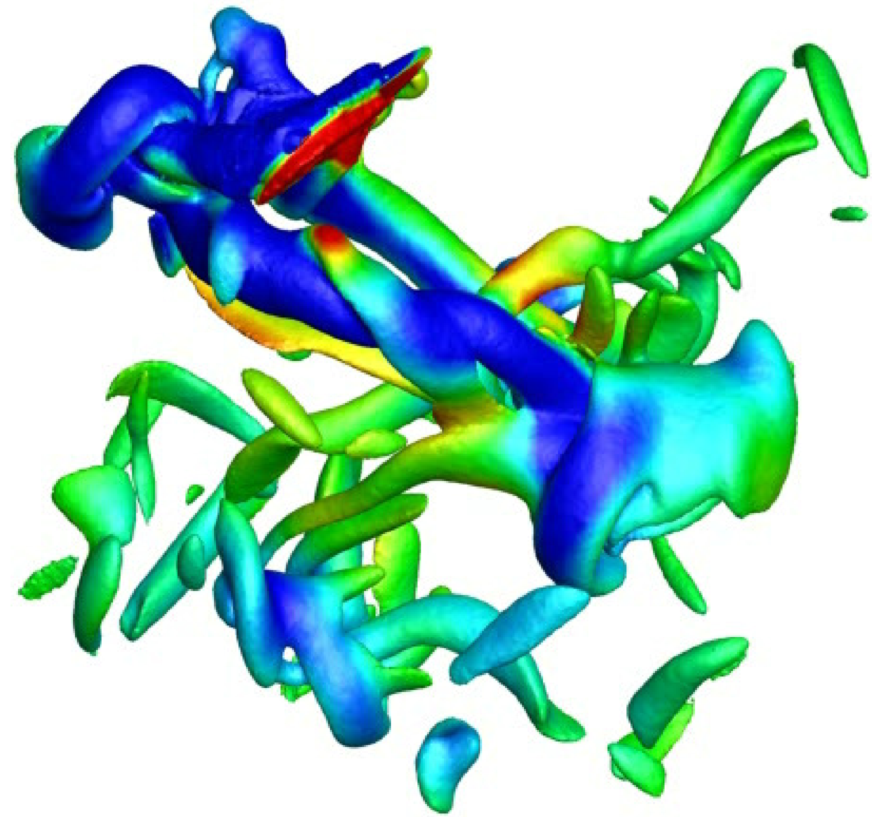
Biologically-Inspired Flapping Wing

Baseline and Optimal Flow Fields

Isosurfaces of Vorticity Colored by Pressure

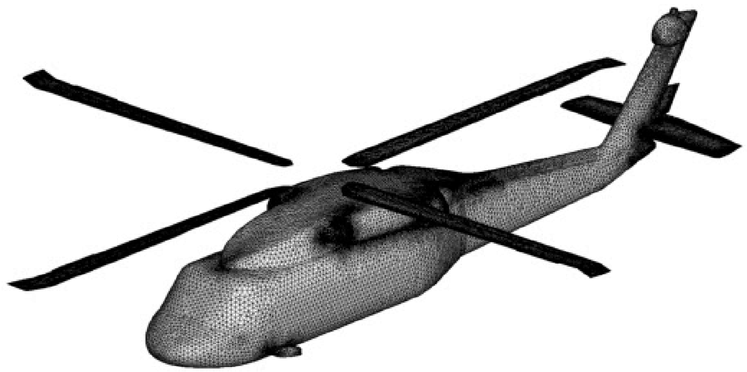


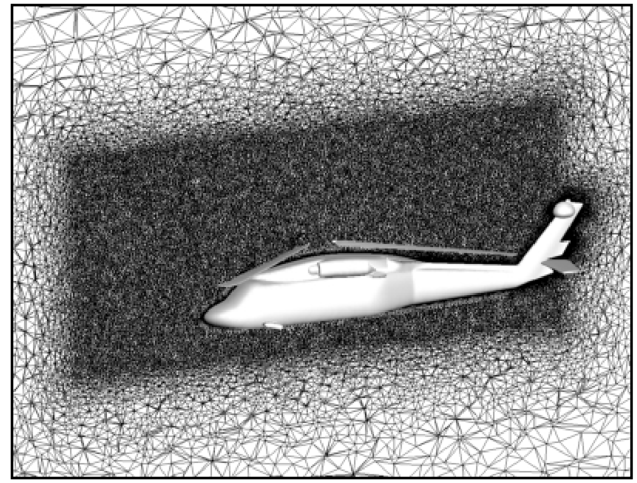
Baseline



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
- BDF2opt scheme run for 2 revs with 15 subiterations per time step
- Time step corresponds to 1º of rotor rotation
- 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 (θ); Collective (θ_c); Lateral cyclic (θ_{1c}); Longitudinal cyclic (θ_{1s})

- Baseline value of all control inputs is zero

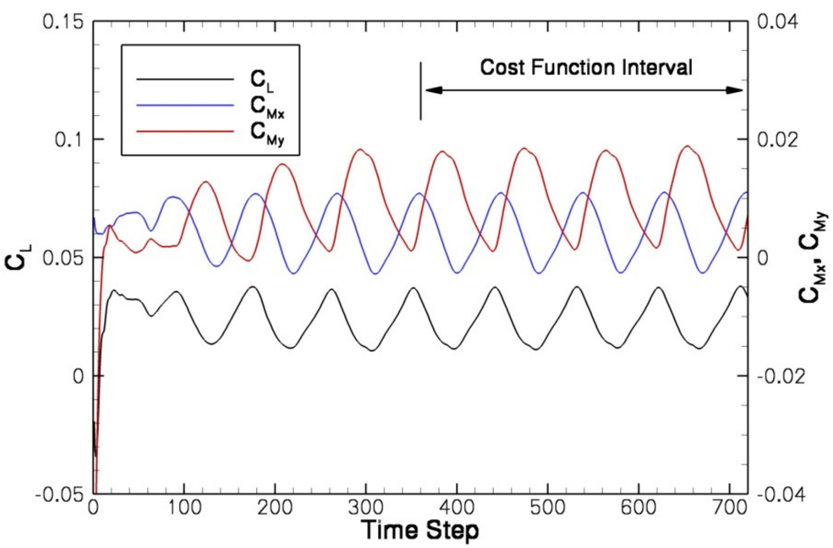
UH-60A Blackhawk Helicopter

Problem Definition

- Baseline conditions yield untrimmed flight with $\bar{C}_L$=0.023 over second rev
- Objective is to maximize $\bar{C}_L$ while satisfying trim constraints over second rev:

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

- 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 ±7º
- Fuselage shape could also be designed, but not pursued here (unknown constraints)



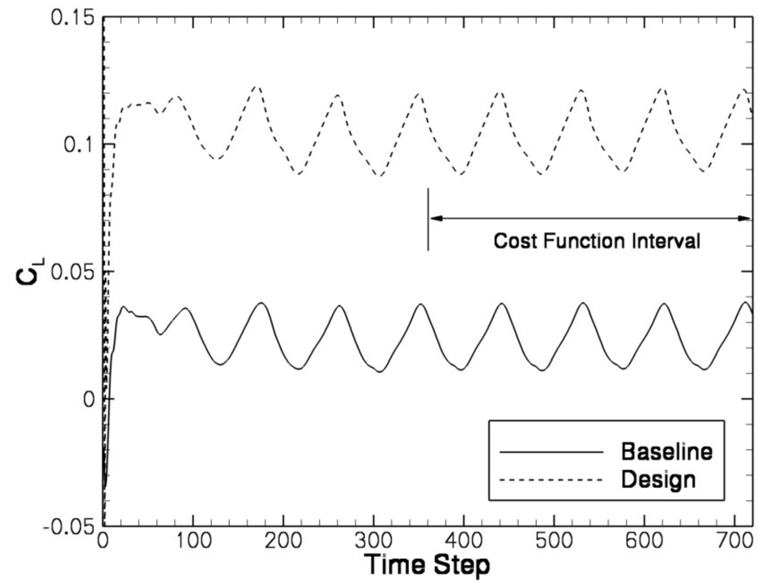
Blade shape design variable locations

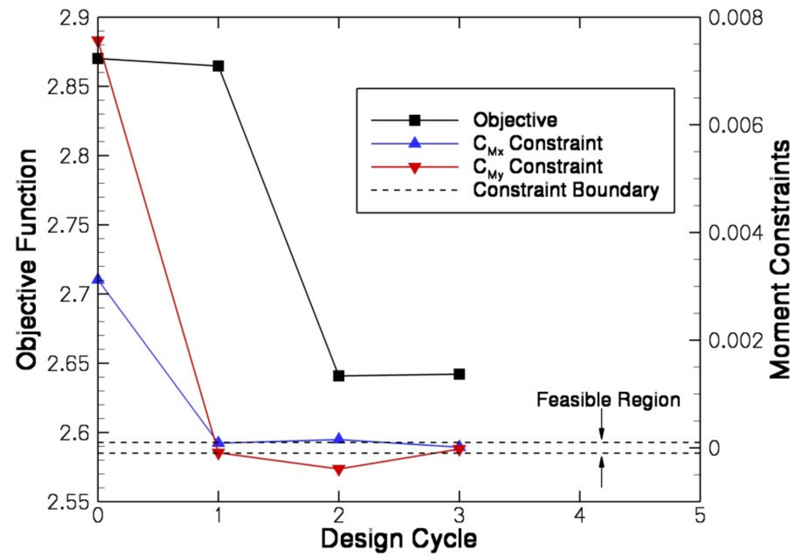
UH-60A Blackhawk Helicopter

Results

- 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

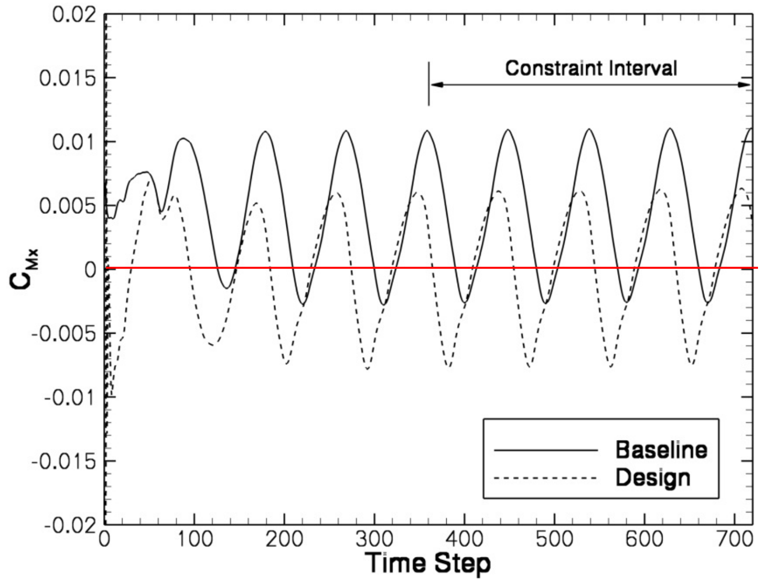
	$\bar{C}_L$	Flow Solves (2 hrs)	Adjoint Solves (3 hrs)	Total Time
Baseline	0.023	-	-	-
Design	0.103	4	4	0.8 days (38,400 CPU hrs)

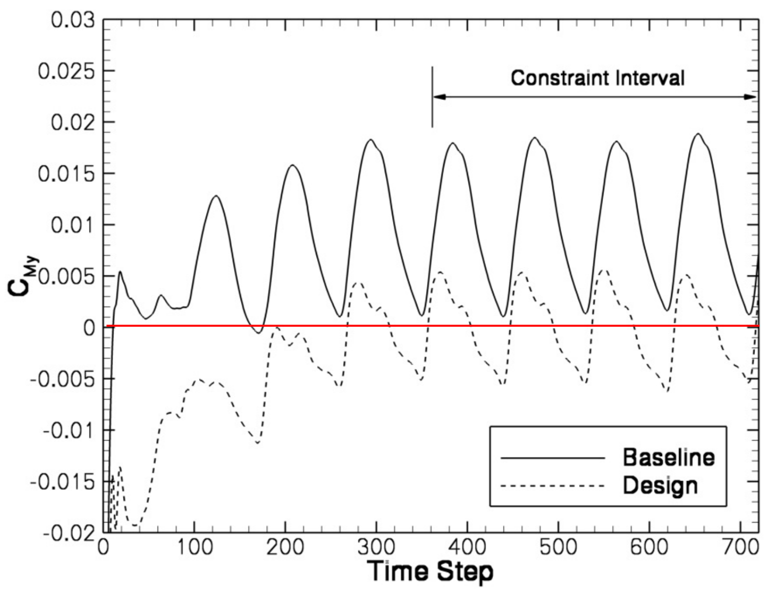


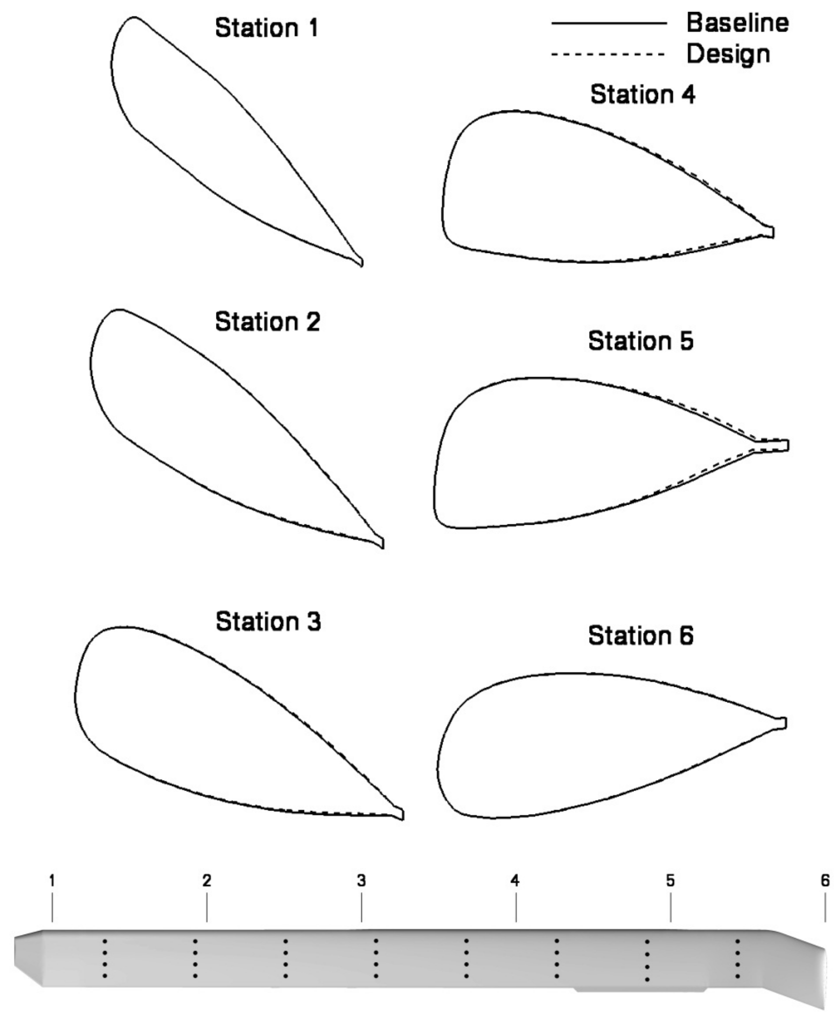


UH-60A Blackhawk Helicopter

Results

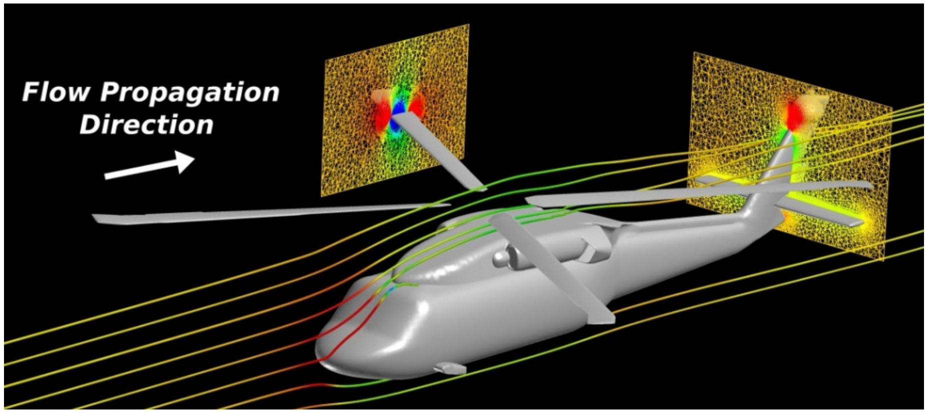


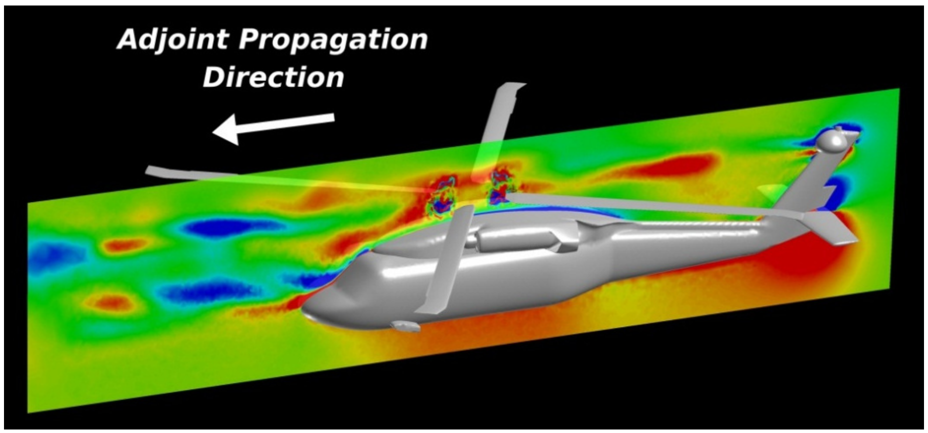




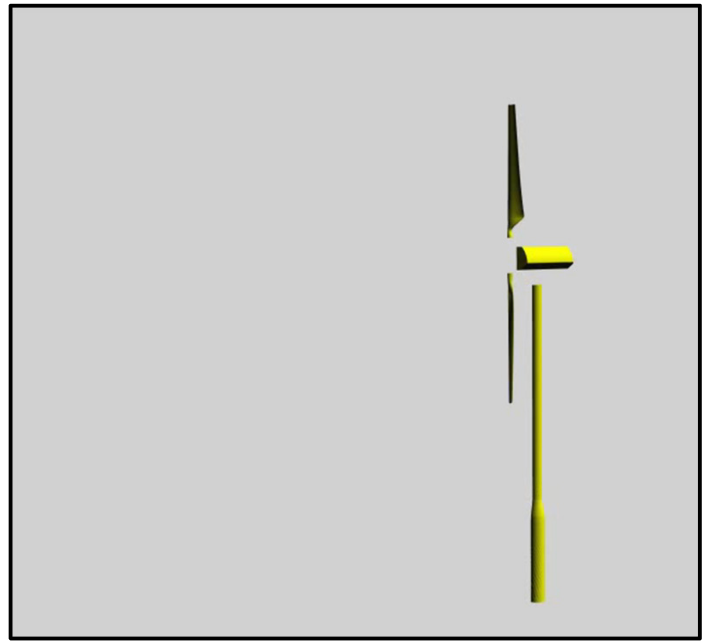
Interpretation of Adjoint Solutions

Helicopter





Wind Turbine



Click for Isosurface Animation

Animations shown in reverse physical time

- Adjoint shows sensitivity of objective function to local disturbances in space and time: $\partial f/\partial \mathbf{R}$
- Solution can be used for sensitivity analysis as done here
- May also be used to perform rigorous error estimation and mesh adaptation
  - Traditional feature-based techniques do not identify such regions

Summary and Future Work

Developed adjoint-based design methodology for URANS simulations on dynamic overset grids in HPC environments

- Formulation
- Implementation
- Verification
- Applications

- Locally optimal, checkpointing techniques
- Multi-fidelity optimization algorithms
- Convergence acceleration schemes
- Simultaneous adjoint-based adaptation & design
- Extension of adjoint methods to MDO problems
- Continued leverage of computer science, software development, and HPC advancements

***Also in AIAA-2012-0554*:**
Wind turbine optimization on 2,880 cores using mesh with 87 million elements