Advanced Modeling & Simulation
(AMS) Seminar Series

01 Nov. 2018



Strand Mesh Generation Using Minimum Distance Fields



TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Beatrice Roget **Vinod Lakshminarayan** **Jay Sitaraman**

Science and Technology Corporation Parallel Geometric Algorithms, LLC

Andy Wissink

Aviation and Missile Research, Development and Engineering Center
Moffett Field, CA

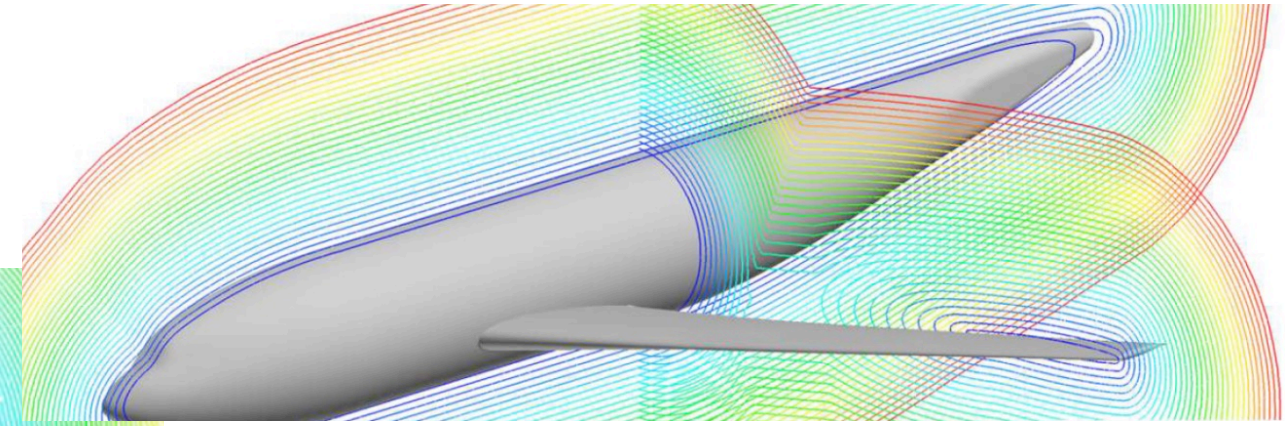
Background

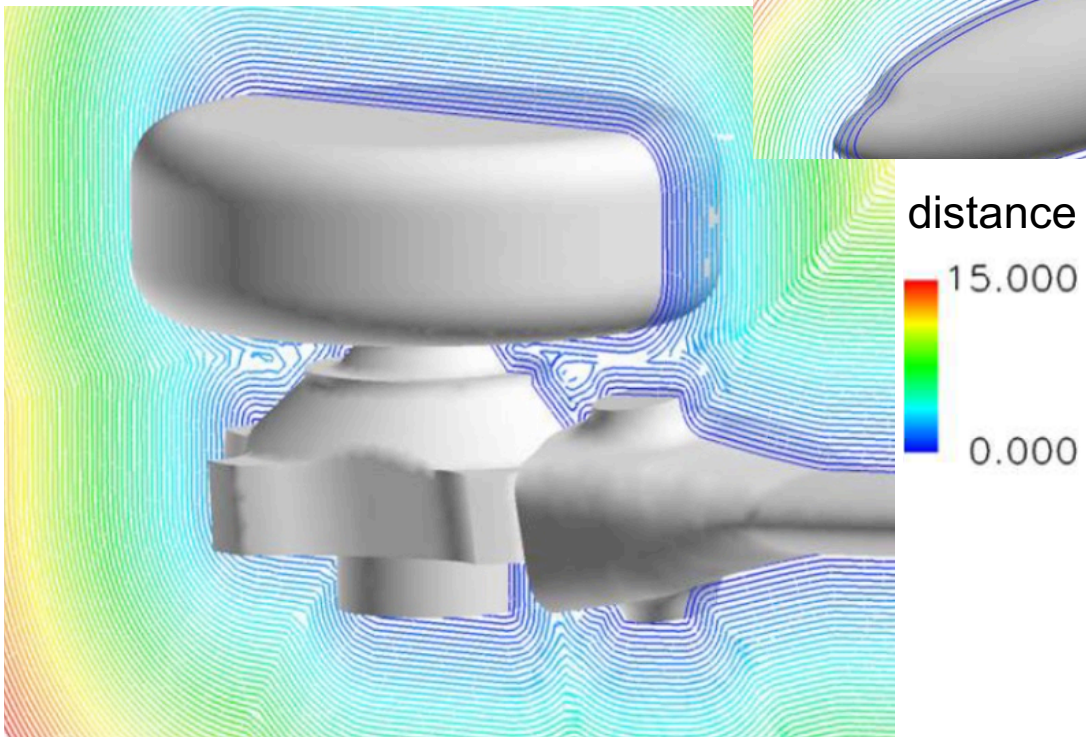
- **ICCFD 7 paper, July 2012, Journal of Computational Physics**

  "Wall Distance Search Algorithm Using Voxelized Marching Spheres"

  B. Roget and J. Sitaraman

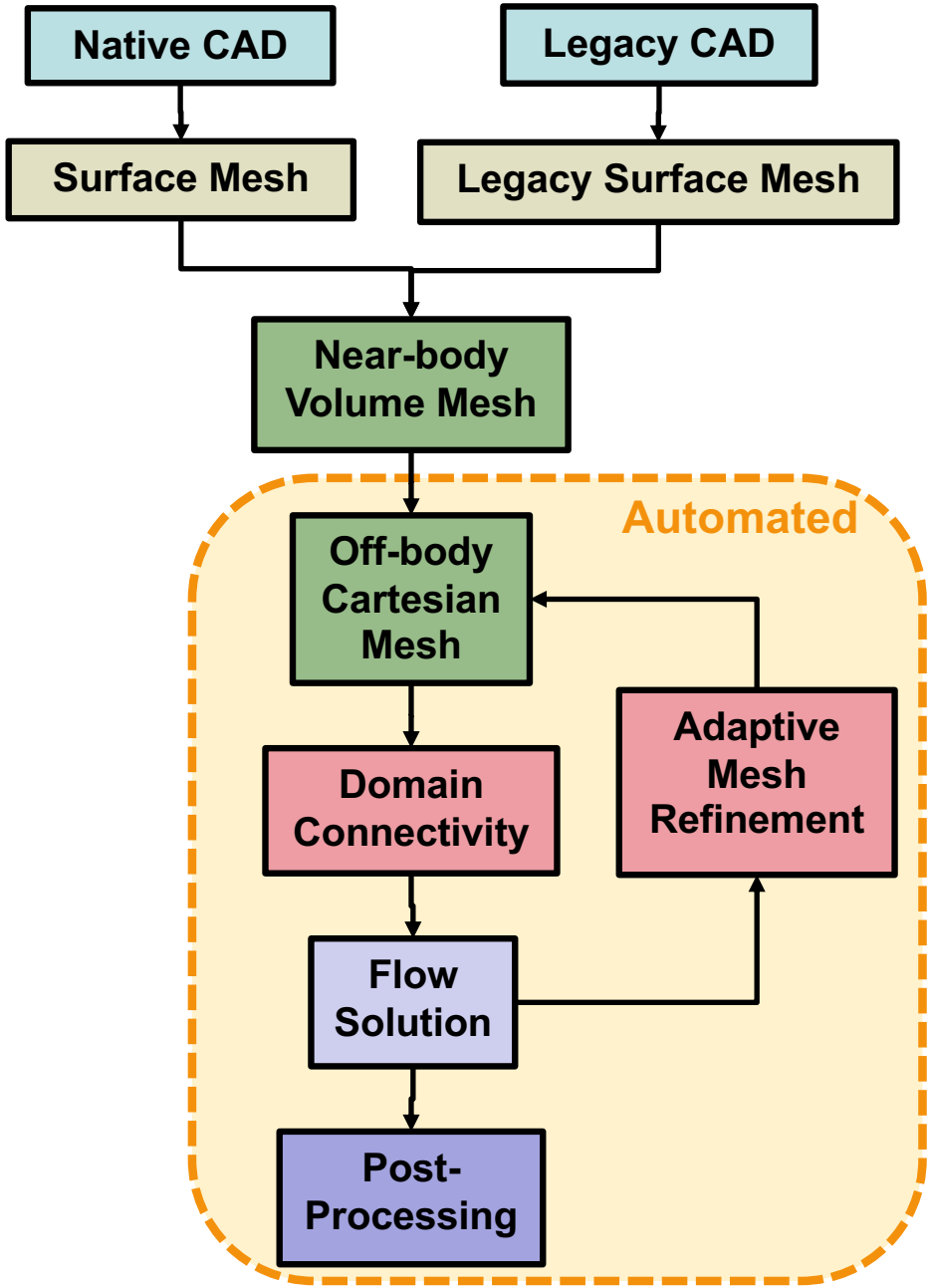
  - Minimum distance to a solid wall (used in turbulence models)
  - Iso-surface of distance from a surface:



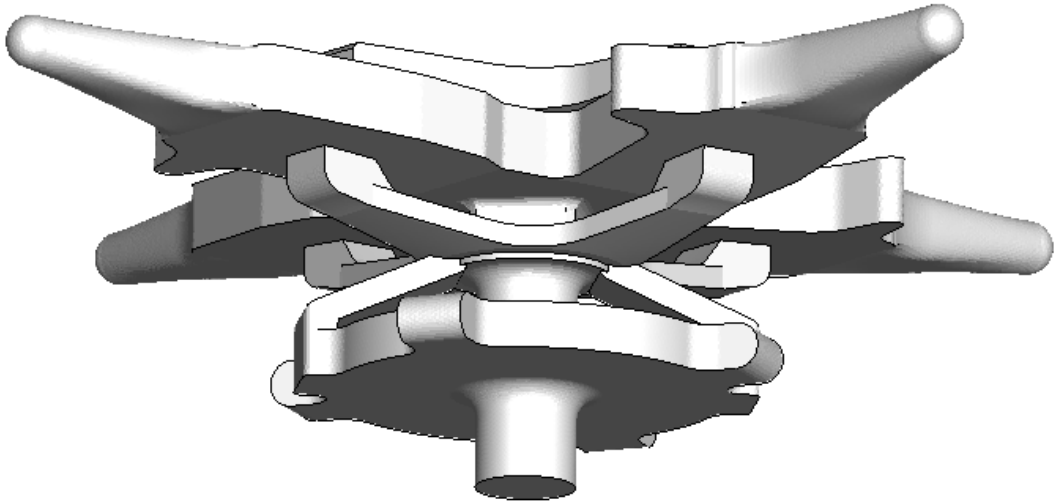


  - Could we apply the iso-surface of distance concept to **mesh generation** ?

Automation



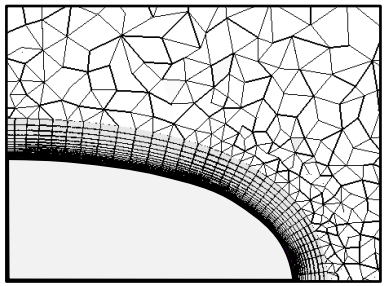
- Meshing automation = top basic research topic in CFD
- High-fidelity rotorcraft simulation without CFD or meshing expertise
- Most tasks fully automated, except near-body meshing



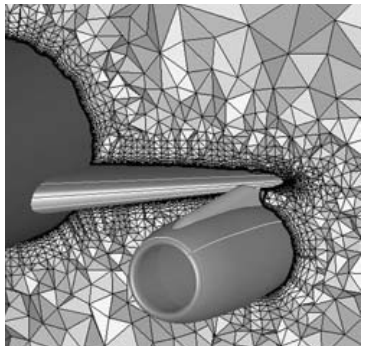
Introduction

Anisotropic mesh required to capture boundary layer flows

- **Advancing front methods:**
  for each prismatic layer,
  - ➢ follow normal direction
  - ➢ check for intersection, merge/fix
    - mostly serial & expensive
    - most robust method currently

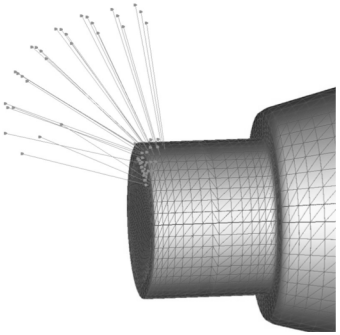


AFLR
Marcum et al.

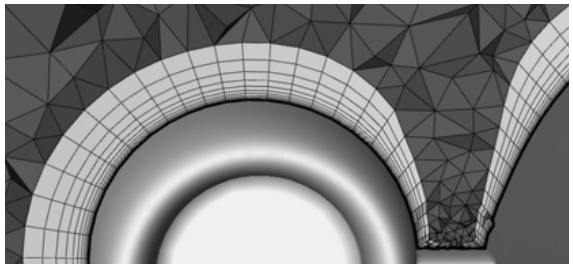


Pointwise

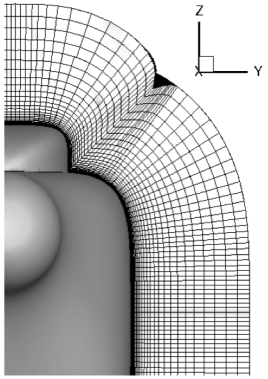
- **Point Placement methods:**
  create directly prismatic envelope
  - ➢ strand meshes
  - ➢ various techniques used for point placement (smoothing, elastic analogy, optimization)



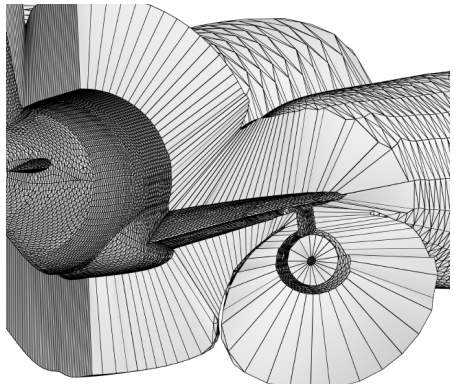
Meakin 2007
(Strand mesh)



Tomac 2013
(SUMO)



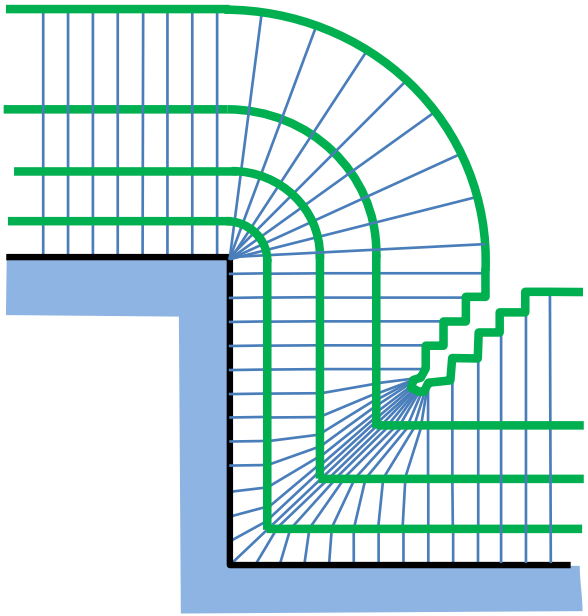
Haimes 2014
(MOSS)



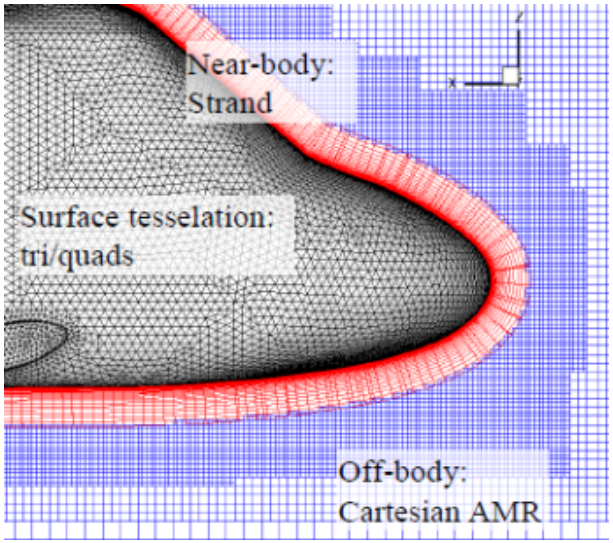
Garanzha 2017

Stand/Cartesian Grid Framework

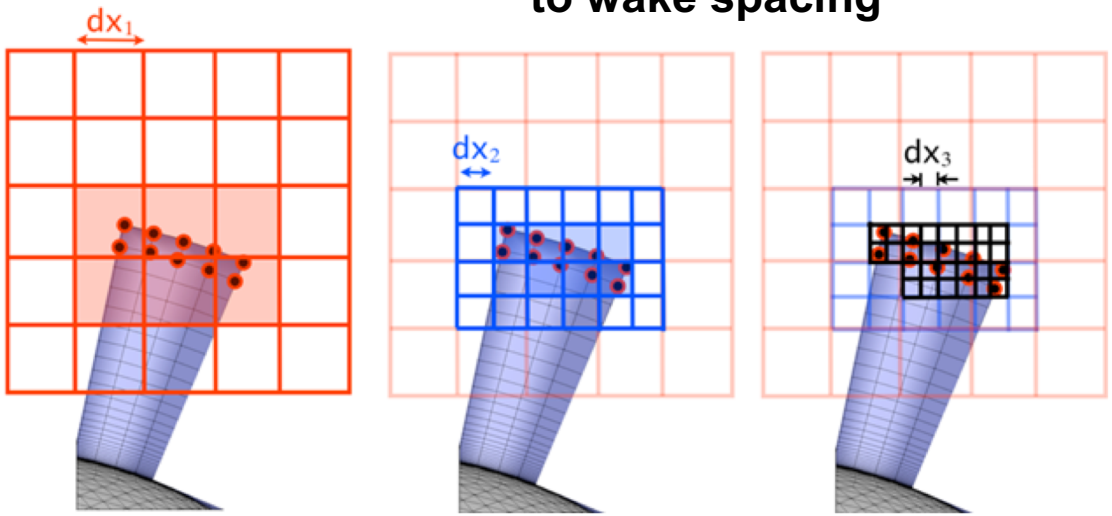
- **Modified Strand mesh**
  - multiple-strands per node
  - multiple levels
  - self-intersection allowed for last level, cleared by domain connectivity module



- **Overset Cartesian Mesh**

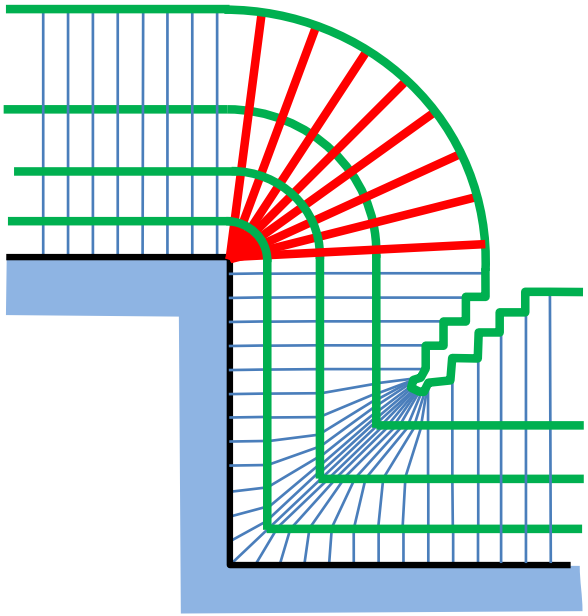


automatic refinement to wake spacing

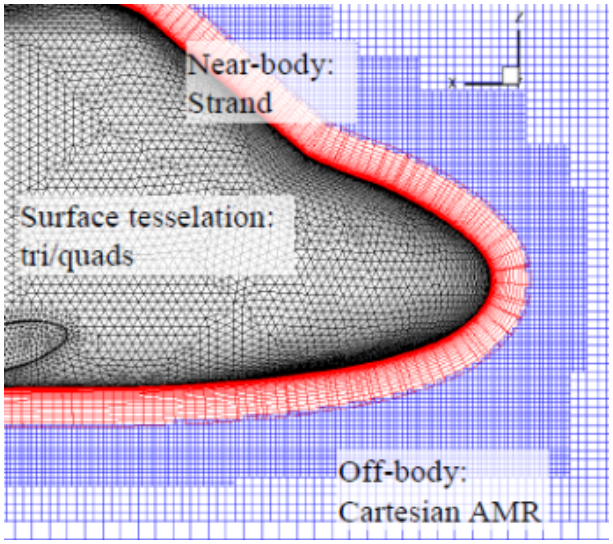


Stand/Cartesian Grid Framework

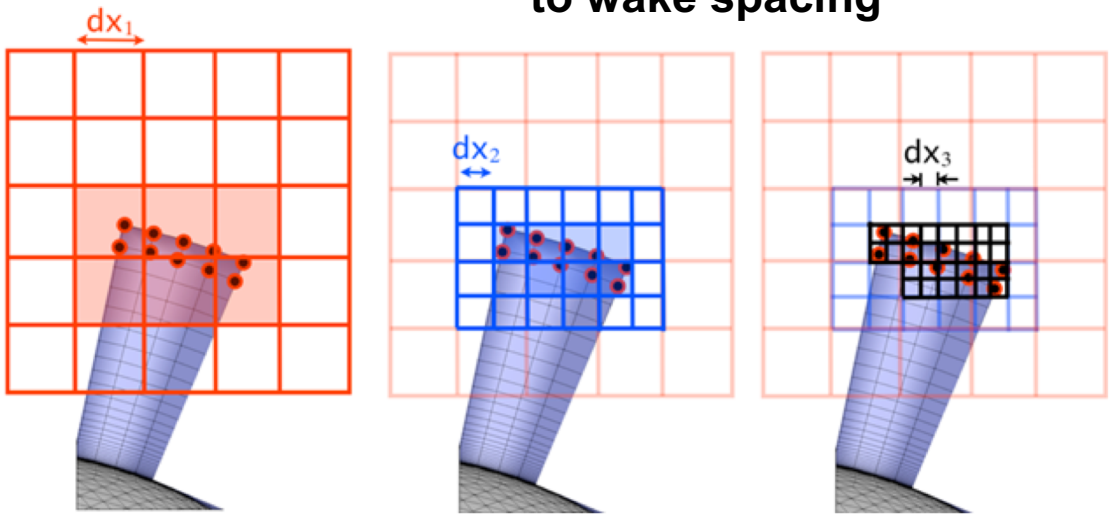
- **Modified Strand mesh**
  - **multiple-strands per node**
  - **multiple levels**
  - **self-intersection allowed for last level, cleared by domain connectivity module**



- **Overset Cartesian Mesh**

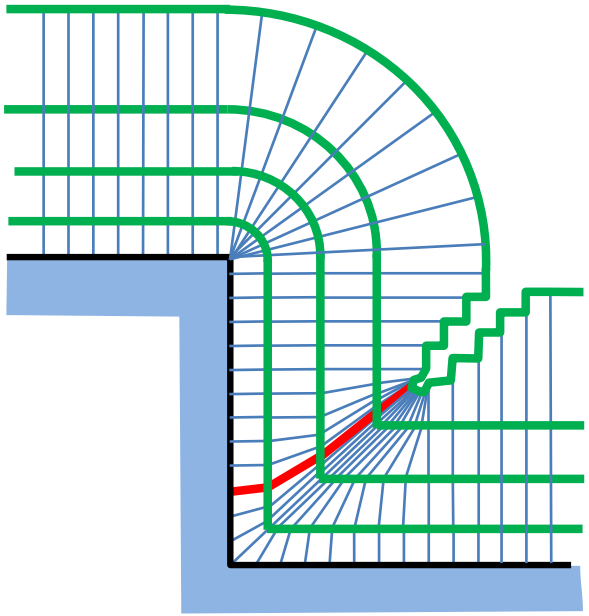


automatic refinement to wake spacing

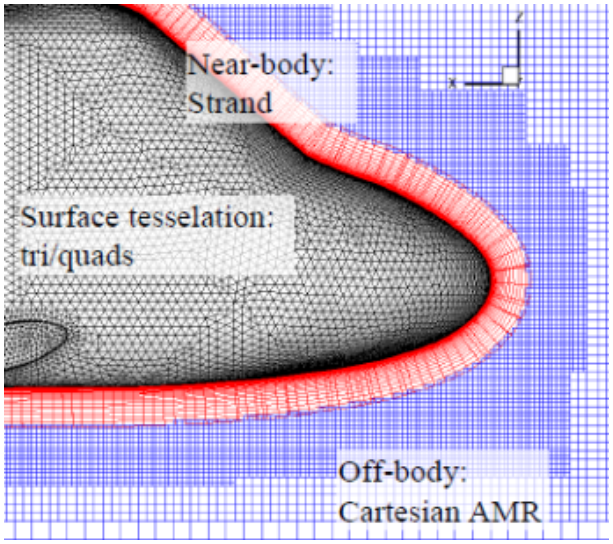


Stand/Cartesian Grid Framework

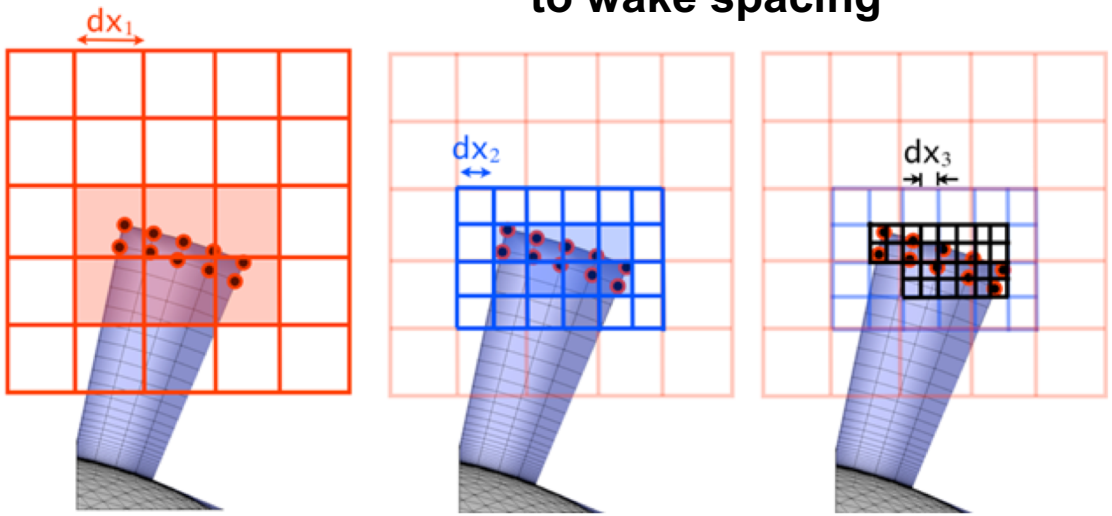
- **Modified Strand mesh**
  - multiple-strands per node
  - **multiple levels**
  - self-intersection allowed for last level, cleared by domain connectivity module



- **Overset Cartesian Mesh**

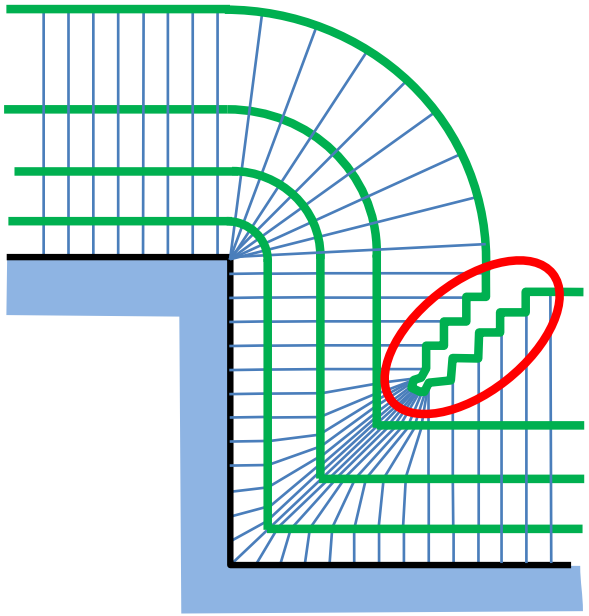


automatic refinement to wake spacing

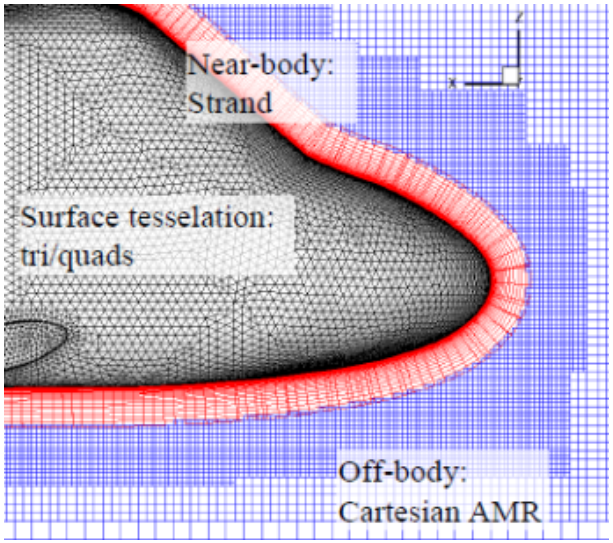


Stand/Cartesian Grid Framework

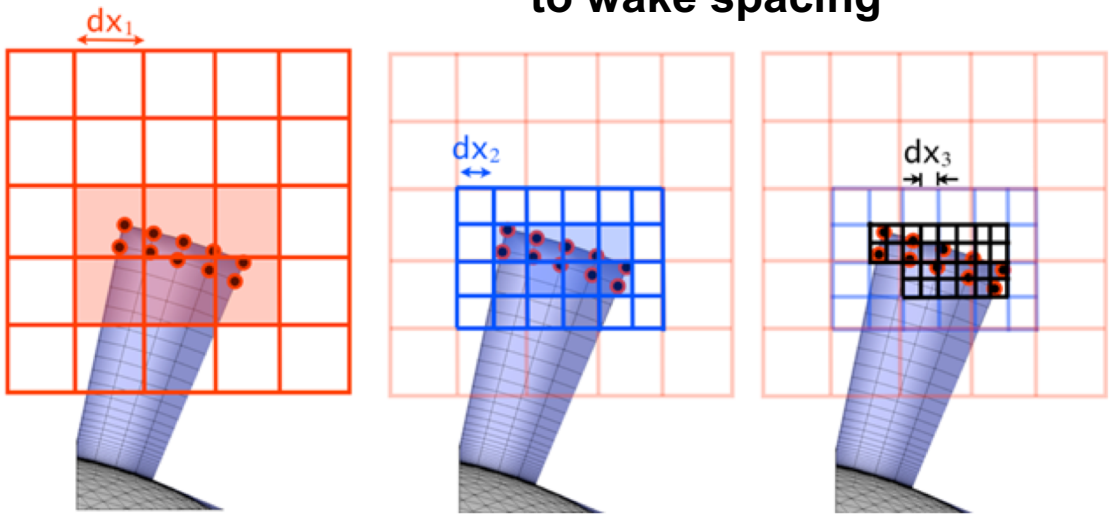
- **Modified Strand mesh**
  - multiple-strands per node
  - multiple levels
  - self-intersections allowed for last level, cleared by domain connectivity module



- **Overset Cartesian Mesh**

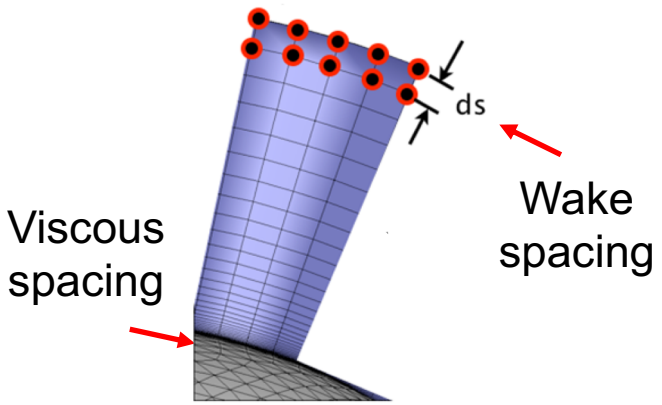


automatic refinement to wake spacing

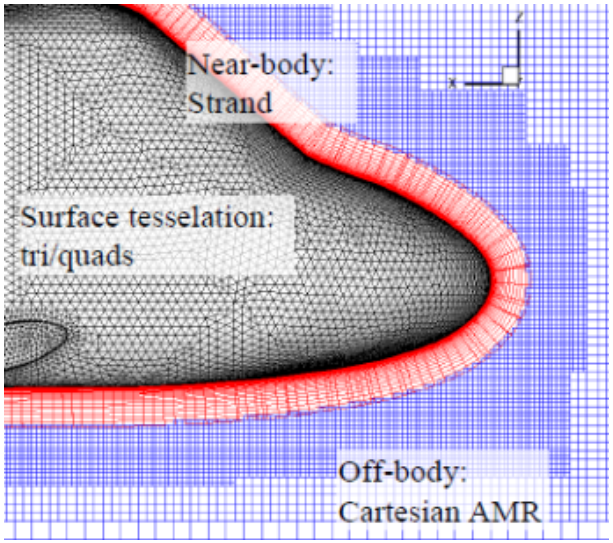


Stand/Cartesian Grid Framework

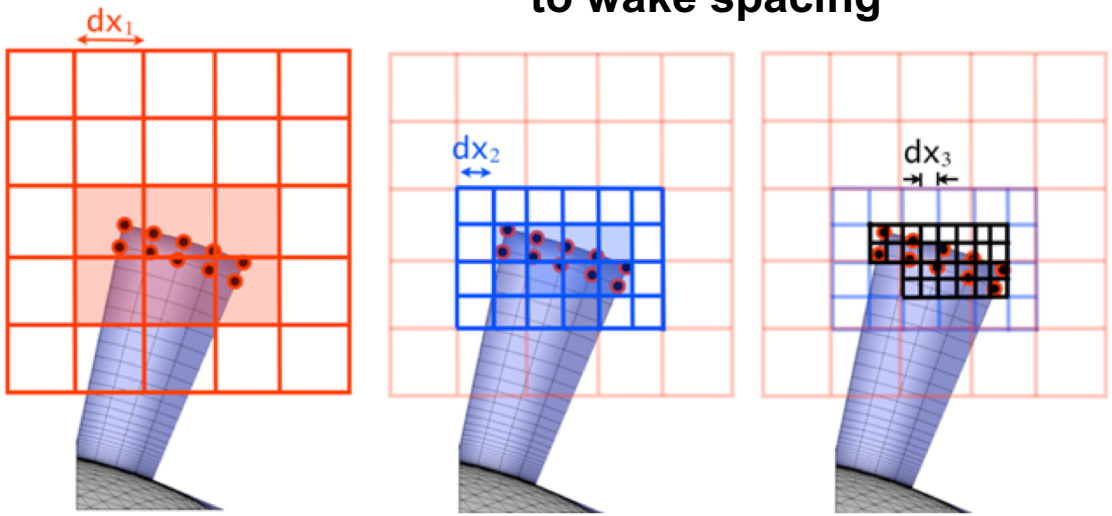
- **Modified Strand mesh**
  - **multiple-strands per node**
  - **multiple levels**
  - **self-intersections allowed for last level, cleared by domain connectivity module**



- **Overset Cartesian Mesh**

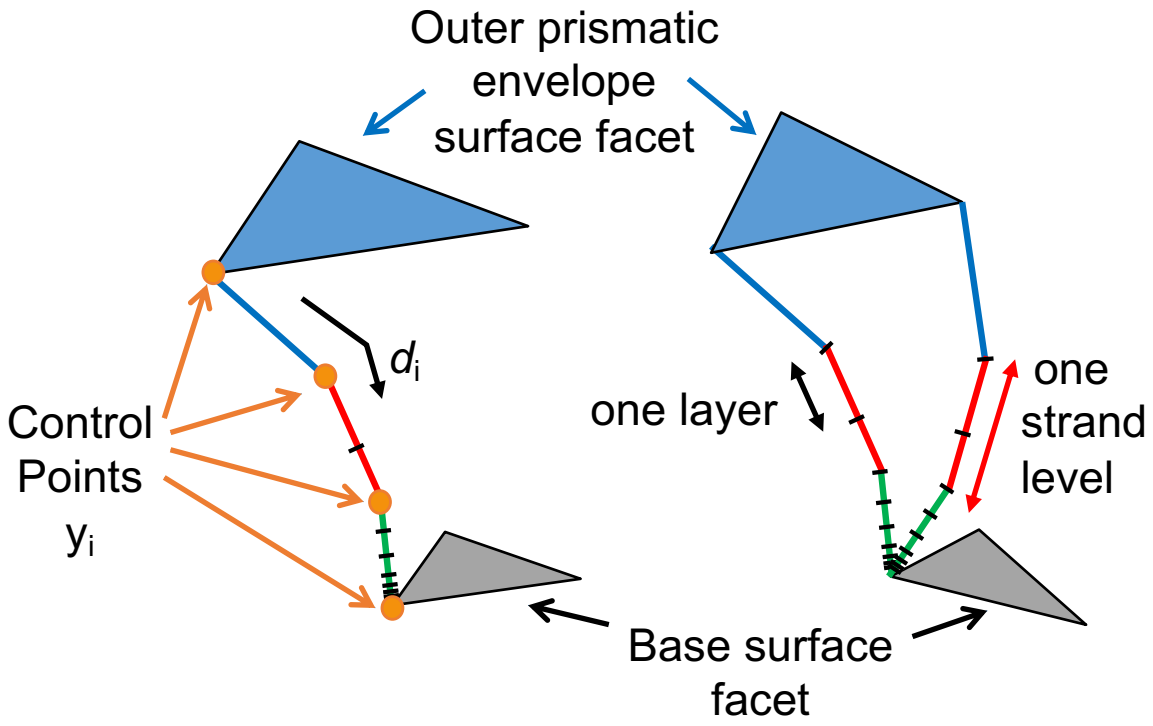


automatic refinement to wake spacing



Strand Mesh Representation

- **Connectivity graph of outer prismatic envelope**
- **Coordinates y_i**
- **Number of control points for each node on envelope**
- **normal distribution d_i**



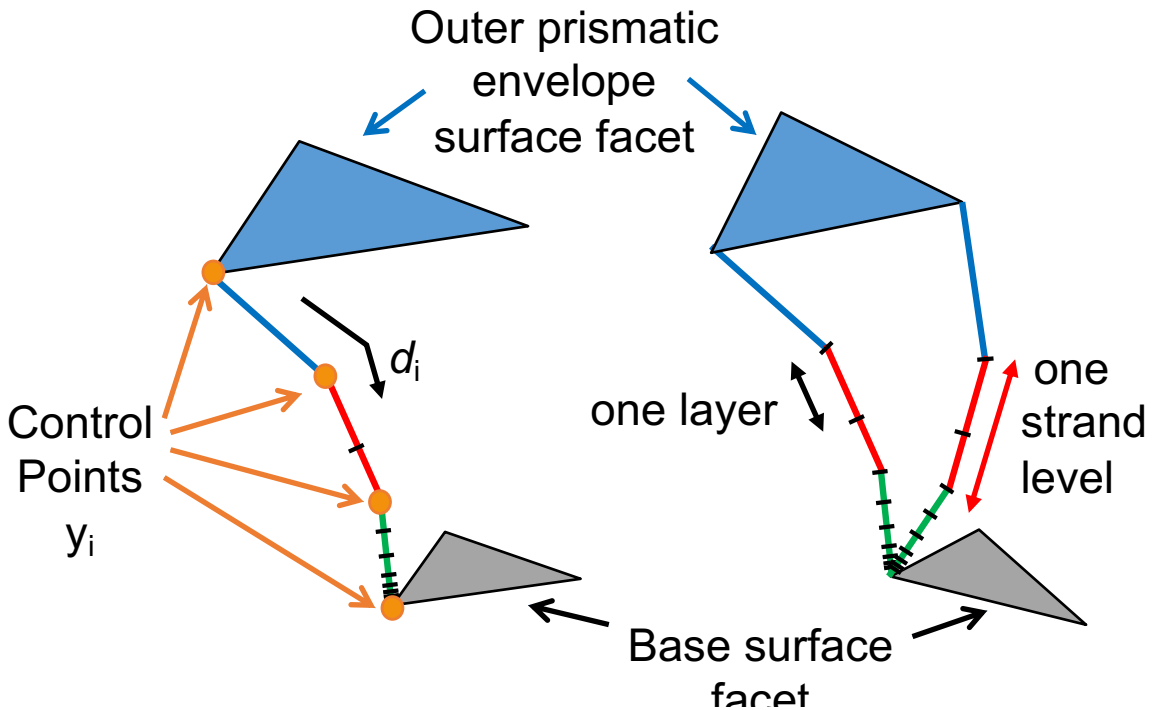
→ **Compact Storage**

Example: 50 million node grid:

- **Unstructured → 4.2 GB**
- **Compact Strand → 0.1 GB (42x less)**

Strand Mesh Representation

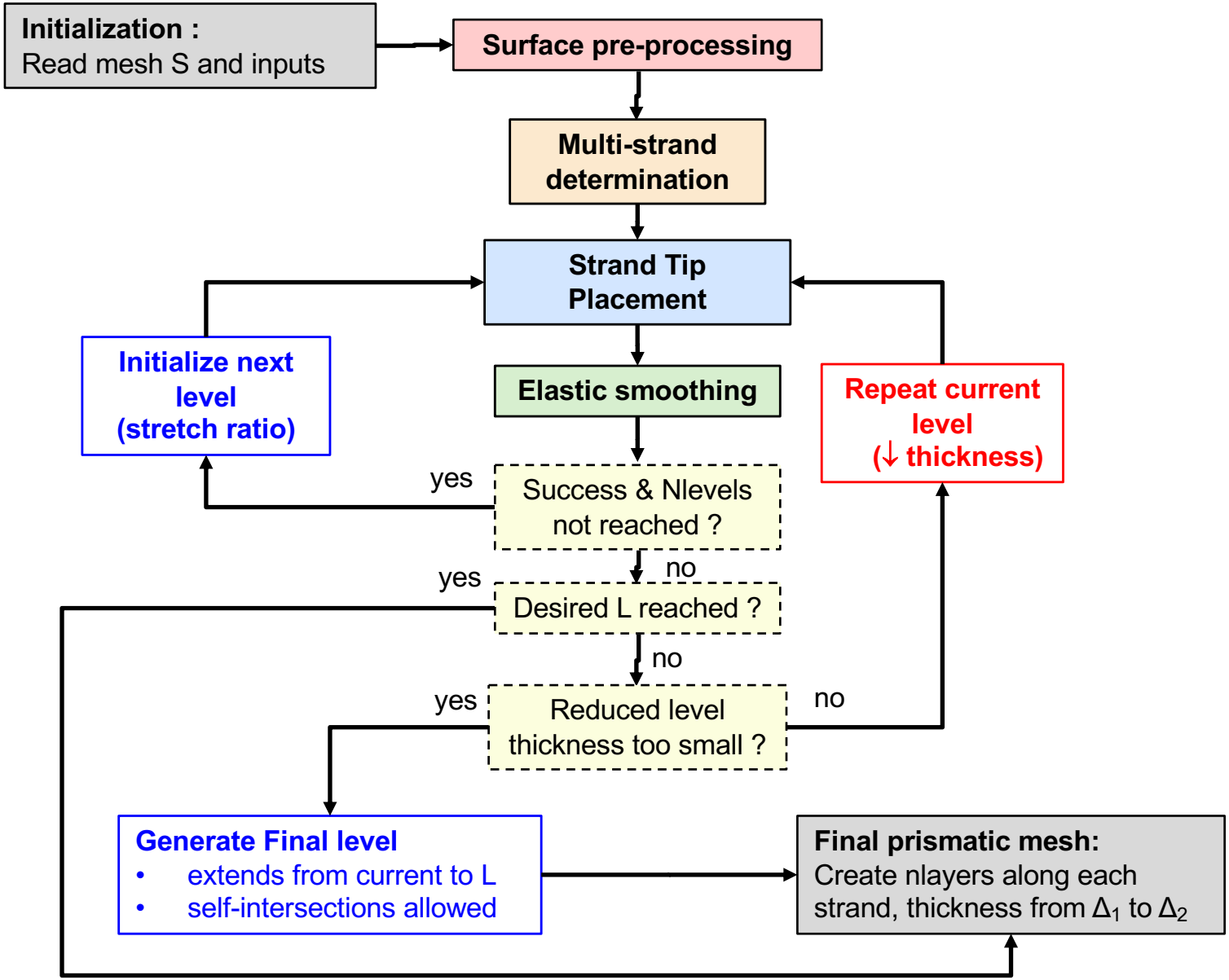
- Connectivity graph of outer prismatic envelope
- Coordinates y_i
- Number of control points for each node on envelope
- Normal distribution d_i



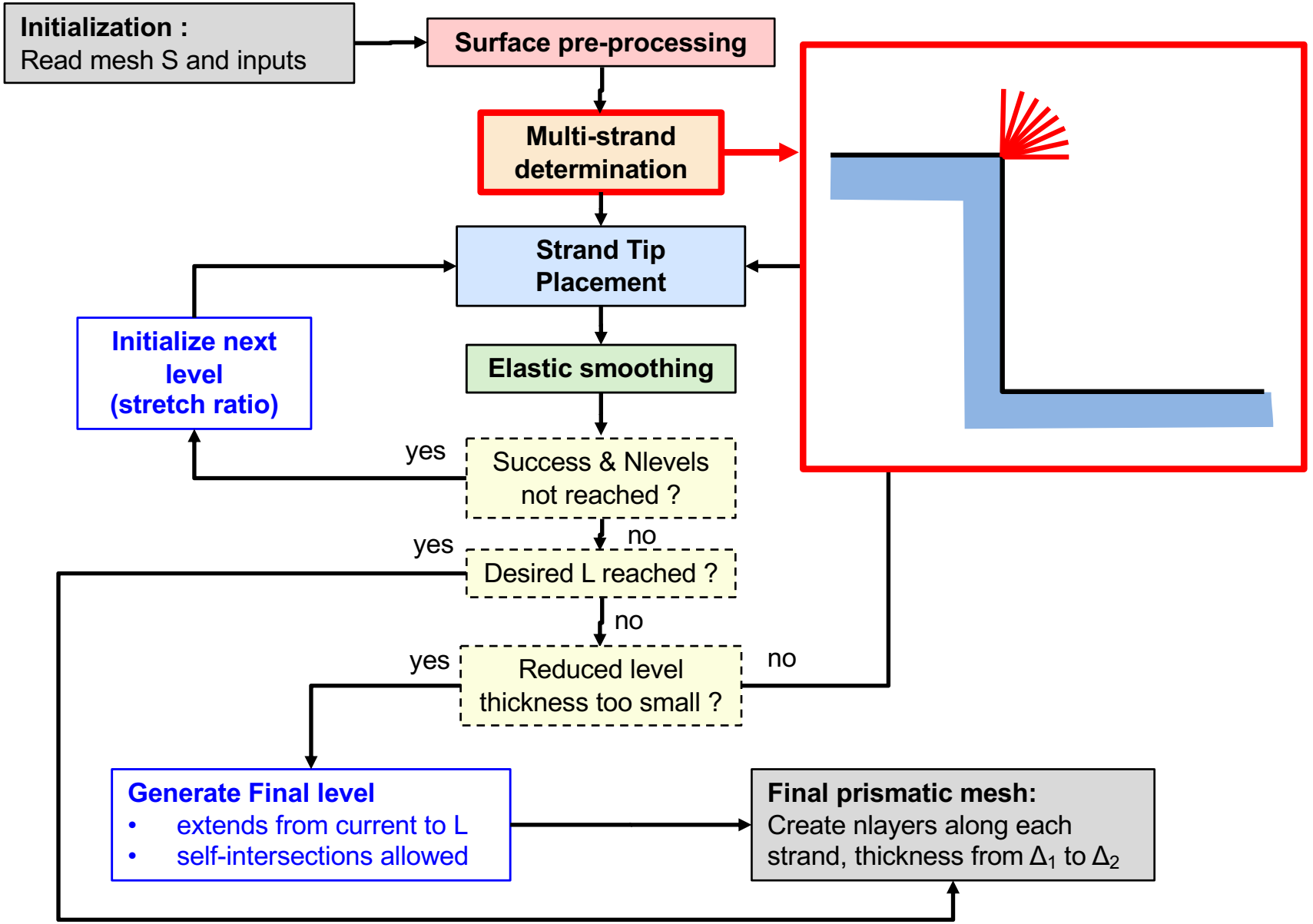
- **Advantages of strands meshes:**
  - **Scalable Domain Connectivity (compact mesh in each proc.)**
  - **Efficient line-based solver technology**
  - **Automated scalable volume meshing from CAD/surface tessellation**

Meshing Method

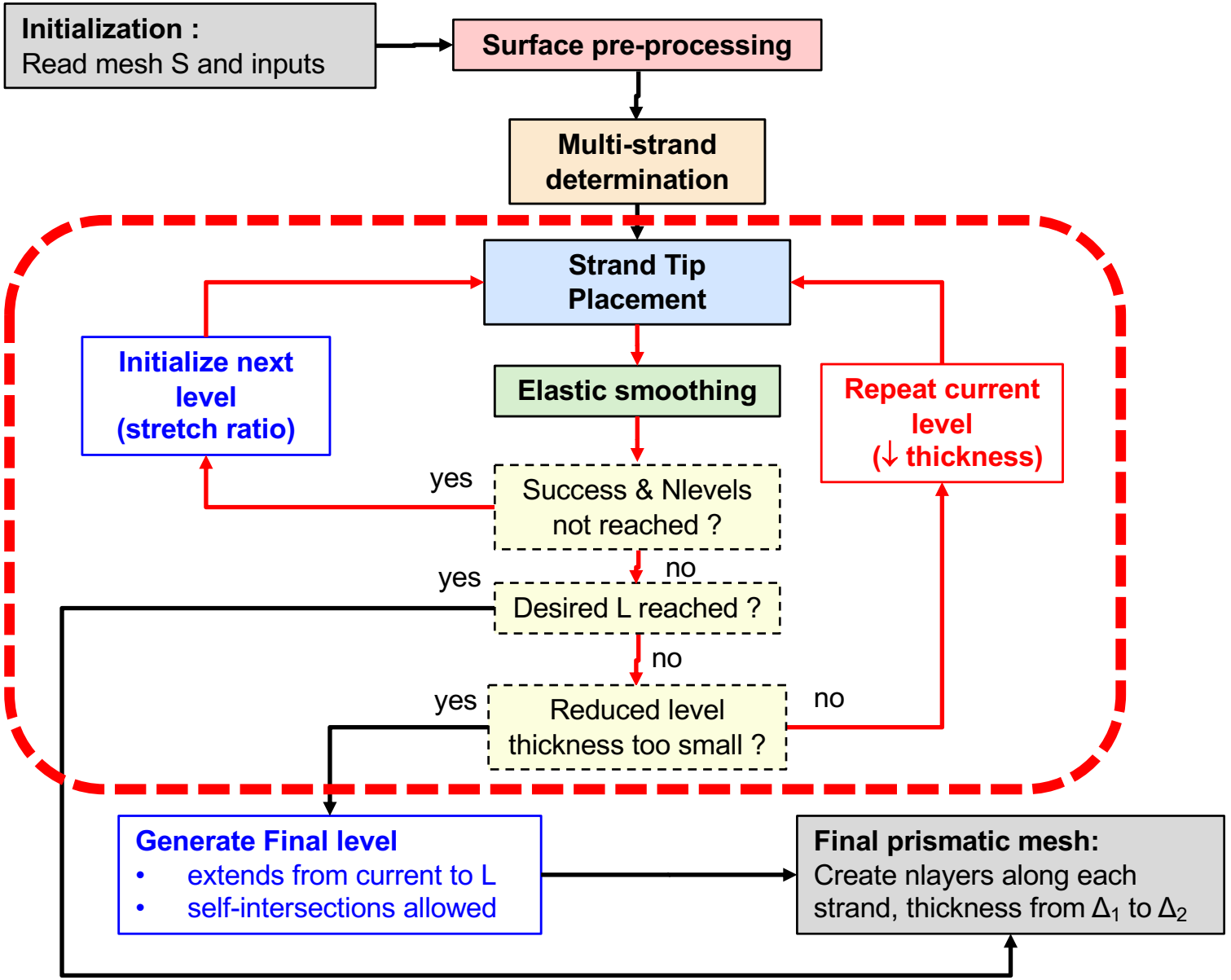
Meshing Method Overview



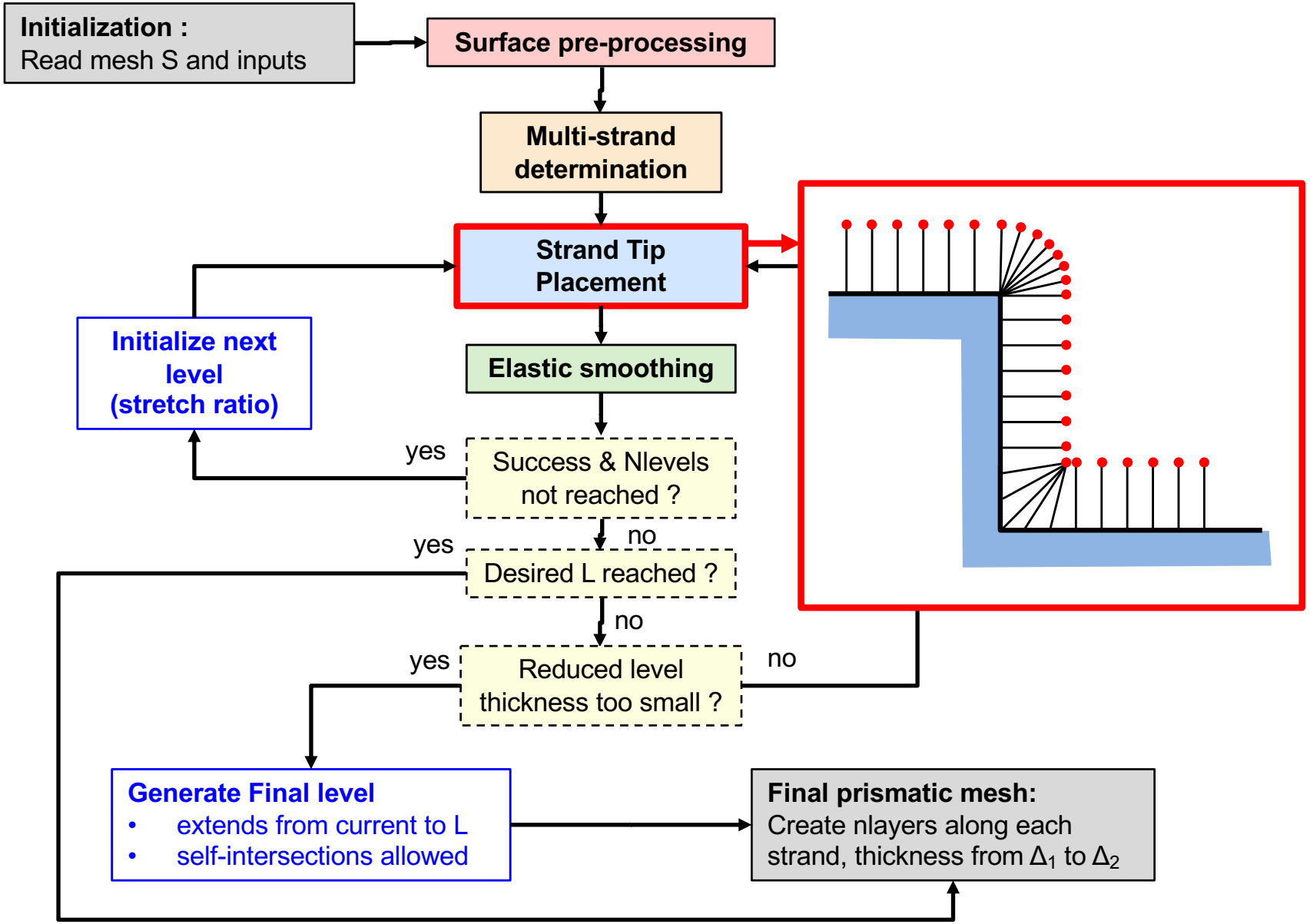
Meshing Method Overview



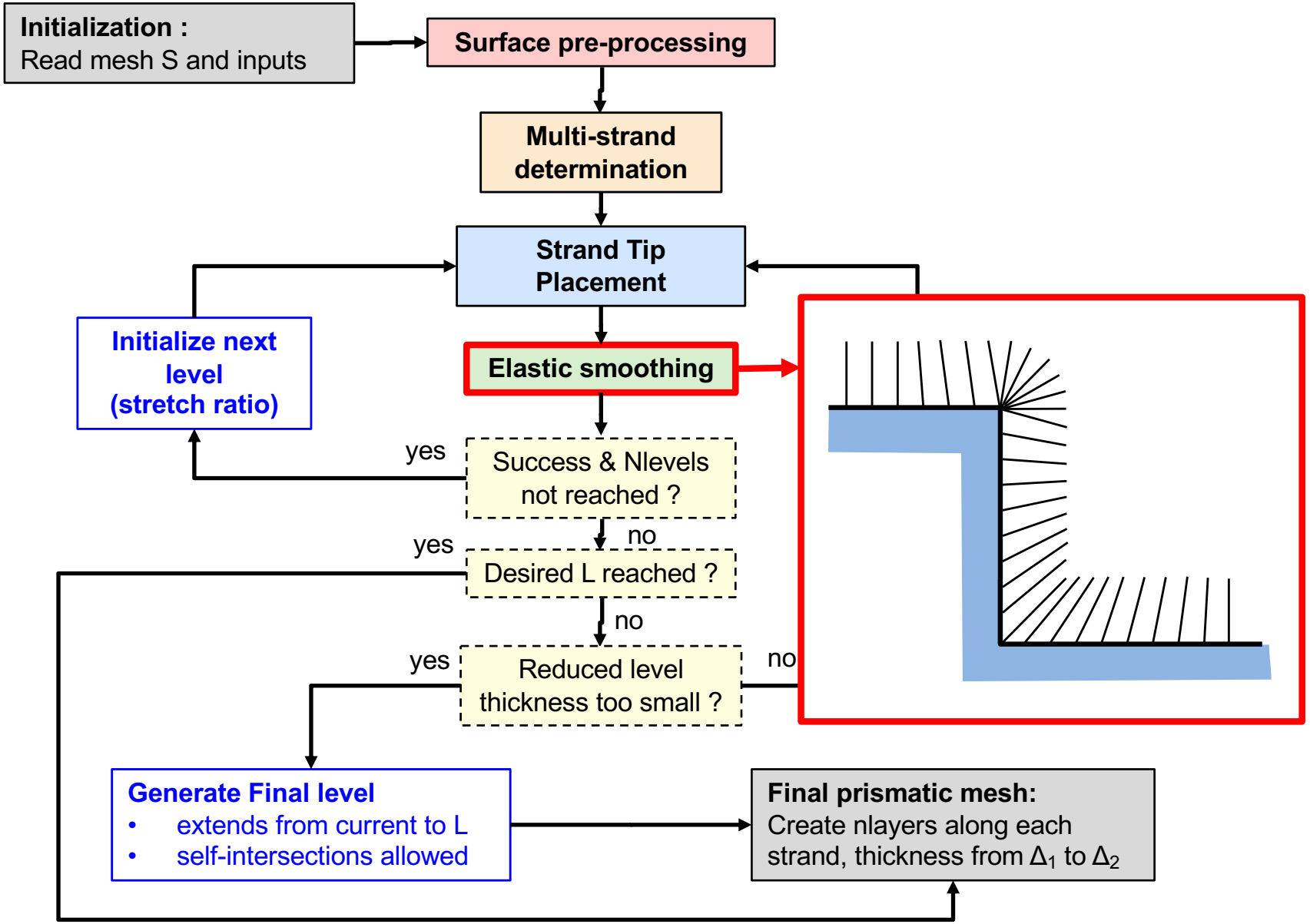
Meshing Method Overview



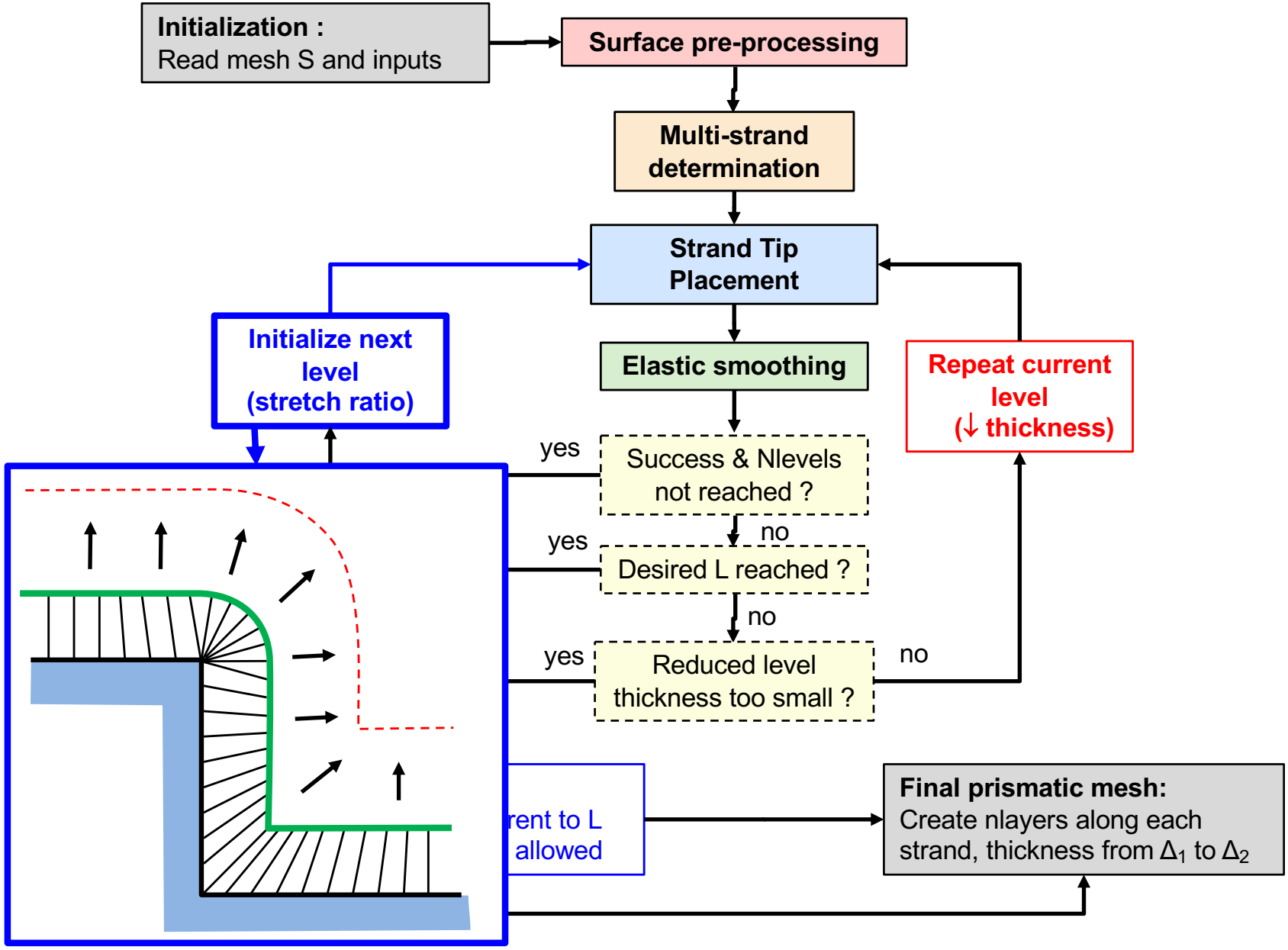
Meshing Method Overview



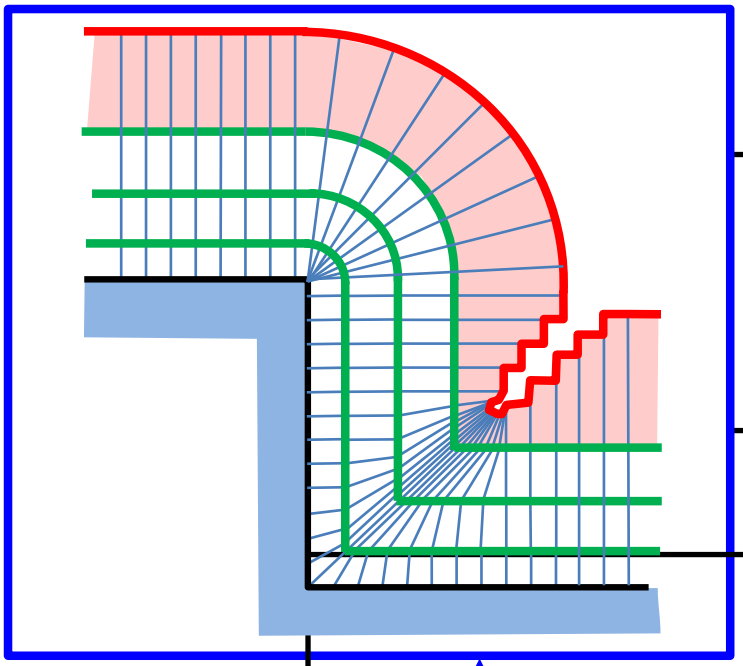
Meshing Method Overview



Meshing Method Overview



Meshing Method Overview



Initialization :
Read mesh S and inputs

Surface pre-processing

Multi-strand determination

Strand Tip Placement

Elastic smoothing

Success & Nlevels not reached ?

yes

no

Desired L reached ?

yes

no

Reduced level thickness too small ?

yes

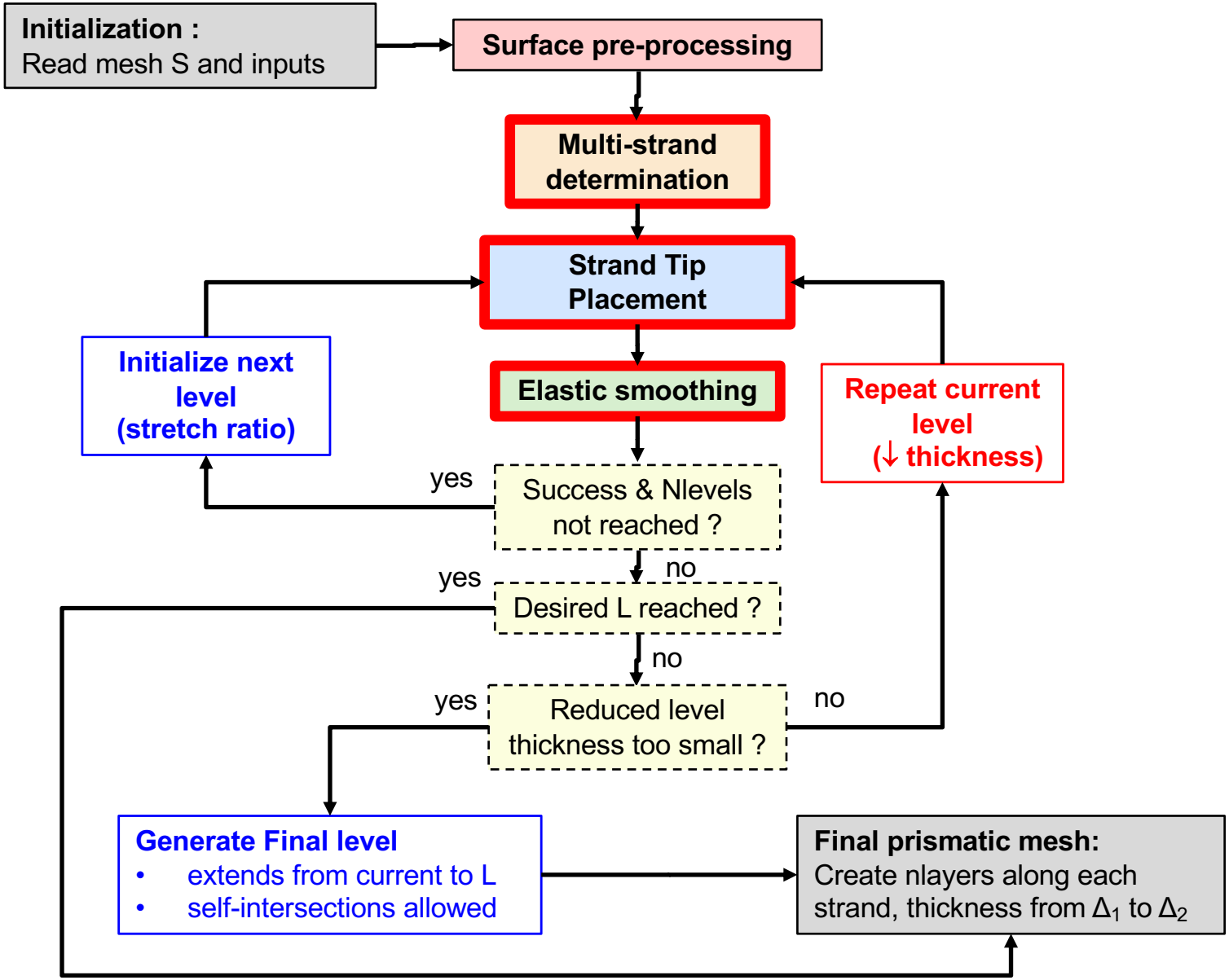
no

Repeat current level (↓ thickness)

Generate Final level
- extends from current to L
- self-intersections allowed

Final prismatic mesh:
Create nlayers along each strand, thickness from Δ_1 to Δ_2

Meshing Method Overview



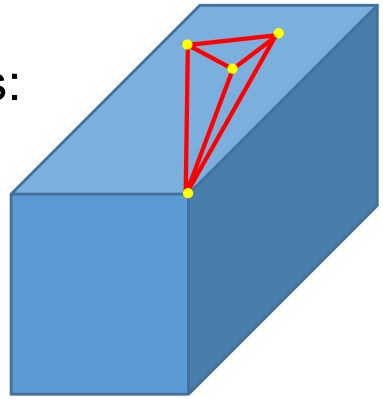
Multi-strand Determination

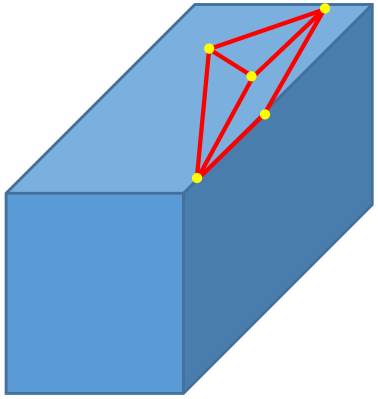
- **Initial mesh level only, improve mesh quality (better coverage)**
- **Surface nodes on convex edges (>45°): multiple strands**
- **Multi-strand cells along convex edges and corners**

multi-strand cells:

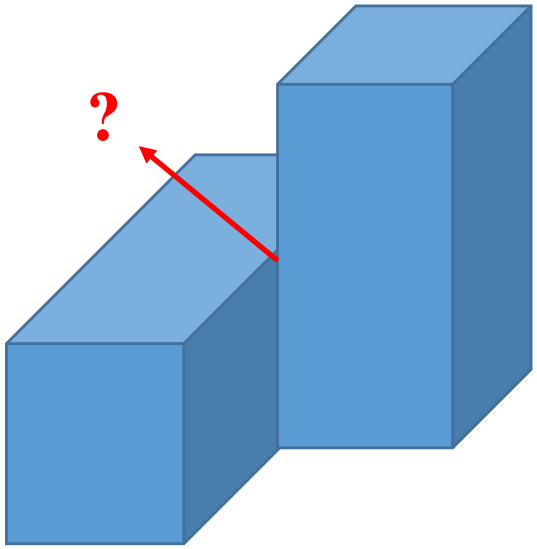
on a corner (tetrahedral)

along an edge ("wedge")



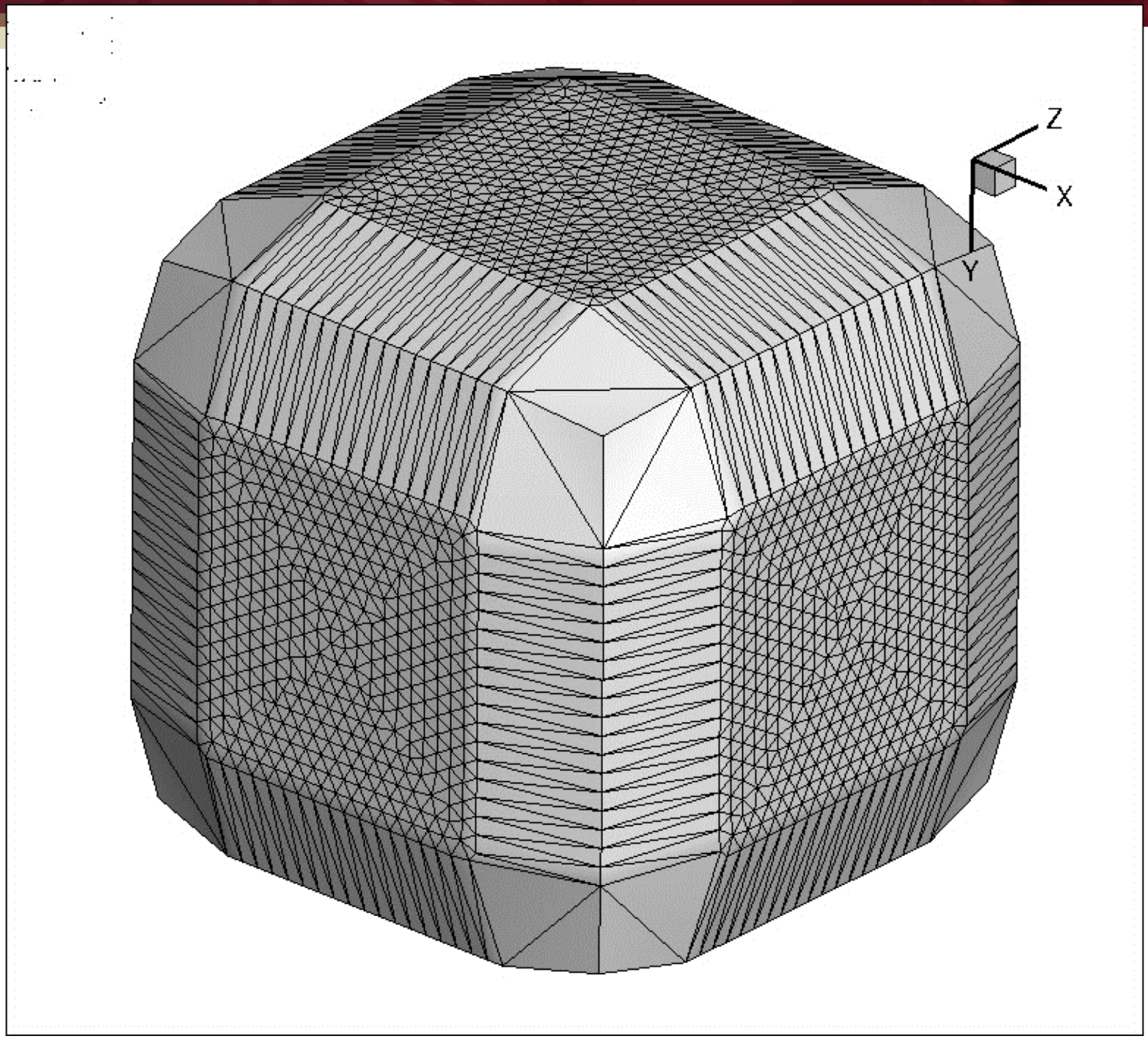


point with no visibility region

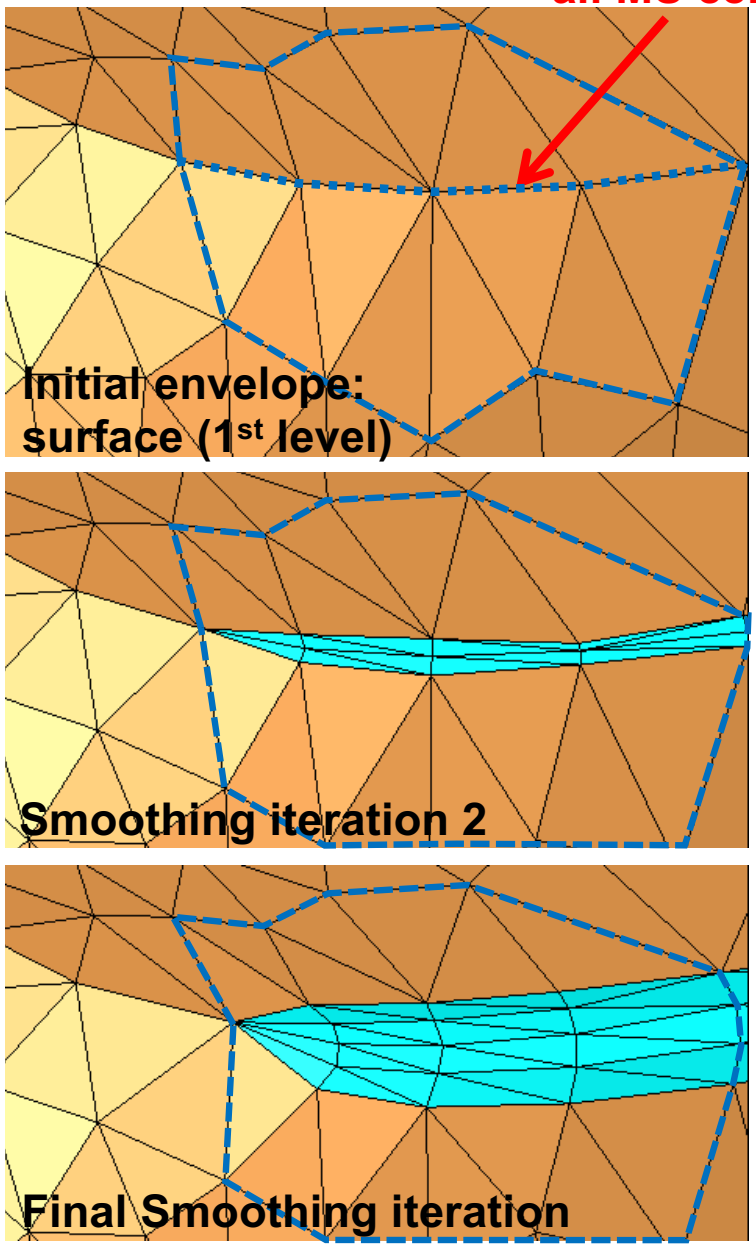


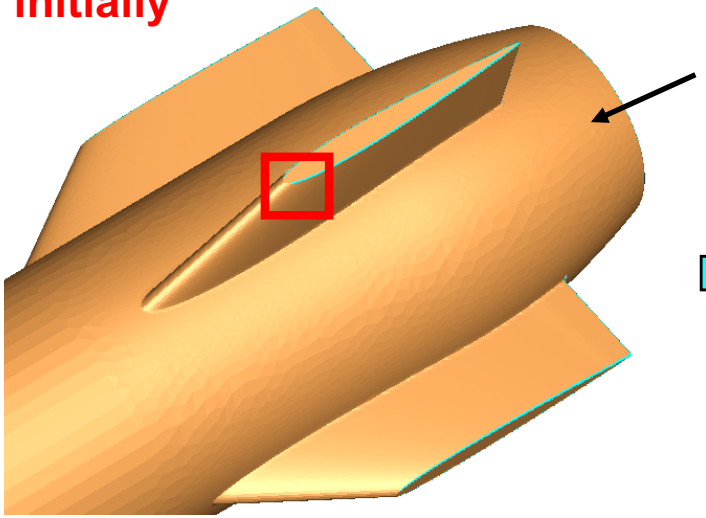
- **Implementation:**
  1. **modify boundary surface connectivity**
  2. **initial multi-strands coincide with original strand,**
  3. **smoothing algorithm spreads multi-strand to cover convex regions**

Cube Example



Multi-strands along convex edges

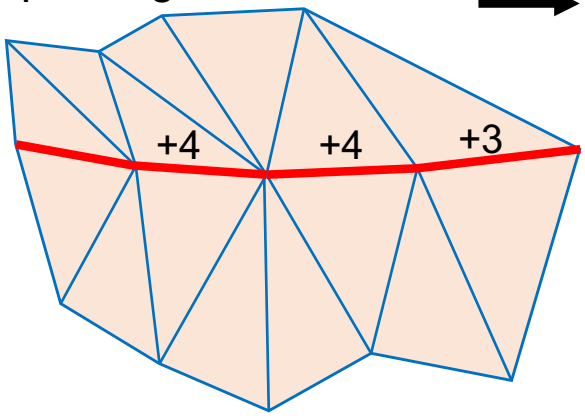




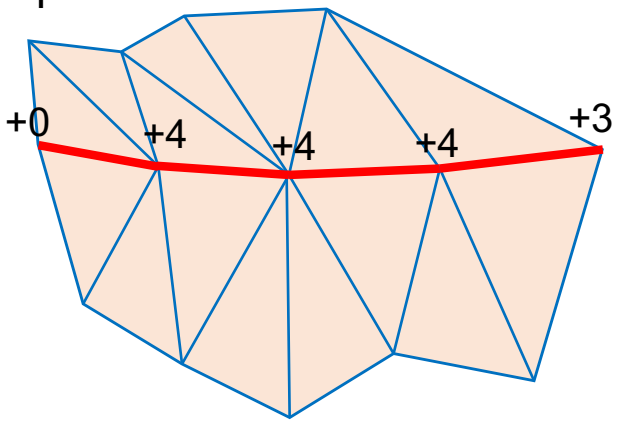
Envelope surface at a small distance from original surface

Multi-strand cells

Number of additional strands per edge:

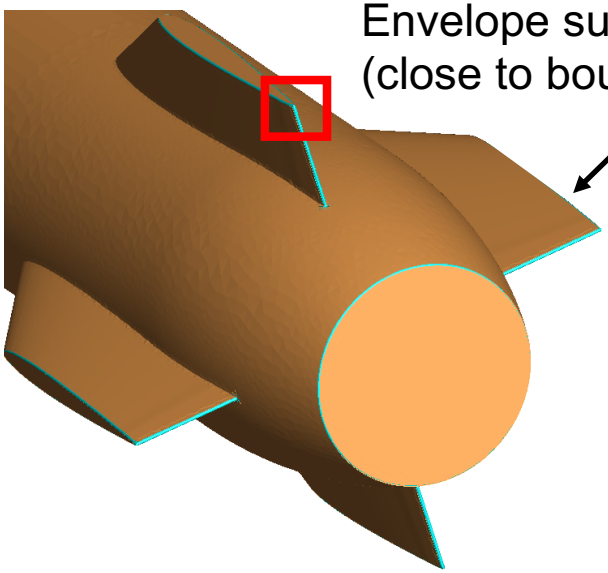


Number of additional strands per node:



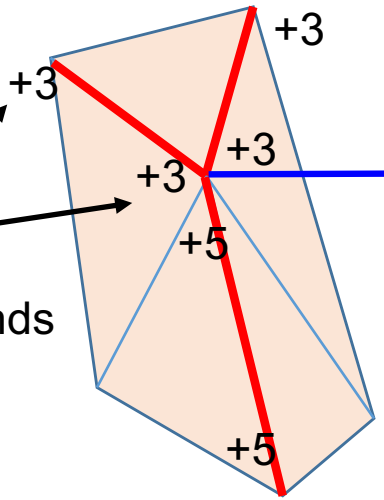
Convex edge

Multi-strand cells generation



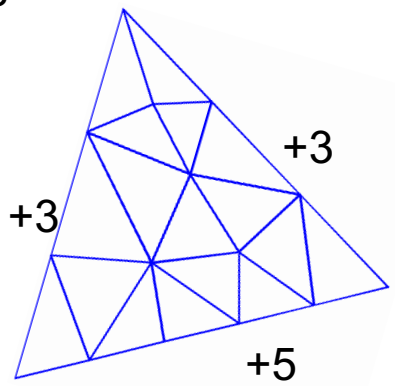
Envelope surface for level 1
(close to boundary surf.)

Convex edge

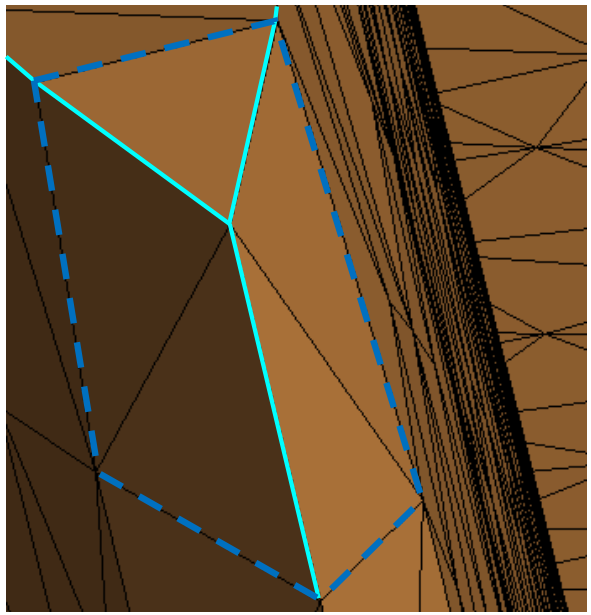


Number of additional strands

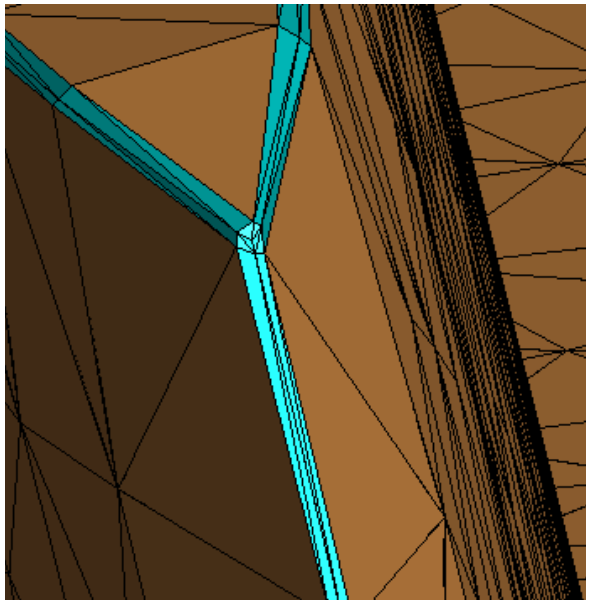
2D Delaunay triangulation:



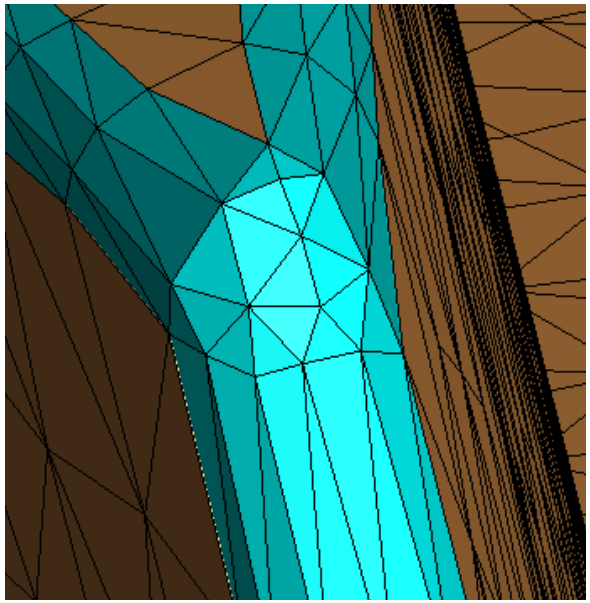
Multi-strand cells



Smoothing iteration 1

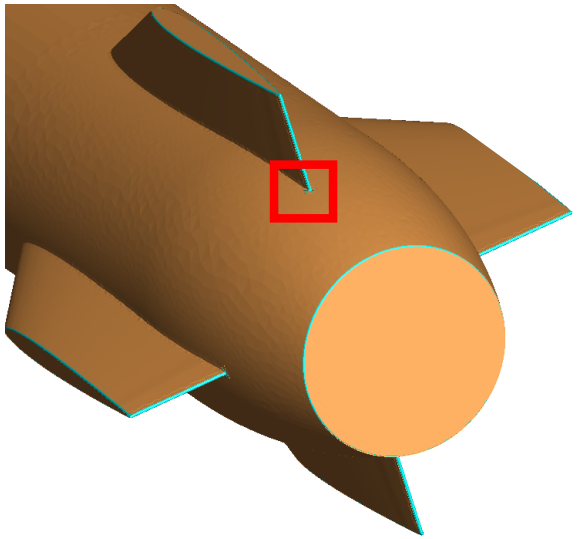


Smoothing iteration 2

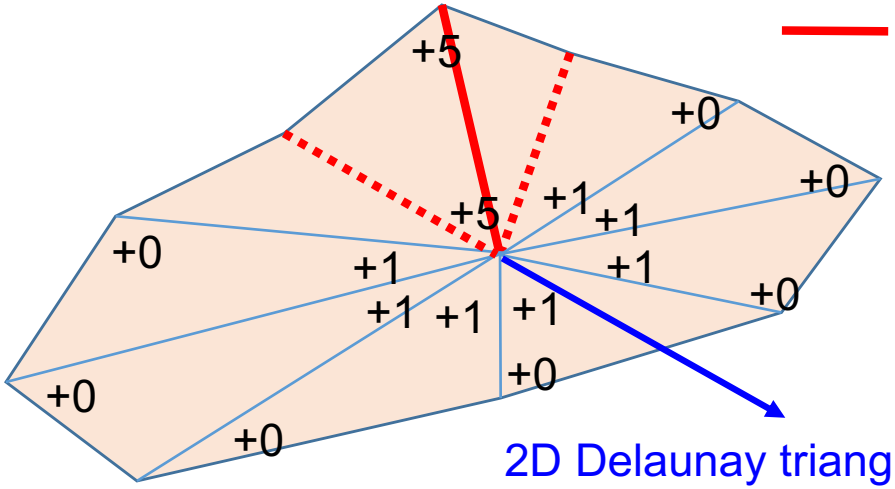


Final smoothing iteration

Multi-strands around convex/concave corners



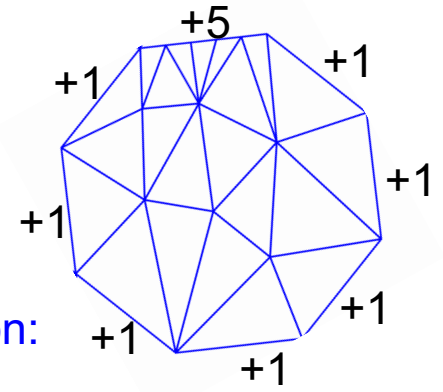
Number of additional strands:

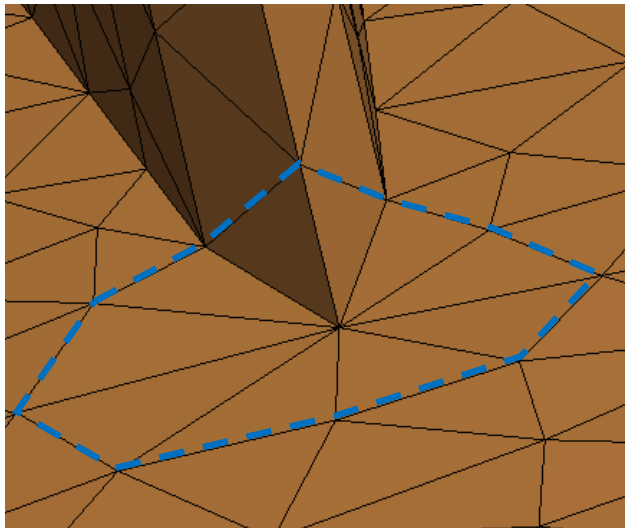


Concave edges

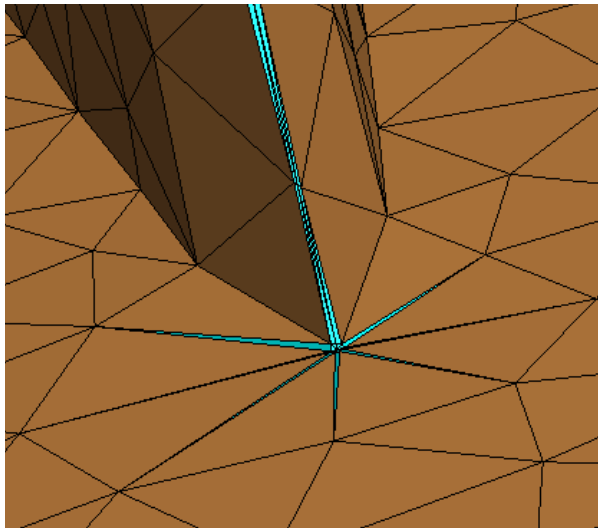
Convex edge

2D Delaunay triangulation:

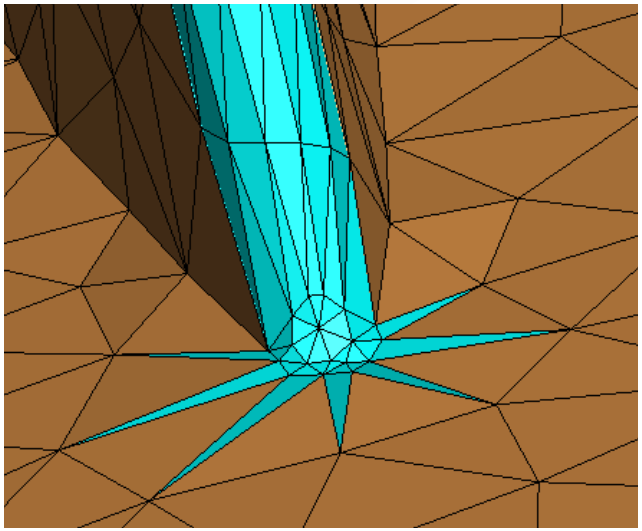




Initial envelope surface
(1st level)



Smoothing iteration 2

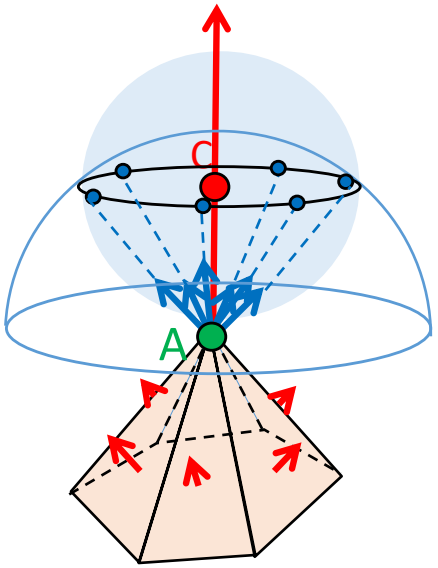


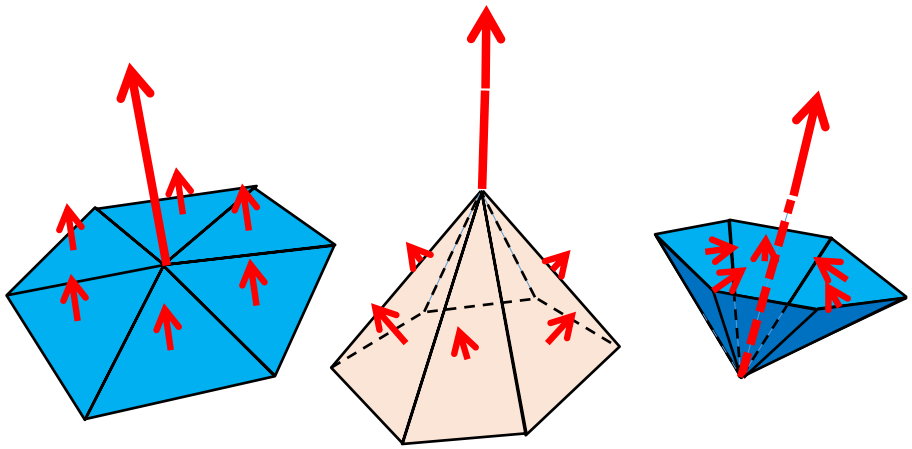
Final Smoothing iteration

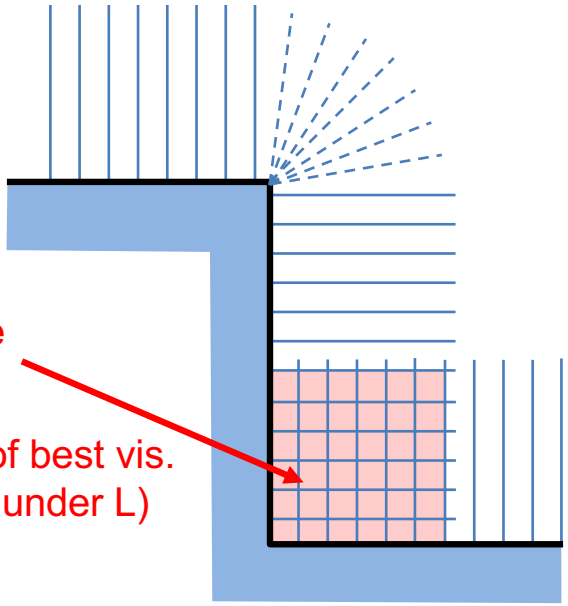
Strand Tip Placement

Direction of best visibility / Iso-surface of Distance Field

- **Direction of best visibility:**
  - minimizes smallest angle with neighboring face normals
  - easy to compute (MNN algorithm from Aubry, 2007)
  - good in convex regions
  - strand collisions in concave regions

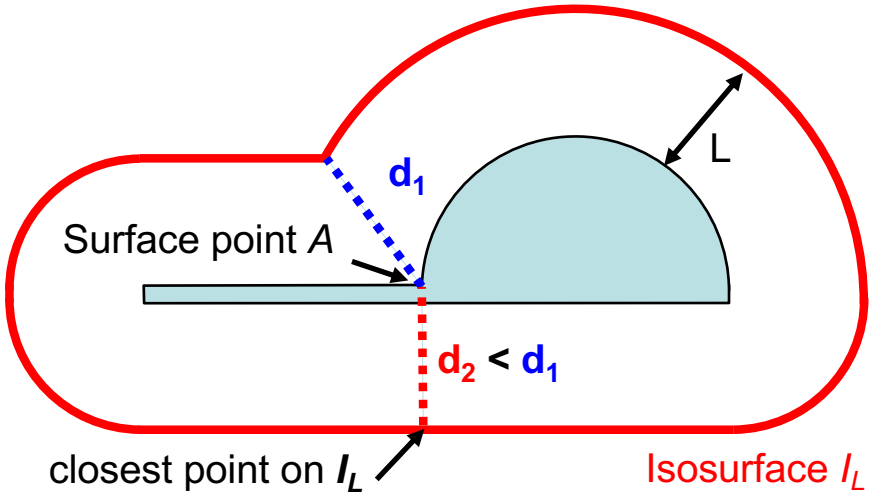


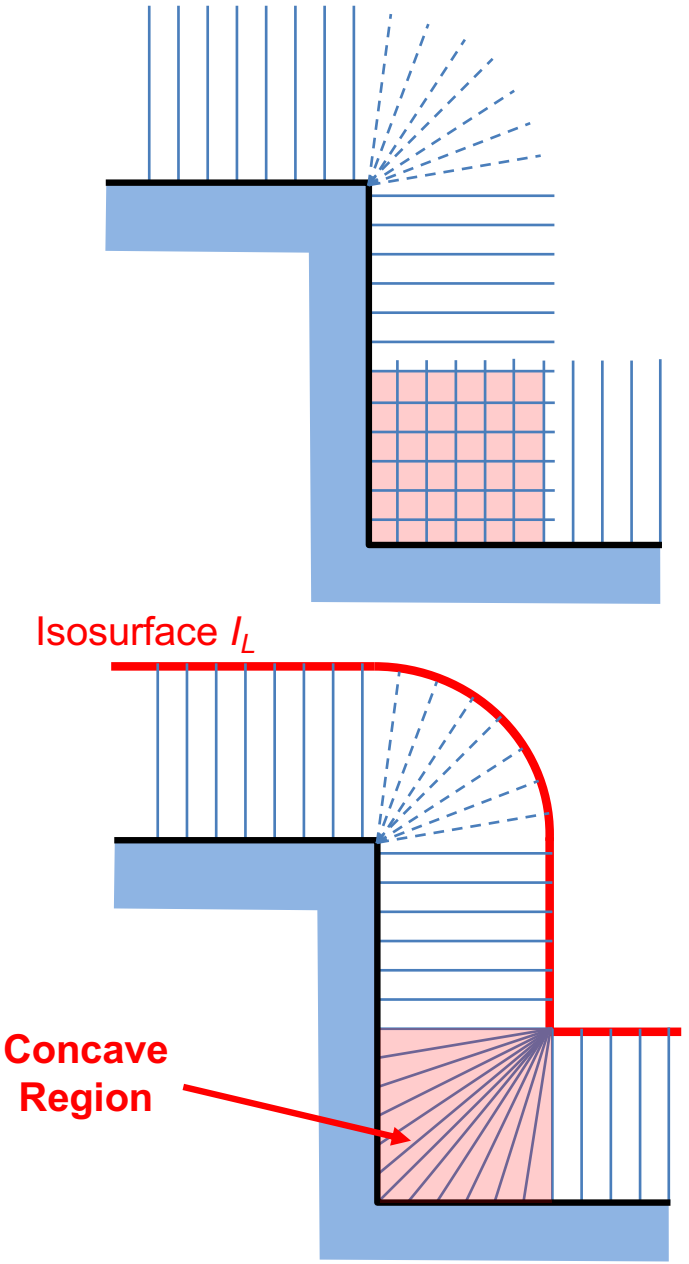




Strand Tip Placement

- **Direction to closest vertex on the iso-surface of distance (CLOVIS)**
  - automatic bending of strands in concave regions
  - can be computed efficiently using geometric algorithms
  - strand can be outside visibility region in some cases:
    → use best visibility strand

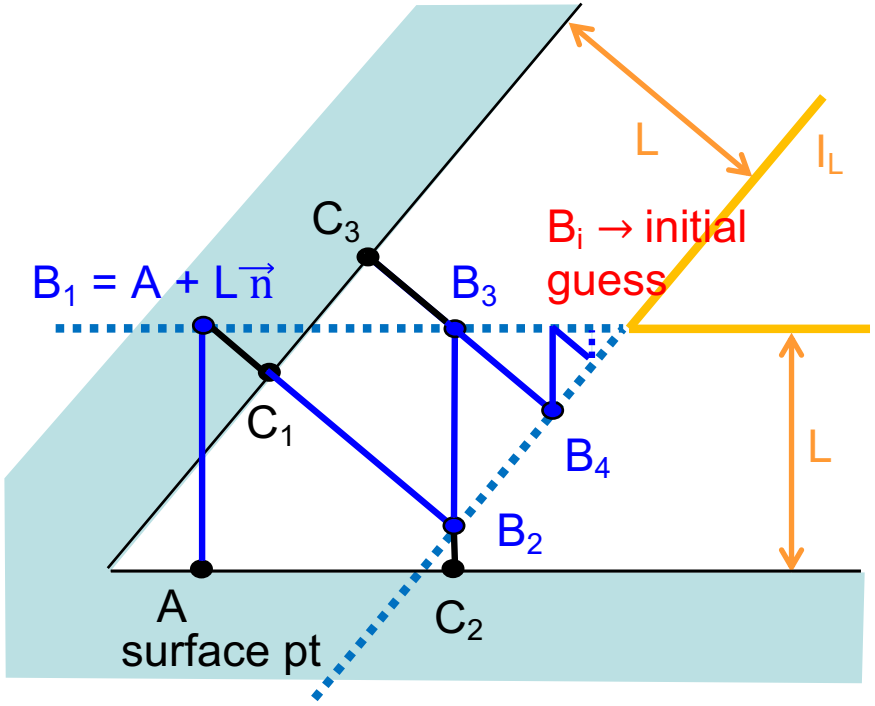




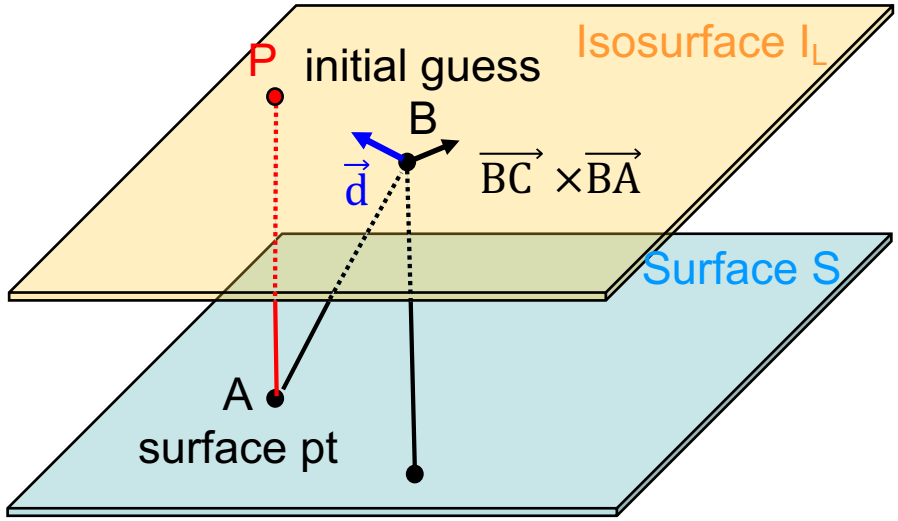
CLOVIS Algorithm

- **Isosurface of distance I_L known implicitly by its mathematical description**
- **Efficient routine (ADT) to compute closest point on S (known explicitly)**
- **Closest point on I_L found in 2 steps (initial guess + search)**

Step 1: initial guess.



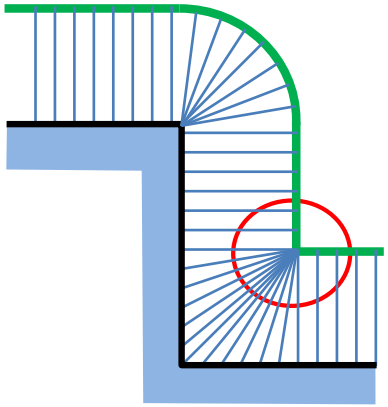
Step 2: marching towards the closest vertex on the isosurface of distance



Elastic Smoothing on the Isosurface of Distance

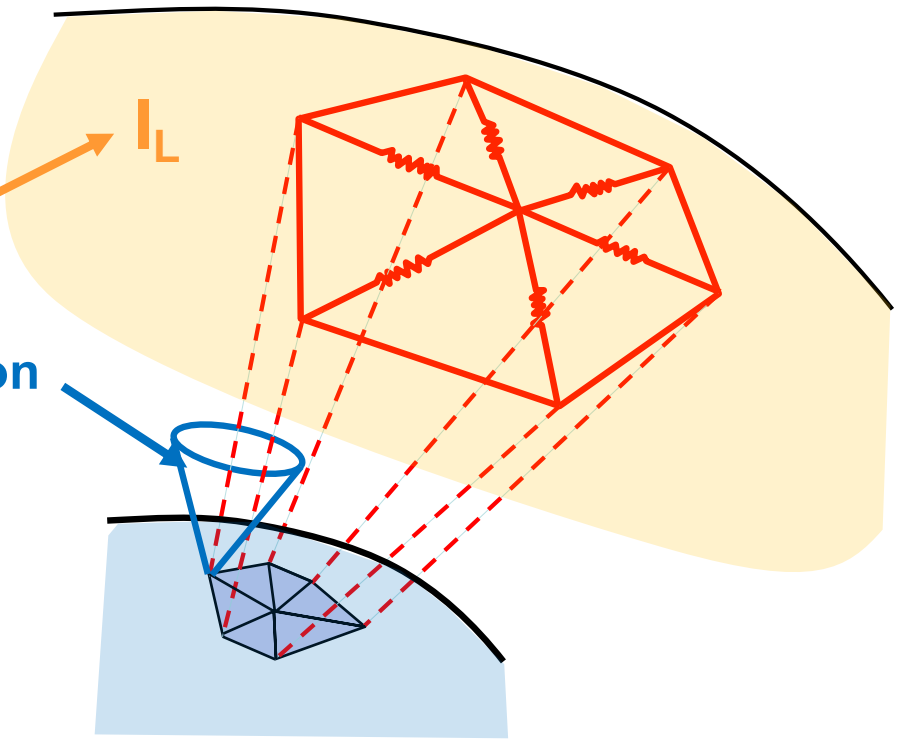
Initial strand mesh obtained from CLOVIS / best visibility direction:

- can have invalid volumes
- often presents too much point clustering on the envelope surface (poor stretch ratio)



⇨ **Apply elastic smoothing algorithm : treat each edge similar to elastic spring, with constraints:**

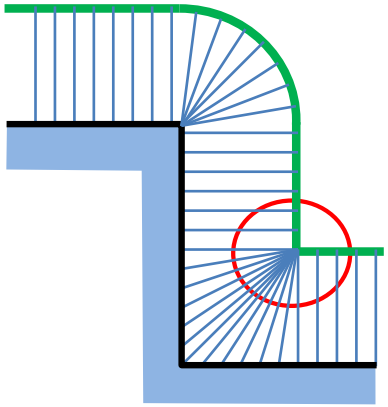
1. strand extremity remains on **Isosurface of Distance at L**
2. strand remains within **visibility region**



Elastic Smoothing on the Isosurface of Distance

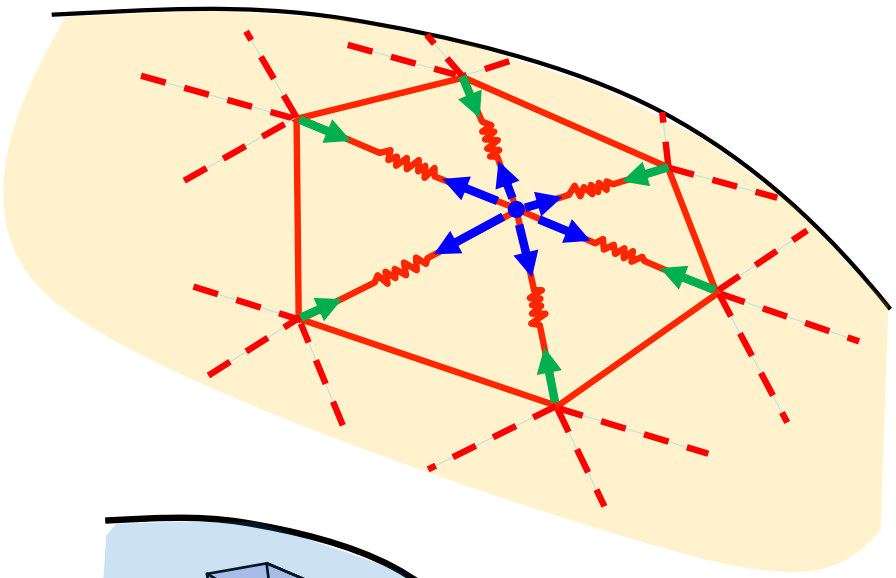
Initial Strand mesh obtained from CLOVIS / best visibility direction:

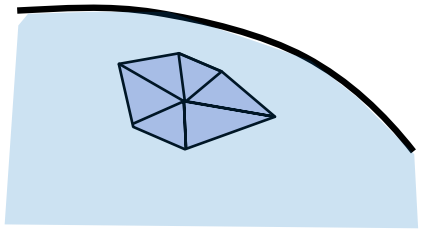
- Can have invalid volumes
- Often presents too much point clustering on the envelope surface (poor stretch ratio)



⇨ **Apply elastic smoothing algorithm : treat each edge similar to elastic spring:**

- Spring rest length = 0 $\vec{F} = K\vec{L}$
- 1st level: spring stiffness K such that nodes are at rest for very short strand lengths
- Final level: uniform stiffness at prismatic envelope

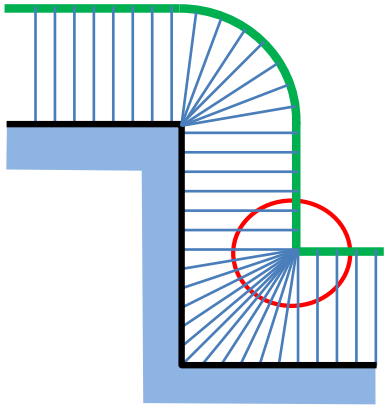




Elastic Smoothing on the Isosurface of Distance

Initial Strand mesh obtained from CLOVIS / best visibility direction:

- Can have invalid volumes
- Often presents too much point clustering on the envelope surface (poor stretch ratio)

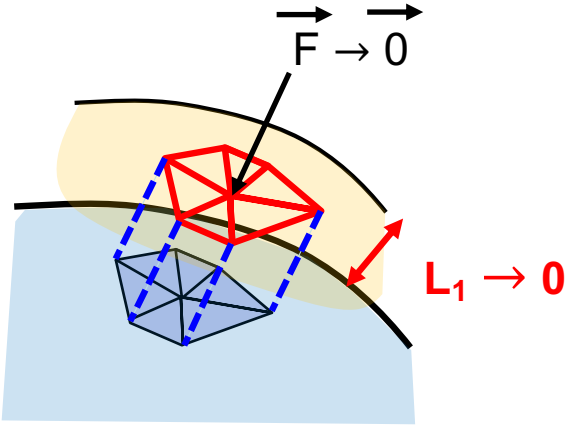


⇨ **Apply elastic smoothing algorithm : treat each edge similar to elastic spring:**

- Spring rest length = 0 $\vec{F} = K\vec{L}$
- 1st level: spring stiffness K such that nodes are at rest for very short strand lengths
- Final level: uniform stiffness at prismatic envelope

1st level thickness → 0

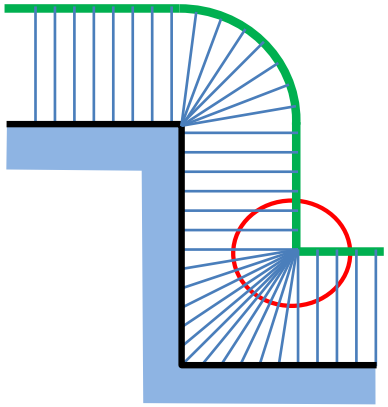
⇒ **Envelope tessellation = boundary tessellation**



Elastic Smoothing on the Isosurface of Distance

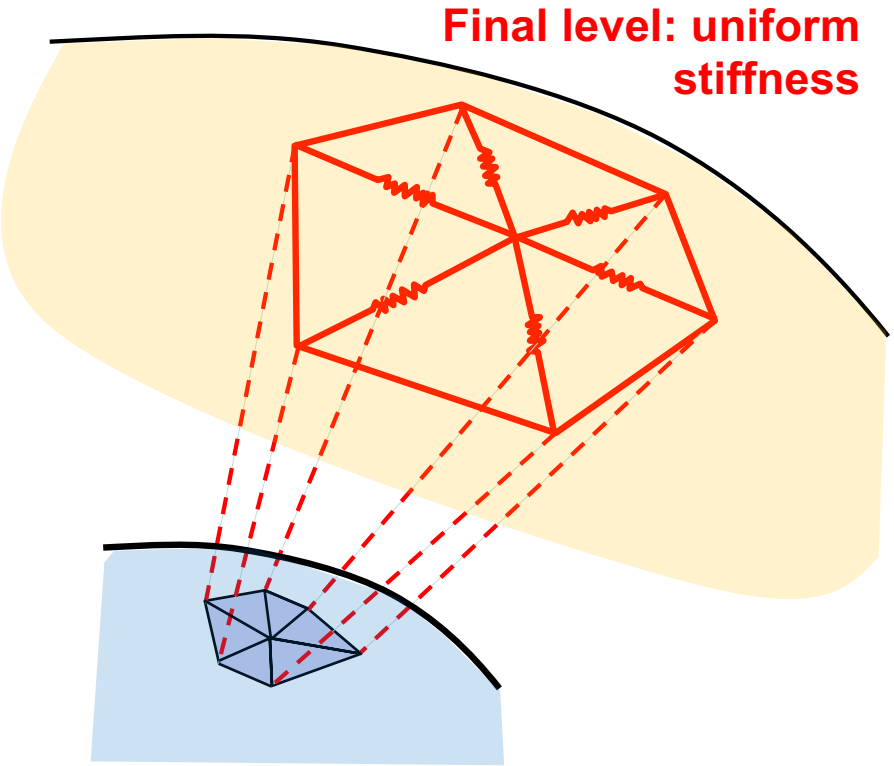
Initial Strand mesh obtained from CLOVIS / best visibility direction:

- Can have invalid volumes
- Often presents too much point clustering on the envelope surface (poor stretch ratio)



⇨ **Apply elastic smoothing algorithm : treat each edge similar to elastic spring:**

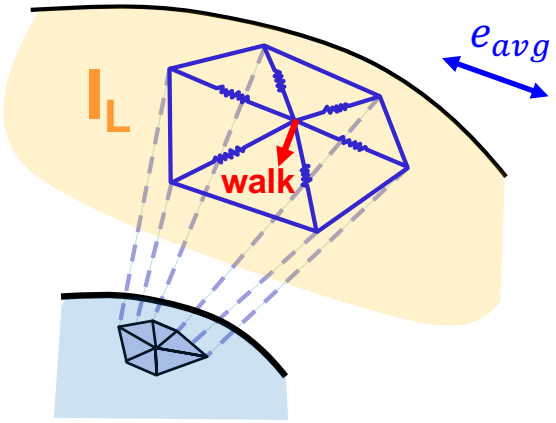
- Spring rest length = 0 $\vec{F} = K\vec{L}$
- 1st level: spring stiffness K such that nodes are at rest for very short strand lengths
- Final level: uniform stiffness at prismatic envelope

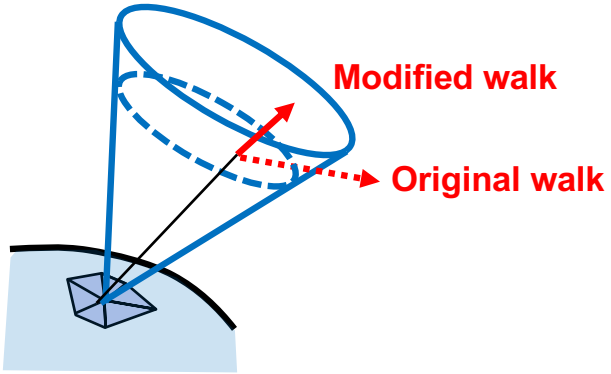


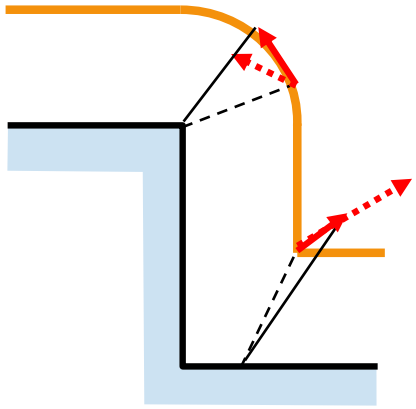
Smoothing Method

Explicit time marching iterations:

- **Walk length:** such that no vertex walks more than 20% average length of surrounding edges:
  - $\vec{p}^{i+1} = \vec{p}^{i} + \frac{0.2}{rmax}\vec{F}^{i} \qquad rmax = \max_{k} \frac{\left|\overrightarrow{F^k}\right|}{e^k_{avg}}$
- **Visibility constraint applied:**
  - Walk component normal to local visibility cone removed
  - If still outside visibility region, walk shortened
- **Isosurface of distance constraint applied:**
  - Convex region vertex: flush point to isosurface
  - Concave region vertex: can move above isosurface, limit applied to strand length







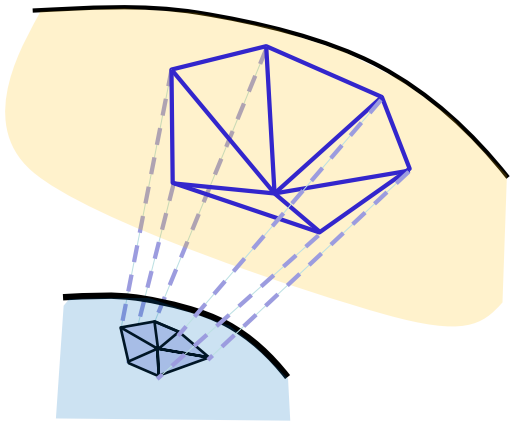
Smoothing Method: Termination

Smoothing iterations performed until:

- Maximum number of iterations reached
- Maximum walk under threshold
- Envelope quality not improving in last 20 iterations + mesh is valid (no cell quality under 0.2)

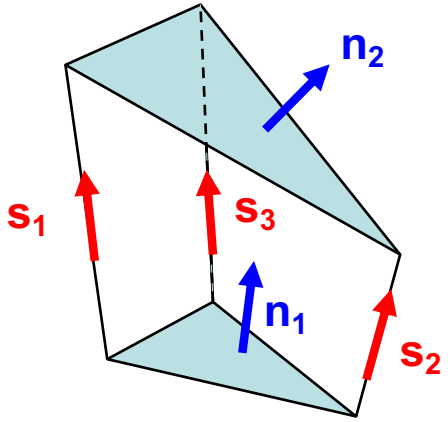
Envelope quality:

Inverse of maximum stretch ratio of face areas around a node on envelope surface:



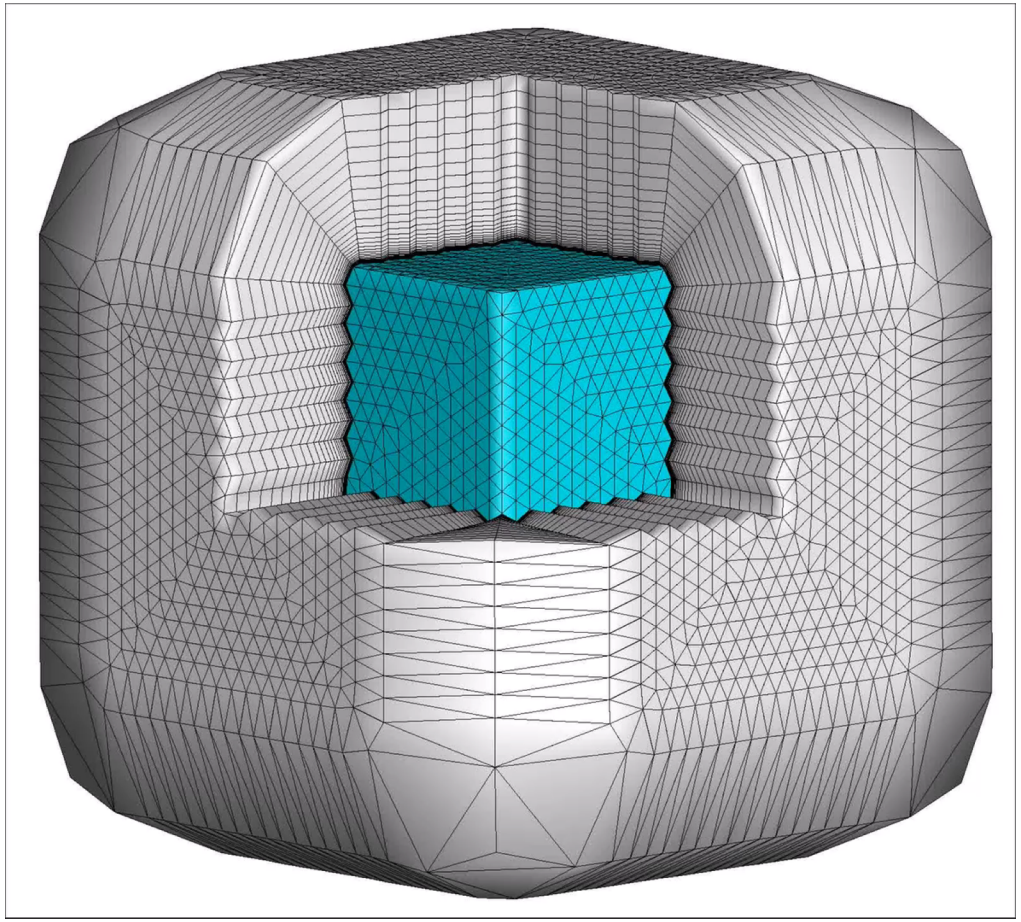
Cell quality:

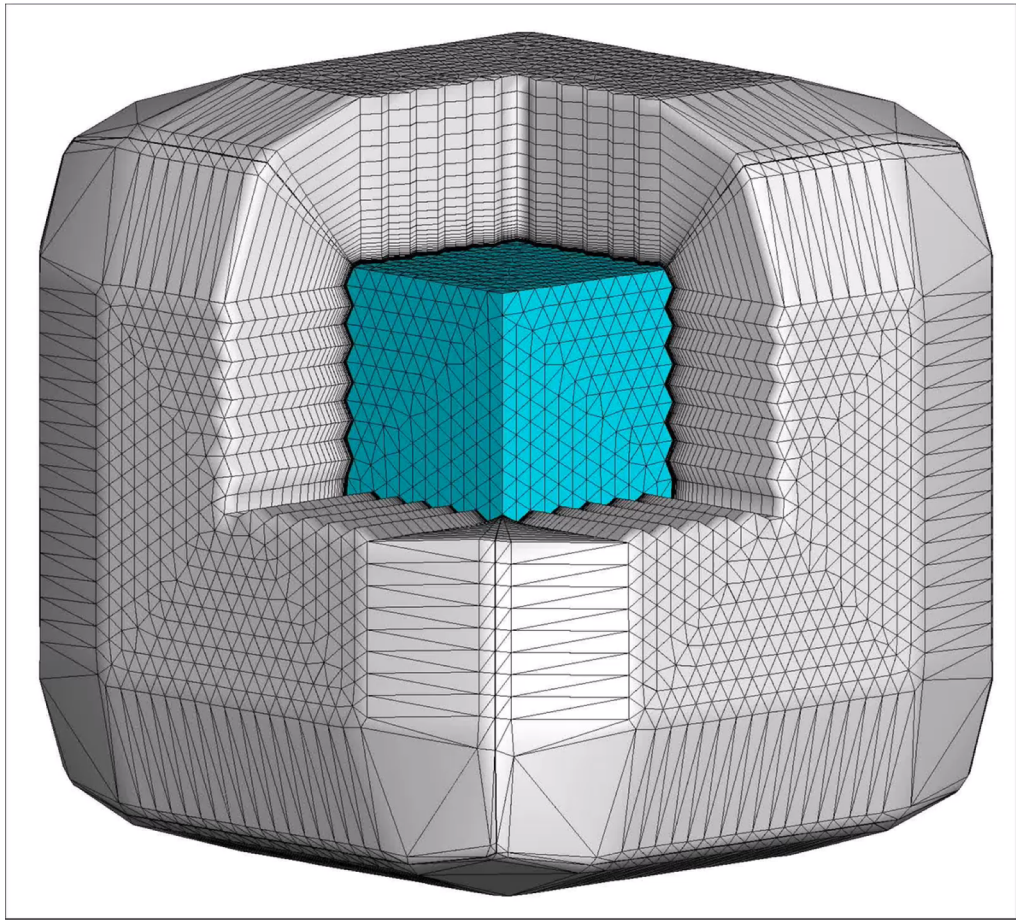
Minimum dot product of strand vectors → and normal vectors →



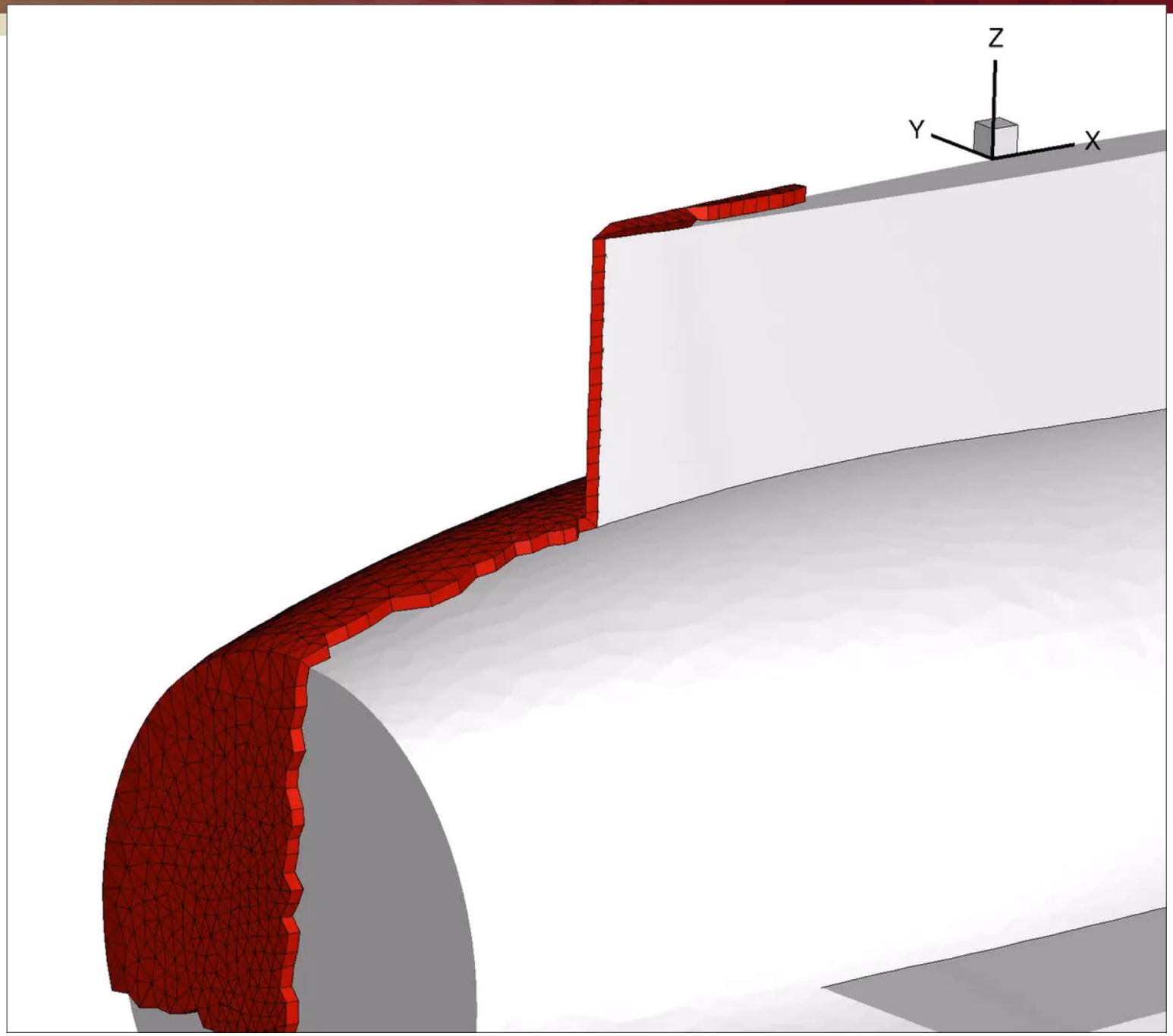
$$Q = \min_{i,j} \left(\vec{s_i} \cdot \vec{n_j} \right)$$

Cube Example

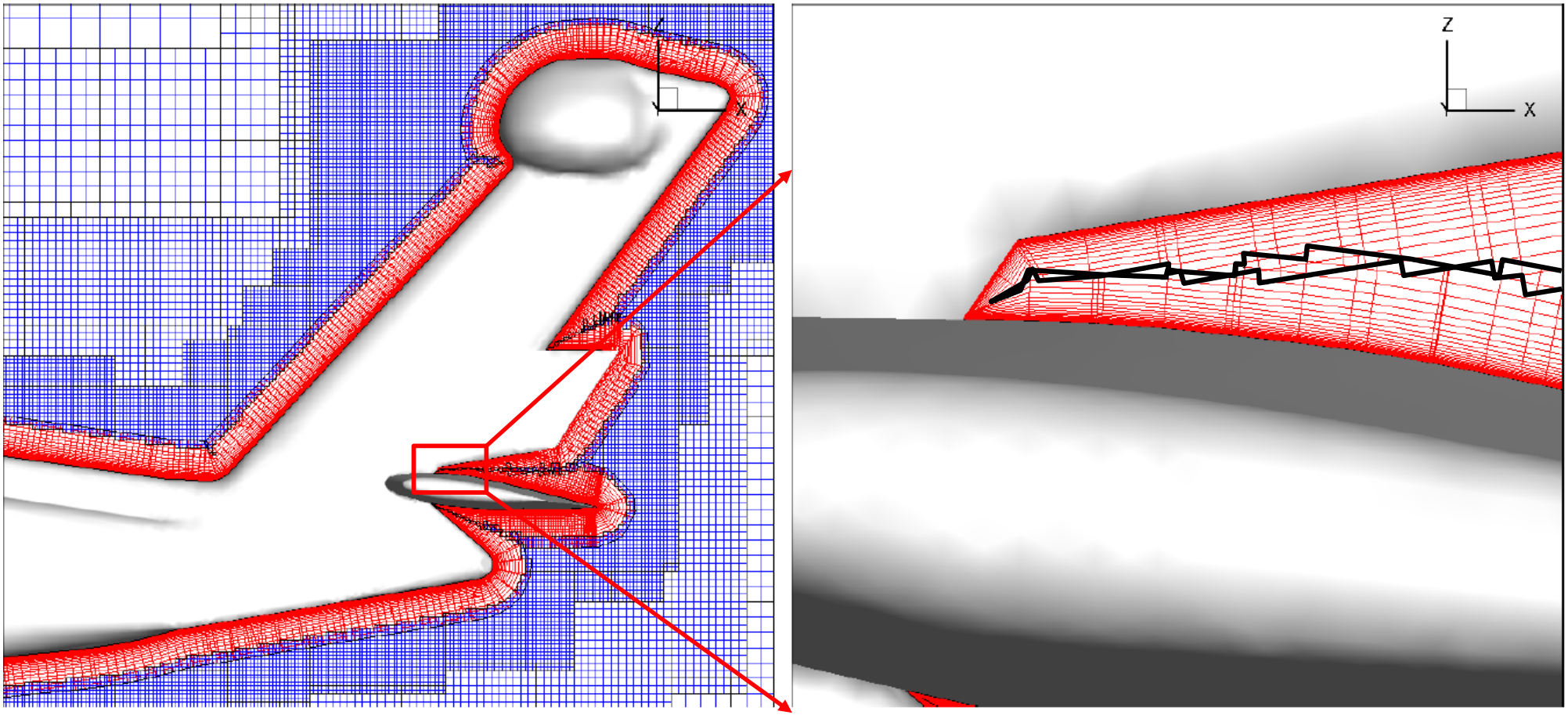




Example 2: Fin Base/Body



Complex Geometries with Self-Intersecting Meshes



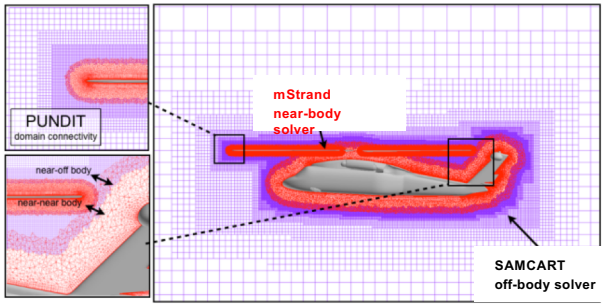
- Final level built by simple extrusion from last level + Lagrangian smoothing
- Complex geometry cannot be meshed without self-intersecting meshes.
  - Requires solution to strand collision to find which cells within the strand mesh solve, interpolate or are removed (OSCAR module in Helios)

Results

Helios Code

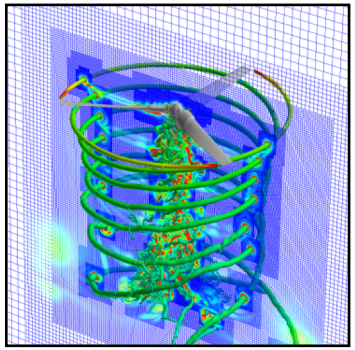
HPCMP CREATE™ AV Helios Helicopter Overset Simulations

Dual Mesh Paradigm



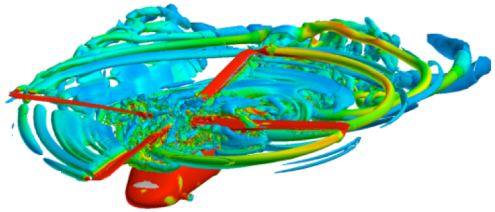
Near-body – mStrand, Overflow, FUN3D, …
Cartesian off-body - SAMCart

Adaptive Mesh Refinement



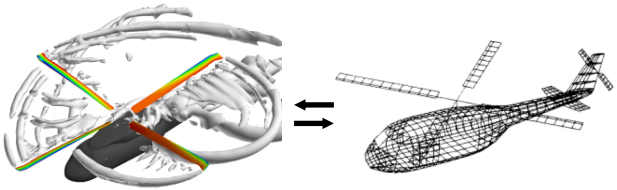
To resolve wake

Moving Body Overset



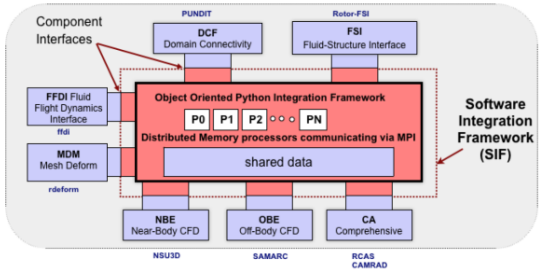
Rotor-Fuselage and Multi-rotor moving mesh support

CFD/CSD Coupling



RCAS and CAMRAD Structural Dynamics and Trim coupling

Advanced Software Infrastructure



Python-based infrastructure readily supports addition of new software

High Performance Computing



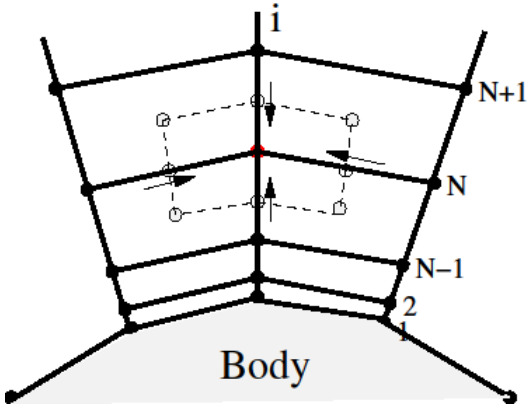
Runs on HPC hardware with focus on parallel scalability

- High-fidelity rotorcraft simulation for government and industry

Solution Process in Helios

Near-body strand solver mStrand

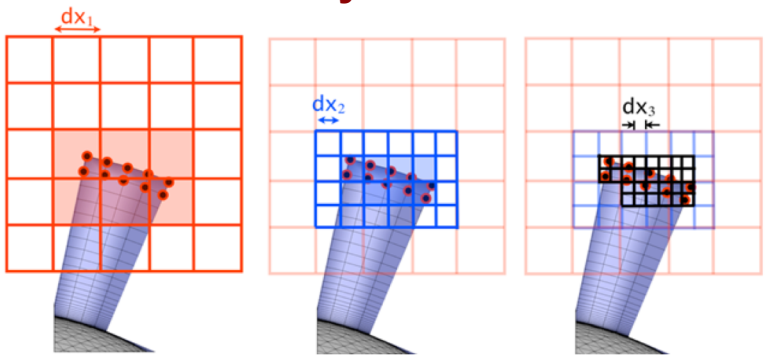
- Fully parallel 2nd order vertex centered FV solver
- Surface grid can be quads or/and triangles
- General prismatic mesh in normal direction
- MUSCL reconstruction + Riemann solver
- 2nd order full Navier-Stokes term
- 1st order implementation of SA turbulence model
- Linear solver using preconditioned GMRES



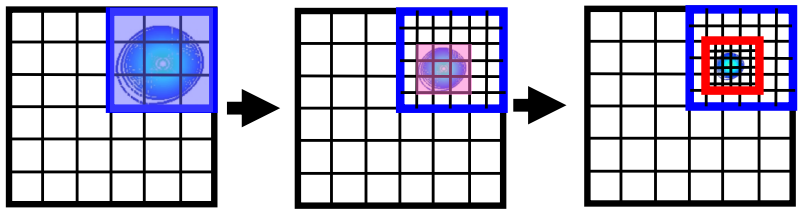
Off-body Cartesian Solver SAMCart

- Parallel mesh adaptive capability by SAMRAI library
- Each grid block solved using CART
- Higher order central difference scheme
  - ➔ 6th order with 5th order dissipation for inviscid term
  - ➔ 4th order for viscous terms
- SA turbulence model with DES capability
- LUSGS or diagonalized ADI implicit operators

Geometry refinement

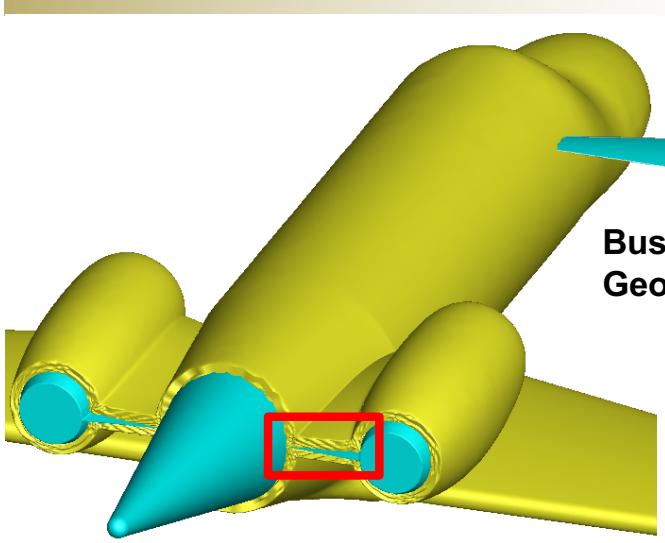


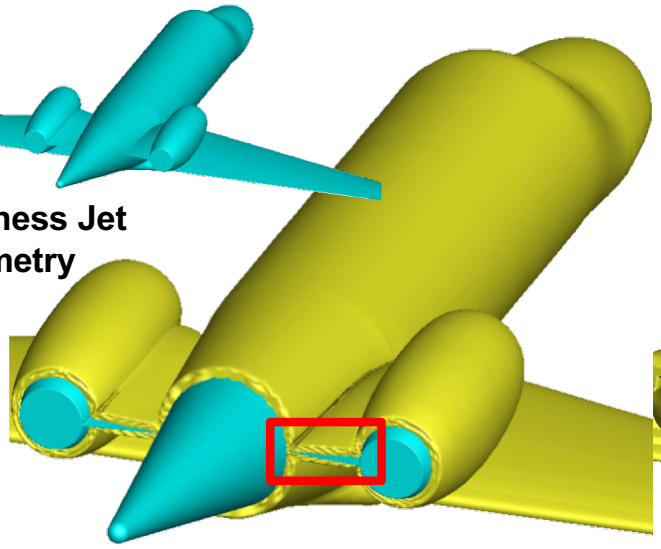
Solution refinement



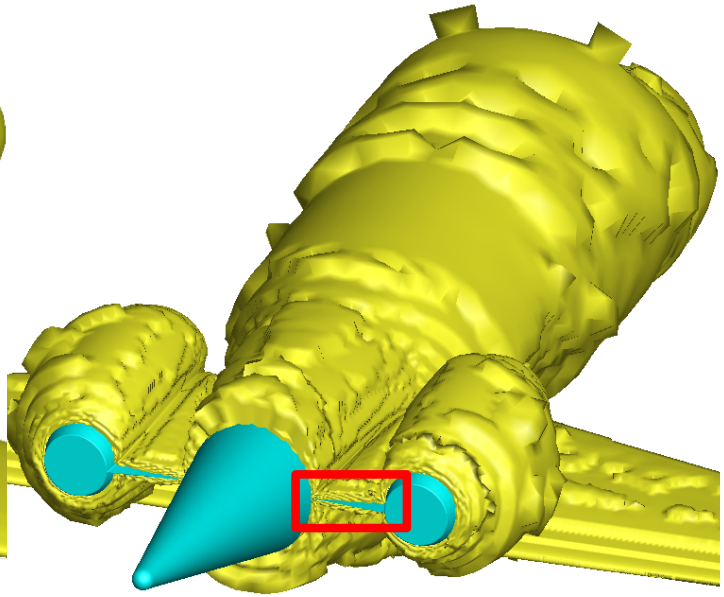
Berger-Colella style block structured AMR

Business Jet
Meshes





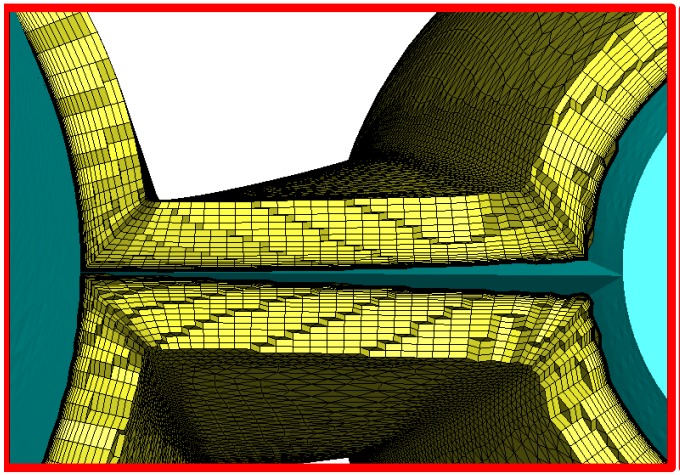
Business Jet
Geometry

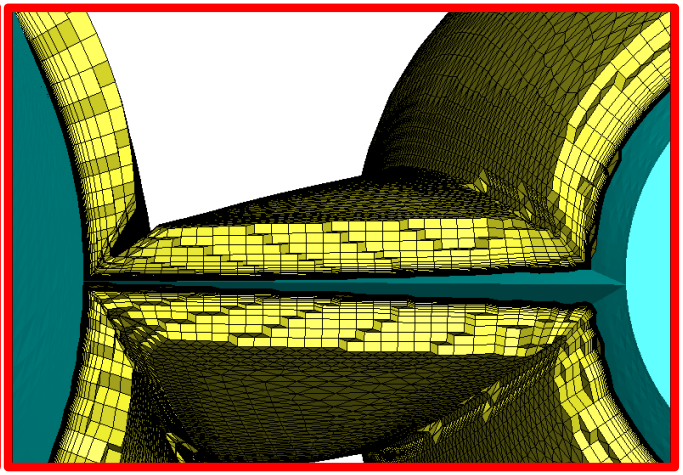


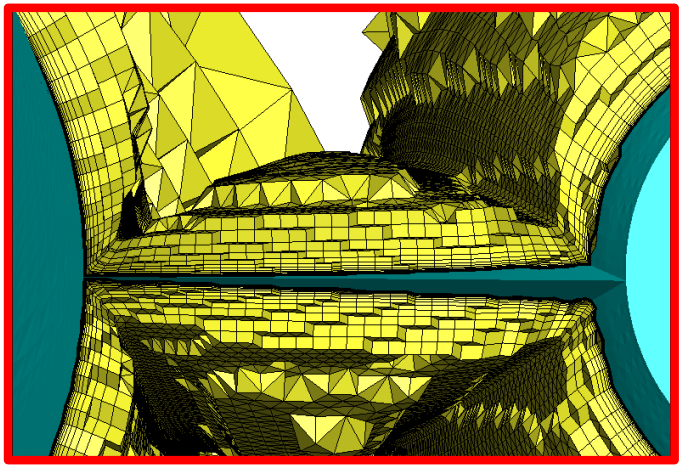
CLOVIS + smoothing

Advancing Front Method, AFLR3

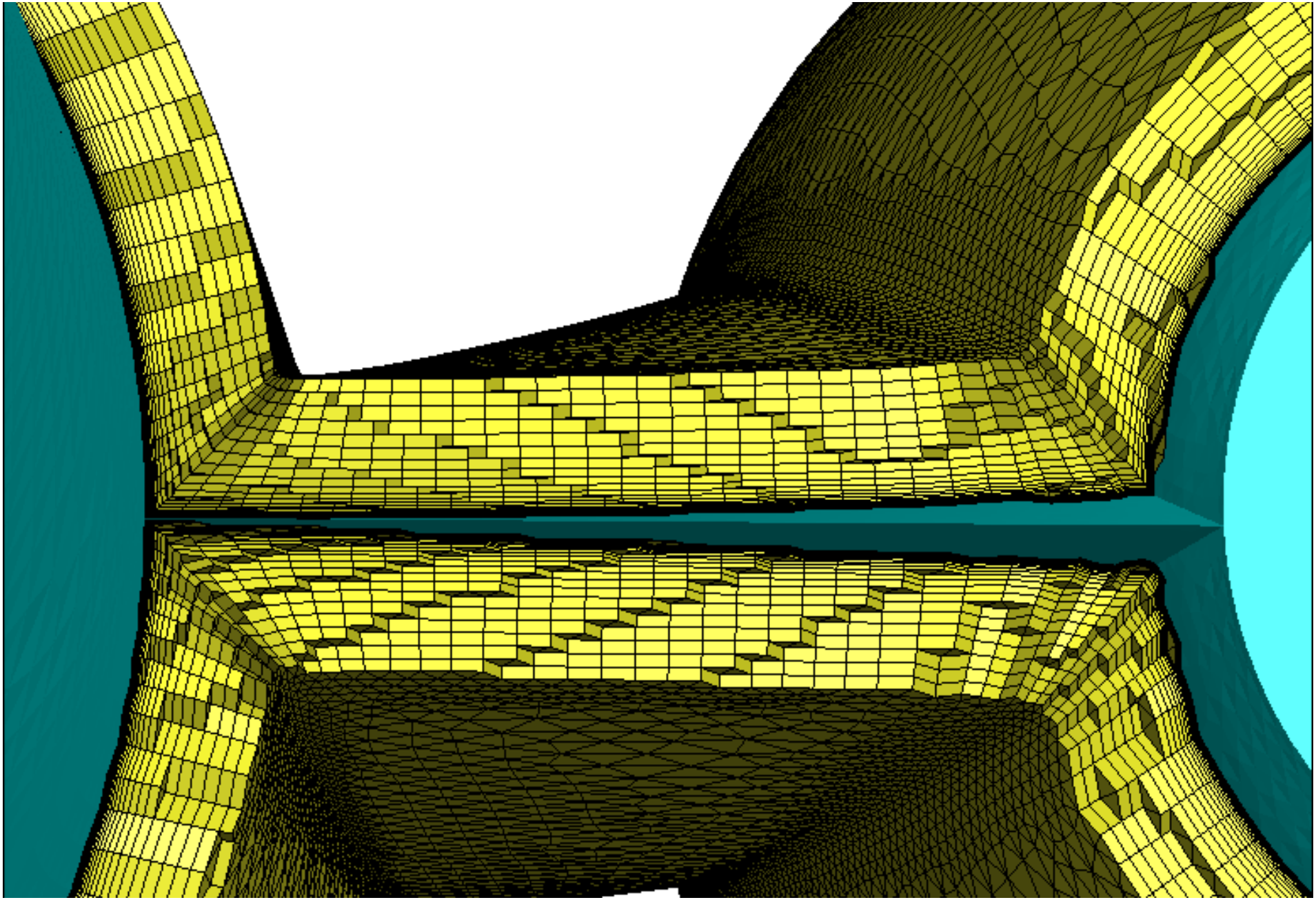
Pointwise (T-rex)



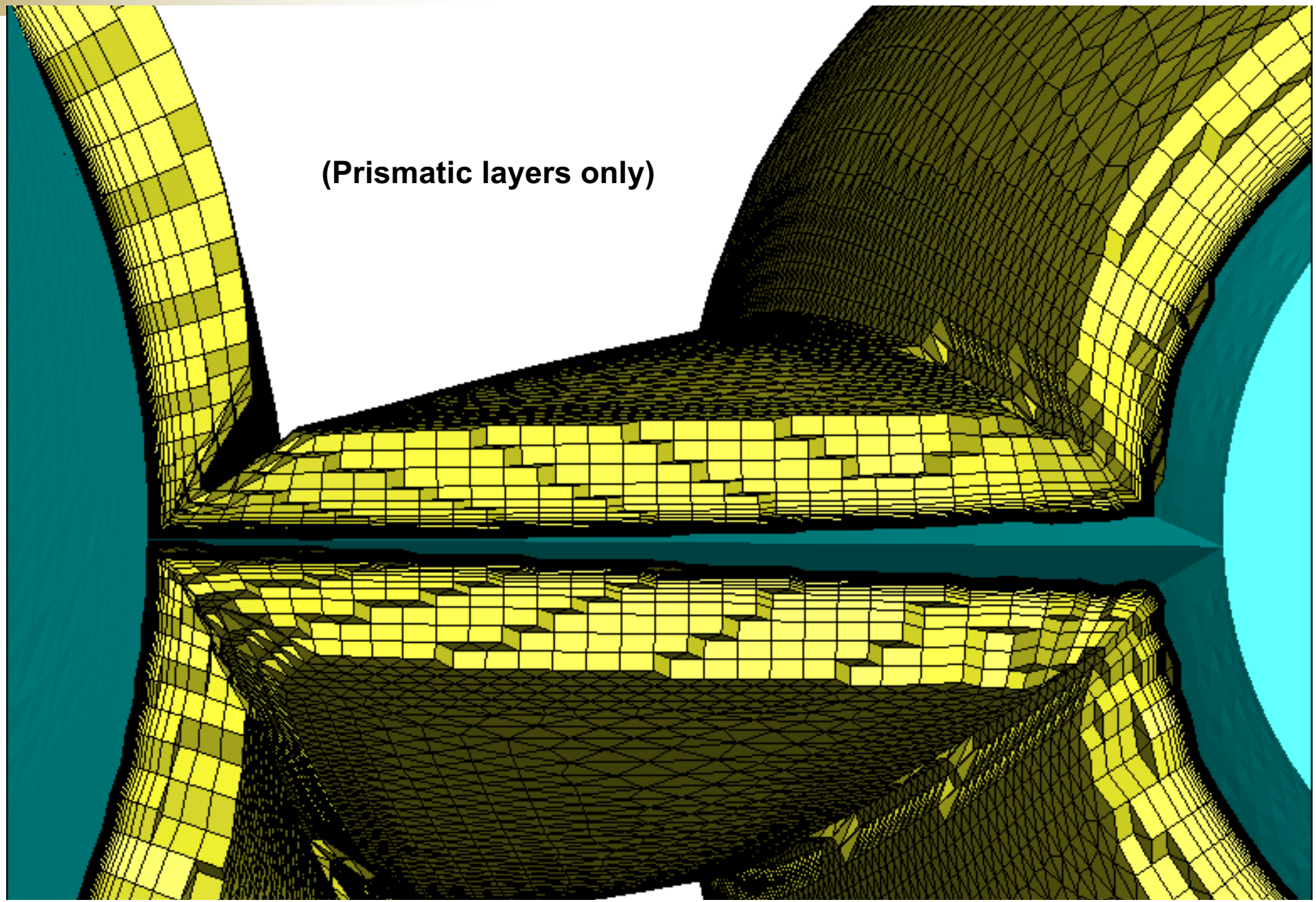




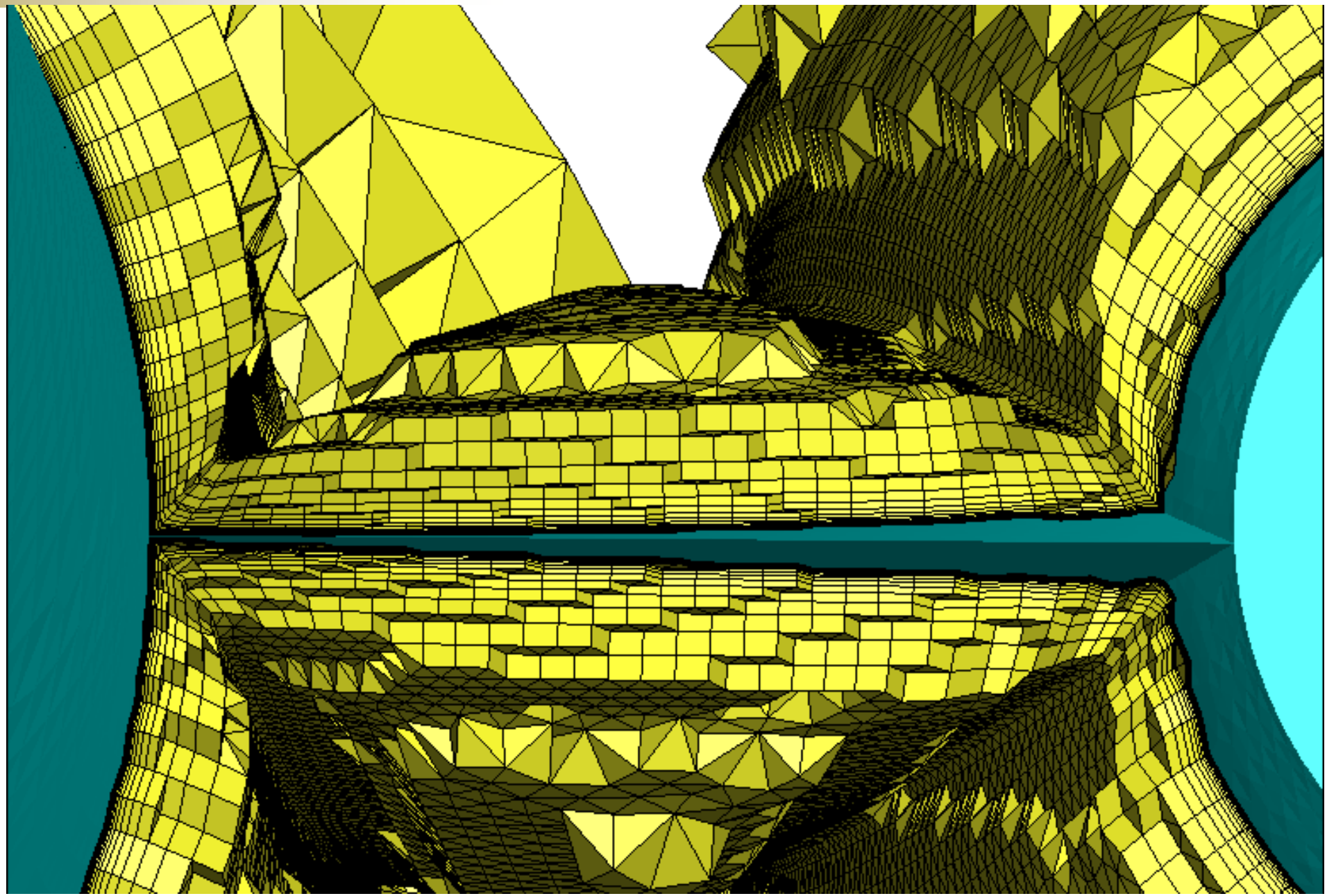
CLOVIS + Smoothing



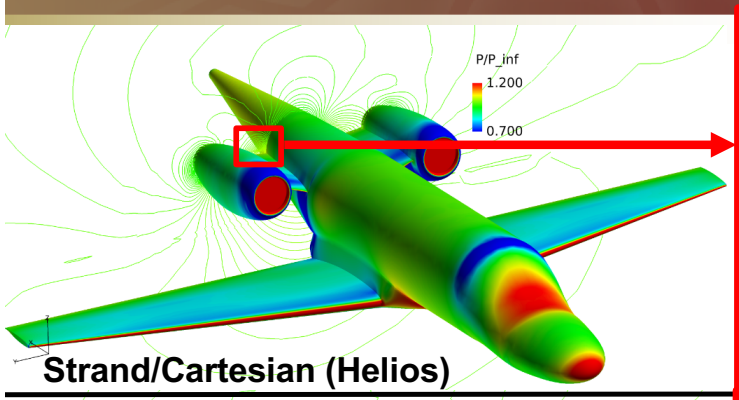
Advancing Front Method
AFLR3



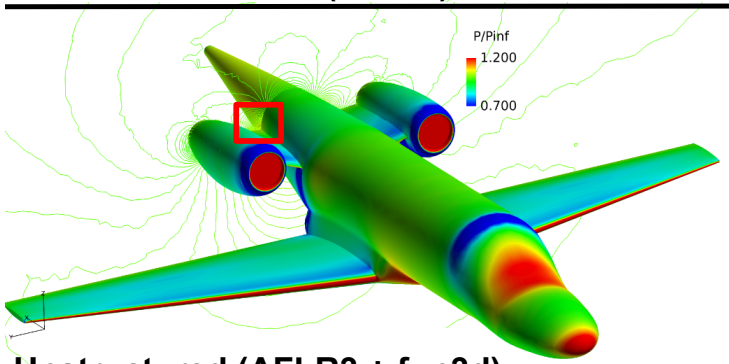
Pointwise
T-rex



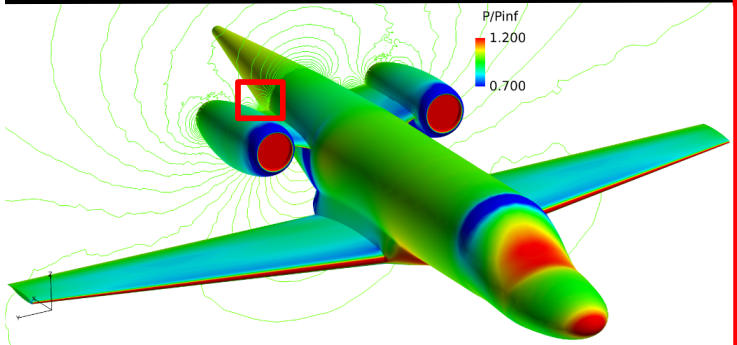
Business Jet
Flow Solutions



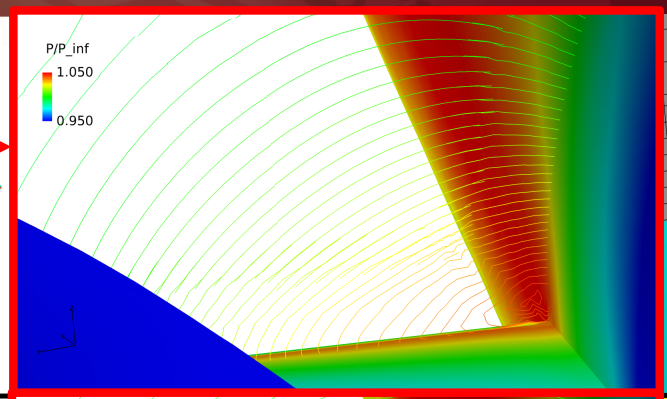
Strand/Cartesian (Helios)

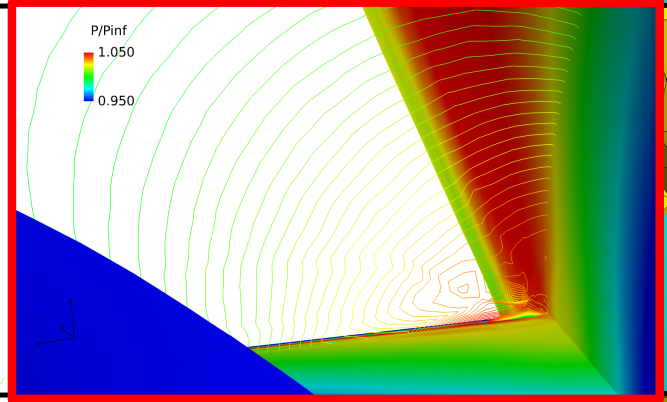


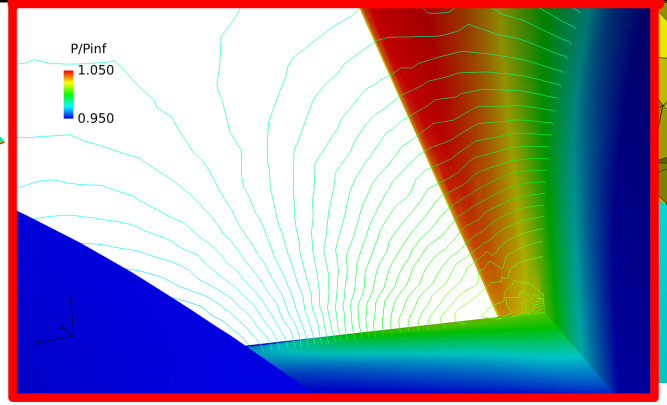
Unstructured (AFLR3 + fun3d)

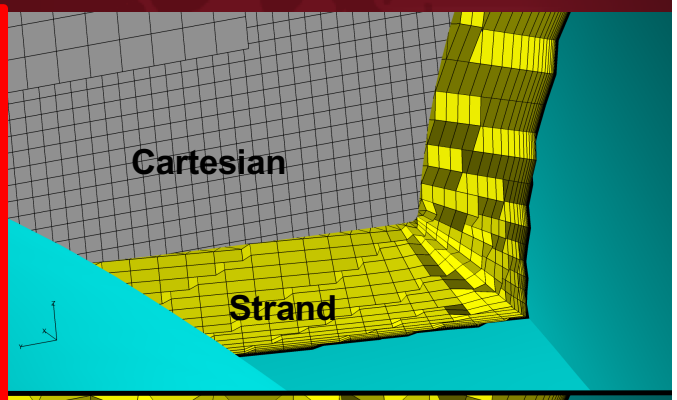


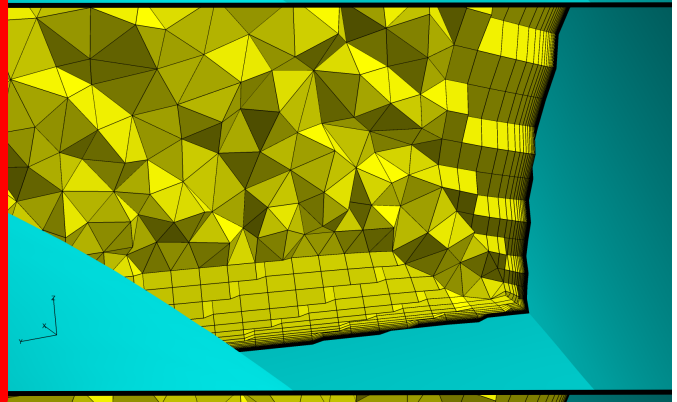
Unstructured (Pointwise + fun3d)

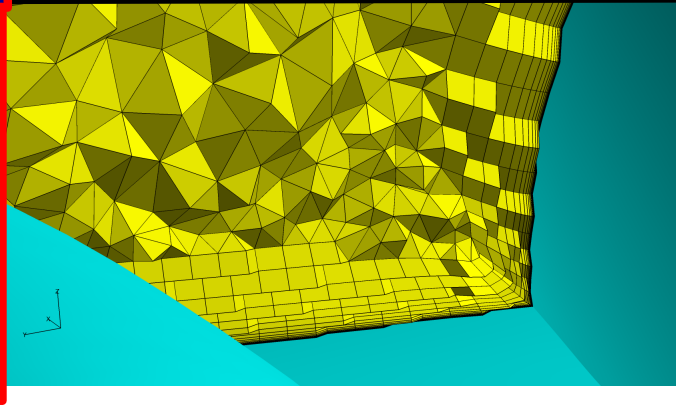




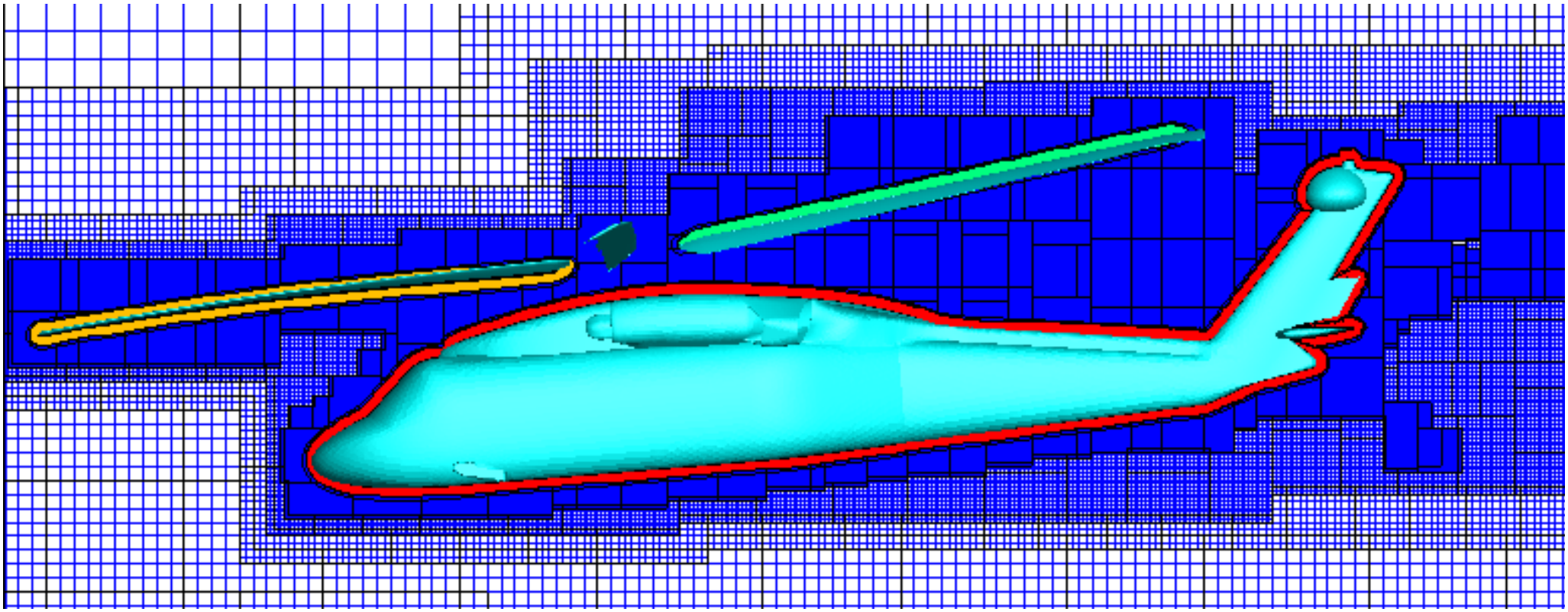








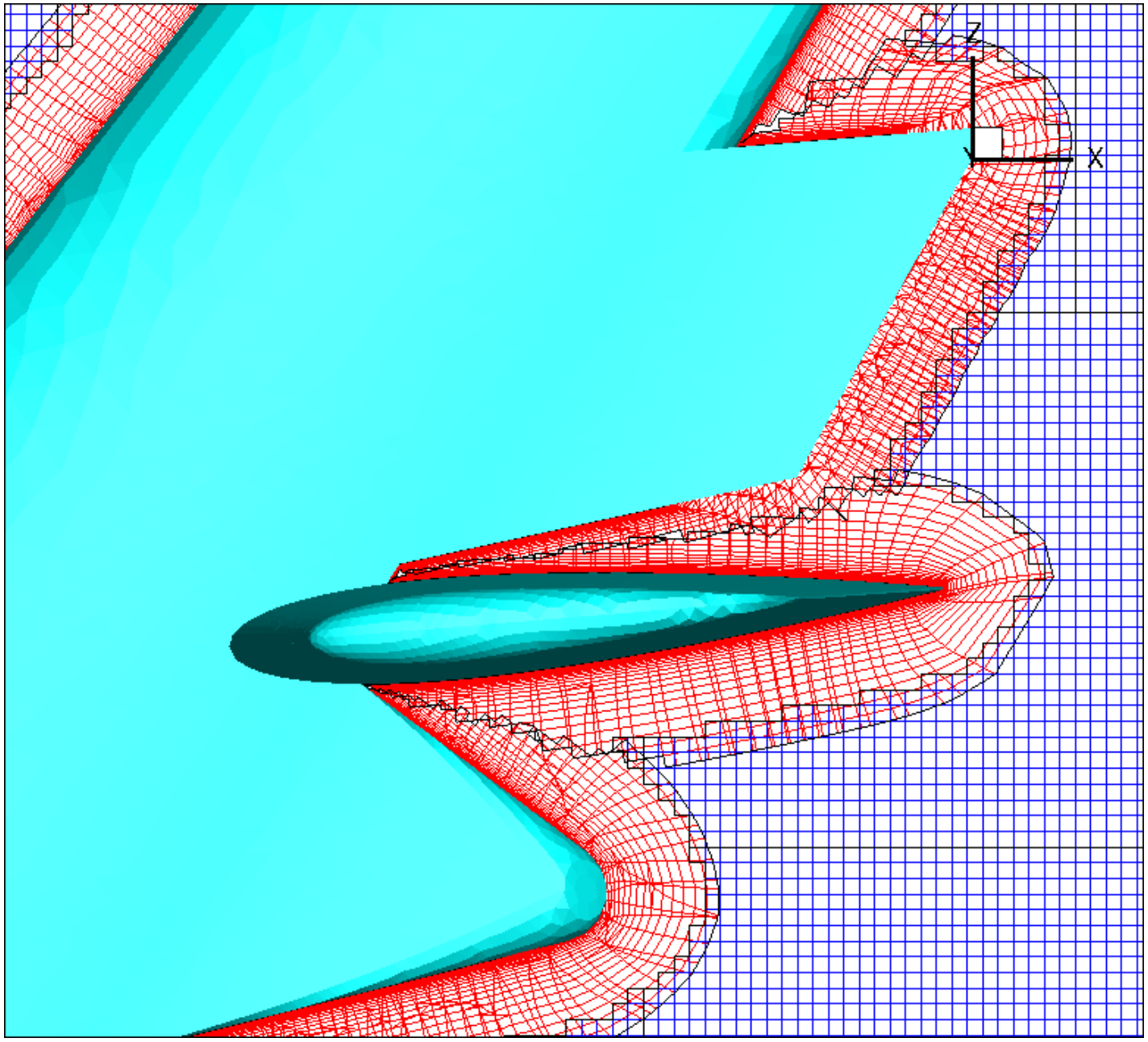
UH-60
Meshes



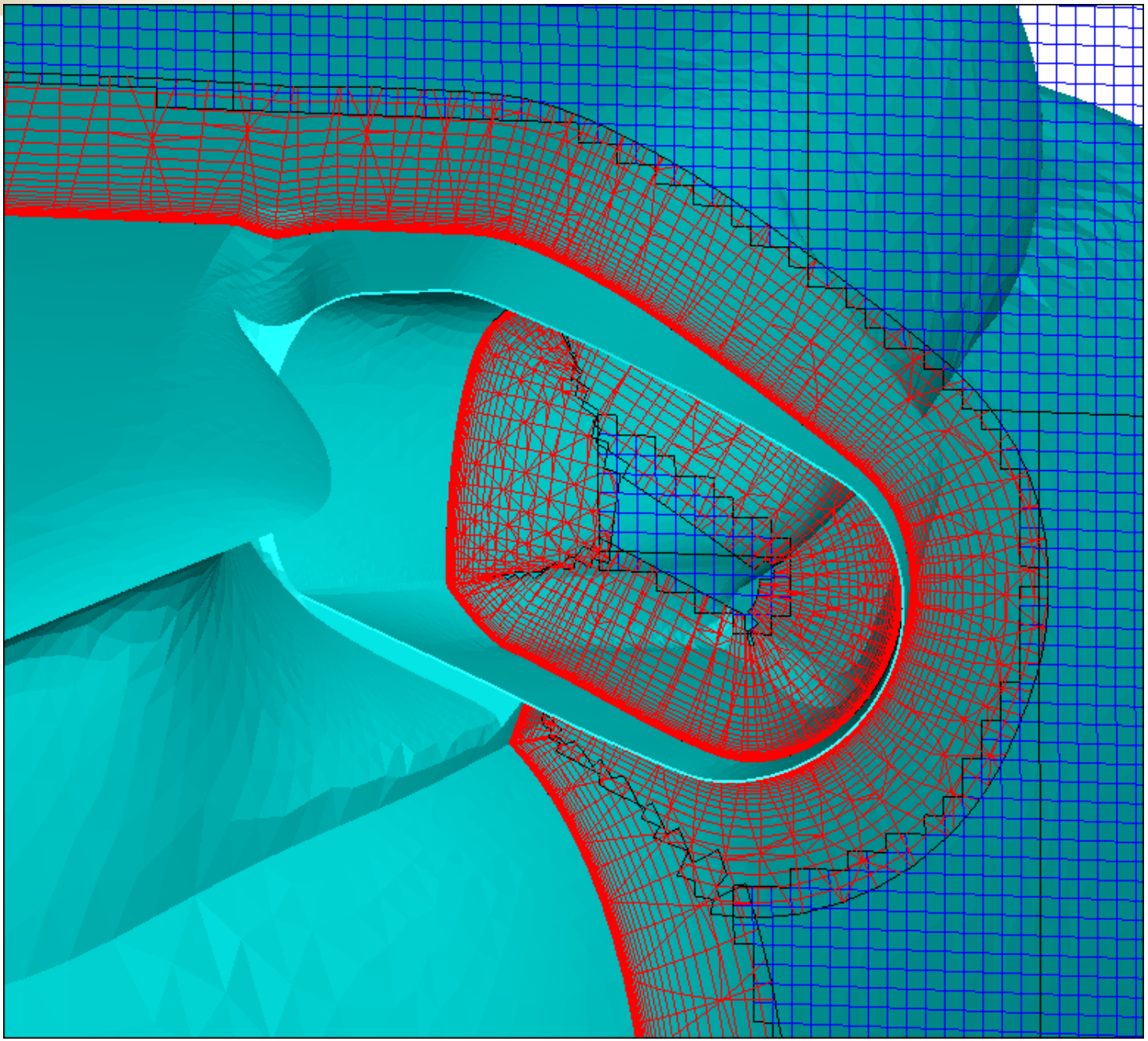
- UH-60 rotor + fuselage in high-speed forward flight
  advance ratio = 0.368

- Blade aeroelastic response + trim computed by RCAS

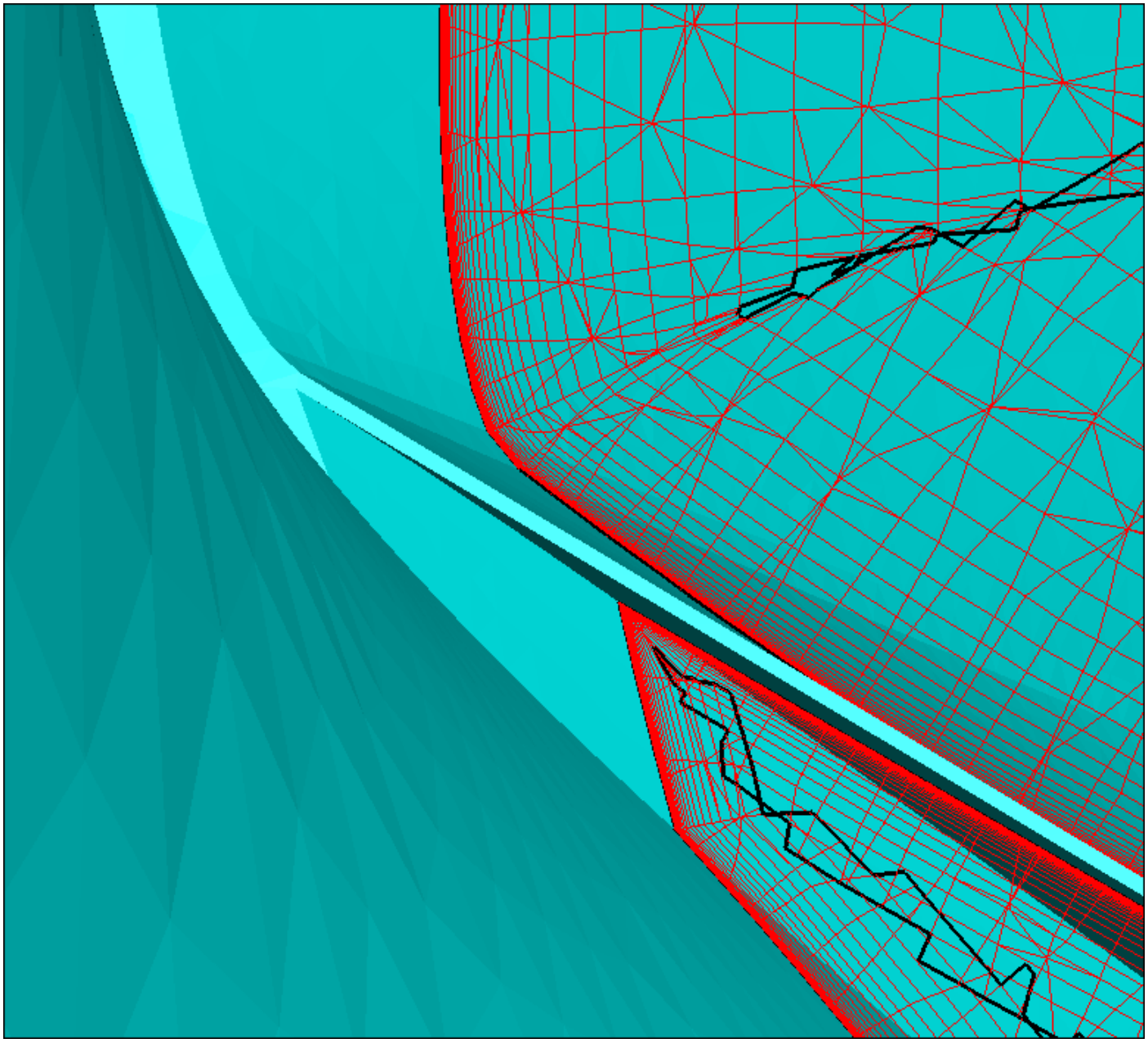
UH-60
Horizontal Tail Detail



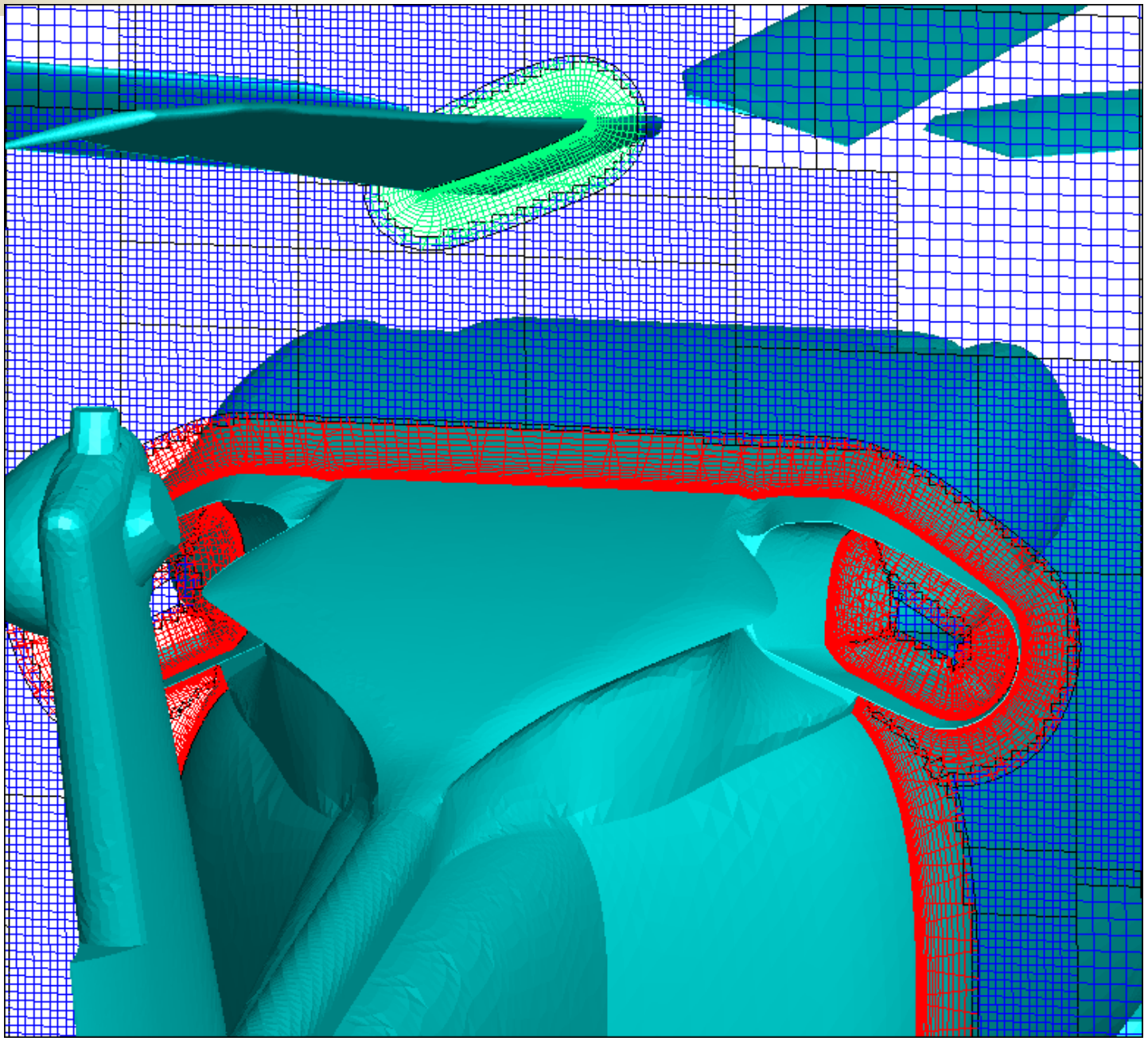
UH-60
Inlet Detail



UH-60
Inlet Corner Detail

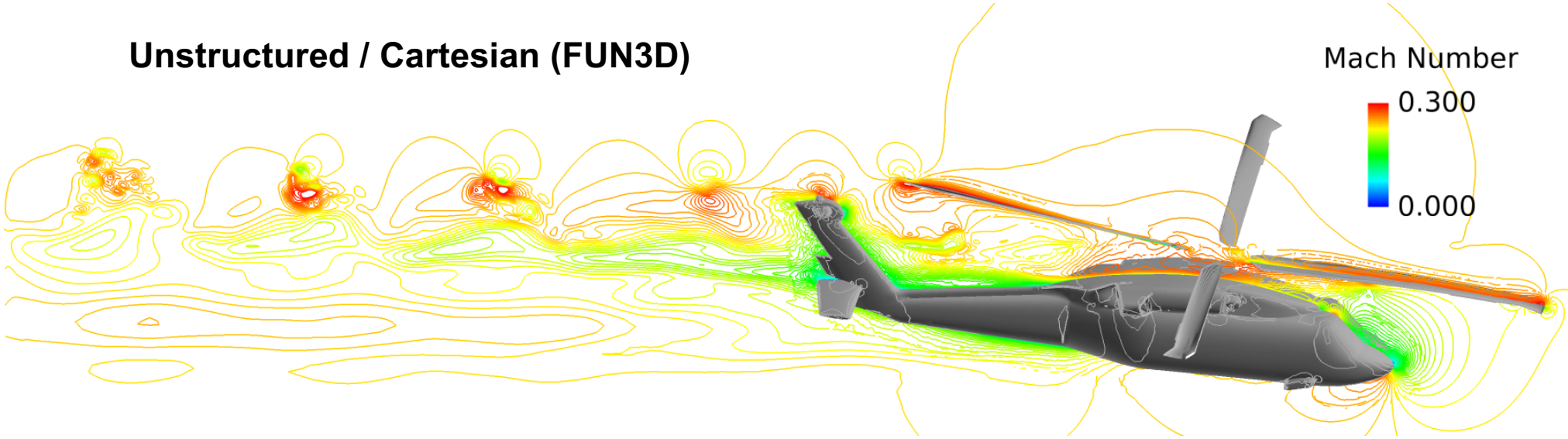


UH-60
Overall View

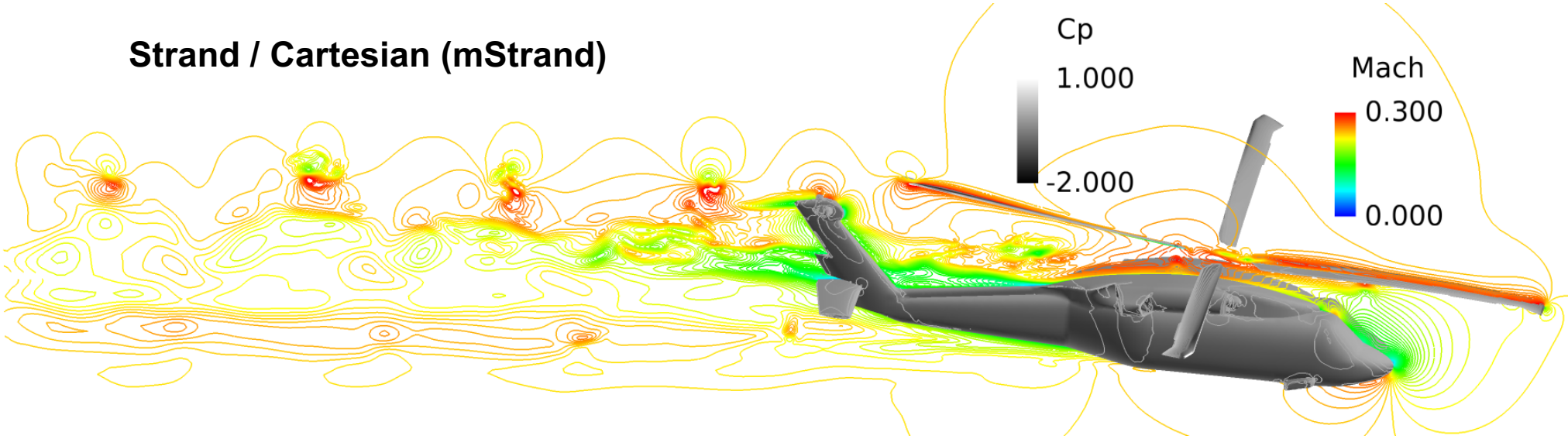


Flow Solutions

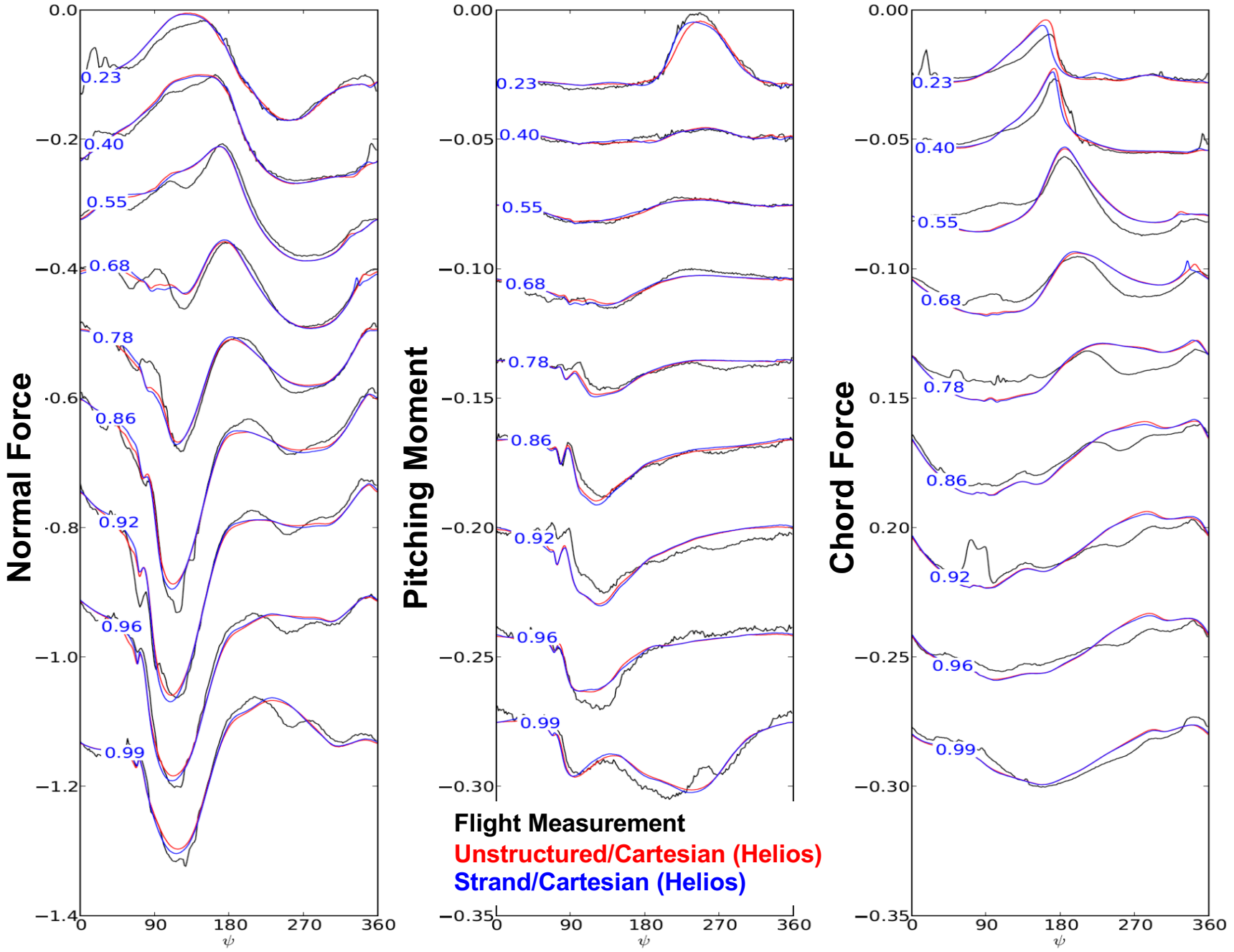
Unstructured / Cartesian (FUN3D)



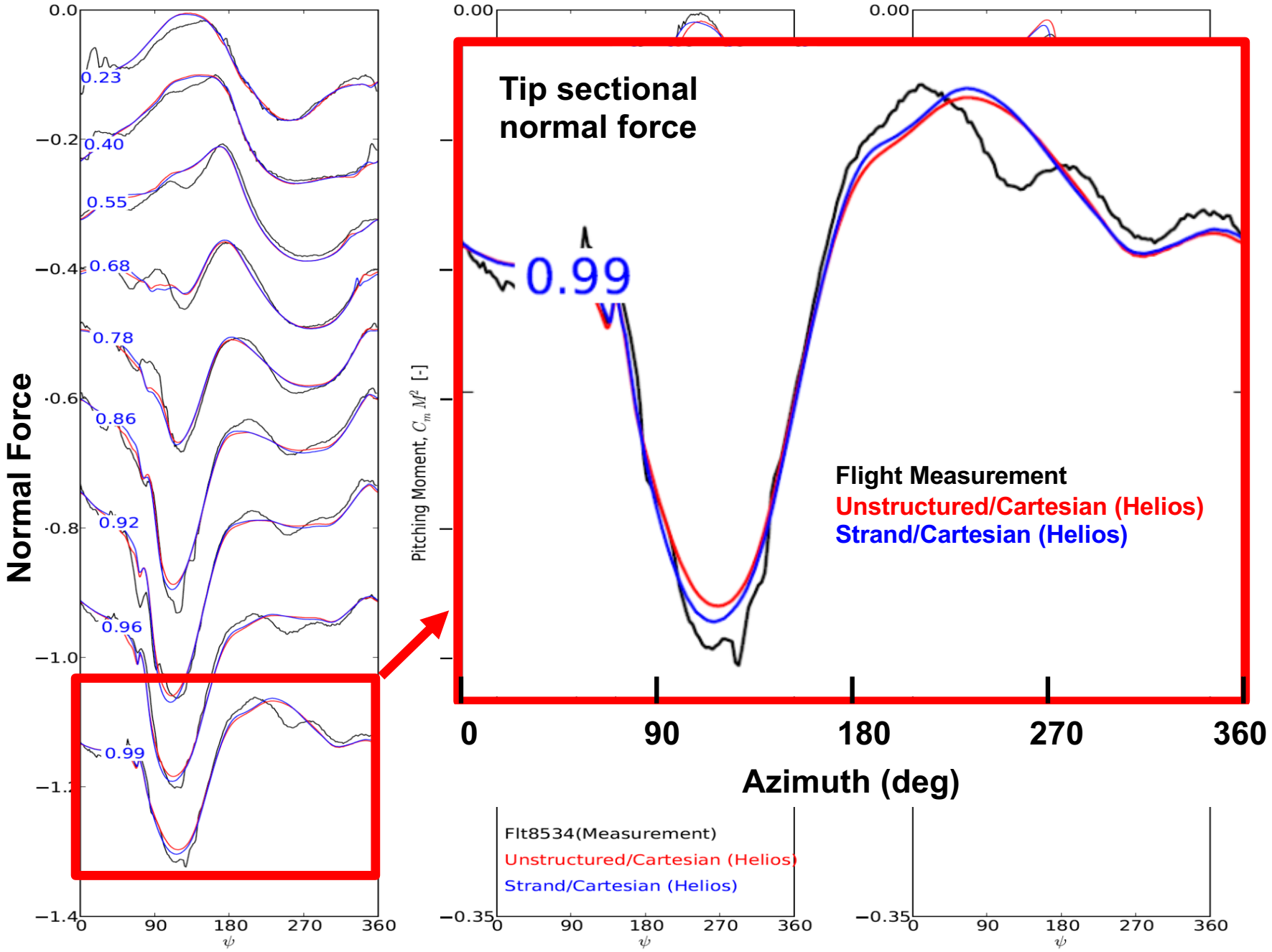
Strand / Cartesian (mStrand)



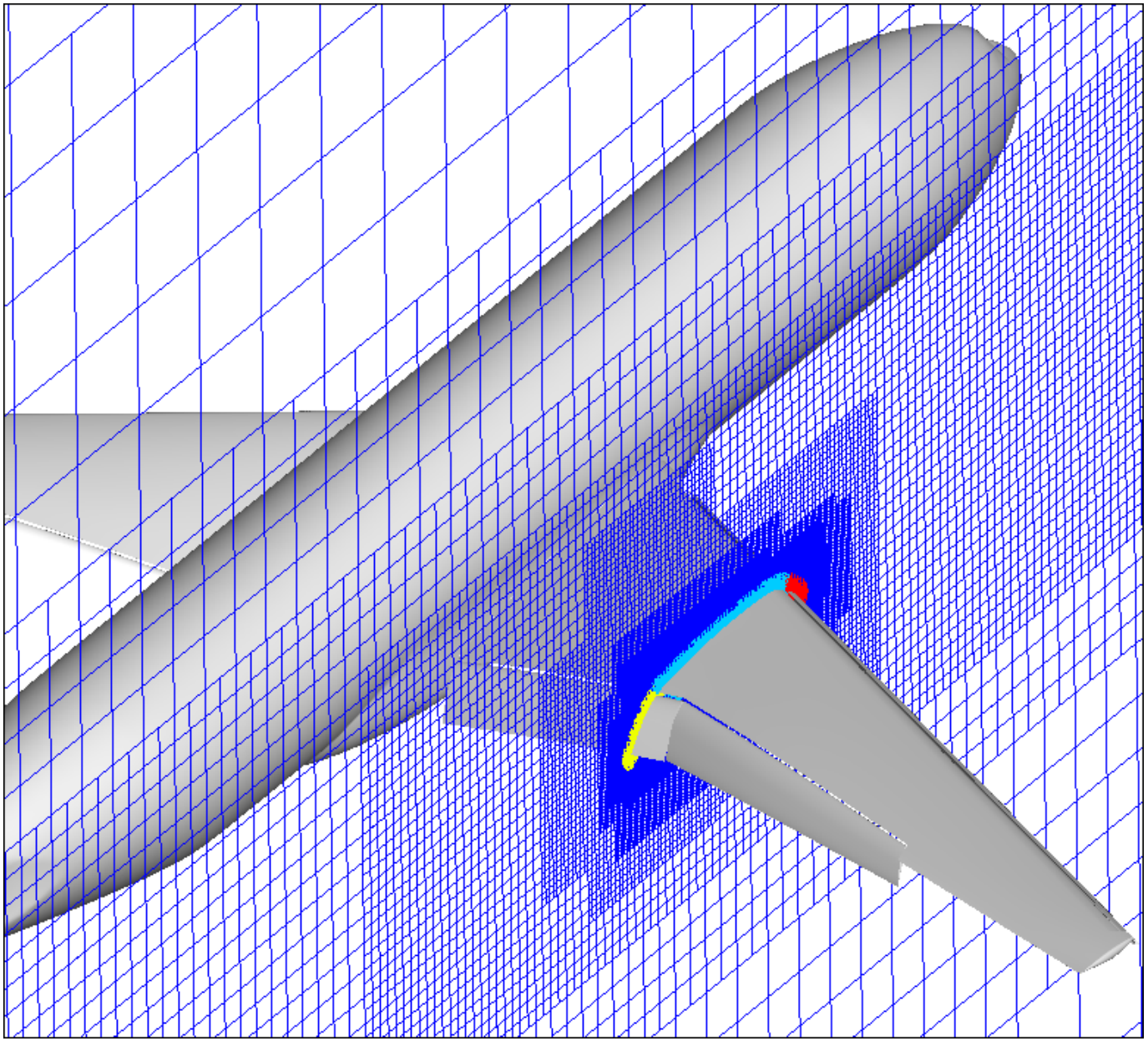
Sectional Aerodynamic Loads



Sectional Aerodynamic Loads



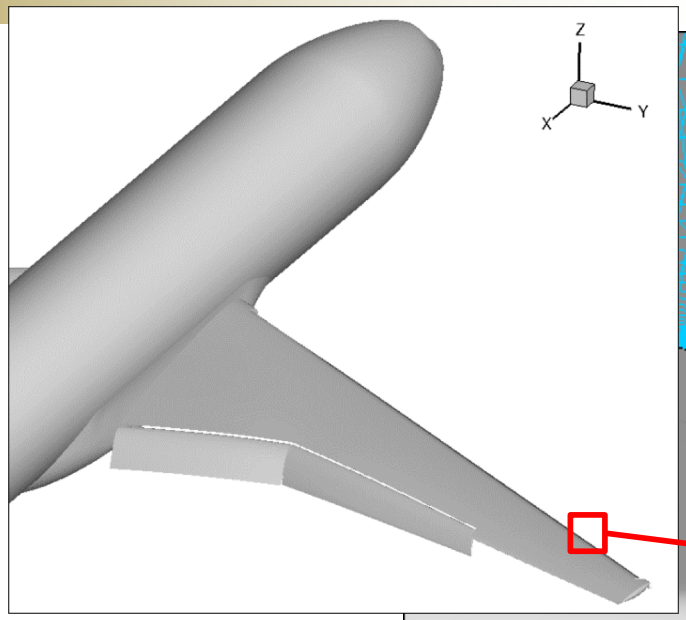
High-Lift CRM
AIAA HL workshop

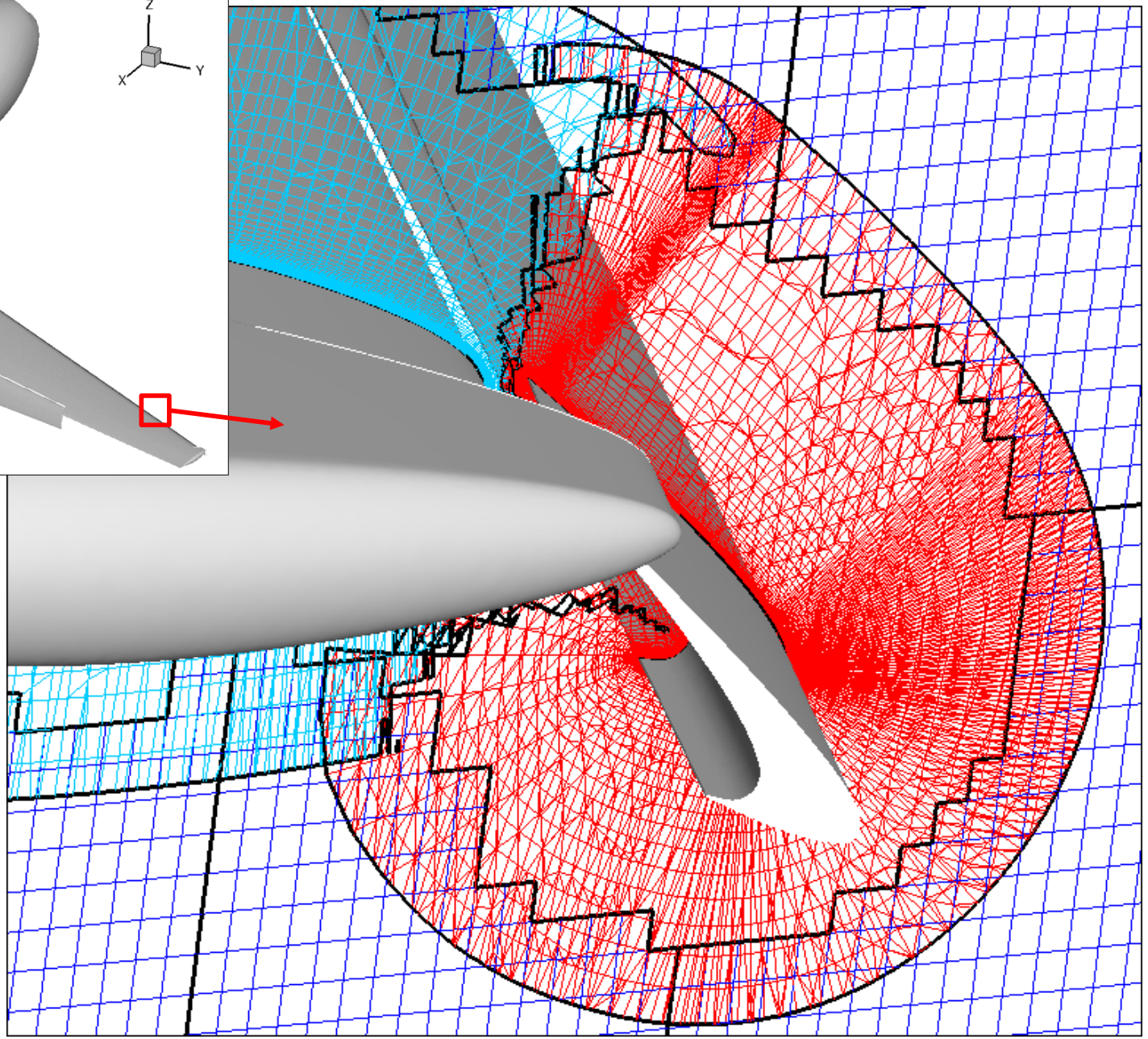


- Wing-body high lift configuration:
  - Fuselage + wing
  - 2 flaps
  - 1 slat
- Slat deployed 30°
- Flaps deployed 37°
- Flow solution at
  aoa = 8° and 16°

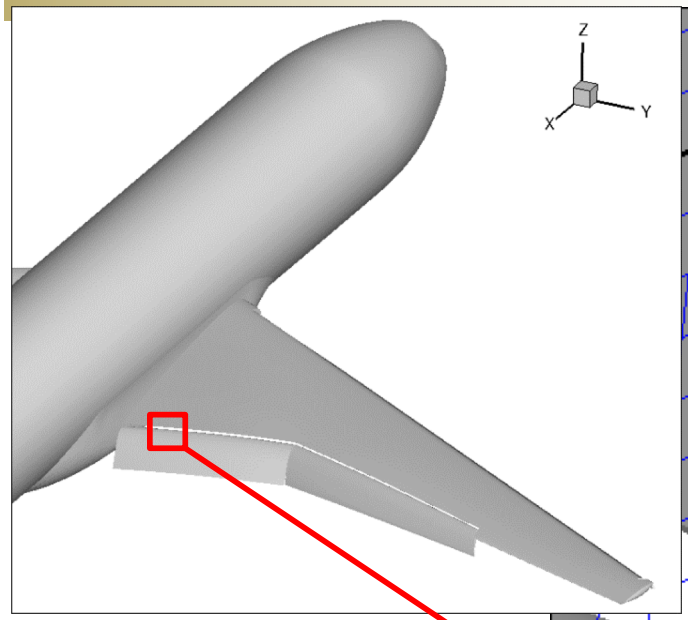
  M=0.2

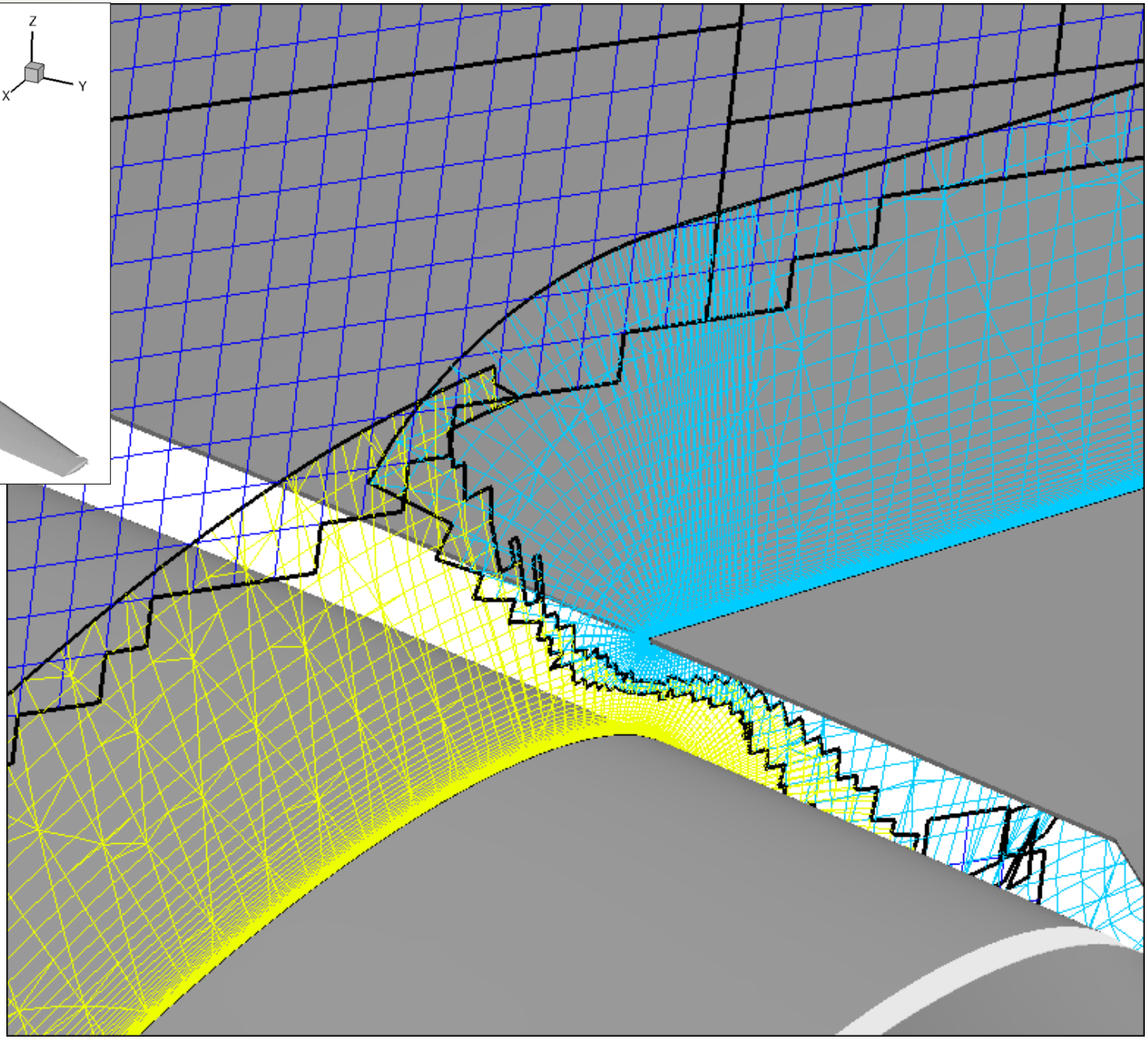
Wing LE / Slat



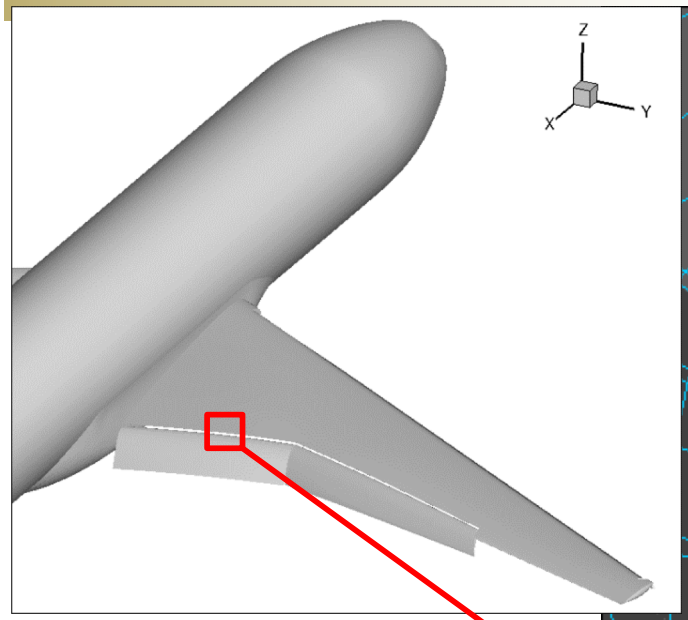


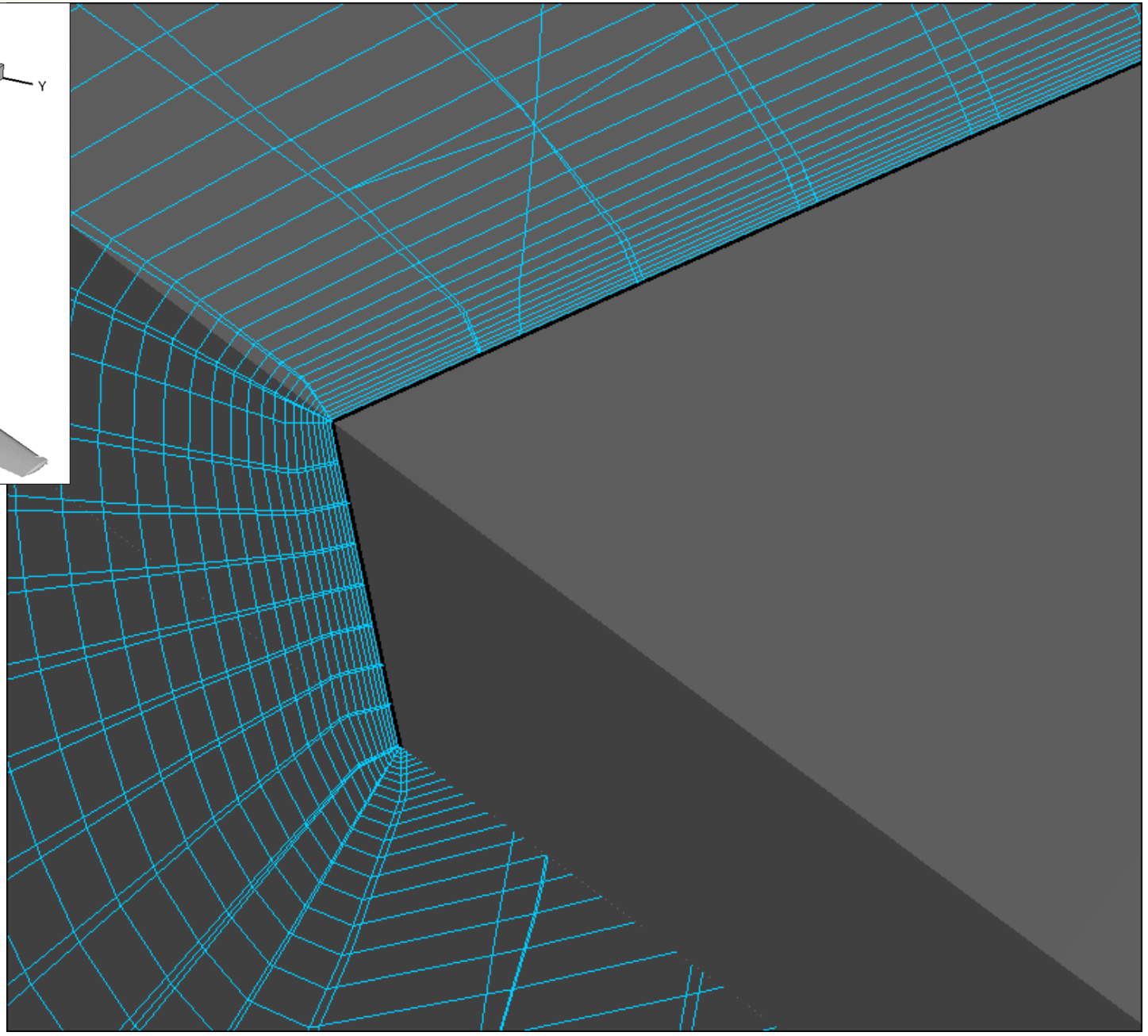
Wing TE / Flap 1



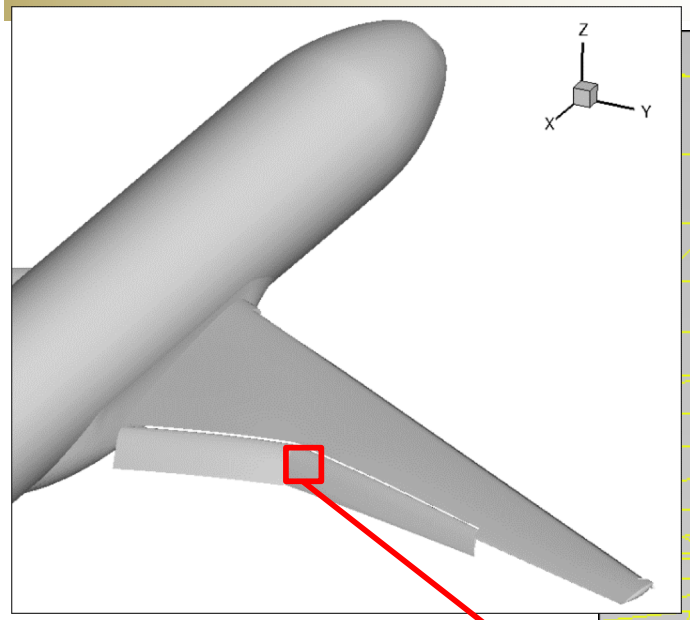


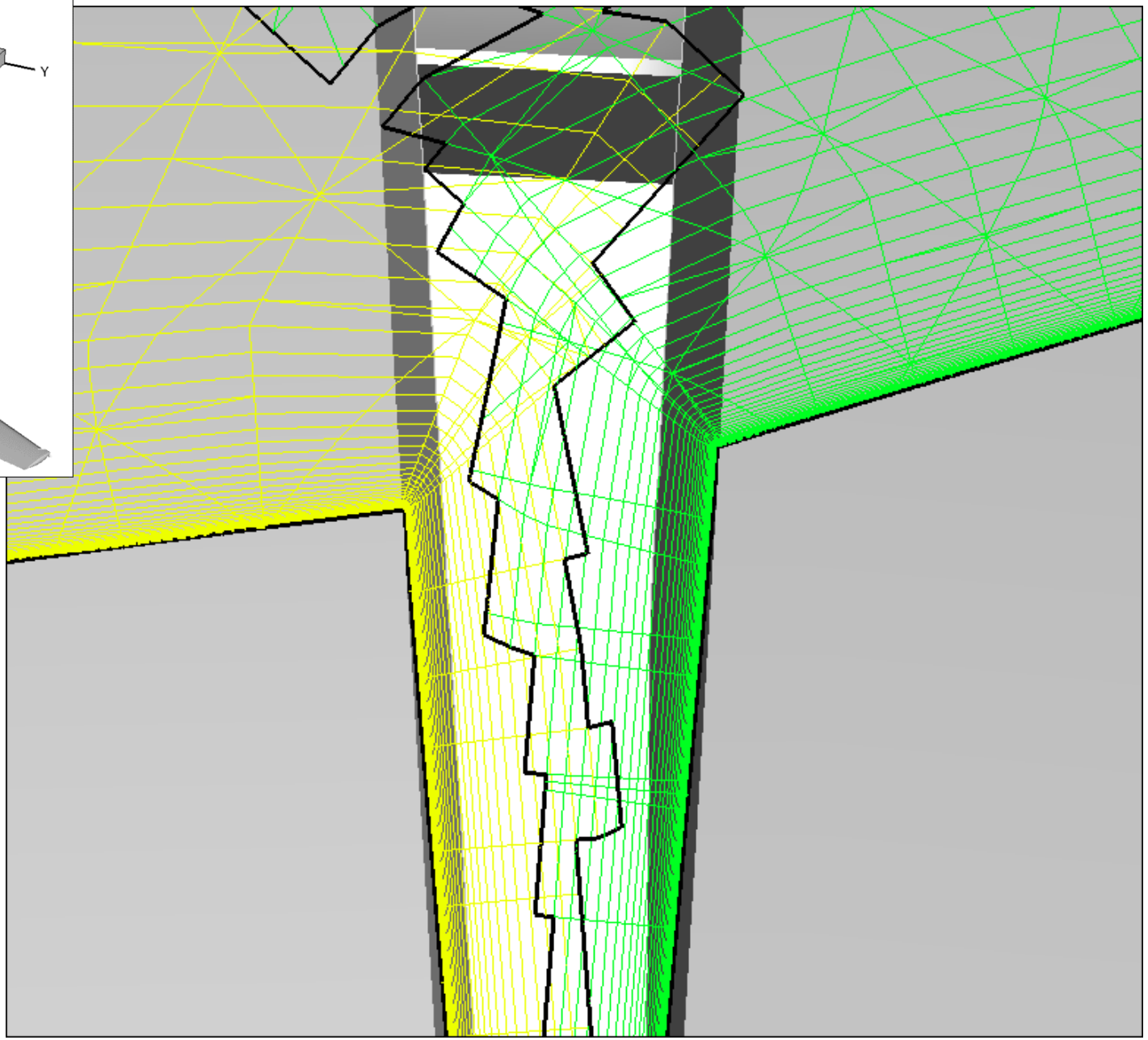
Wing Trailing Edge



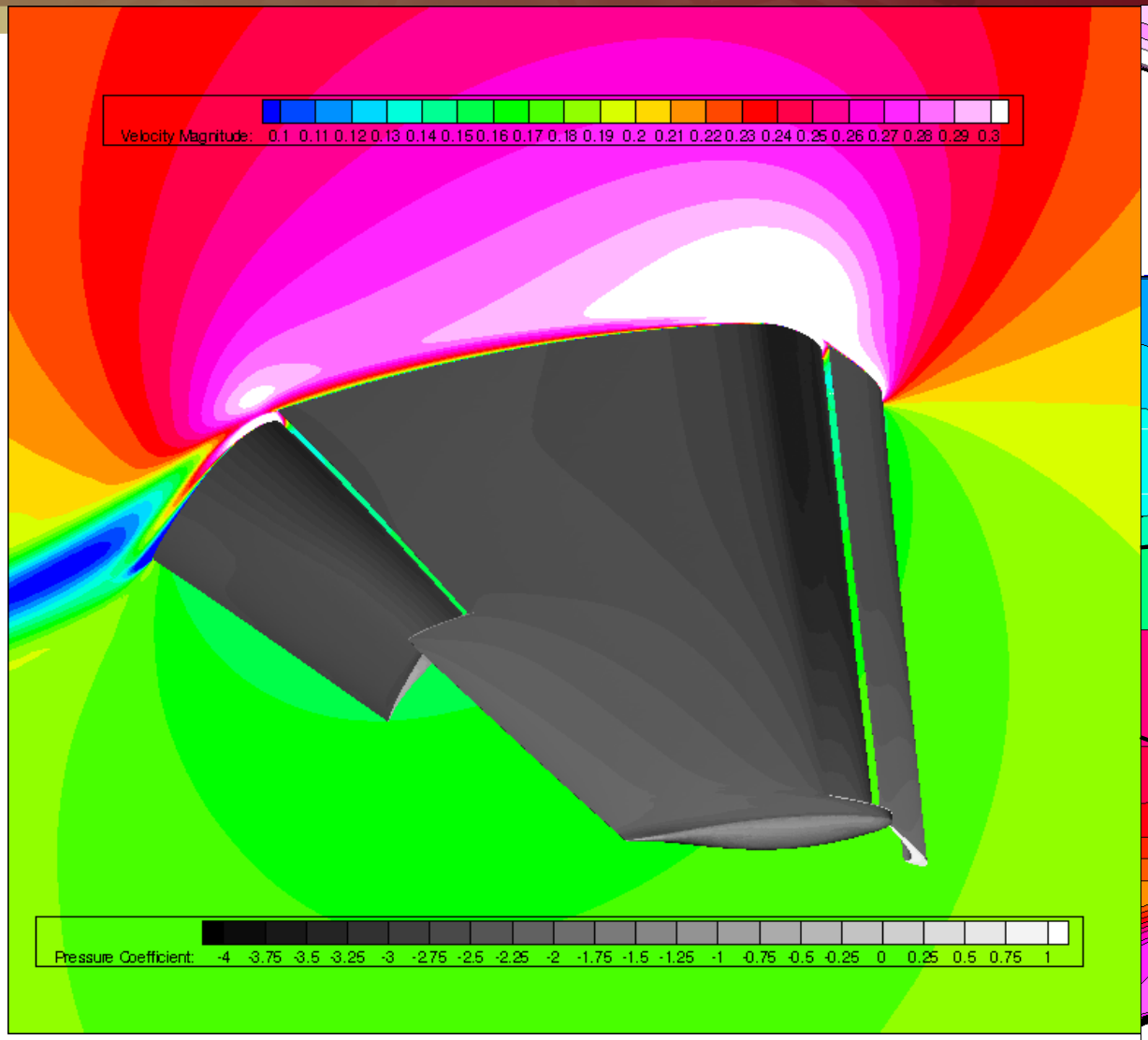


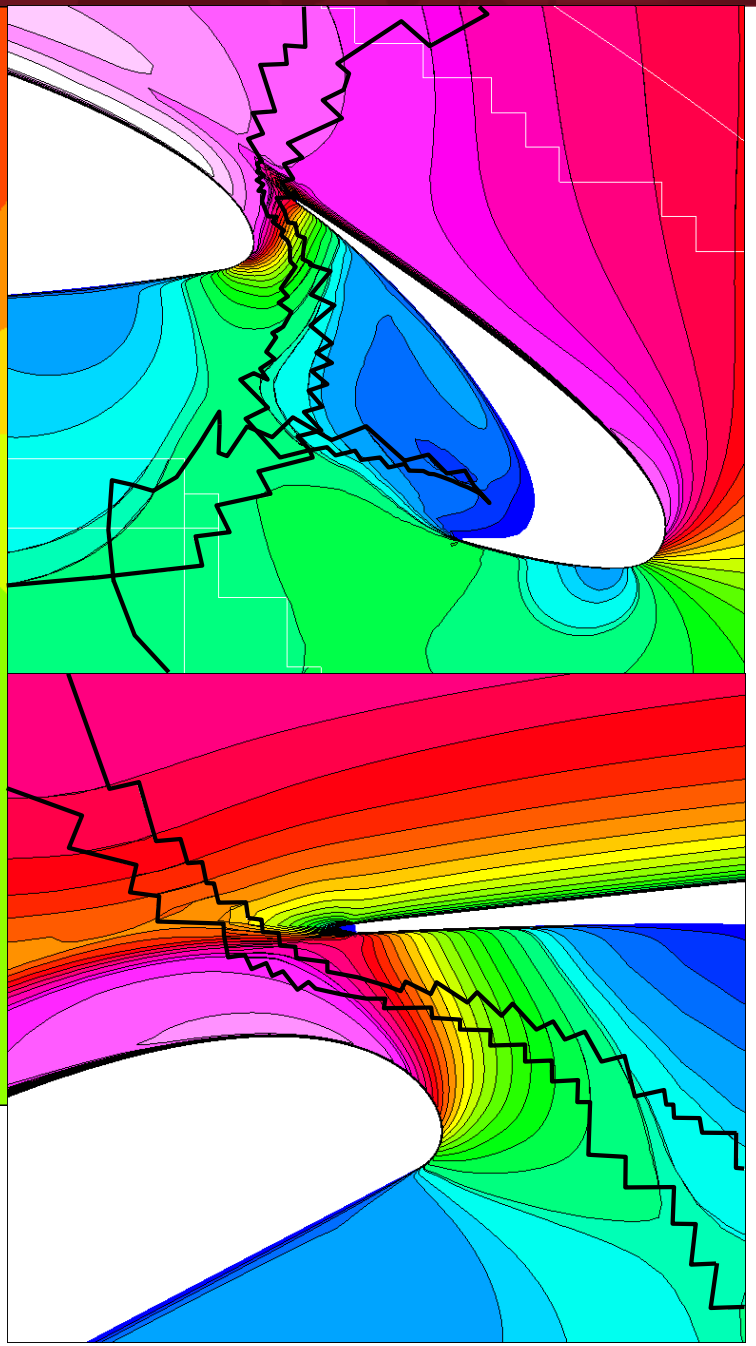
Flap 1 / Flap 2 Gap





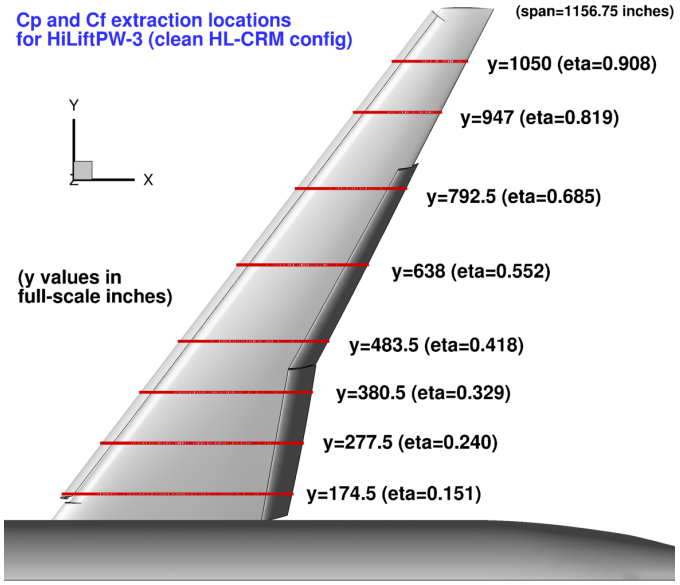
HLCRM
Flow Solution

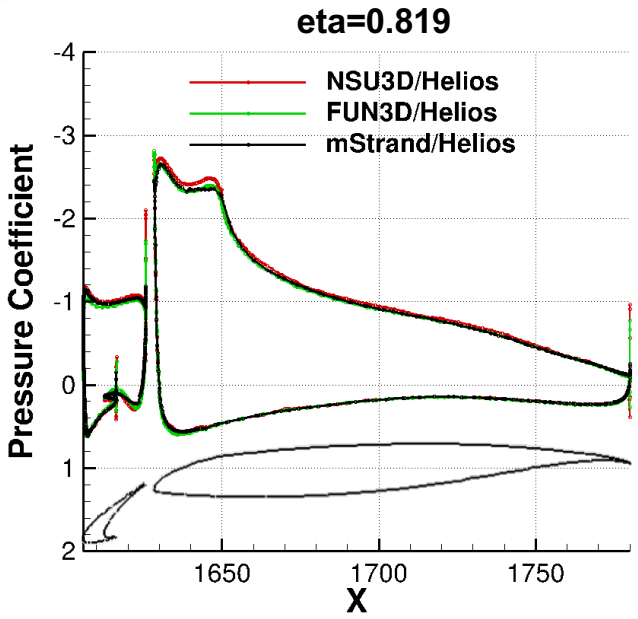


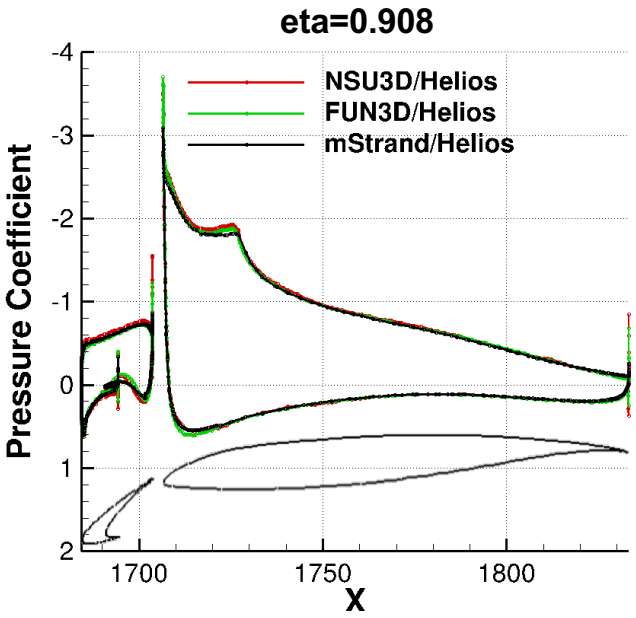


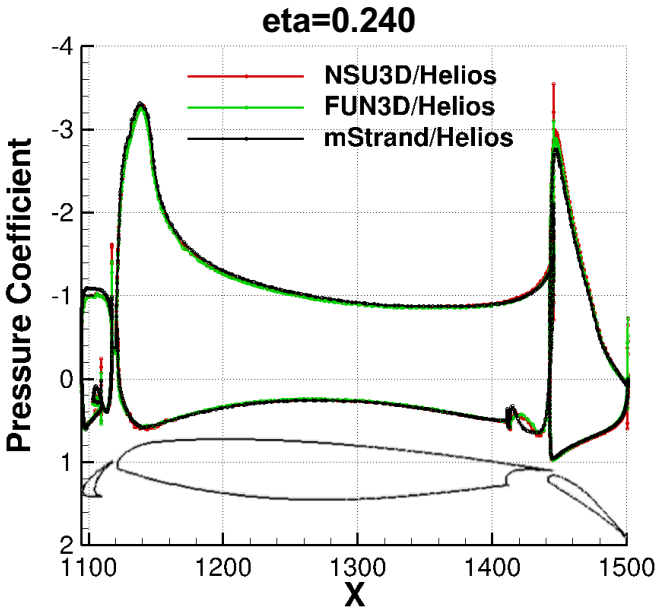
- Slat deployed 30°
- Flaps deployed 37°
- Flow solution at aoa = 8° , M =0.2

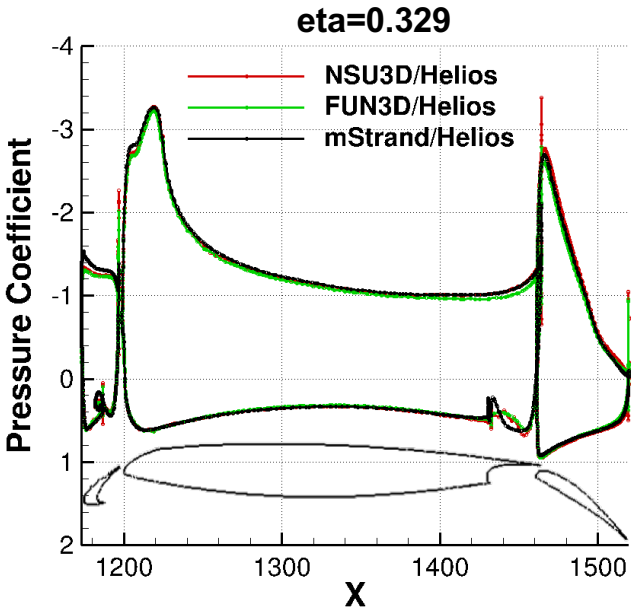
HLCRM
Pressure Coefficient

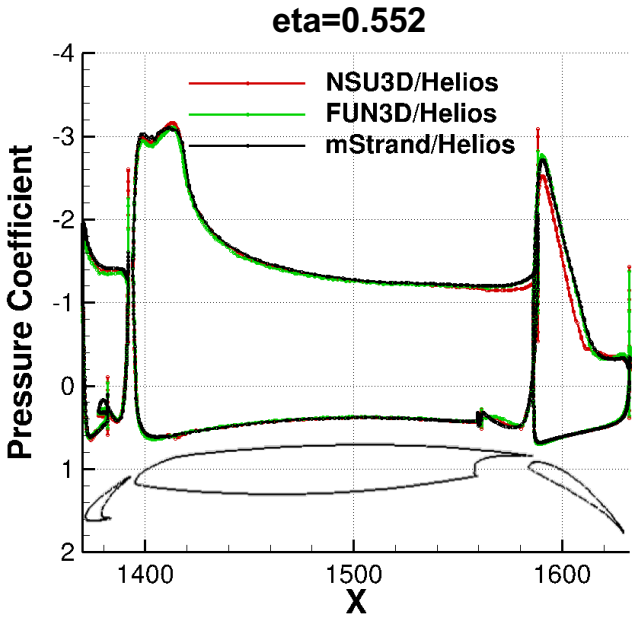




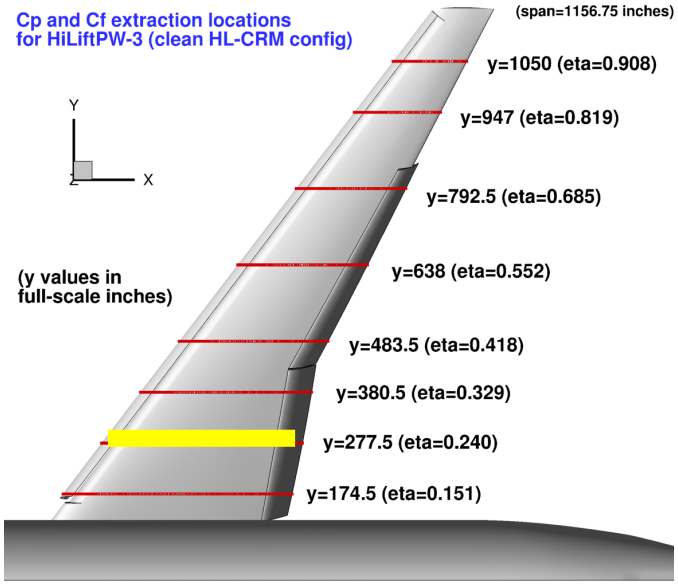








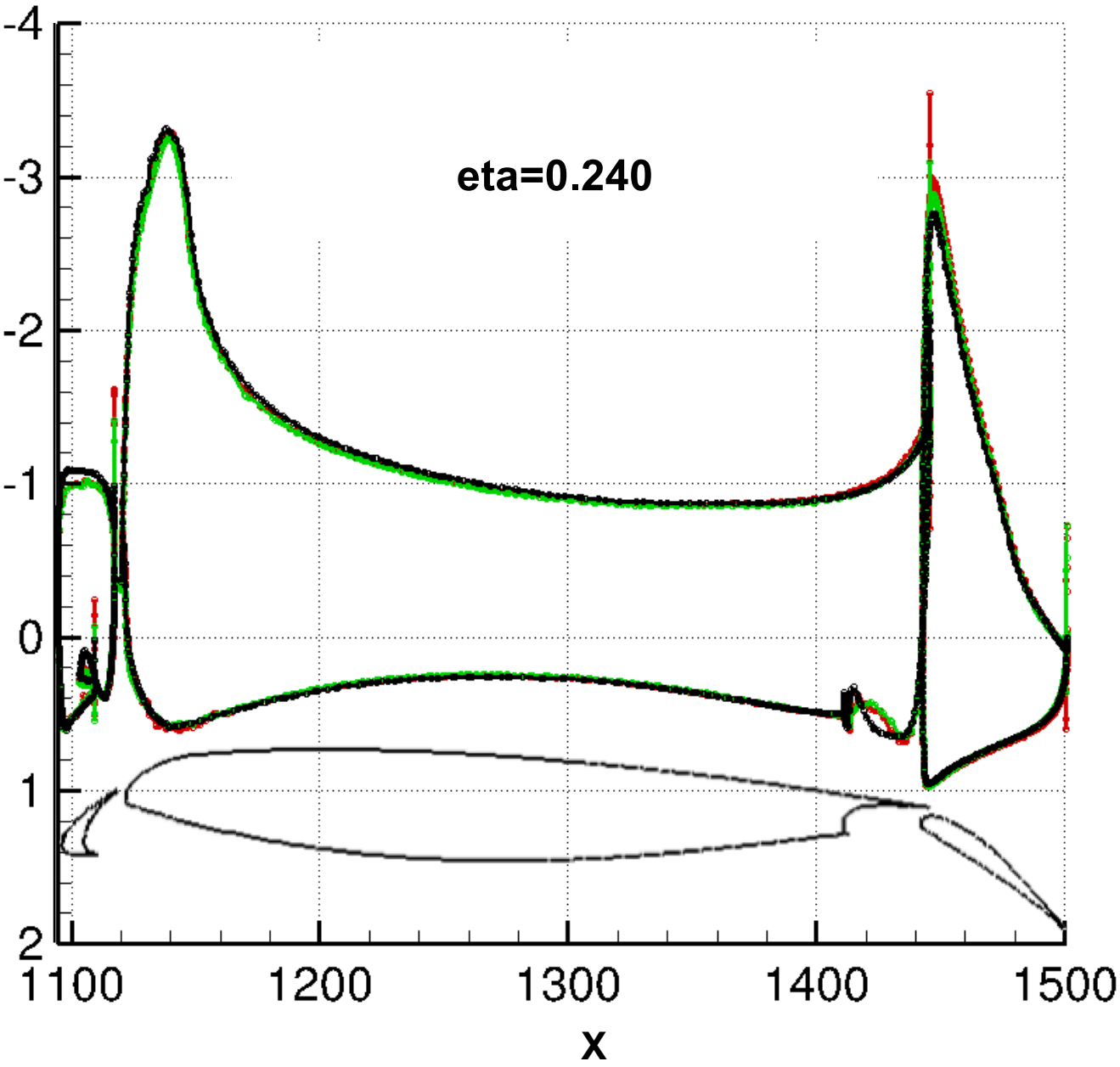
HLCRM
Pressure Coefficient



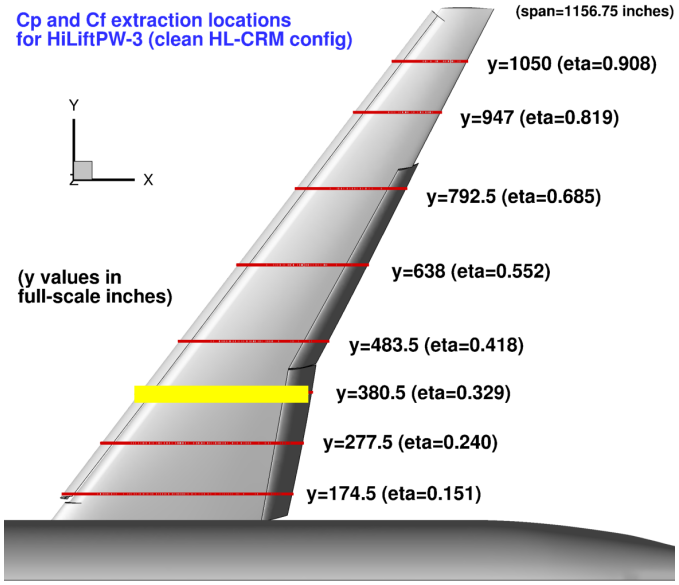
NSU3D / Helios

FUN3D / Helios

mStrand / Helios



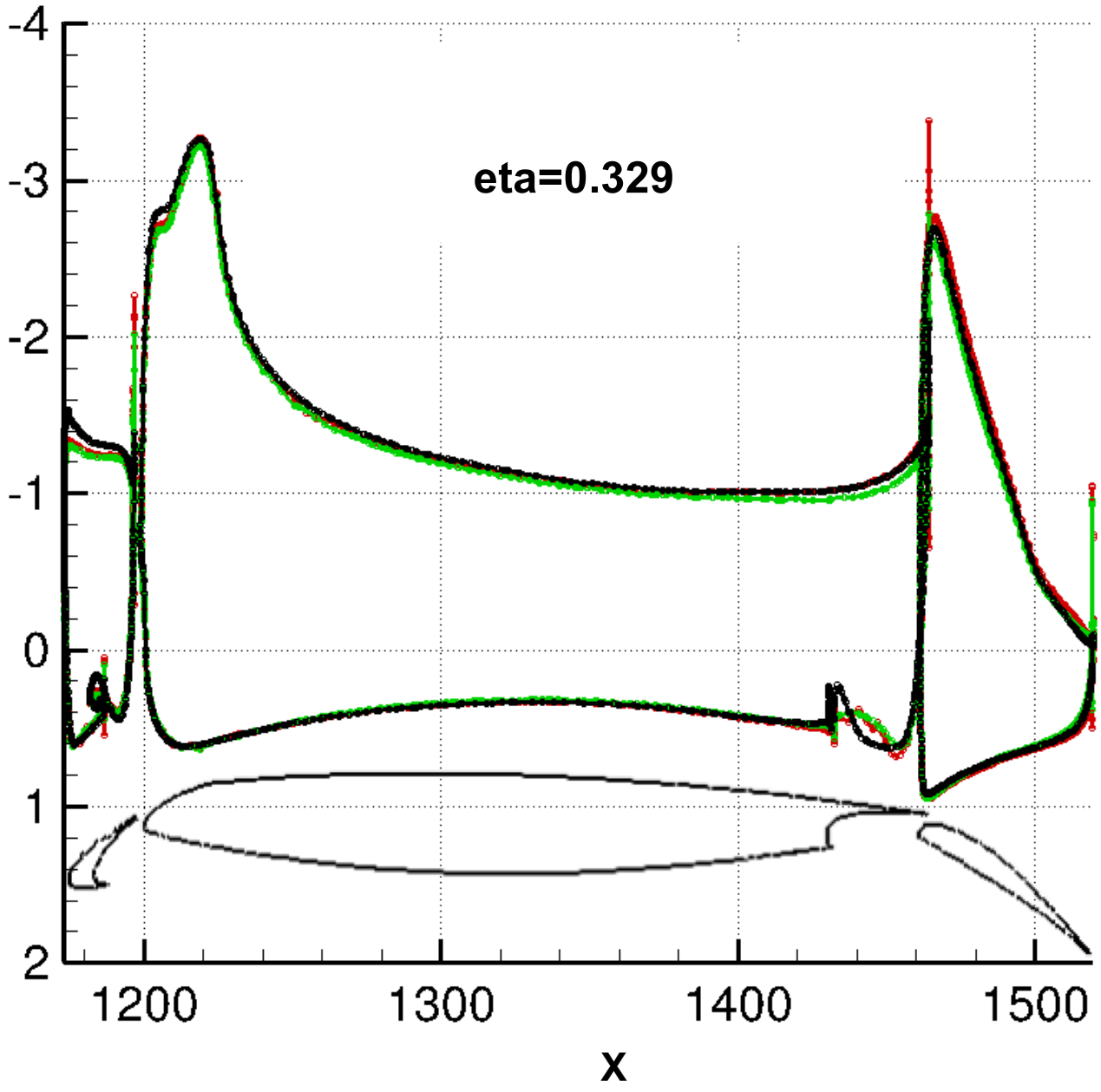
HLCRM
Pressure Coefficient



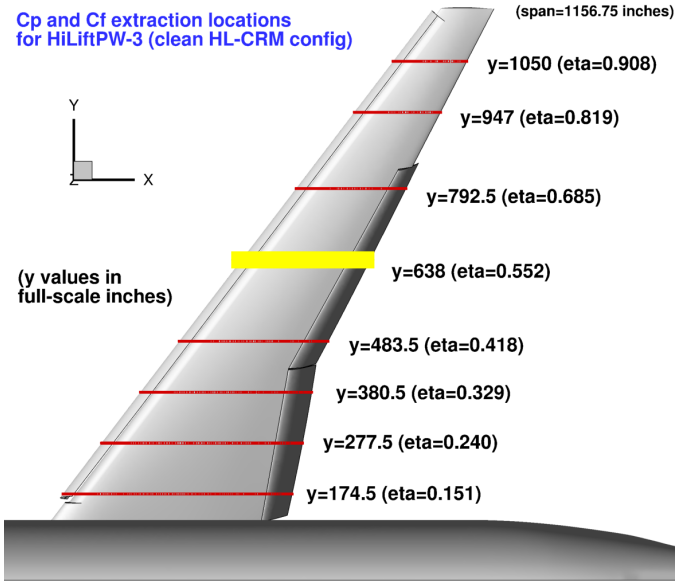
NSU3D / Helios

FUN3D / Helios

mStrand / Helios



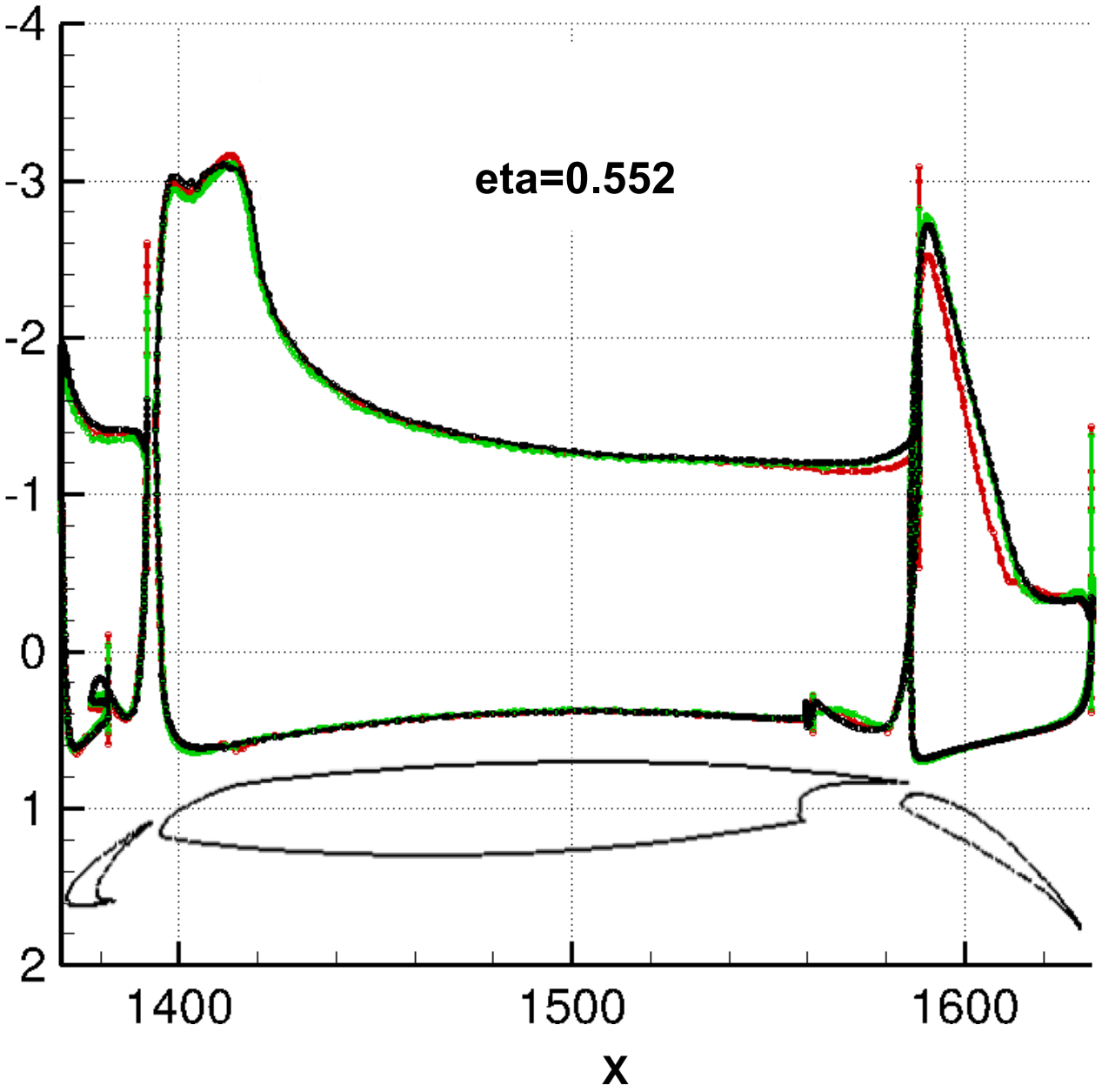
HLCRM
Pressure Coefficient



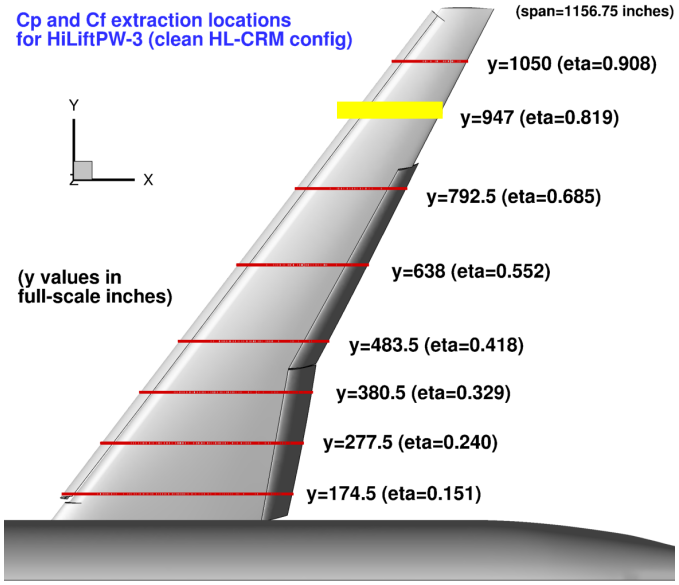
NSU3D / Helios

FUN3D / Helios

mStrand / Helios



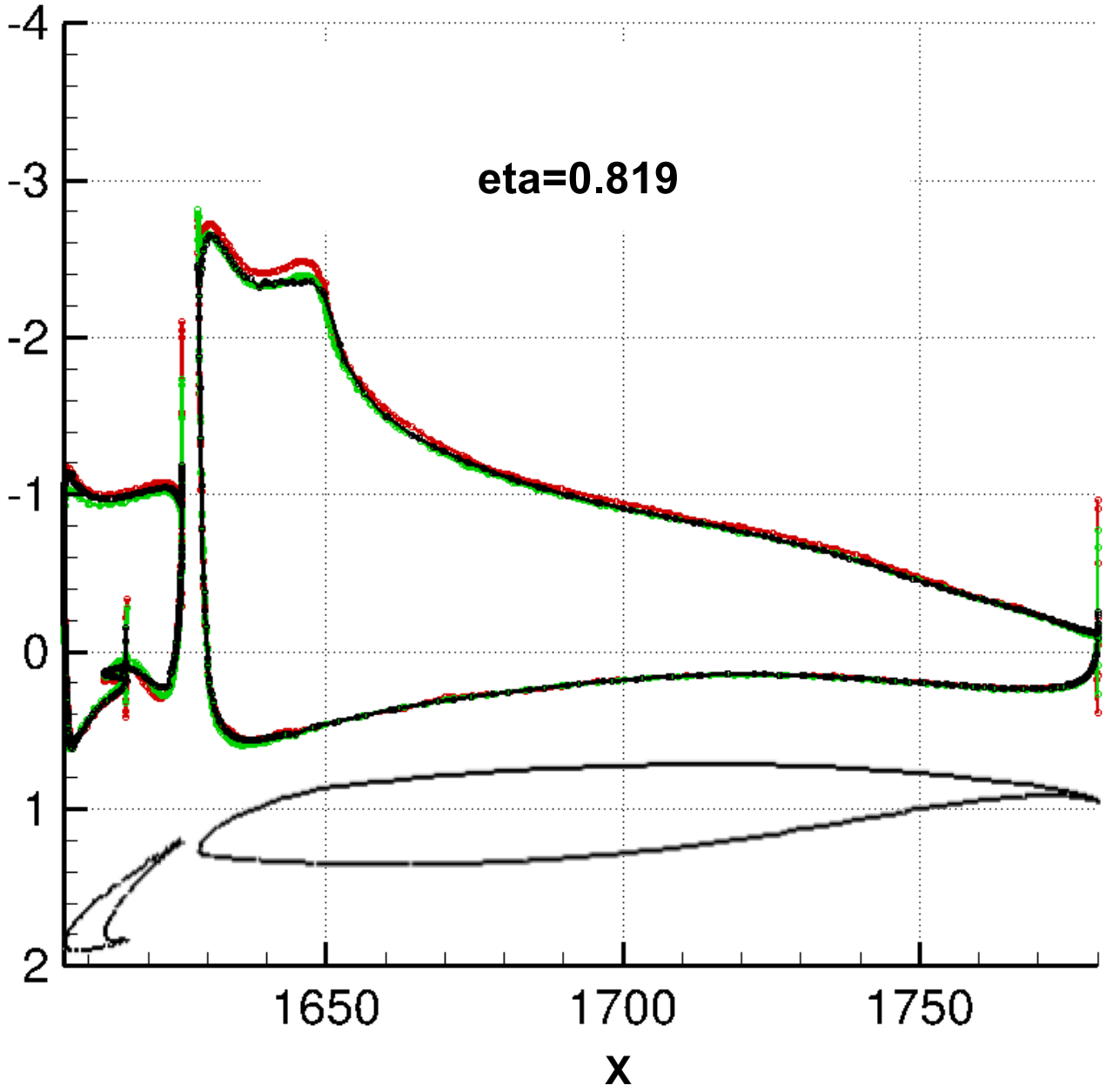
HLCRM
Pressure Coefficient



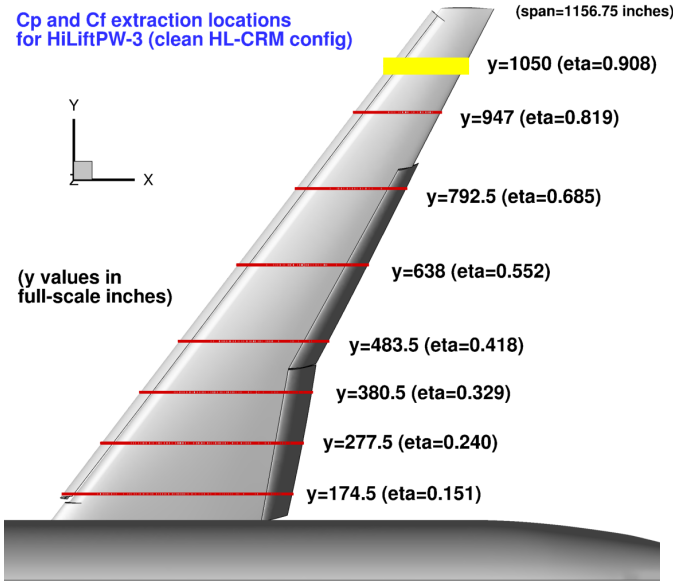
NSU3D / Helios

FUN3D / Helios

mStrand / Helios



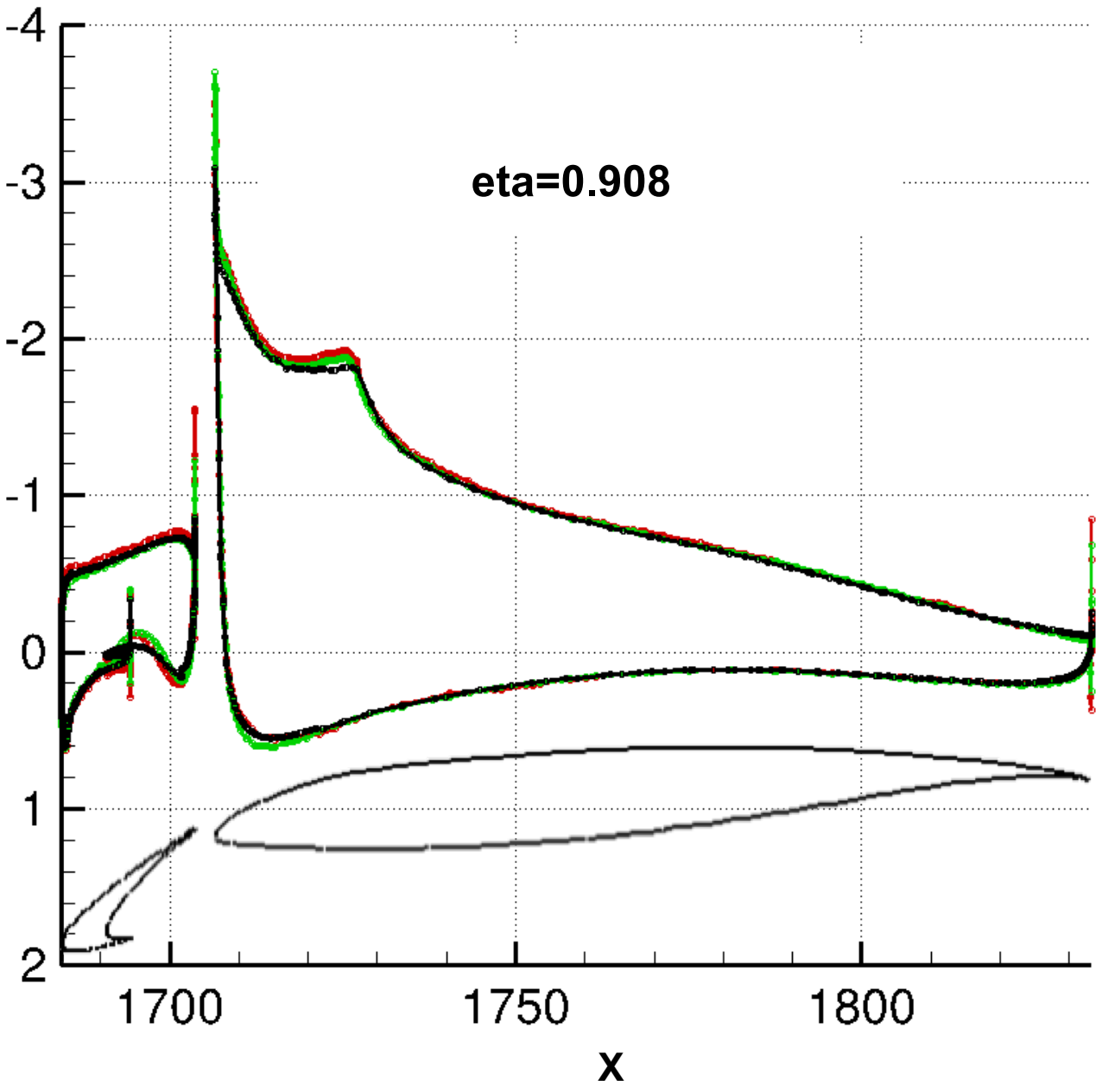
HLCRM
Pressure Coefficient



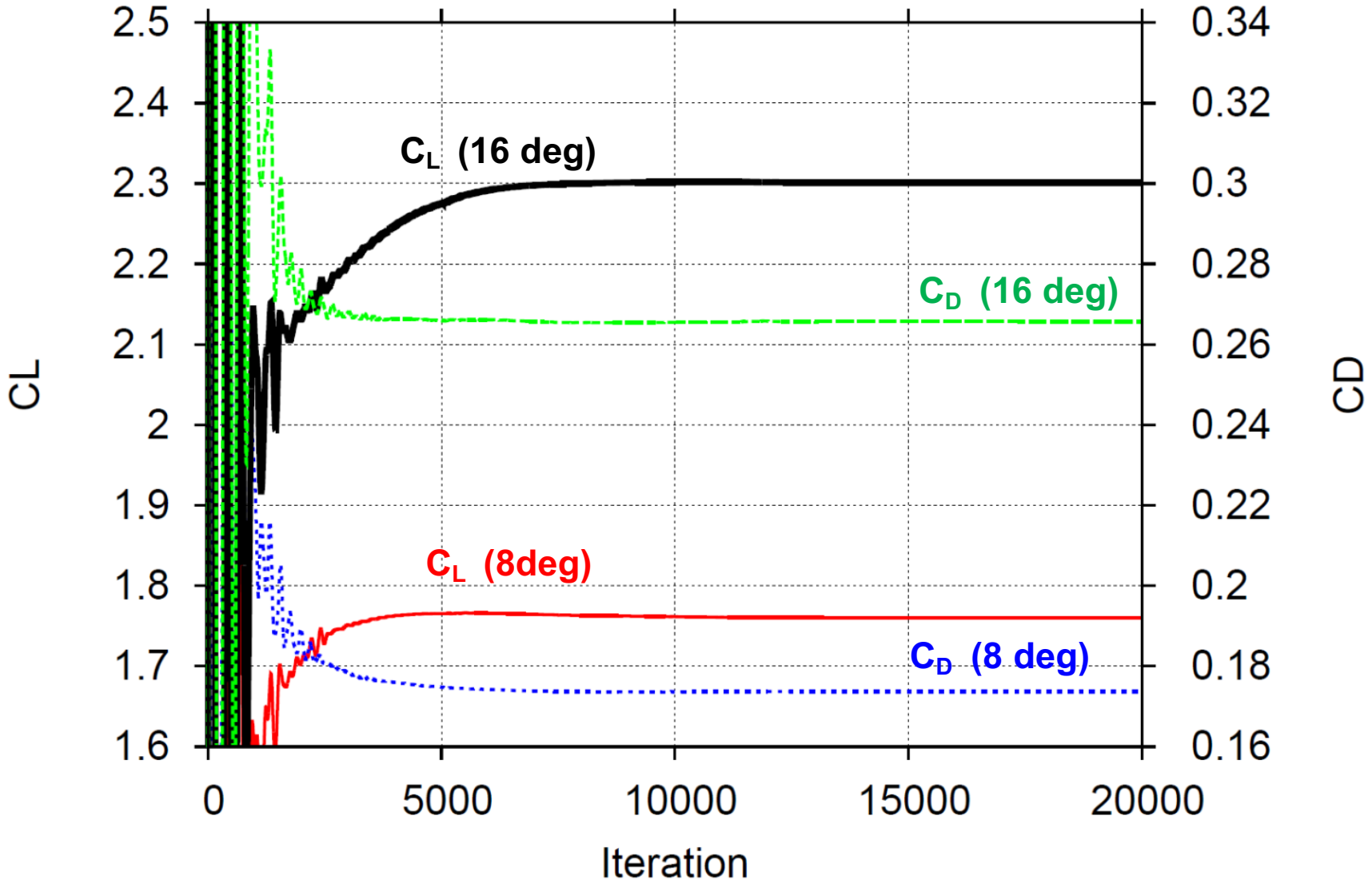
NSU3D / Helios

FUN3D / Helios

mStrand / Helios

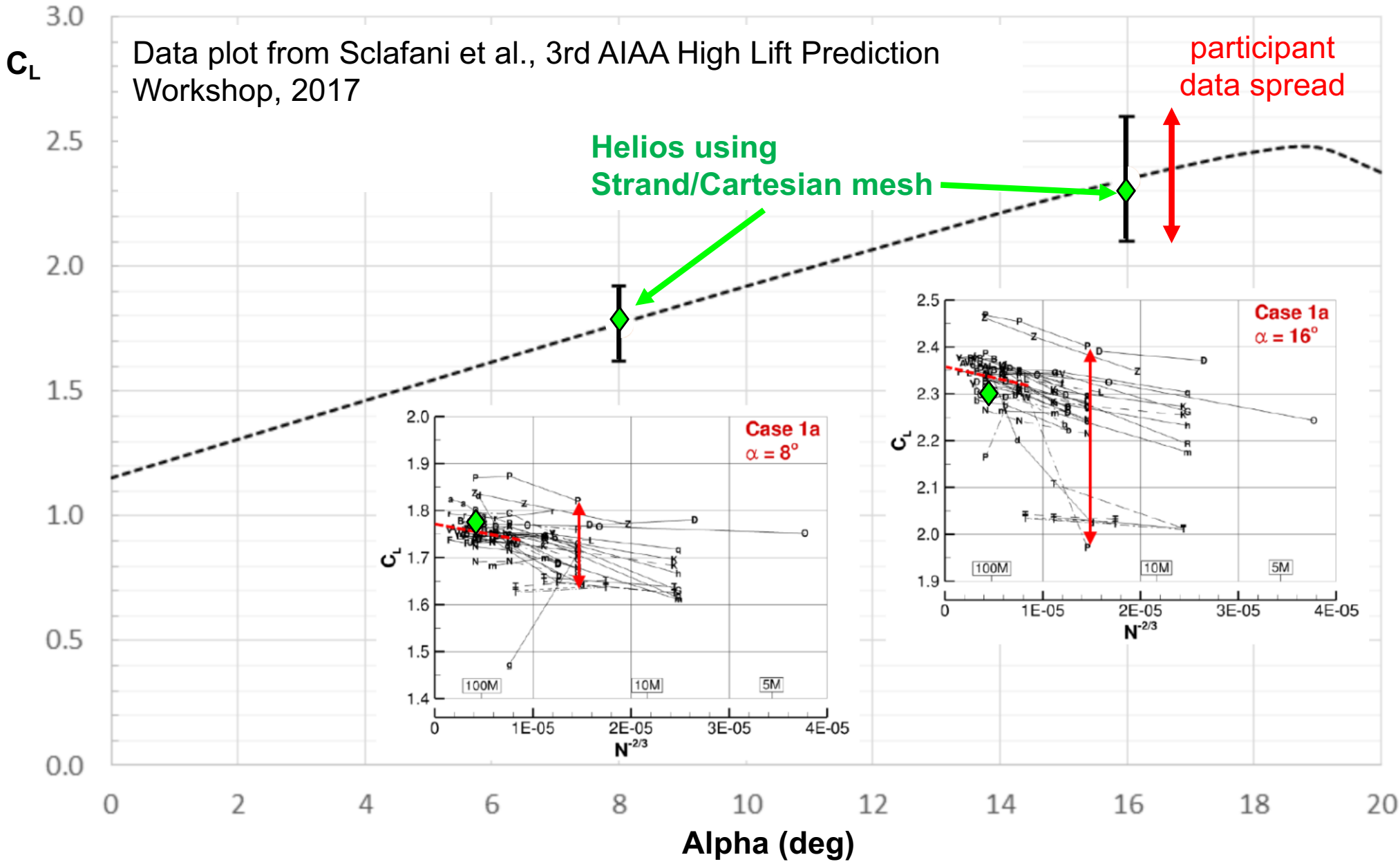


Lift and Drag Convergence (mStrand)



Convergence in 10,000 iterations

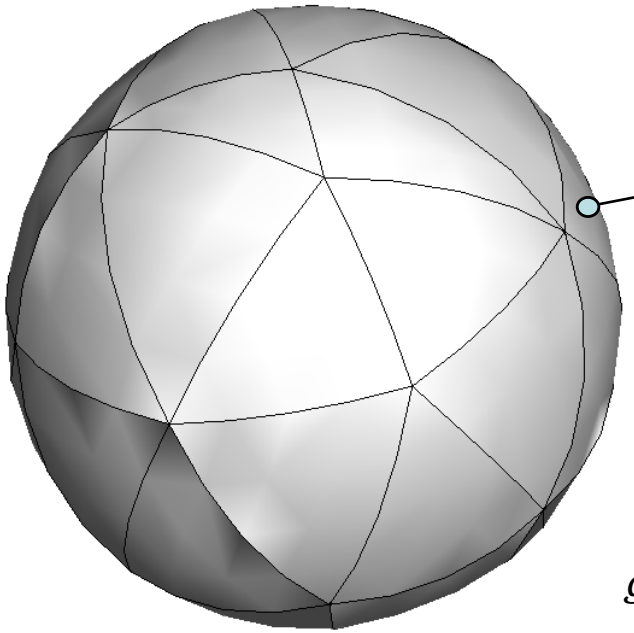
CL Comparison with Other Analysis



Extension to High-order

- **Strand meshing method well suited to high-order mesh generation: can be done directly as part of initial meshing (as opposed to the more common a posteriori approach)**

- **Need a function for exact minimum distance to a surface represented by curved elements:**
  → Bound constrained minimization of natural coordinate location, with a sub-triangle (u,v) as starting point

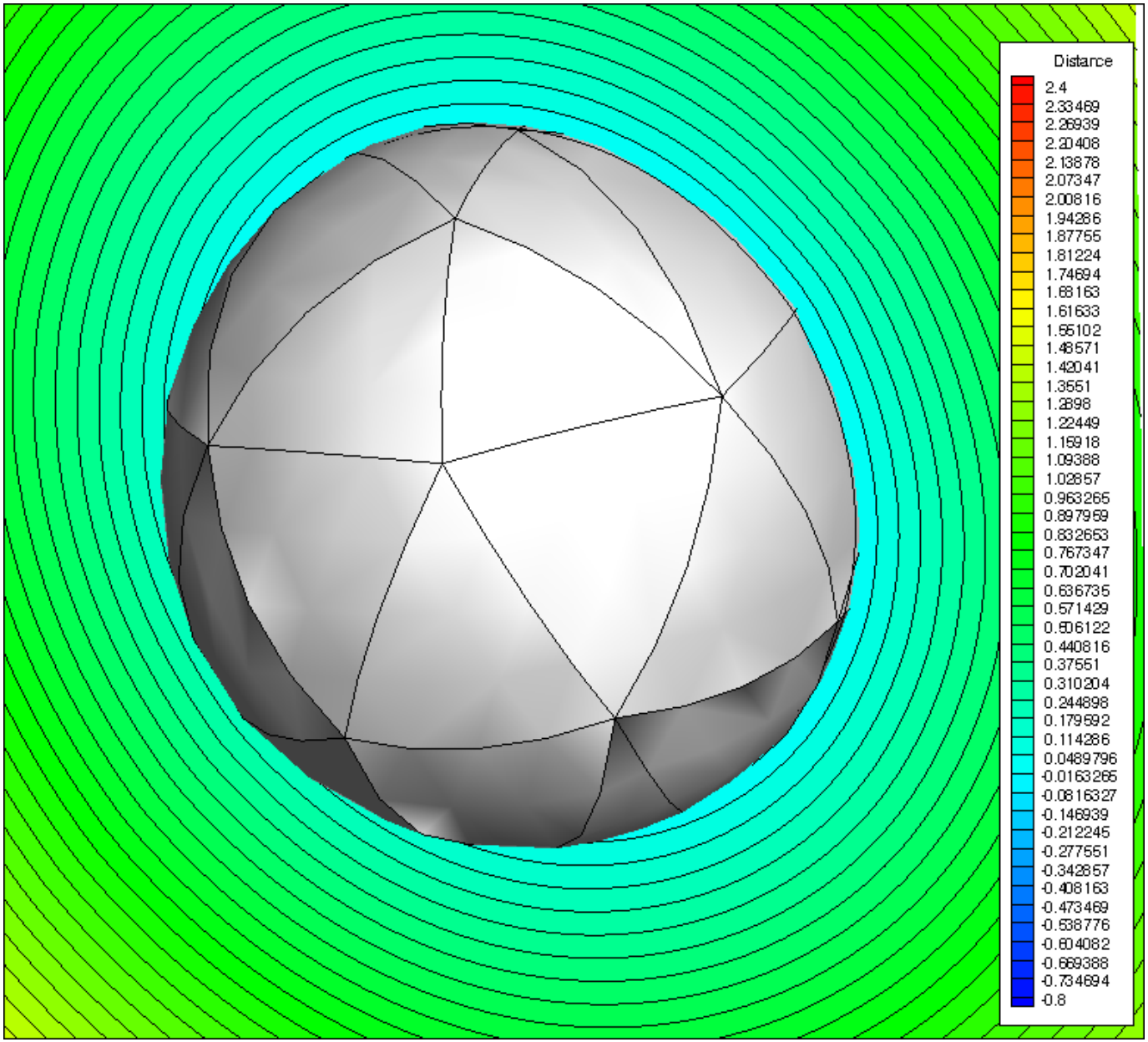


Curved sphere (6-noded triangles)

$g(x)$ = minimum distance between point x and the curved surface tessellation

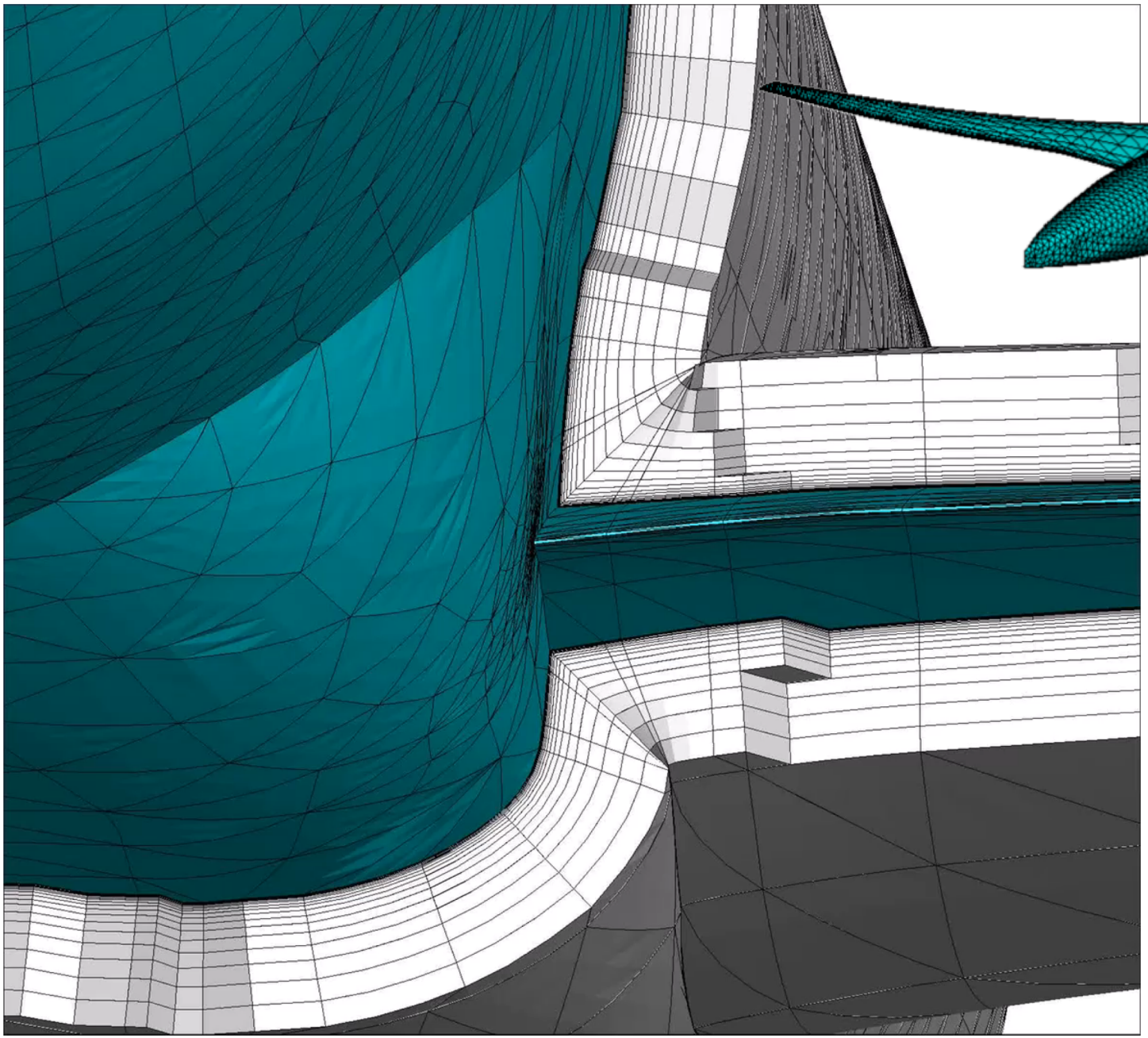
$$\min_{g(x)=L}(|x-x_0|),\ g(x)=\min_{y\in\partial\Omega}(|x-y|)$$

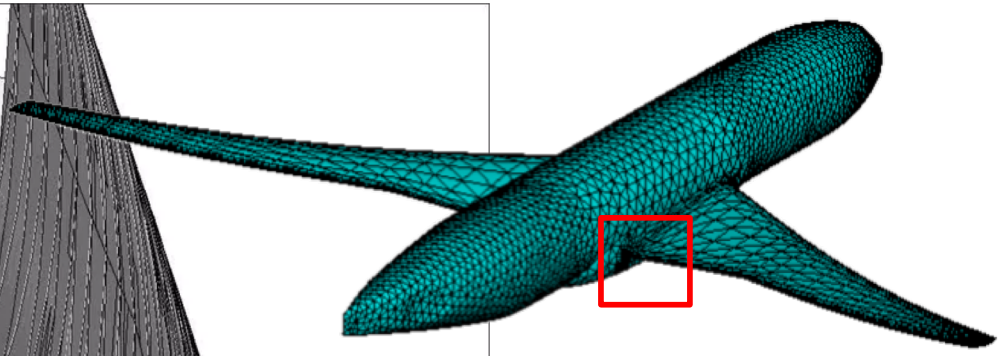
Solution using constrained minimization



Minimum distance contours on a regular grid around a quadratic sphere surface

High-order meshing Animation for NASA CRM

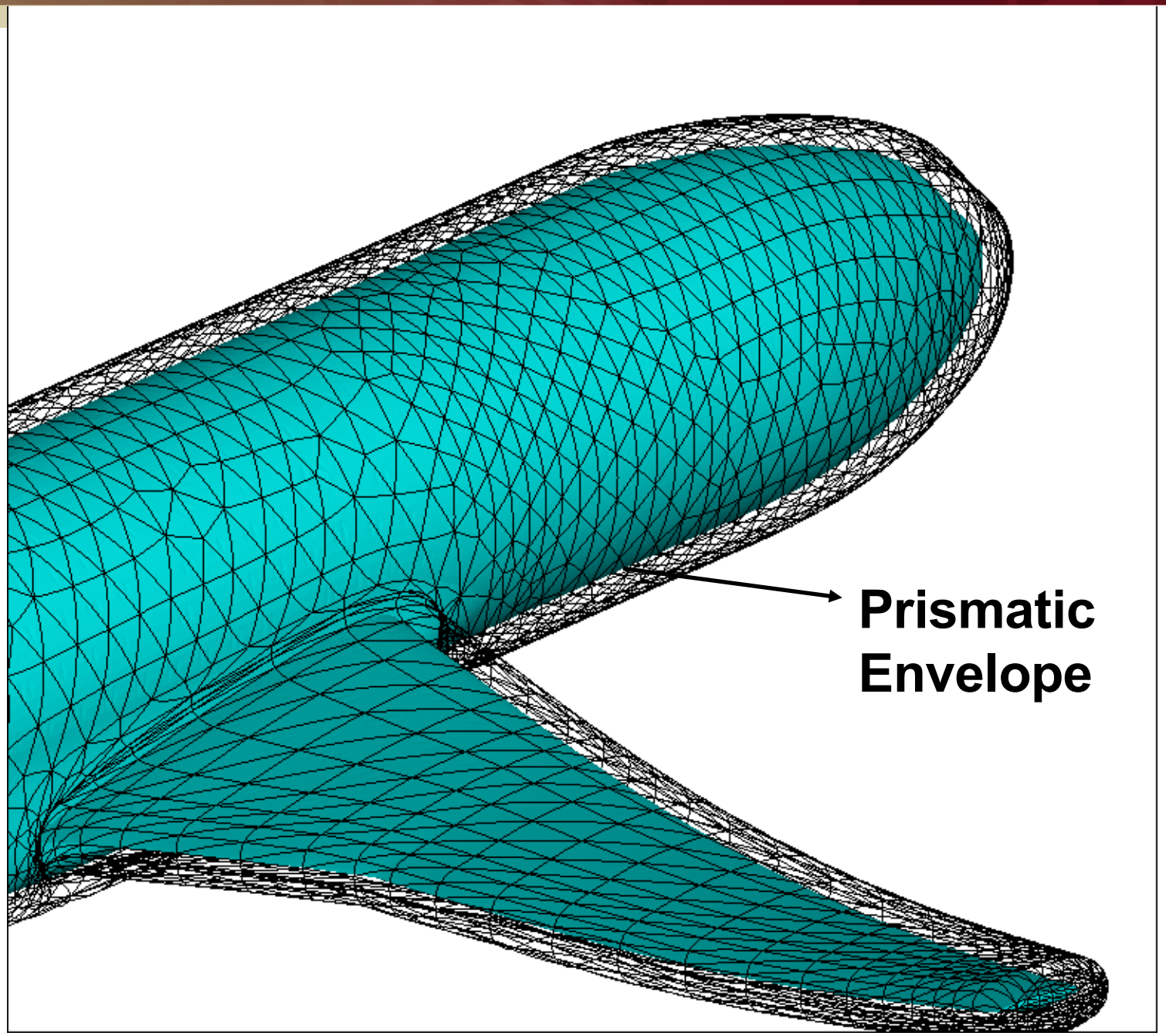




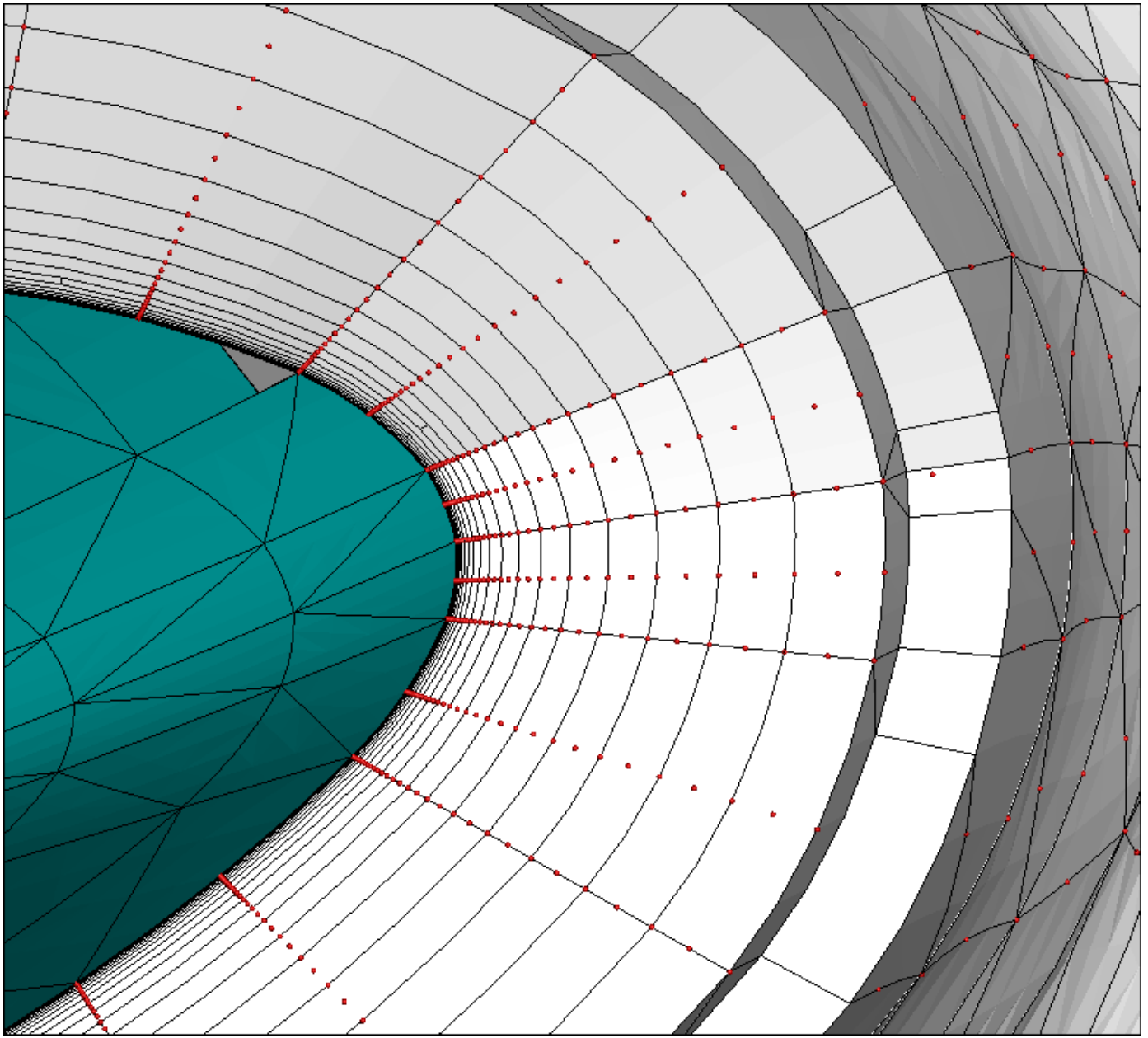
CRM wall surface described as a set of 6-noded triangles in VTK format (High-order workshop)

Prismatic envelope is also a set of 6-noded triangles. Each node is placed and smoothed on the exact min distance iso-surface

NASA Common Research Model



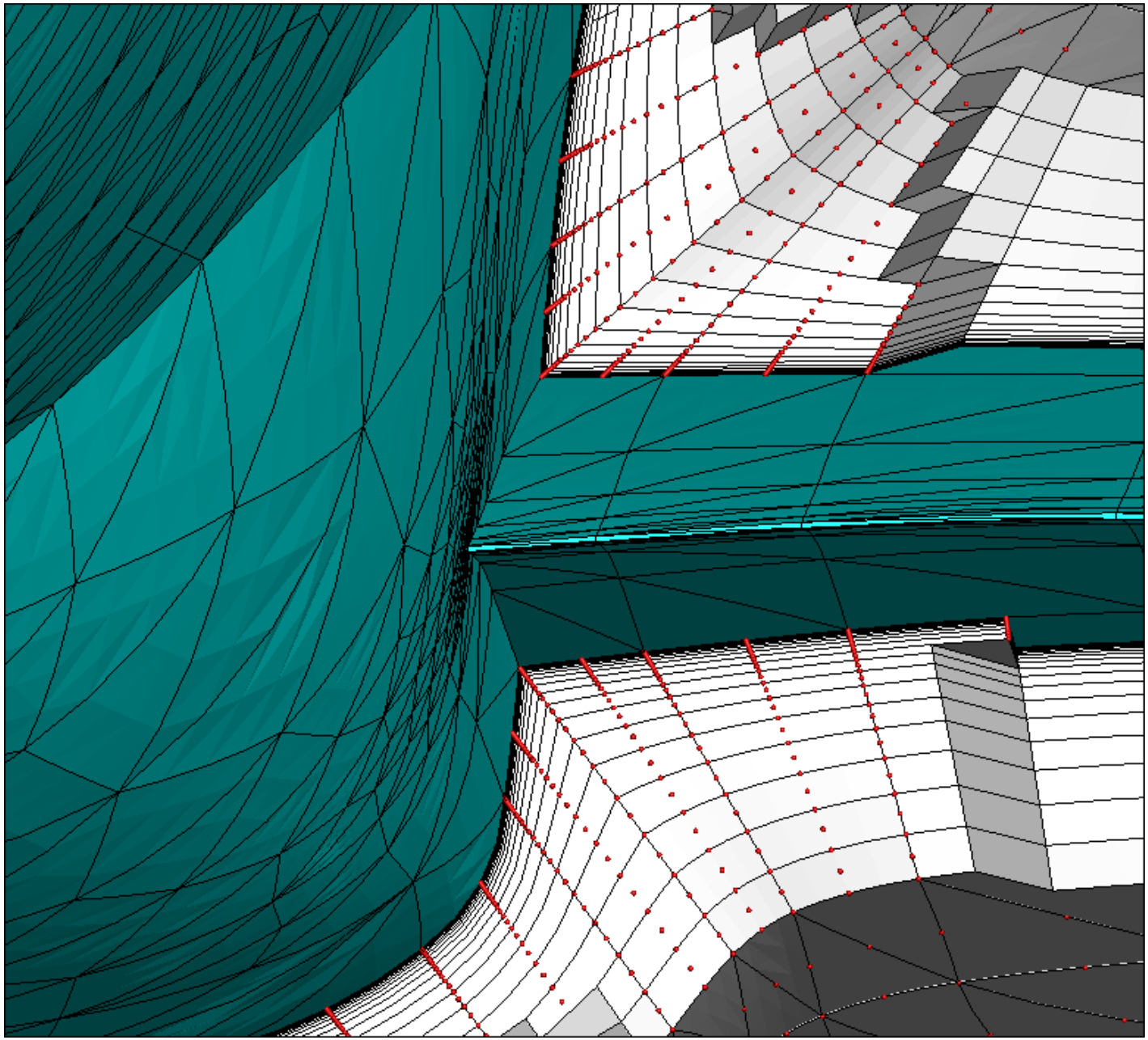
NASA CRM
Wing Leading Edge



Output:

18 node-prisms
CGNS format

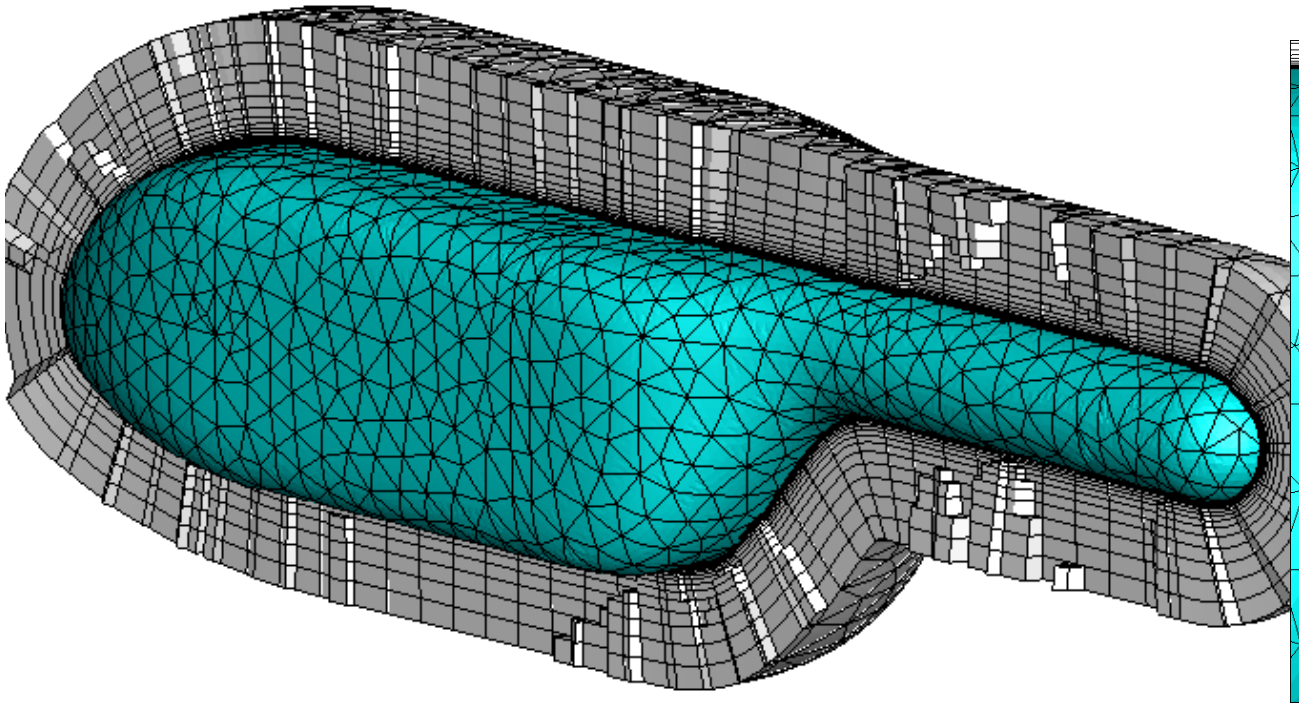
NASA CRM
Wing Fuselage intersection

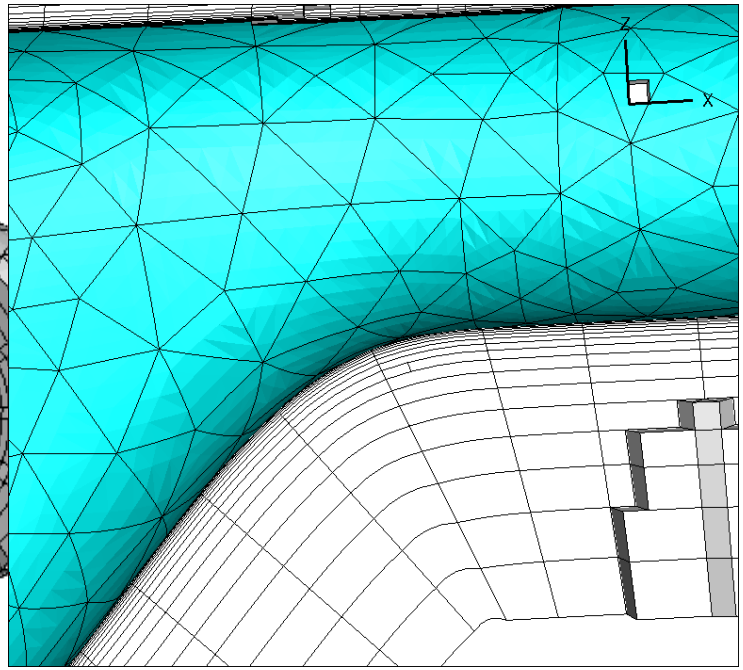


Output:

18 node-prisms

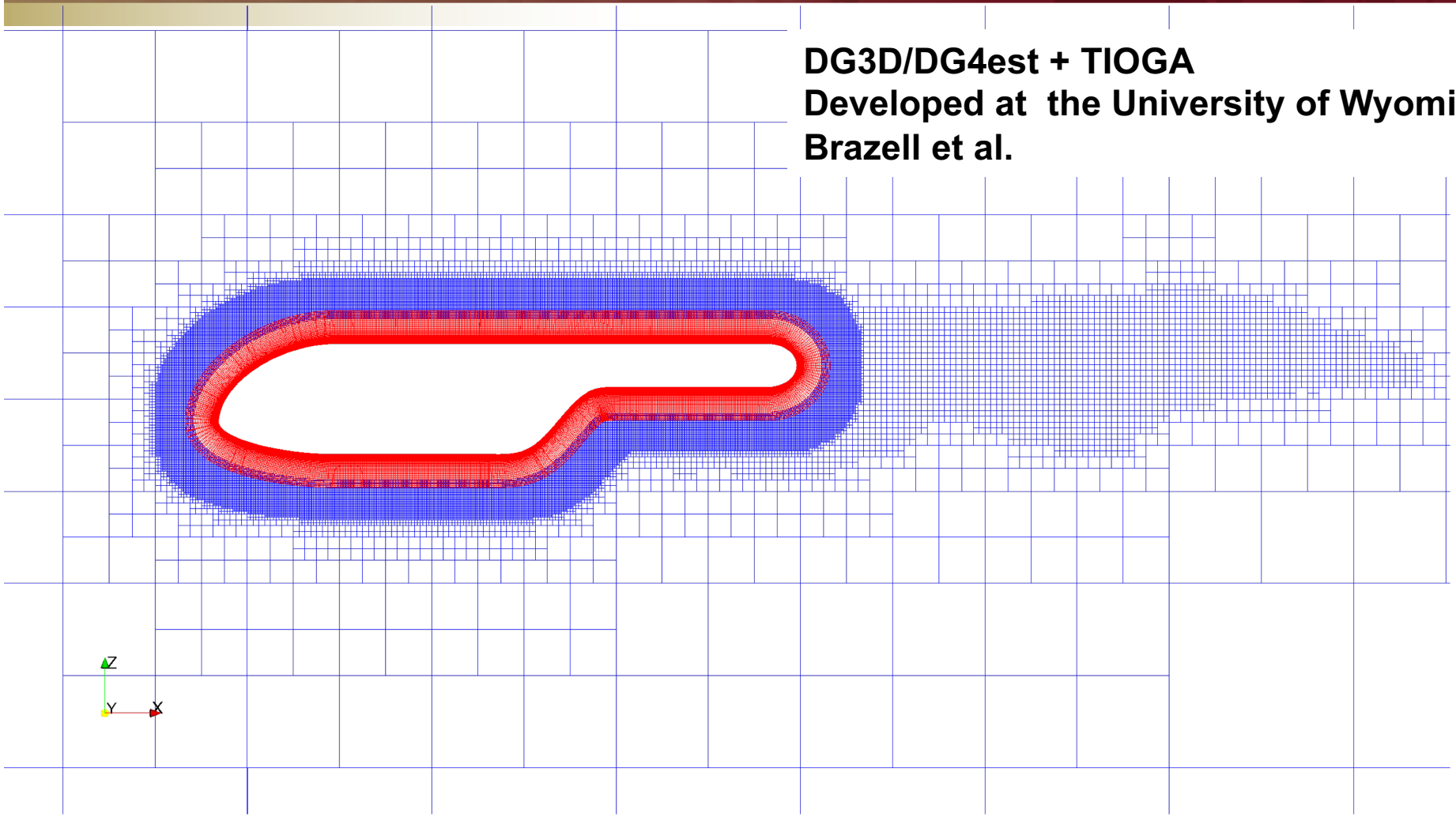
ROBIN high-order mesh (coarse mesh)





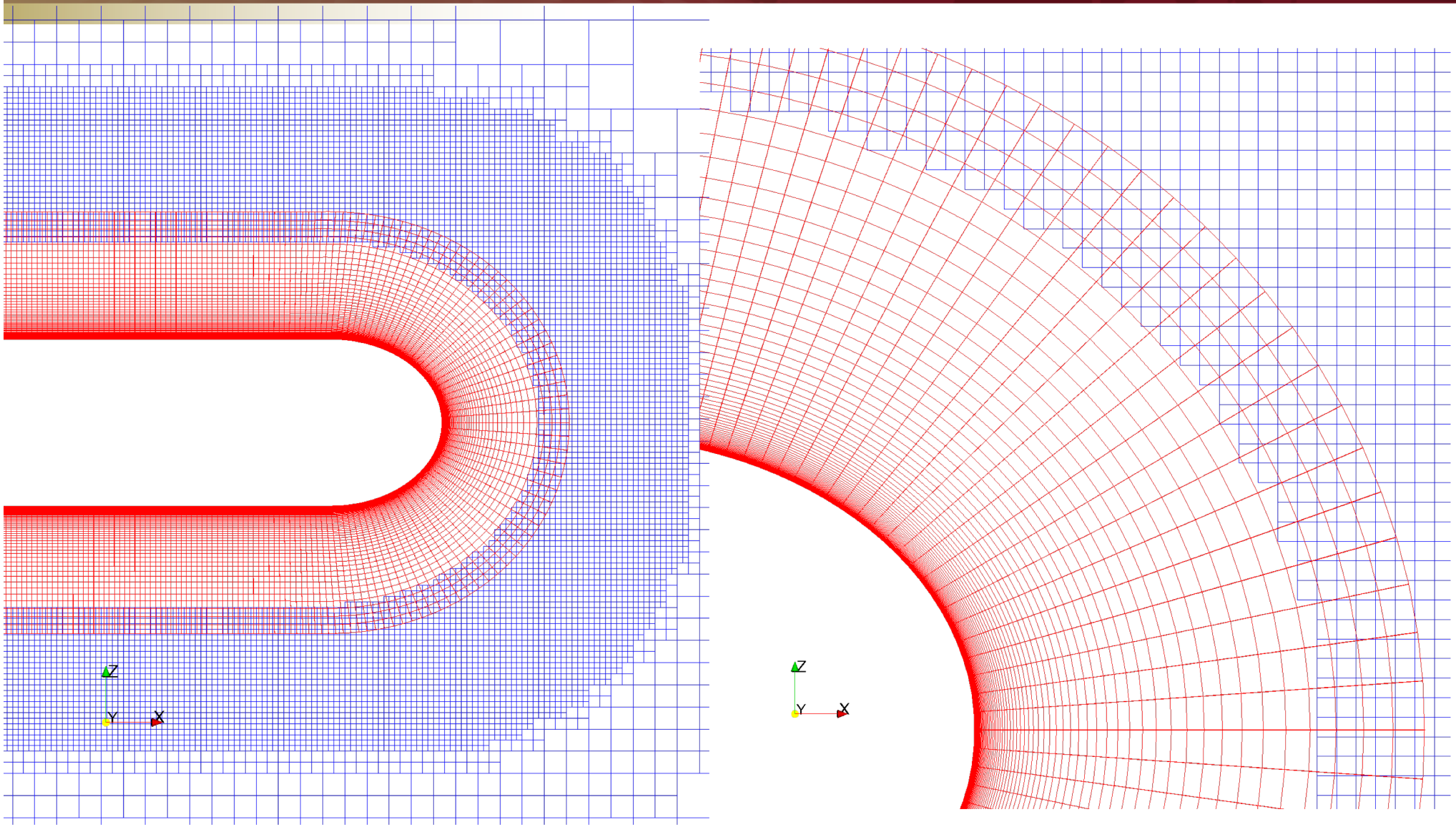
Dual mesh solution methodology

DG3D/DG4est + TIOGA
Developed at the University of Wyoming
Brazell et al.



- p=2 (3rd order) Discontinuous Galerkin in the near-body
- p=2,3,4 (3-5th order) h-p adaptive Discontinuous Galerkin in the off-body
- 225 million DOF/field (15 M near-body + 210 M off-body)
- High-order overset from Brazell, Sitaraman and Mavriplis (JCP 2016)

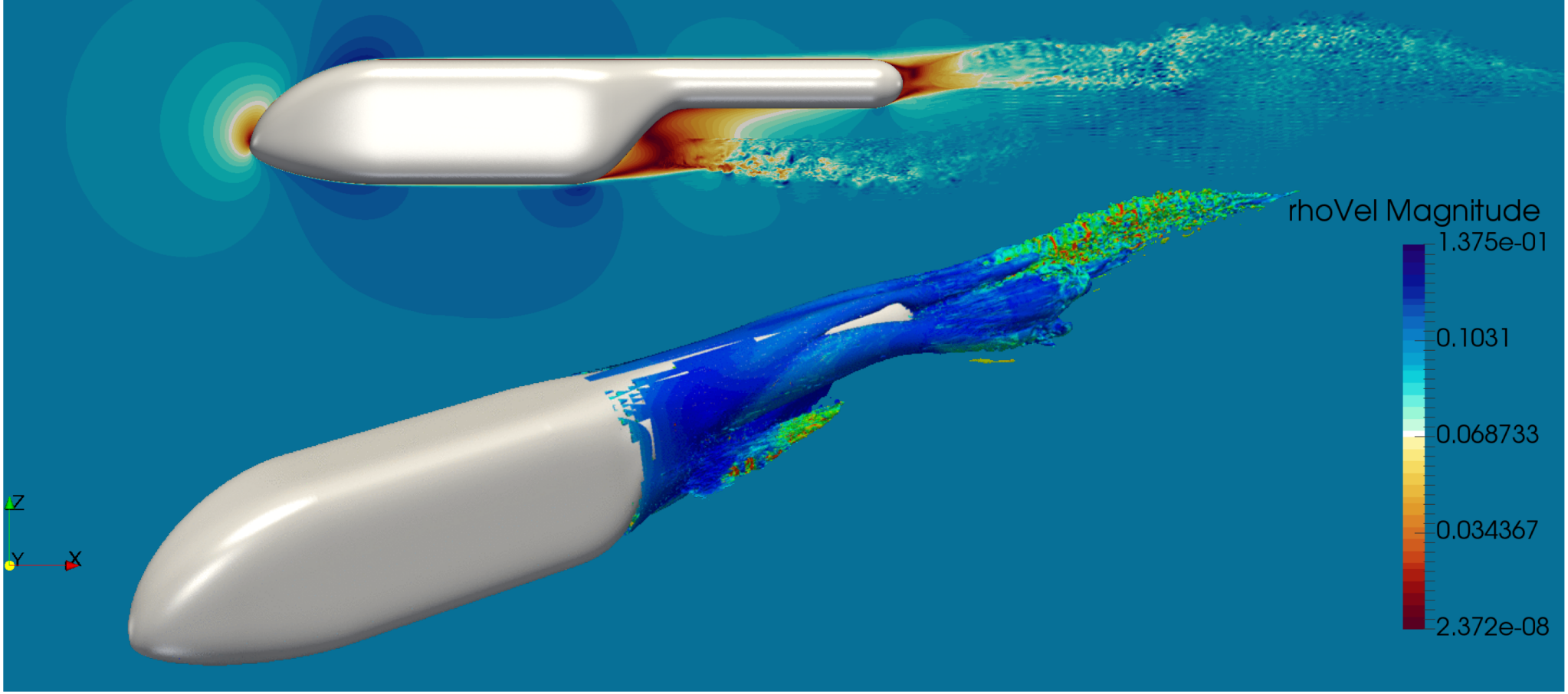
Mesh zoom



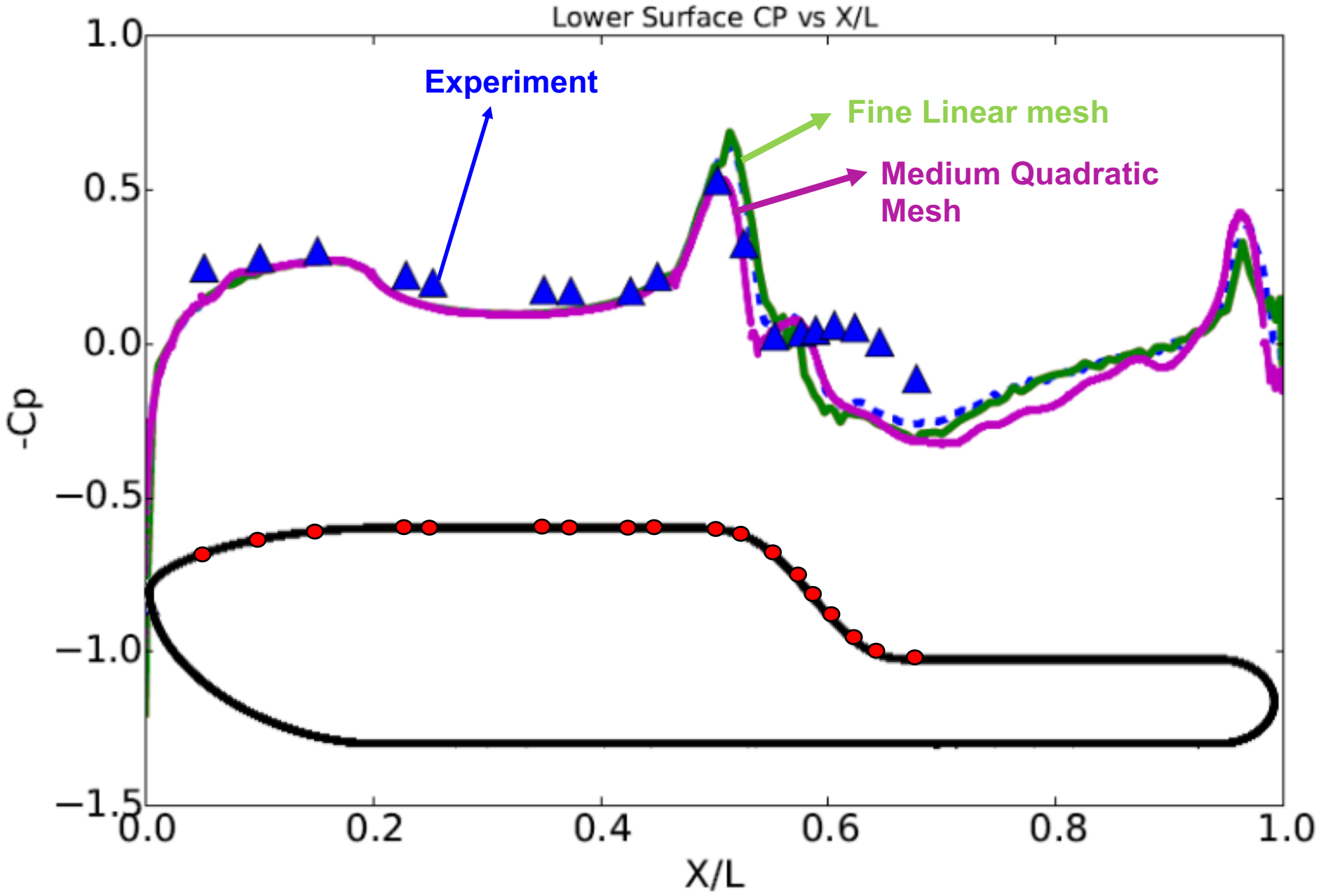
Robin flow solutions

High-order Detached Eddy Simulation for separated flow problem, using curved prismatic grids for near-body

p=2 near the wall
h-p adaptive (p=2,3,4) away from the wall



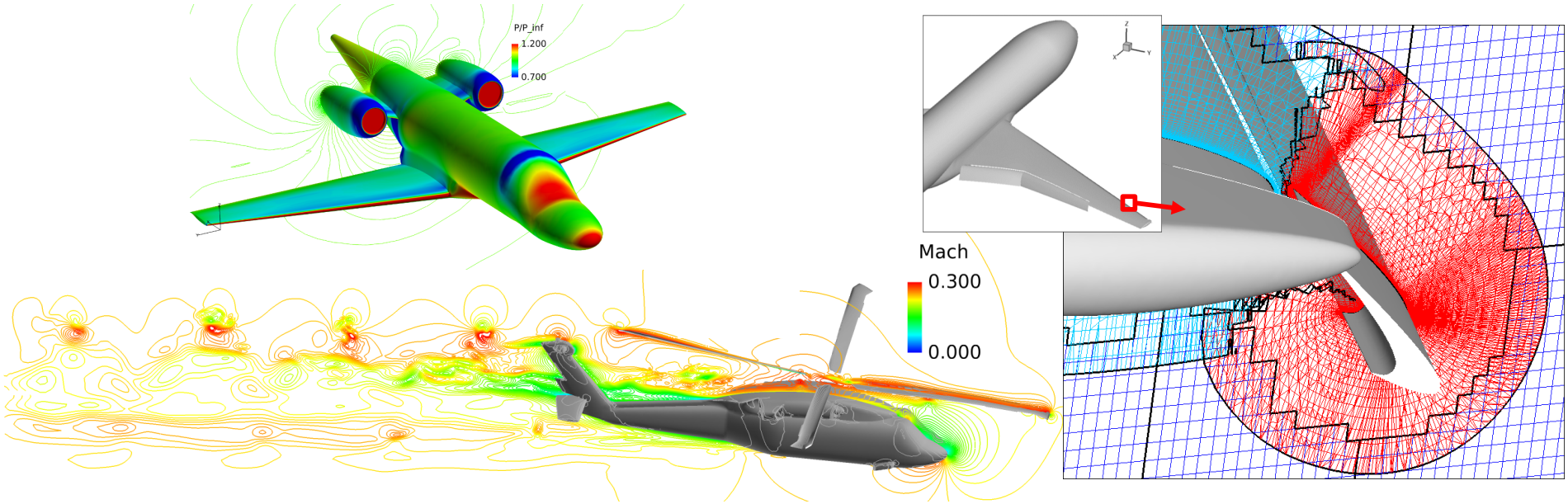
Quantitative comparison



Concluding Remarks

- **Progress towards fully automated Strand meshing for complex geometries**
  - Closest Vertex on Isosurface of Distance I_L
  - Elastic Smoothing constrained on I_L
  - Strand collision problem solved via intra-mesh domain connectivity

- **Validated solution accuracy by comparing with flight test data and fully unstructured computations**



Future Work

- Apply Strand method to generate higher order meshes with curved edges
- Explore using cell quality gradient as alternate smoothing method
- Explore merging of elements to push further extent of fully closed prismatic mesh
- Further improve automation of decision logic (number of levels / level thickness)
- Surface adaption



Acknowledgements

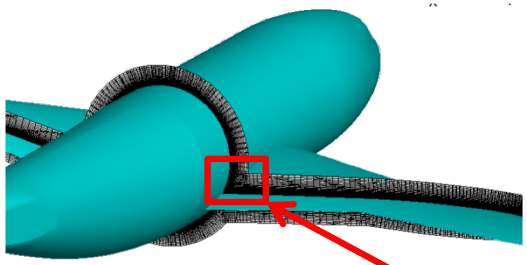
Support from the HPCMP CREATE A/V program and Capstone team

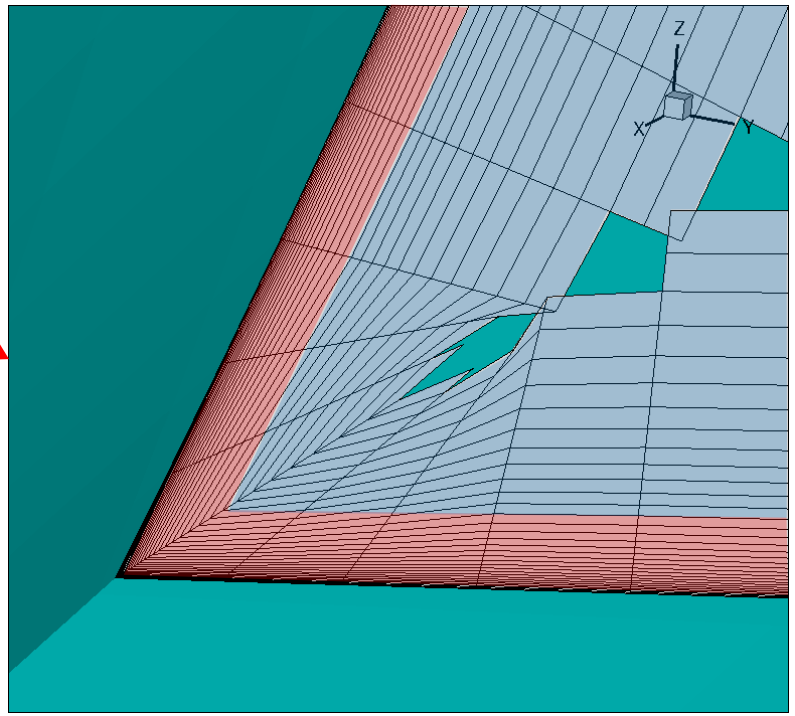
Meshing Method Overview

In order to allow meshing of complex geometries, Strand Meshes are built in 2 steps:

1. **Inner layer, can be very close to surface (covering boundary layer): must be fully closed (no self-intersection)**

2. **Outer layer, extends larger distance to allow interfacing with Cartesian grid: can be self-intersecting**



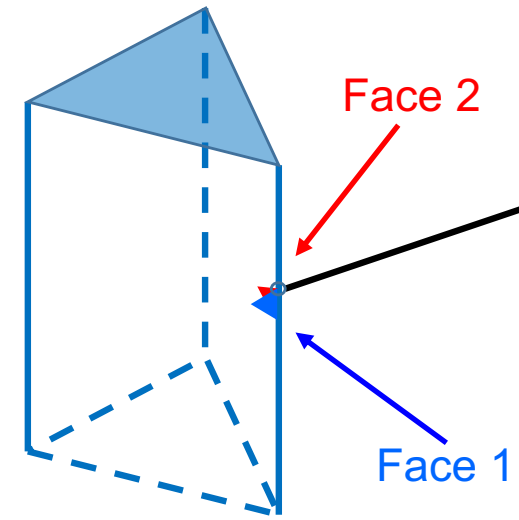


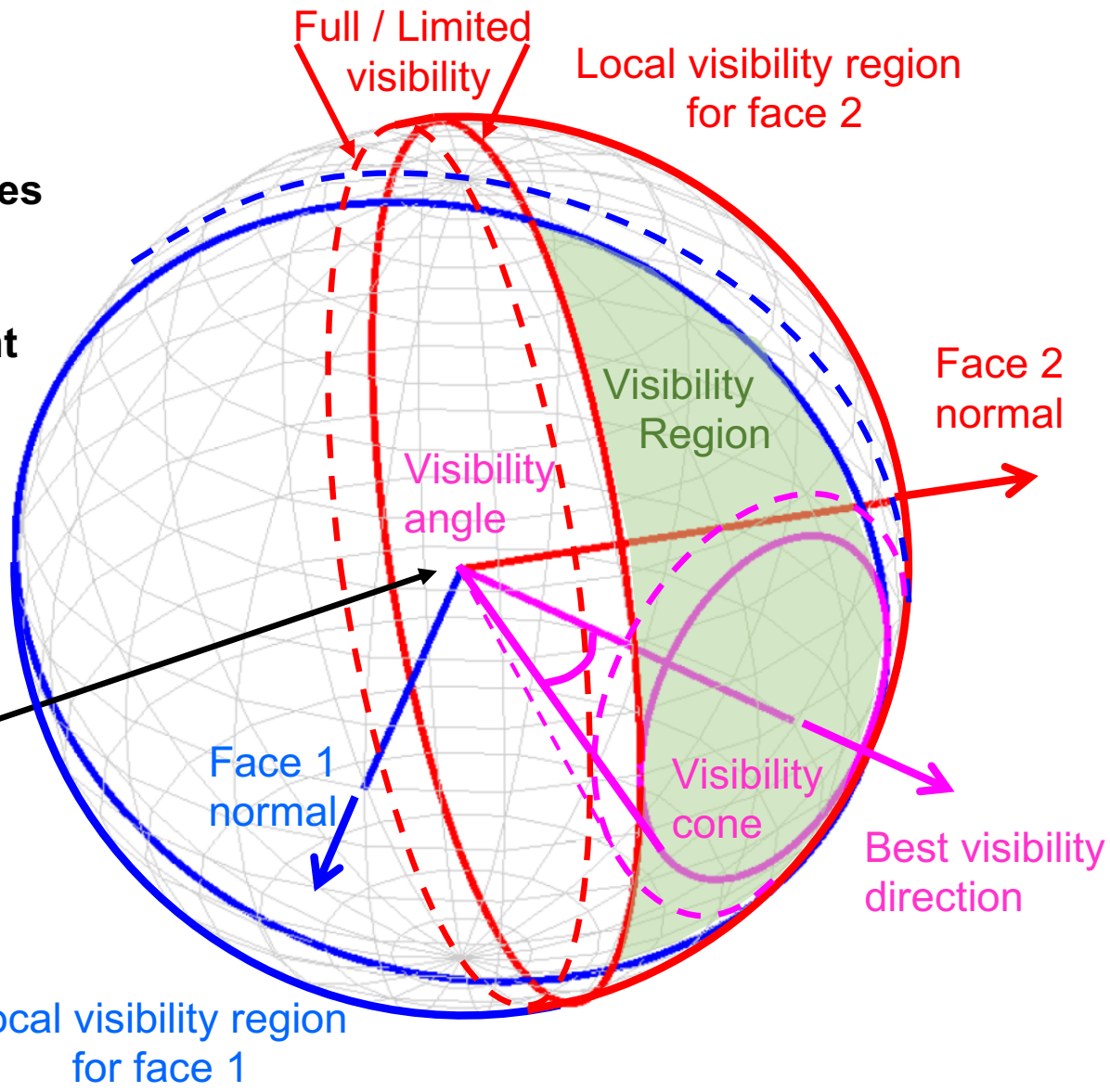
Outer layer:
- Extrusion from 1st layer
- intra-mesh domain connectivity to clear self-intersections

Inner layer:
- fully closed Strand mesh
- may consist of a few levels (< 10)

Visibility Region

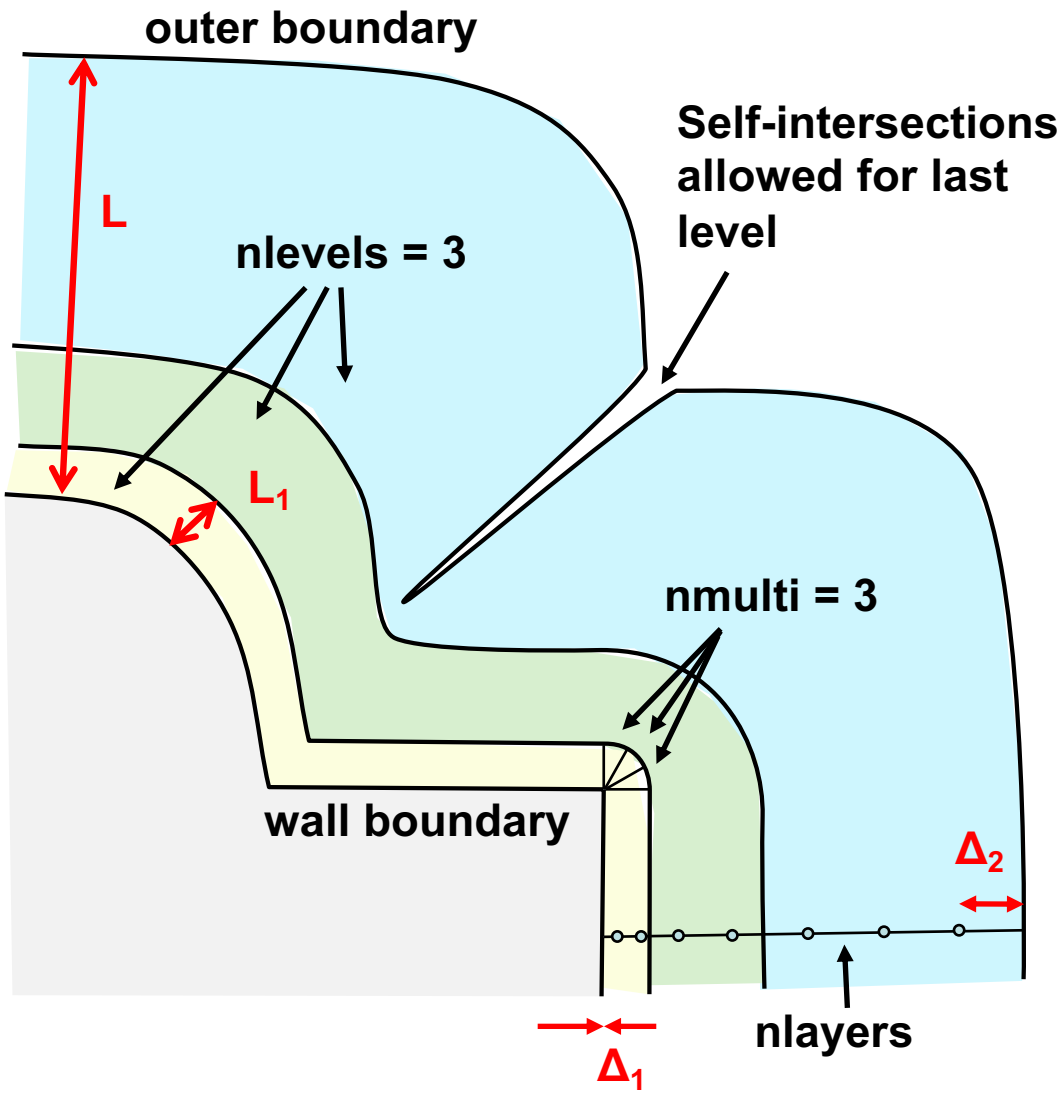
- **Region with no strand/surface intersection**
- **Intersection of local visibility cones for each neighbor face**
- **Limited visibility region constraint enforced during smoothing**





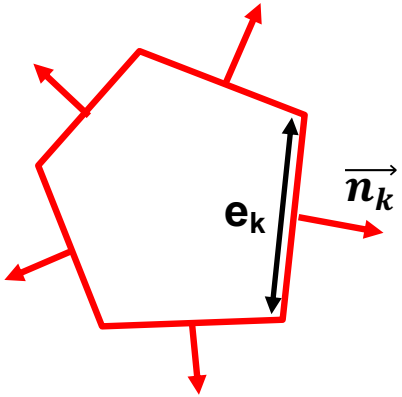
Required Input

S	Surface tesselation
L	desired extent from boundary surface to the prismatic envelope
Δ_1	spacing required at the wall
Δ_2	spacing at the prismatic envelope
nlayers	number of layers in the distribution function
nmulti	maximum number of multi-strands from a node
nlevels	maximum number of strand levels
L_1	thickness of the initial level
s	stretch ratio for the thickness of each level

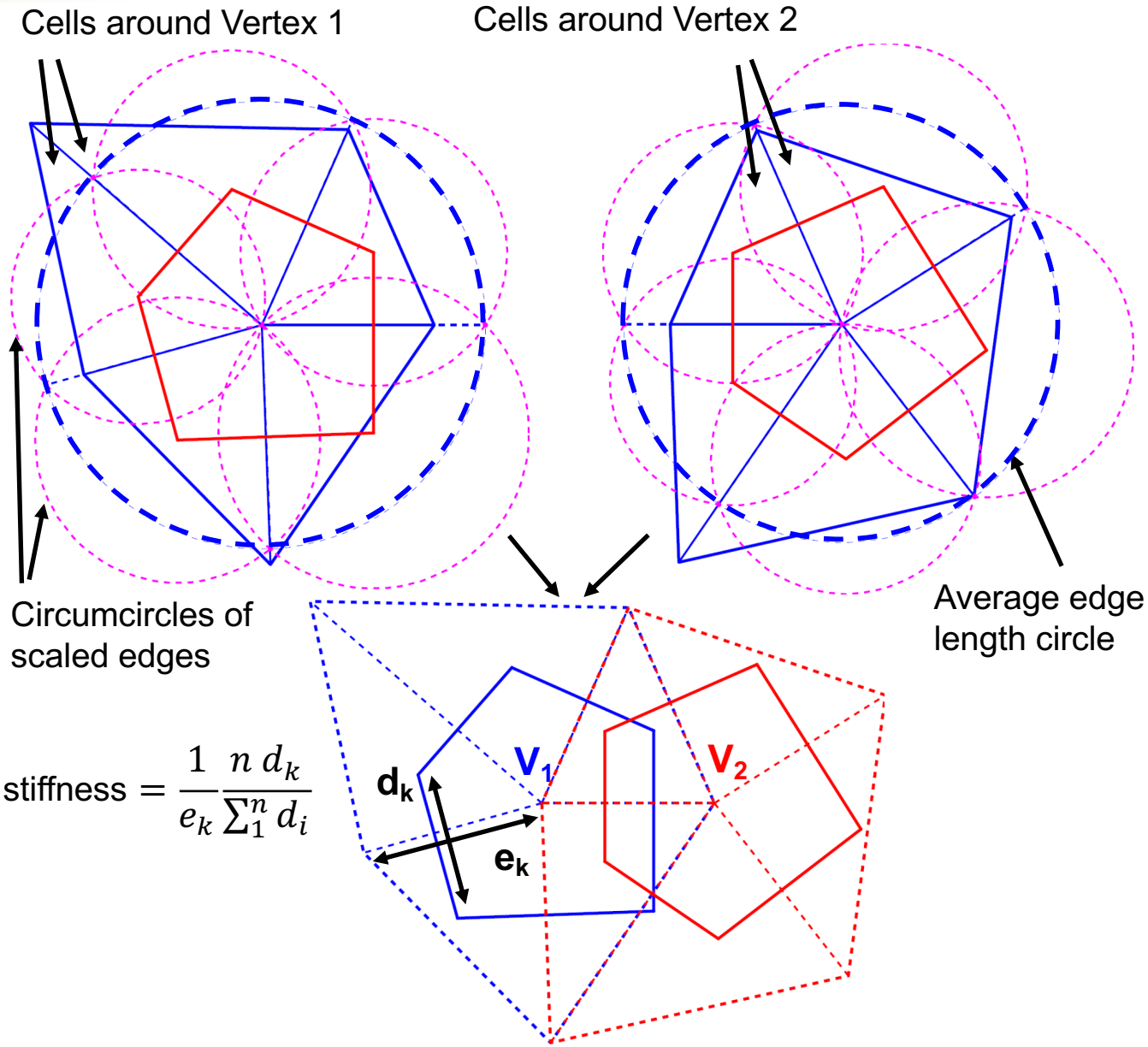


Spring Stiffness

$$\sum e_k \overrightarrow{n_k} = \vec{0}$$



- **Springs are in equilibrium for very short initial strands**
- **Dual-value stiffness: one for each edge extremity**
- **For equilateral triangles, stiffness = 1/(edge length)**

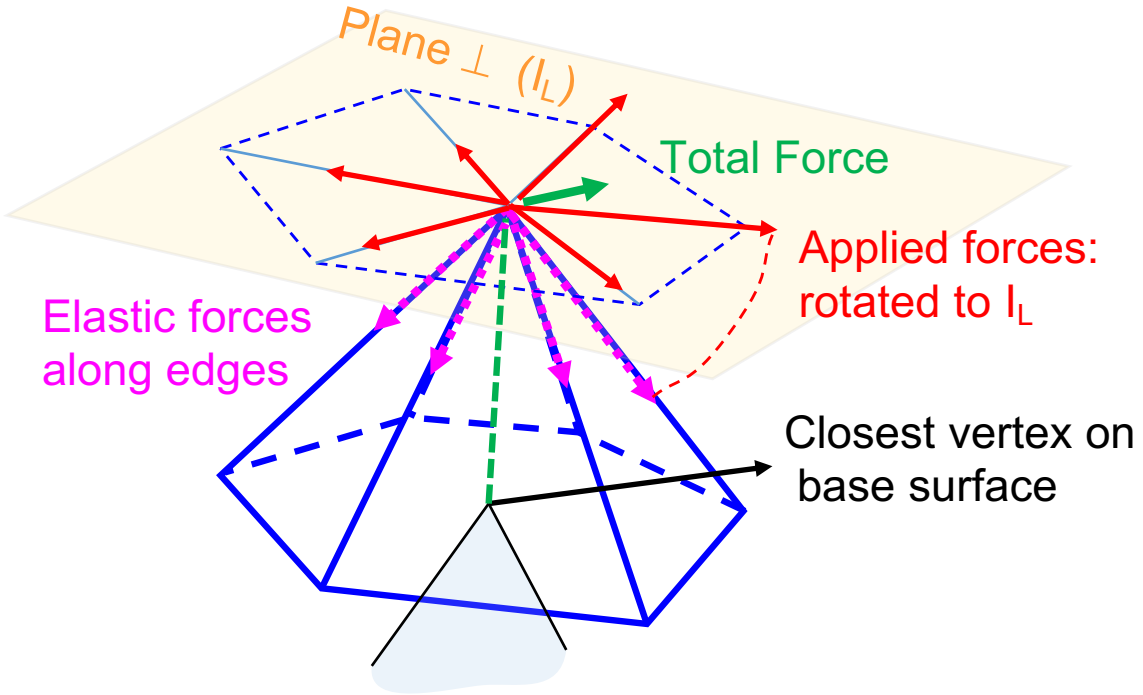


$$\text{stiffness} = \frac{1}{e_k} \frac{n\, d_k}{\sum_1^n d_i}$$

- **Stiffness varies linearly to be uniform at prismatic envelope**

Elastic Force Computation

- **Vertex in Convex Region**:
  - Forces constrained to isosurface of distance



- **Vertex in Concave Region**:
  - No force constraint, elastic forces applied along actual edge
  - Vertex tends to move above isosurface of distance, reducing concavity for next level

