Unsteady, Unstructured Overset Mesh Adaptation with an Efficient Parallel Localization Scheme

Rajiv Shenoy, CRAFT Tech

Marilyn Smith, Georgia Tech

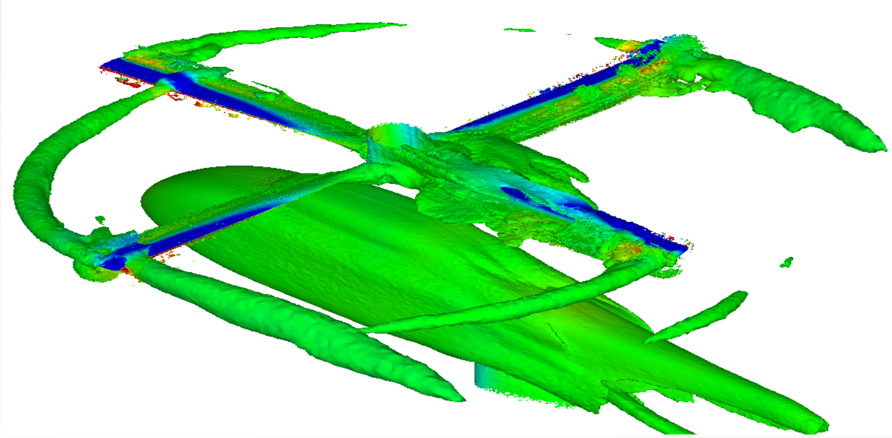
Michael Park, NASA Langley

Georgia Institute of Technology

Nonlinear Computational Aeroelasticity Lab

Introduction

- Complex wake physics require high fidelity methods for short and long age wake computations
- Applications include rotorcraft and wind turbine wakes
- Researchers have improved accuracy and capabilities:
  - **Mesh adaptation** is more efficient than uniform refinement
  - **Overset grids** enable moving body functionality and is popular for dynamic simulations
- Unstructured grids permit body-fitting of complex geometries



Computational Method

FUN3D – developed at NASA Langley

- Mixed-element unstructured, node-centered, finite volume solver
- Second-order accurate in space
- Second-order implicit time-integration (BDF2opt)
- Several turbulence models including hybrid RANS-LES (HRLES) – *Lynch, (GT 2011)*
- Overset functionality with DiRTlib/SUGGAR++ (PSU) – *O'Brien, (GT 2006)*
- Metric-based grid adaptation for tetrahedral elements

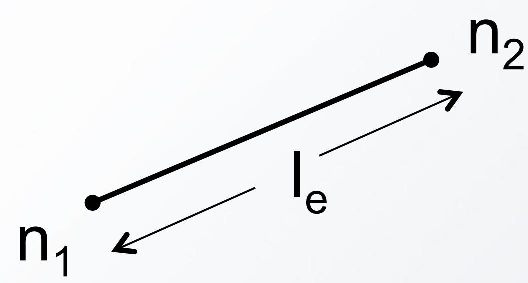
Feature-Based Error Estimators

Vorticity Magnitude

$$F_{e,\omega} = \ell_e \frac{|\omega|_{n_1} + |\omega|_{n_2}}{2}$$

Pressure Difference

$$F_{e,p} = \ell_e \left| p_{n_1} - p_{n_2} \right|$$



Q-criterion: $\frac{1}{2}\left(\|\Omega\|^2 - \|S\|^2\right)$ Separates regions of high rotation rate **Ω** from high strain rate **S**

Nondimensional Q-Criterion
Kamkar et al. (JCP 2012)

$$F_{e,Q-crit.} = \max_{n_1,n_2}\left[\frac{1}{2}\left(\frac{\|\Omega\|^2}{\|S\|^2} - 1\right)\right]$$

Metric-Based Adaptation

Intensity at each node →

$$\hat{I} = \max_{edges}\left(\frac{F_e}{F_{tol}}\right)$$

F_e: *error-estimate formulation (calculated on each edge)*

F_{tol}: *user-defined tolerance*

Such that **~10 nodes** resolve vortex core

- By obtaining adaptation intensity at each node, isotropic spacing is obtained
- Anisotropy reduces cost by stretching element
  - Introduced by computing a Hessian-based grid metric
  - The grid metric gives shape to each grid element
- Adaptation mechanics are performed based on the grid metric
  - Boundary layer mechanics are currently unavailable
  - More details in *Park et al. (AIAA, 2008)*

Extension to Overset Grids

Component grids

Stationary Grid (Fuselage) | **Near-body dynamic grids (Blades)** —**Assembly**→ **Composite grid**

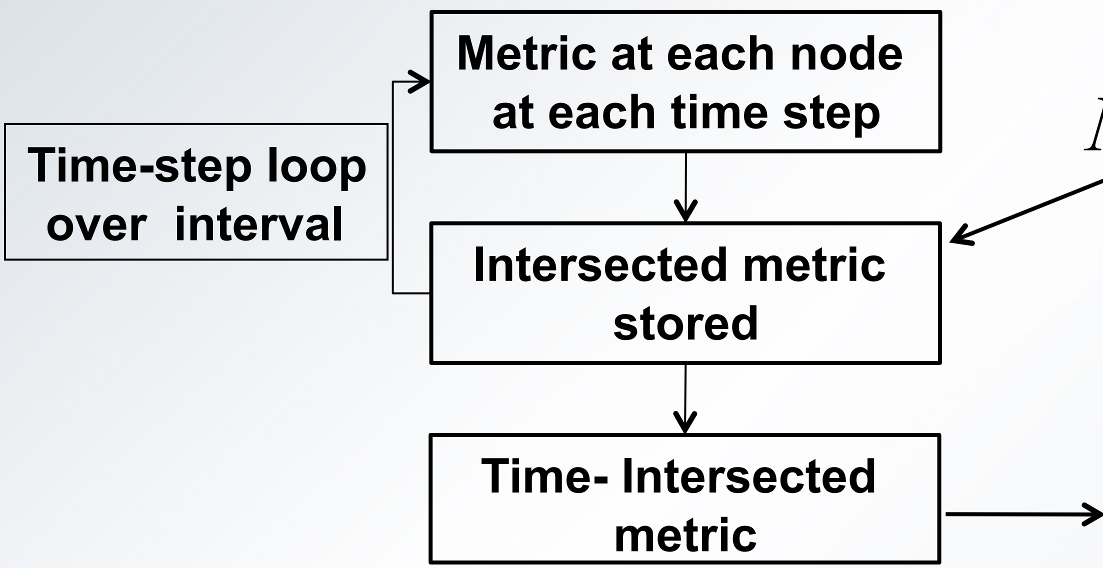
1. Adaptation performed on the **composite grid**
   - **All component grids** are **adapted** outside of the boundary layer
2. Hole-cutting of adapted grid is handled by SUGGAR++

   Adapted composite grid broken down into component grids and SUGGAR++ is called to obtain new domain connectivity
3. Designed a node indexing protocol for agreement between FUN3D and SUGGAR++

Time-Dependent Adaptation

Adapt grid over time-evolving interval(s)

Based on *Alauzet et al: AIAA 2007*



Grid Adaptation

Metric Intersection

$$M_{i,\max} = \bigcap_{n=1}^{N} M_{i,n}$$

Store the max. metric magnitude (highest error) at each node

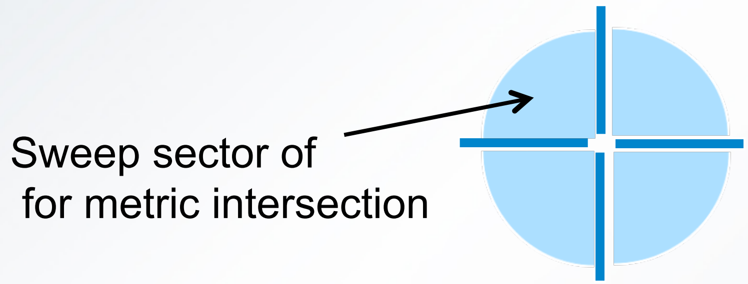
To handle implicit time integration:

1. Back plane metrics are included in metric intersection
2. Back plane grid motion is updated

Time-Dependent Adaptation

- Periodic interval adaptation
  - Rigid, prescribed motion
  - Adapt over a period and use new grid to get improved predictions and repeat until convergence

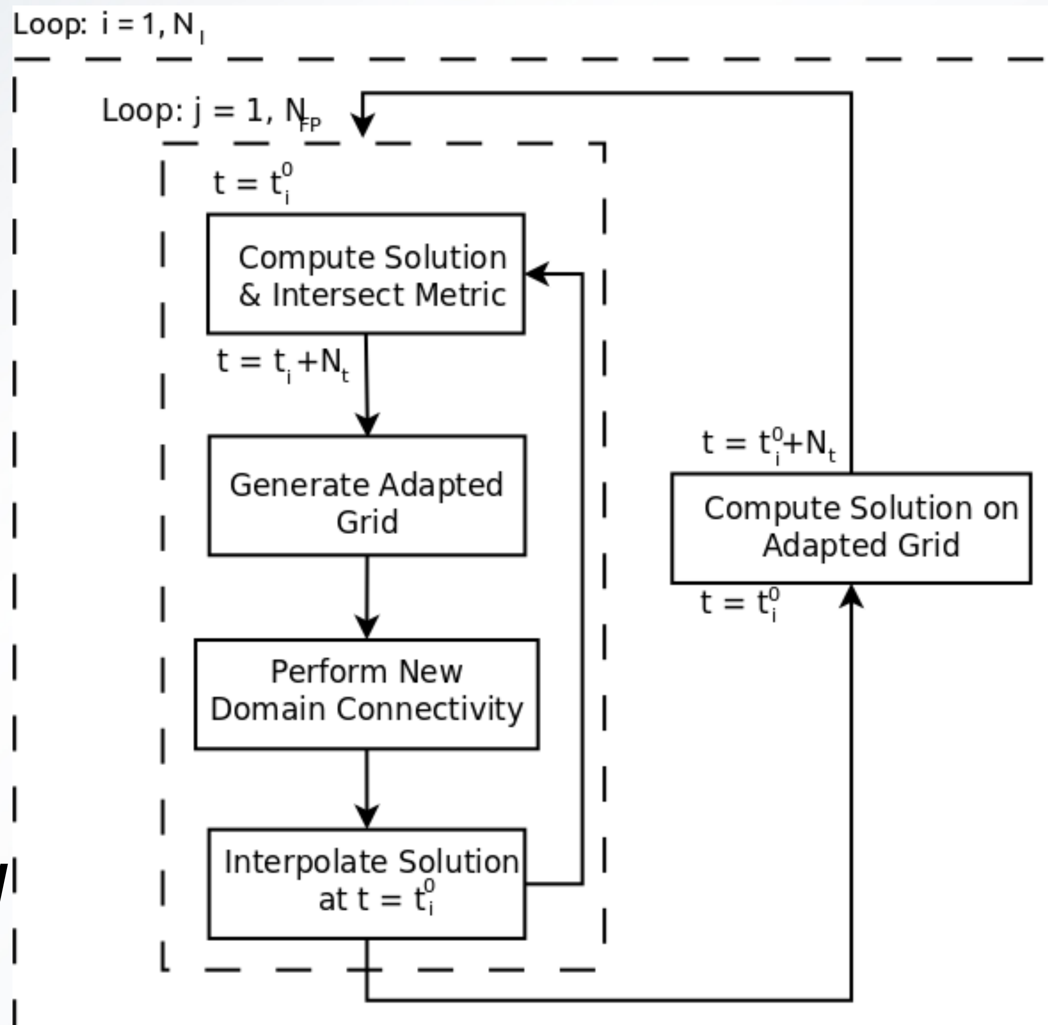
Rigid-body rotorcraft – $1/N_{blades}$ rev. after periodicity achieved



- General (non-periodic) interval adaptation
  - Adapted over several time intervals using a transient fixed-point algorithm (*Alauzet et al, AIAA 2007)*
  - For each interval, perform adaptation and solution transfers and compute improved solution on adapted grid

Transient-Fixed Point Algorithm

	Definition
N_{FP}	No. of fixed point iterations
N_I	No. of adaptation intervals
t_i^0	Start time of interval *i*



Unlike previous methods, back plane complexity is handled

Efficient Localization Scheme

- Developed a novel *parallel* scheme to efficiently search for interpolation stencils over massively distributed systems
- Compatible with mixed-element overset grids
- Uses collective communication (one thread per processor)
- Features that make the scheme fast:
  - Relies on **neighbor walks**, so searches are linear in space
  - **Parallel advancing front** keeps search space small *(new feature)*
- To avoid search failures, there are robustness features:
  - **Hierarchical prioritization** prevents search failure on realistic geometries *(new feature)*
  - **Random selections** to terminate cyclic searches
  - Defaults to **kd-tree method**, *Lynch et al. C&F 2014*

Box Grid Test Case

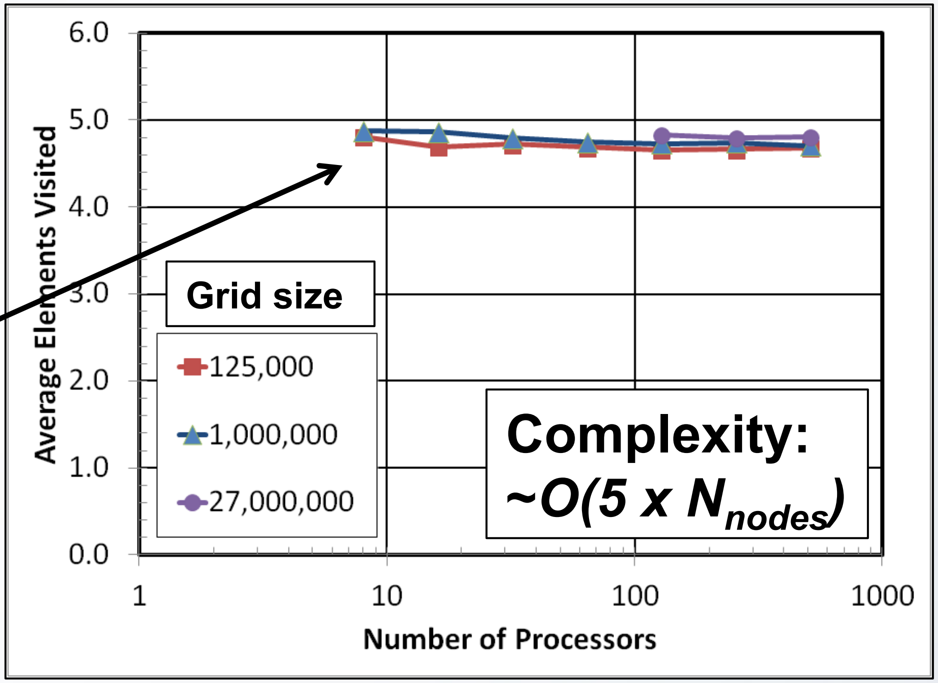
- Simple box domain with no geometry
- Test for cost and parallelization
- Three grids: 125k, 1M, and 27M nodes
- Number of processors: 8 … 512
- Averages done on five trials for consistency (randomness of neighbor walk)
- Assessments performed on NASA Langley's K cluster
- Monitor **complexity** and **required wall time**

Box Grid Localization Results

Serial complexity is $O(C \times N_{nodes})$ - Alauzet et al. (IJNM 2010)

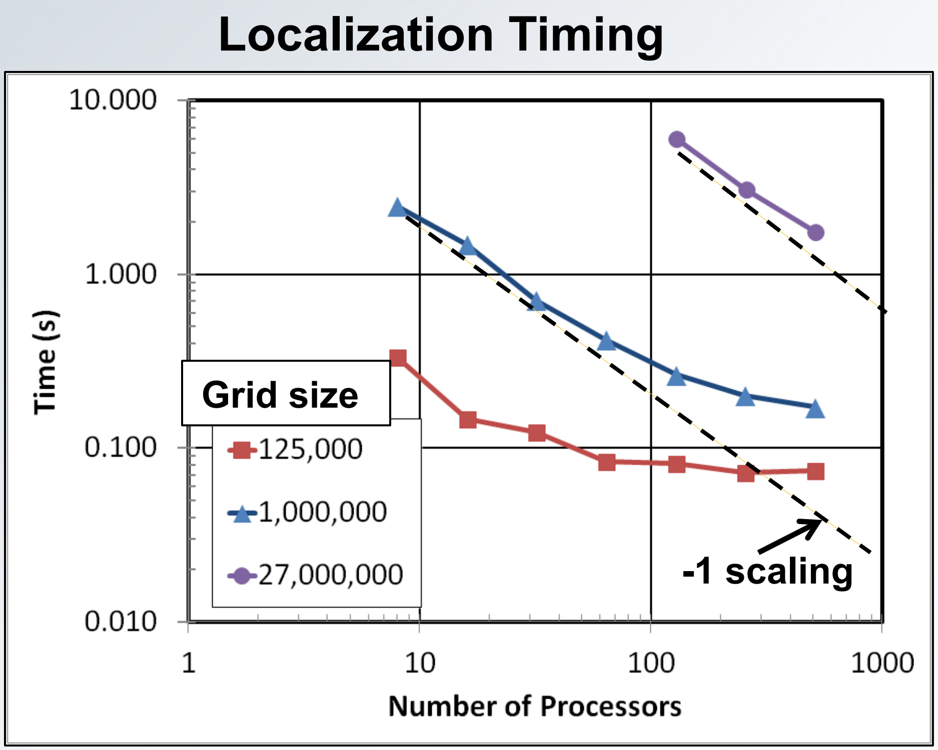
- *C* should be approximately the average number of elements surrounding a node
- Observed *C* is approx. 5 for all decompositions
- **Consistent parallel complexity** of the localization scheme is confirmed

Average number of searches



Nonlinear Computational Aeroelasticity Lab

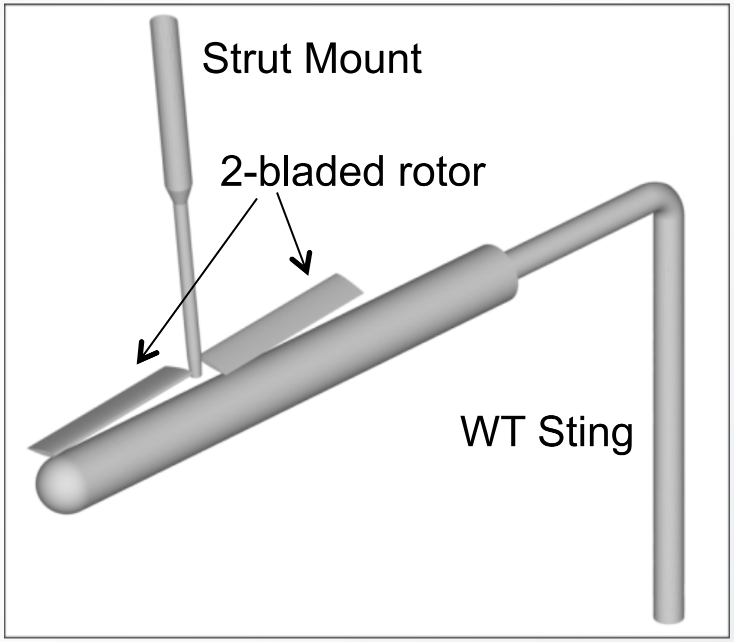
Box Grid Localization Results



- Time expected to decrease monotonically and ideal scaling should have a power -1
- Observed power is approx. **-0.89**
- Method is scalable with solver tested
- Localization is *cost effective* with respect to most processes that the solver handles

GT Rotor-Airfame Configuration

- Simple geometry to study RFI effects
- *Brand: PhD Thesis, 1989* for exp. data:
  - Time-averaged fuselage pressures
  - Instantaneous fuselage pressures

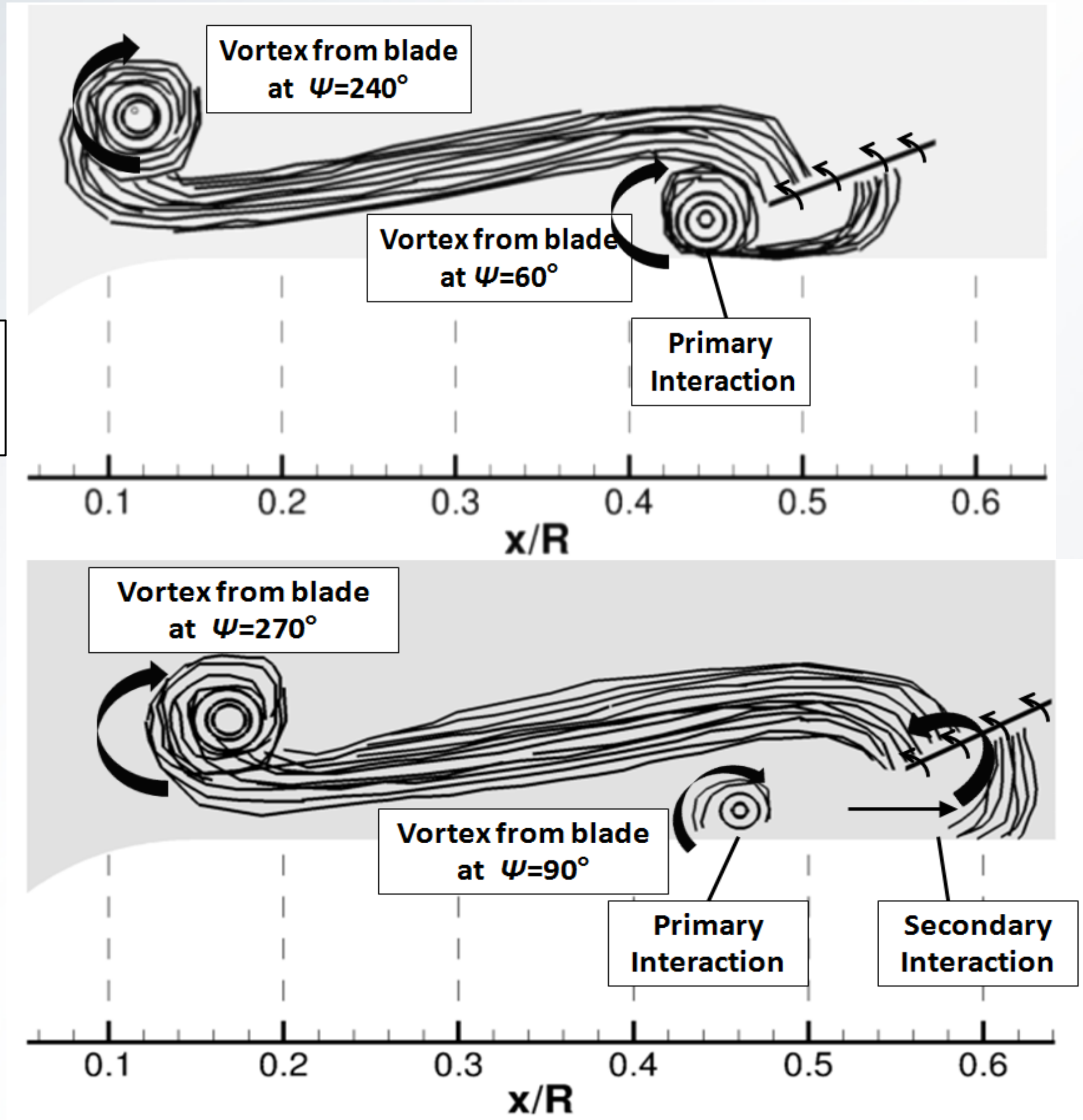


- Advance ratio of 0.10
- Time-step equivalent to 1º azimuth
- Hybrid RANS-LES computations
- Periodic time interval: 180º sweep
- Adaptation interval study

Method	Tolerance (F_{tol})
$\|\omega\|$	0.001
Δp	0.003
Nondim. Q-criterion	0.01

Vortex-Fuselage Impingement Physics

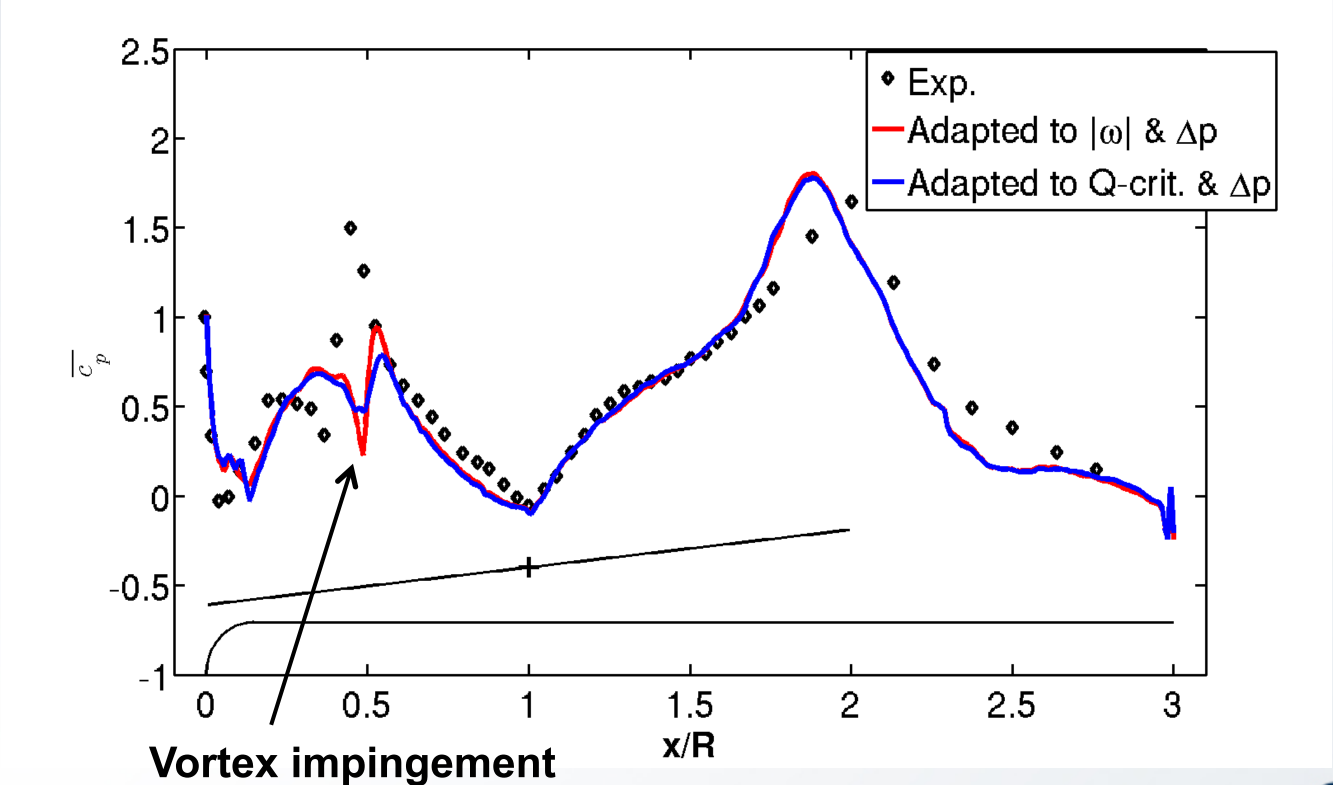
Sketch of Fuselage Symmetry Plane



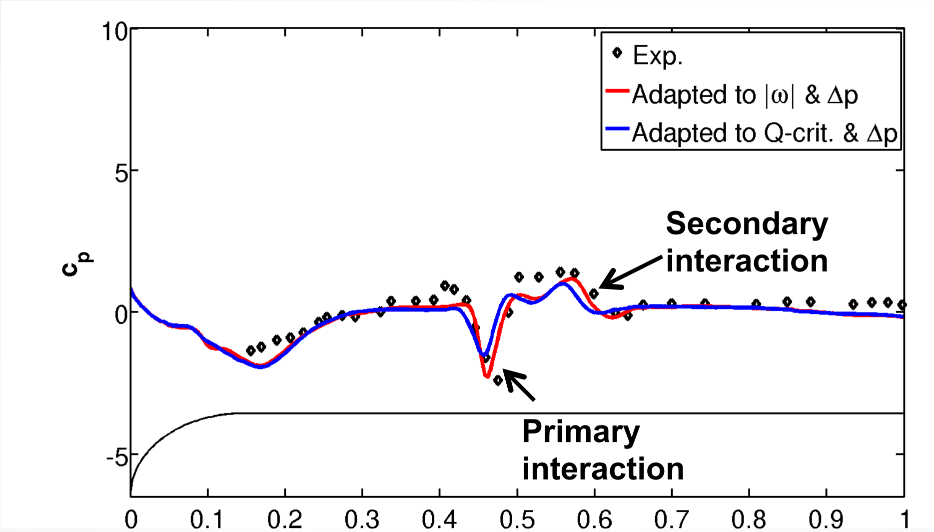
Average Pressures

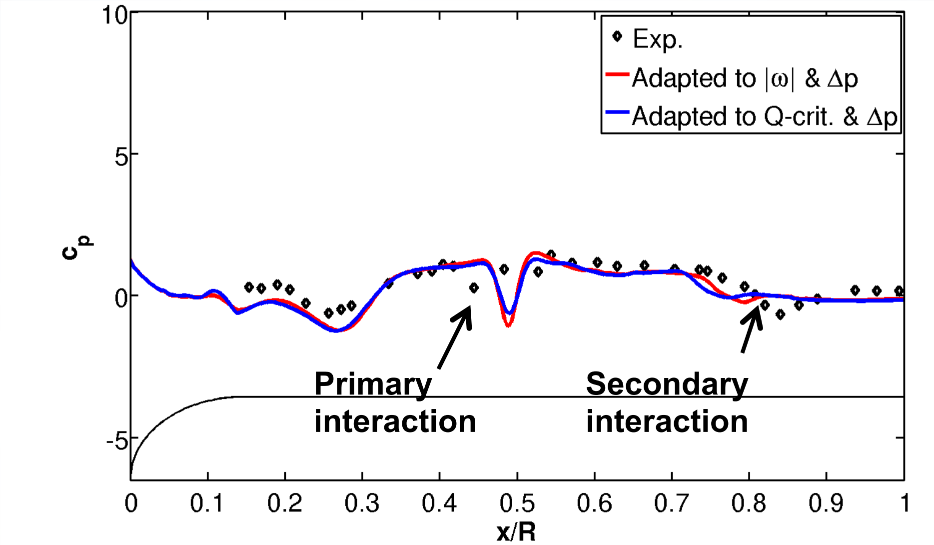
Upper fuselage centerline pressures are time-averaged

Vorticity mixed scheme: |ω| & Δp
Q-crit. mixed scheme: Q-crit. & Δp



Instantaneous Pressures

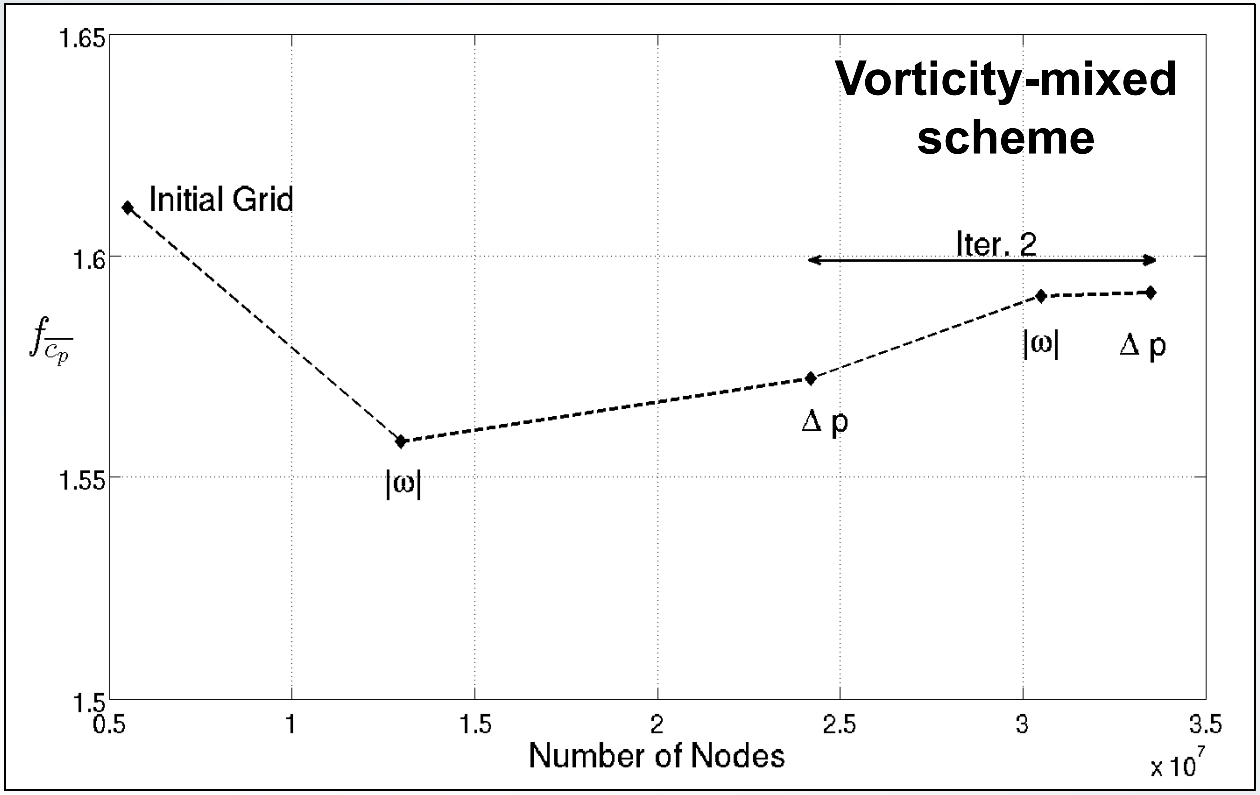




Adaptation Convergence

Integrated averaged centerline pressure coefficient

$$f_{\bar{c}_p} = \int_0^{3R} \bar{c}_p \, dx$$



Iter. 2: ~1.0% change

Adaptation Interval Sensitivity

- Investigated effect of time intervals:
  - 5° requiring 72 intervals/rev
  - 15° requiring 24 intervals/rev
  - Periodic (180°) adaptation – 1 interval same grid re-used
- Vorticity-mixed scheme used

Wall time/rev. (hours) on 480 processors

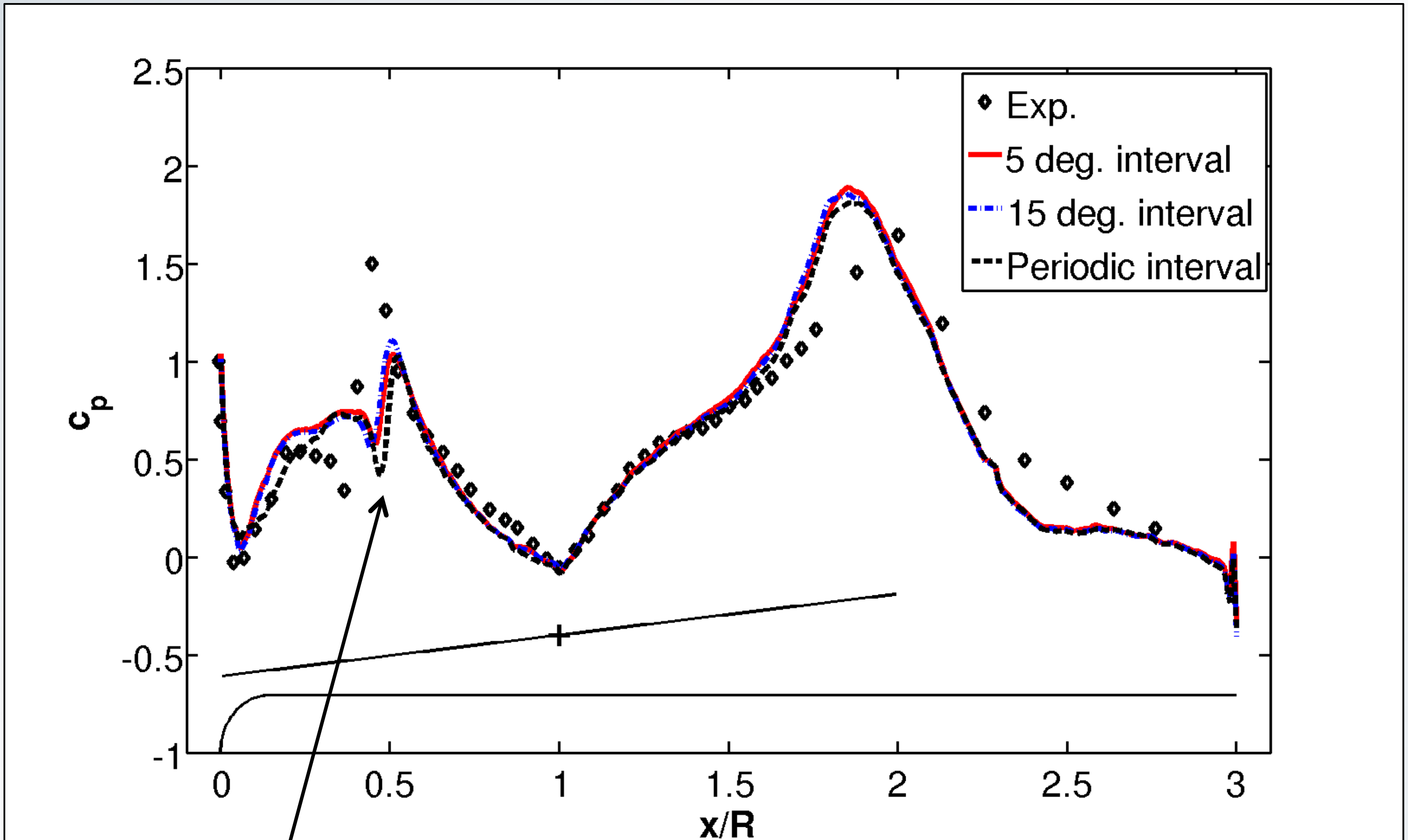
	5 deg	15 deg	Periodic
Flow Solver	24.0	22.8	16.7
Adaptation & Interpolation	7.2	4.2	0.3
Domain Connectivity	13.2	4.8	1.2
Total	**44.4**	**31.8**	**18.2**

2.4x *cost of periodic case*

1.75x *cost*

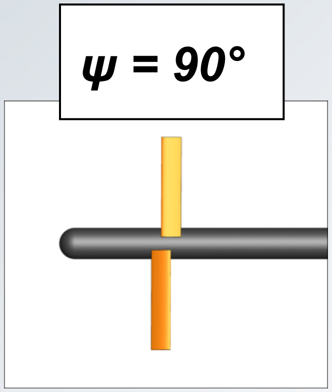
Note: Substantial cost increase due to overhead tasks; may be better streamlined

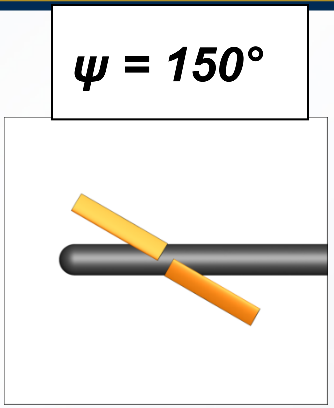
Interval Sensitivity (Average Pressures)

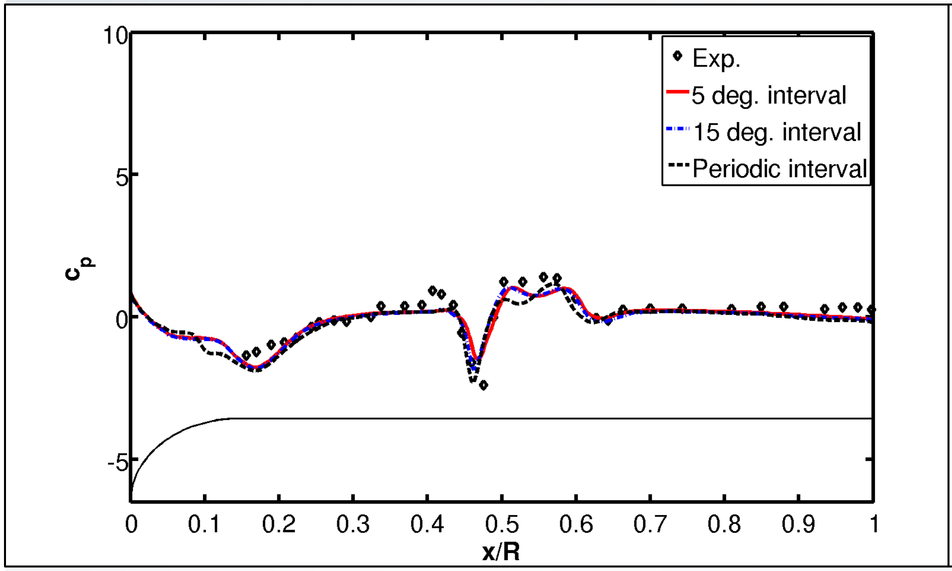


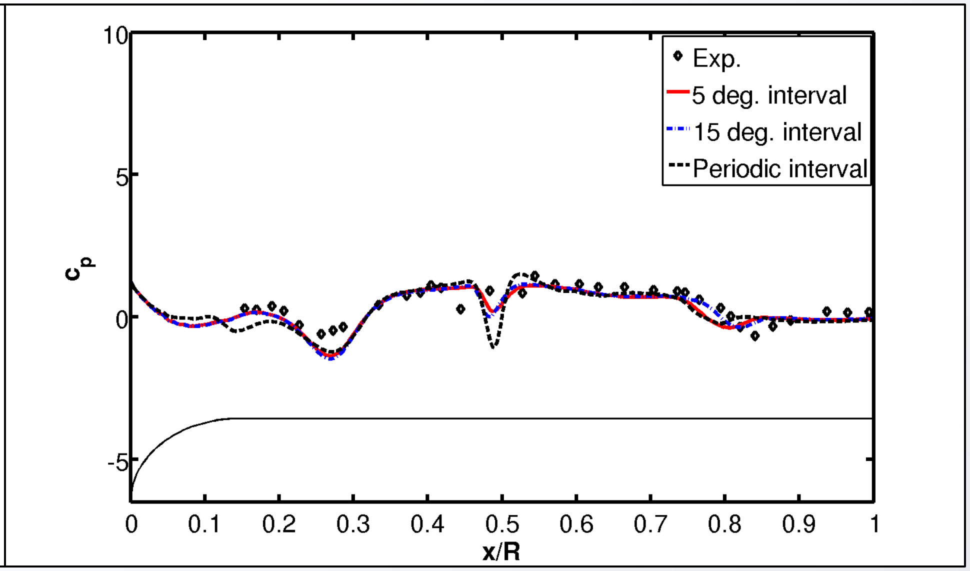
Minor difference in impingement location

Interval Sensitivity (Instantaneous Pressures)



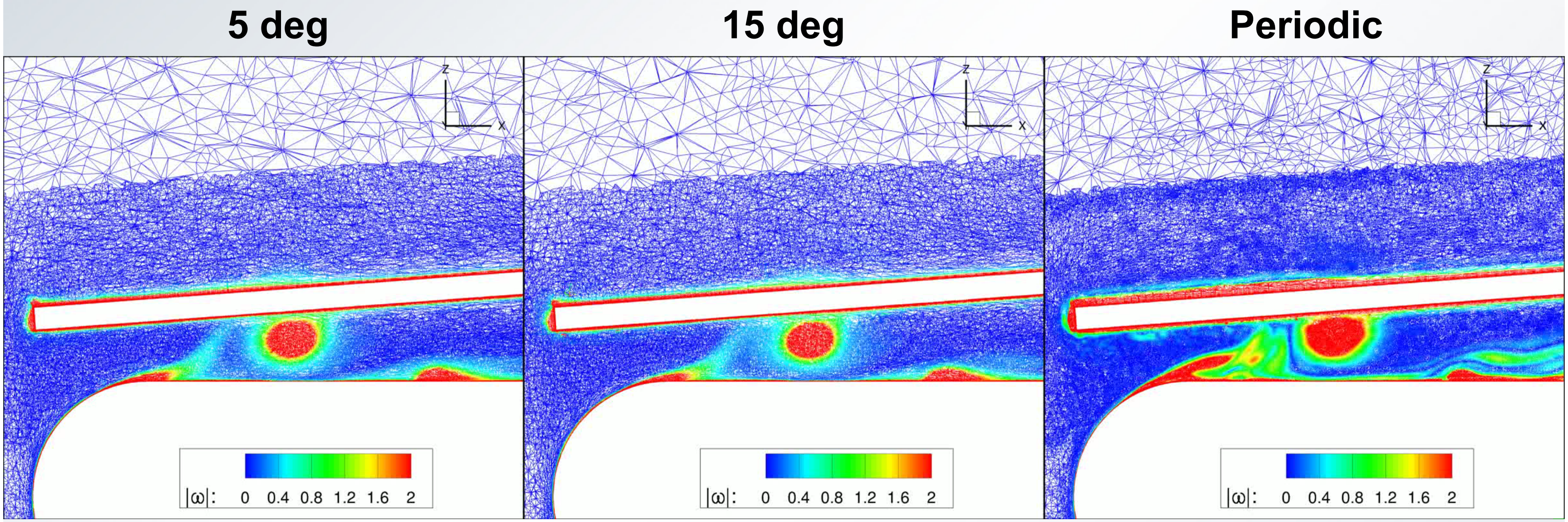






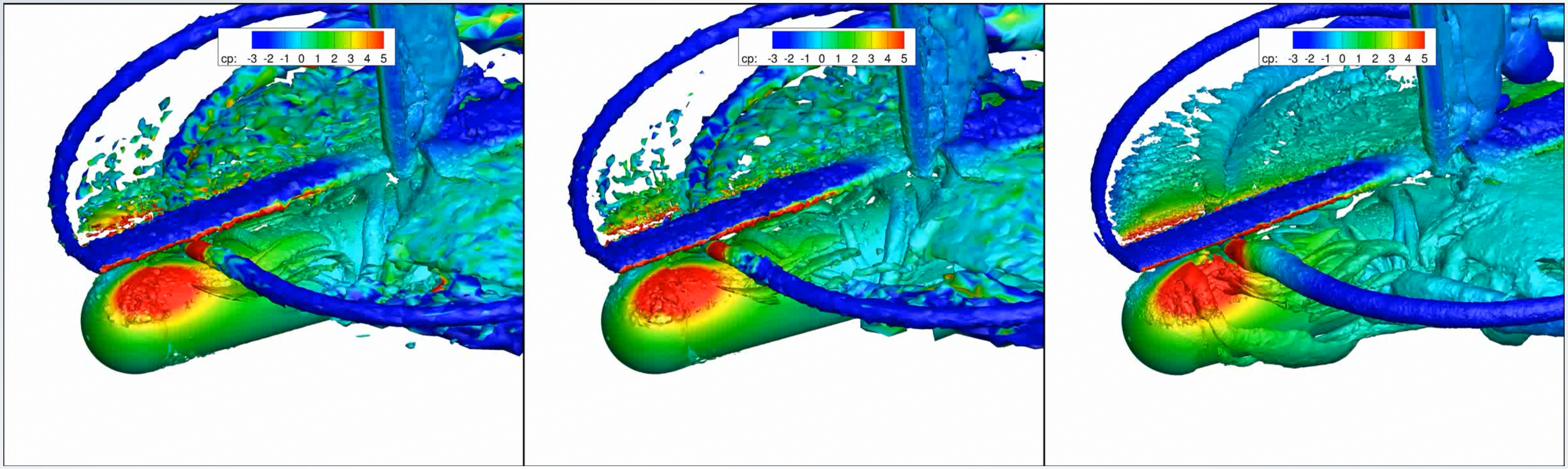
Consistent agreement and small differences in location and magnitude observed

Vorticity Mesh Contours



Q-Criterion Iso-Contours

5 deg 15 deg Periodic



More Information

All GT Theses and many datasets/presentations are freely available at:

https://smartech.gatech.edu/

(once you create an account – it is free, but you must register)

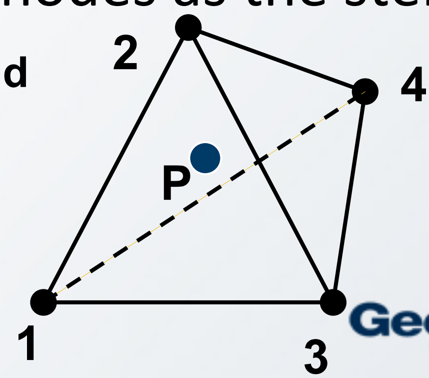
Rajiv Shenoy's Thesis:

https://smartech.gatech.edu/handle/1853/51796

Localization & Interpolation

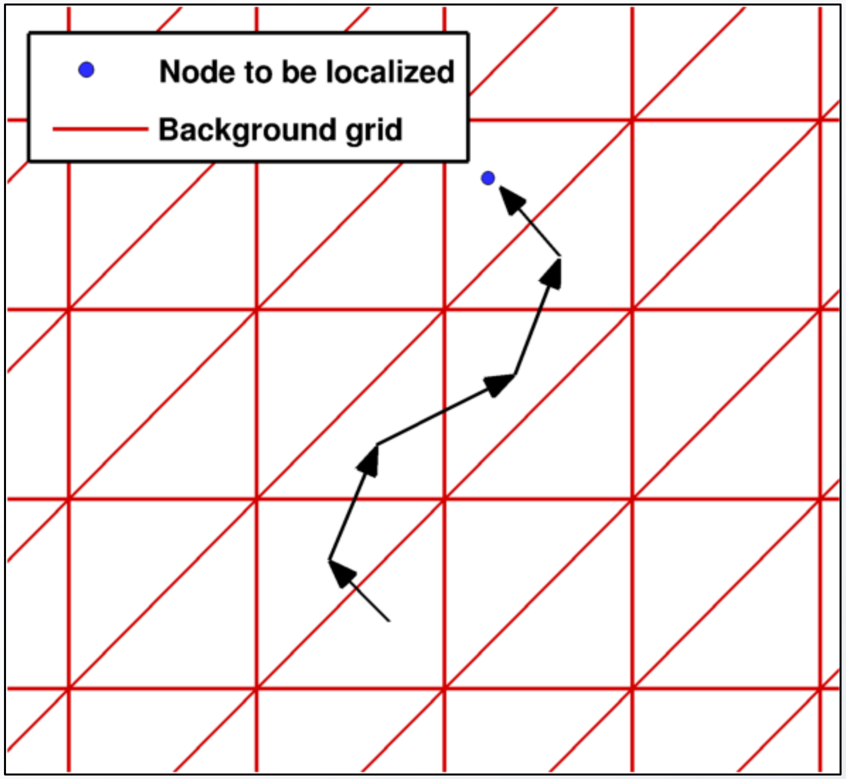
- **Background Grid:** Solution is known on this grid
- **New Grid:** Solution needs to be transferred to this grid
- **Localization:** Search process of a new grid node to find an enclosing element of background grid
- **Barycentric coordinates** (β_i) provide a natural basis to localize a node to a tetrahedral element
  - Provides information for search path (next step)
  - Serve as weights using the 4 tetrahedron nodes as the stencil
- **Interpolation:** $\phi_I(P) = \sum_{i=1}^{4} \beta_i \phi_i$

Interpolated solution: $\phi_I(P)$; Weights: β_i; Background solution: ϕ_i



Neighbor Walk

- Barycentric coordinates provide means of making neighbor steps
- Make steps until node is localized
- If multiple neighbor choices exist, then a step is randomly selected
  - Prevents infinite cyclic walks
- When cyclic walks are not avoidable, the search can default to a kd-tree search for those nodes

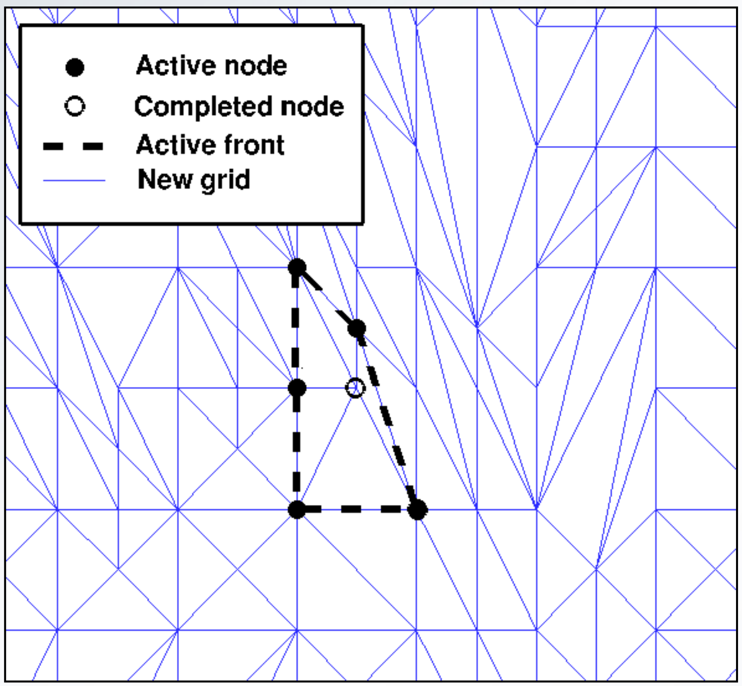


Requires background grid's list of cell adjacencies

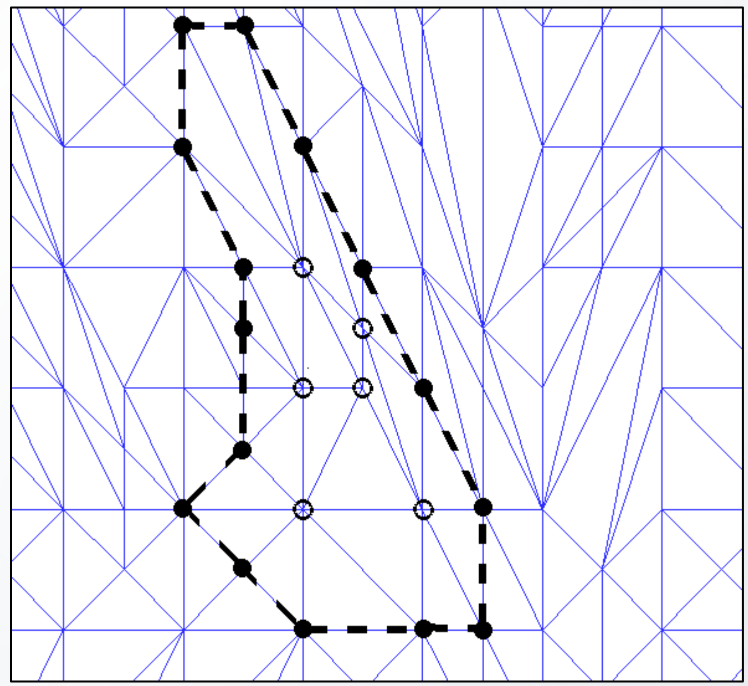
Advancing Front

- Enclosing element becomes a guess for node neighbors
- Subsequent searches become increasingly confined

Seeding the front with 1 node



Advancement of front



Requires new grid's list of node neighbors

Parallelization of Localization Scheme

- Relies on collective communication (MPI)
- Keep track of the *guess element* and *partition number*
- What happens if the walk hits a partition boundary?
  - Communicate to neighboring partition (processor)
  - Information about a boundary node is provided
  - On that partition, one element surrounding that node is *randomly* selected as the guess element and the neighbor walk continues
- Once a node is localized, its **enclosing element**, **partition**, and **weights** are stored for interpolation

Robustness Features

- These searches are linear and can potentially encounter geometry boundaries especially for realistic problems
- Therefore localize nodes in hierarchical fashion:
  - Corner nodes: where three boundary faces coincide
  - Edge nodes: where two boundary faces coincide
  - Surface and volume nodes (bulk of the grid)
- Use kd-tree search method, *Lynch et al. C&F 2014*, to localize **corner** an**d edge nodes**, generally < 1% of grid
- Surface and volume nodes are then localized using the parallel advancing front scheme

Other Enhancements

- Handling mixed-element grids
  - Non-tetrahedral elements are used in the boundary layer
  - Can use barycentric approach by converting elements into tetrahedra (only data structures)
- For overset grids, localize each component grid