Automatic Conversion of Conceptual Geometry to CFD Geometry for Aircraft Design

Wu Li
NASA Langley Research Center, USA
email: w.li@nasa.gov

March 25, 2007

Conceptual aircraft design is usually based on simple analysis codes. Its objective is to provide an overall system performance of the developed concept, while preliminary aircraft design uses high-fidelity analysis tools such as computational fluid dynamics (CFD) analysis codes or finite element structural analysis codes. In some applications, such as low-boom supersonic concept development, it is important to be able to explore a variety of drastically different configurations while using CFD analysis to check whether a given configuration can be tailored to have a low-boom ground signature. It poses an extremely challenging problem of integrating CFD analysis in conceptual design. This presentation will discuss a computer code, called iPatch, for automatic conversion of conceptual geometry to CFD geometry.

In the article "Translation Time" (Mechanical Engineering, October 2006), Jean Thilmany wrote that "*... analysts at the [Sandia] lab spend around 75 percent of their time cleaning up geometry after it's been imported from computer-aided design to analysis software.*" The article provides a telling story about one of major bottlenecks in simulation-based design processes. In general, conceptual aircraft geometry is not as well-defined as a CAD geometry model. In particular, a conceptual aircraft geometry model usually does not define the intersection curves for the connecting surfaces. The computer code iPatch eliminates the gap between conceptual geometry and CFD geometry by accomplishing the following three tasks automatically: (1) use bicubic B-splines to extrapolate (if necessary) each surface in a conceptual geometry so that all the independently defined geometry components (such as wing and fuselage) can be intersected to form a watertight CFD geometry, (2) compute the intersection curves of surface patches at any resolution (up to 10^{-7} accuracy) specified by users, and (3) write the B-spline surface patches and the corresponding boundary points for the watertight CFD geometry in the format that can be directly exported to the meshing tool VGRID in the CFD software TetrUSS. As a result, conceptual designers can get quick feedback on the aerodynamic characteristics of their concepts, which will allow them to understand some subtlety in their concepts and to be able to assess their concepts with a higher degree of confidence. This integration of CFD analysis in conceptual aircraft design will greatly eliminate some uncertainty due to simple analysis codes used to develop the concepts and improve the feasibility/credibility of the final concept.

The presentation will highlight the mathematical challenges of accomplishing the aforementioned three tasks and the computational algorithms used by iPatch.

Automatic Conversion of Conceptual Geometry to CFD Geometry for Aircraft Design

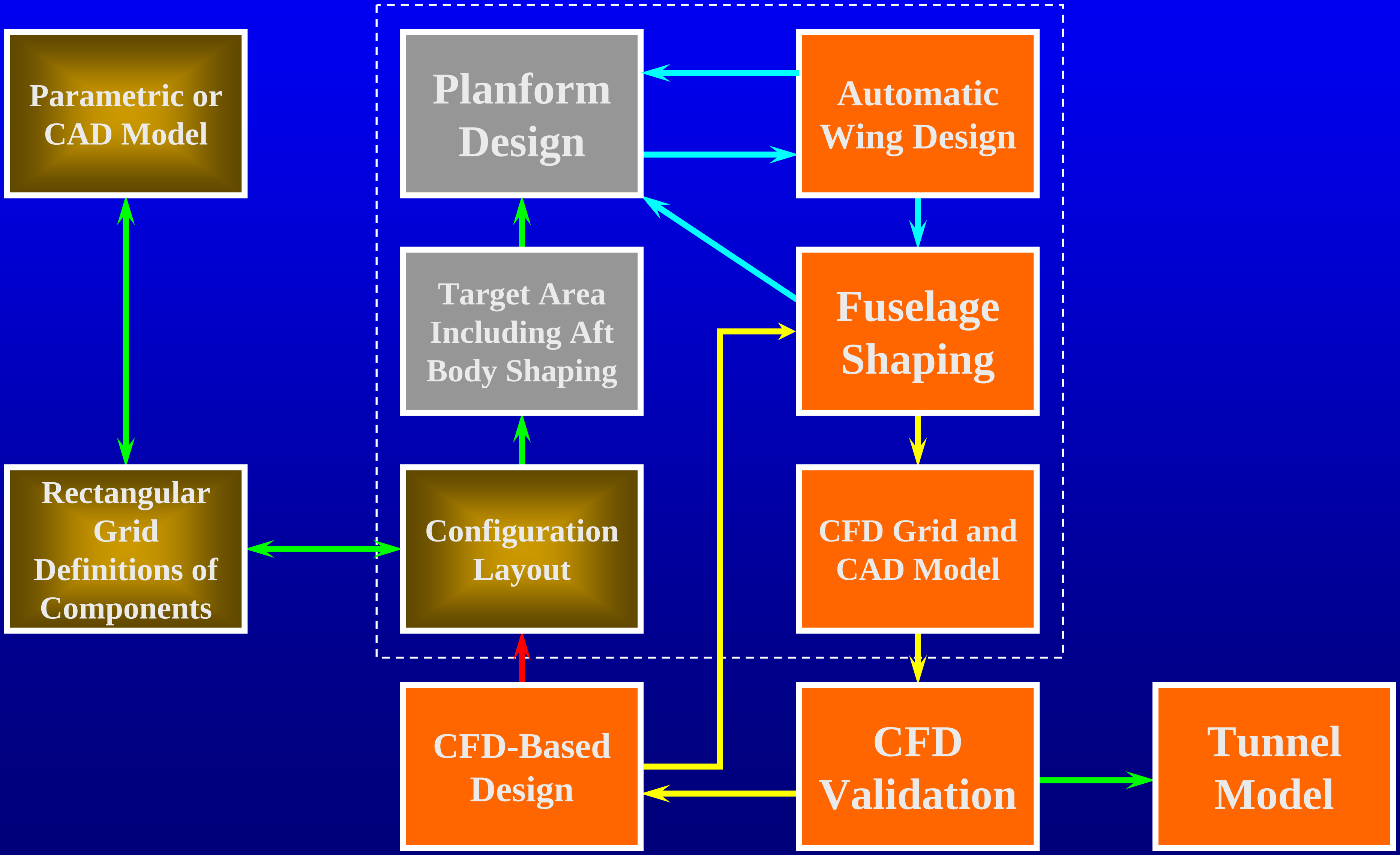
Dr. Wu Li
Aeronautics Systems Analysis Branch
NASA Langley Research Center

SIAM Conference on Mathematics for Industry: Challenges and Frontiers
Philadelphia, October 9–11, 2007

Background

- **NASA Aeronautics Research Mission Directorate Goals**
  - NASA Goal 3E: Advance knowledge in the fundamental disciplines of aeronautics, and develop technologies for safer aircraft and higher capacity airspace systems.
  - NASA Sub-Goal 3E.3. By 2016, develop multidisciplinary design, analysis, and optimization capabilities for use in trade studies of new technologies, enabling better quantification of vehicle performance in all flight regimes and within a variety of transportation system architectures.
- **One of Supersonic Project’s Research Goals**
  - Milestone 3.03.03. Integrated aft end vehicle shaping methodology validated analytically [using CFD etc.], 4Q 2008.

Overview of Integrated Low-Boom and Low-Drag Design Process



Parametric or CAD Model

Rectangular Grid Definitions of Components

Planform Design

Automatic Wing Design

Target Area Including Aft Body Shaping

Fuselage Shaping

Configuration Layout

CFD Grid and CAD Model

CFD-Based Design

CFD Validation

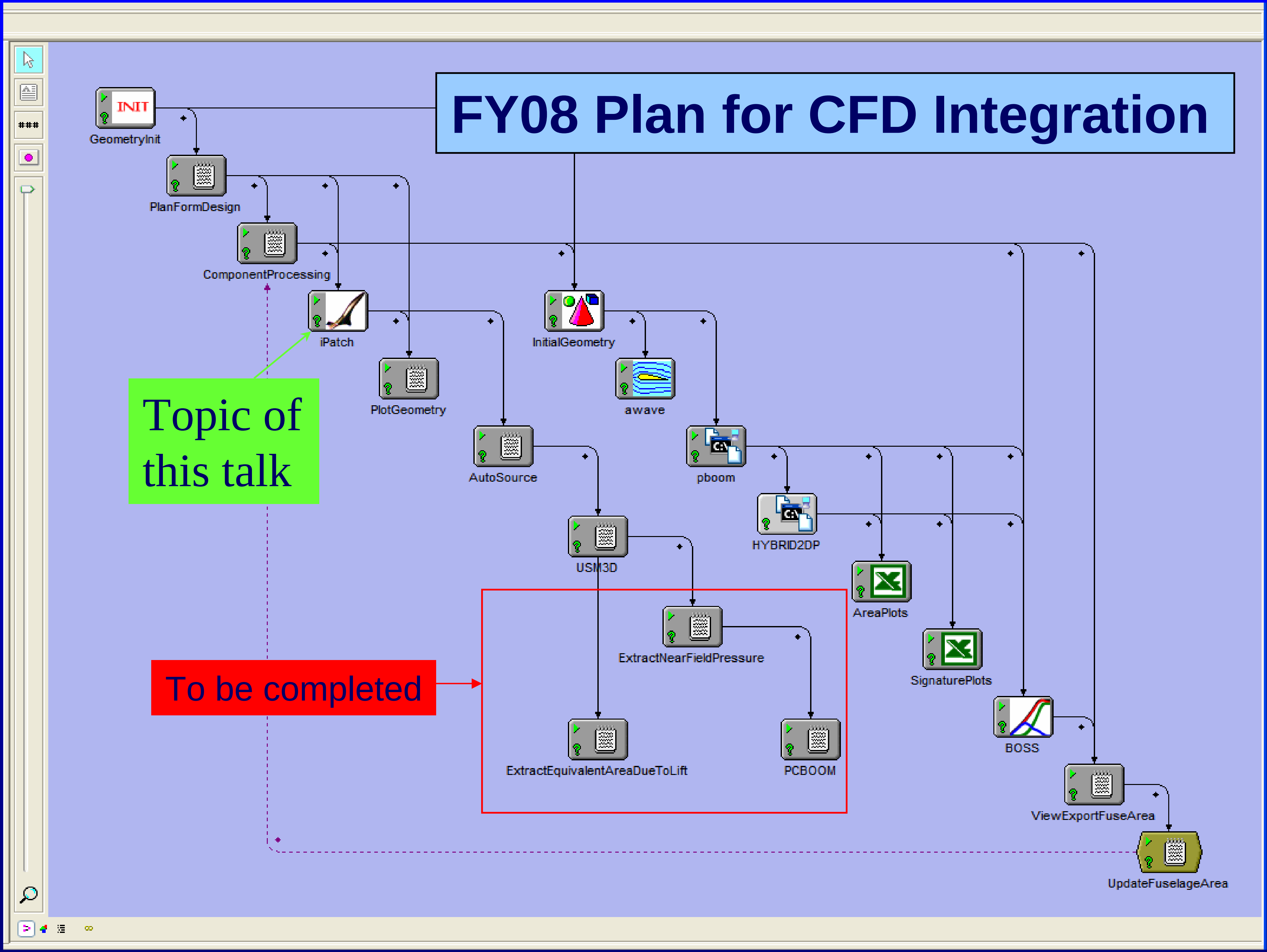
Tunnel Model

Enabling Tools

1. **BOSS** (Boom Optimization using Smoothest Shape Modifications): Modify the fuselage for low-boom concepts in 1-2 hours per plan form
2. **ePatch** (Elastic Patch for fuselage and wing blending based on tension-free PDE surface models): Automatically generate a smooth wing and fuselage blending patch based on 10 control parameters
3. **SintS** (Surface intelligently intersecting Surface): Allow exact intersection (within 1.0e-6 error tolerance) of two (extrapolated) surfaces
4. **iPatch** (Automatic conversion from component geometry to CFD geometry): ePatch + SintS + Patch2CFD (to write the geometry data in the format required by VGRID)
5. **AutoSource**: Automatic source generation of a watertight geometry defined by *B*-spline surface patches and boundary curves in VGRID format
6. **CDISC** (An inverse design tool for cruise efficiency)

Tools in green color will be covered in this presentation.

FY08 Plan for CFD Integration



Topic of this talk

To be completed

Geometry Conversion Issue

- "Closing the Gap Between CAD Model and Downstream Application" (SIAM News, June 23, 1999, Rida Farouki): "… Ken Morgan of the University of Wales, …, presented the following typical breakdown of the effort in a realistic CFD analysis: 1-4 weeks for geometry repair and preparation, 10-20 minutes for surface meshing, 3-4 hours for volume, and about 1 hour for the actual flow analysis."

- "Healing the Wounds of Data Conversion" (CAD User AEC Magazine, March 2000): "… Peter Kerwin, Parasolid's business development manager, calculates that up to 20 percent of models imported into applications using its kernel software contain errors that have to be accommodated for before they can be used."

- "Translation Time" (Mechanical Engineering, October 2006, Jean Thilmany): "... analysts at the [Sandia] lab spend around 75 percent of their time cleaning up geometry after it's been imported from computer-aided design to analysis software."

- "Simulation-Driven Design Benchmark Report - *Getting It Right the First Time"* (Aberdeen Group, October 2006): "Best in class manufacturers are 48% more likely to provide technologies to transfer models from CAD to independent preprocessors for their analysts."

Technical Approaches

- Assumptions: **No three components with a common intersection point and no partial intersection.**
- Each component is a bicubic B-spline surface defined by a rectangular grid:

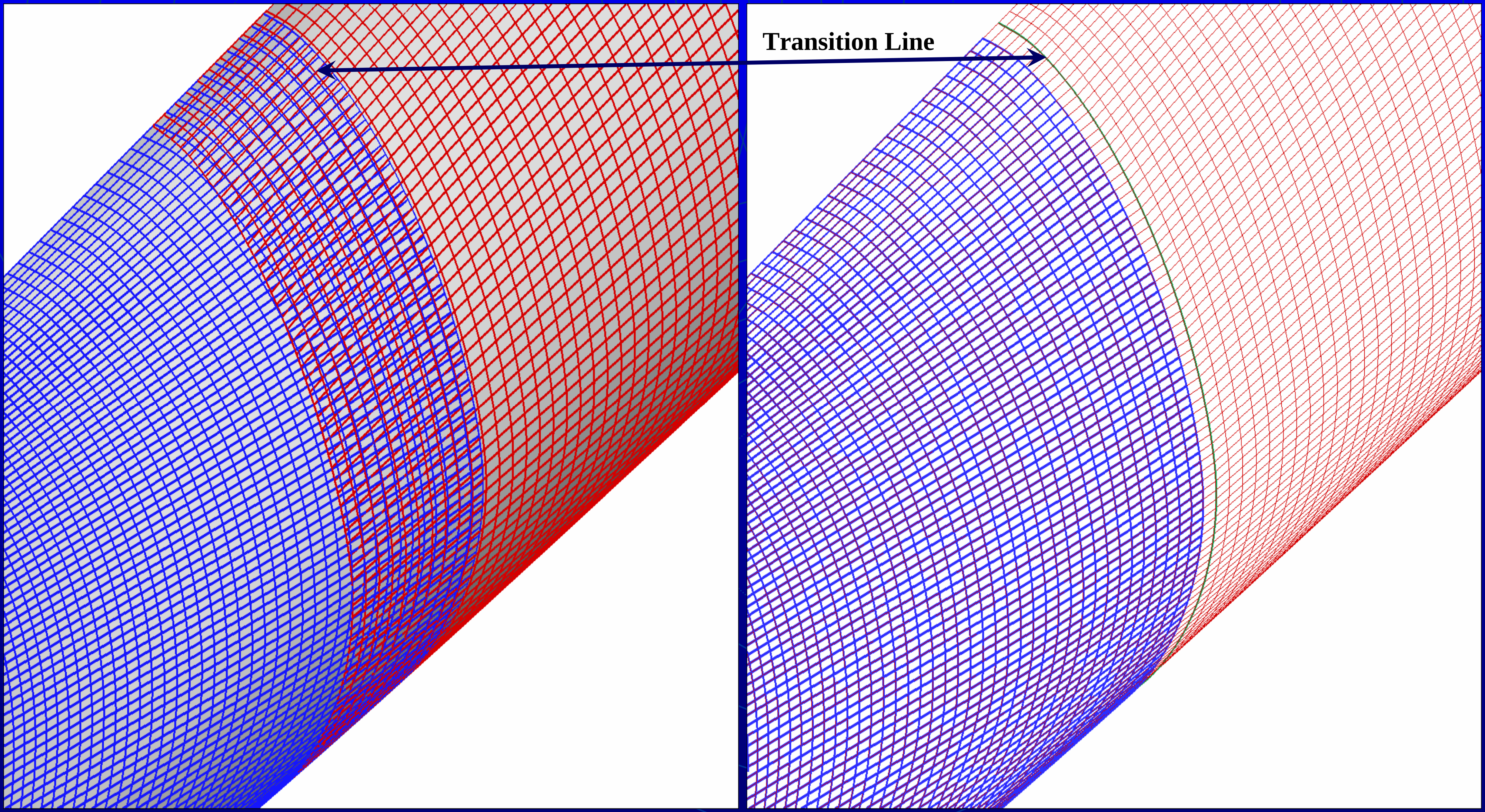
  $$(X_{ij}, Y_{ij}, Z_{ij}) \quad (1 \le i \le m,\ 1 \le j \le n).$$
- User defines the intersection matrix $\Re$:

  $$\Re(k,s) = 1$$

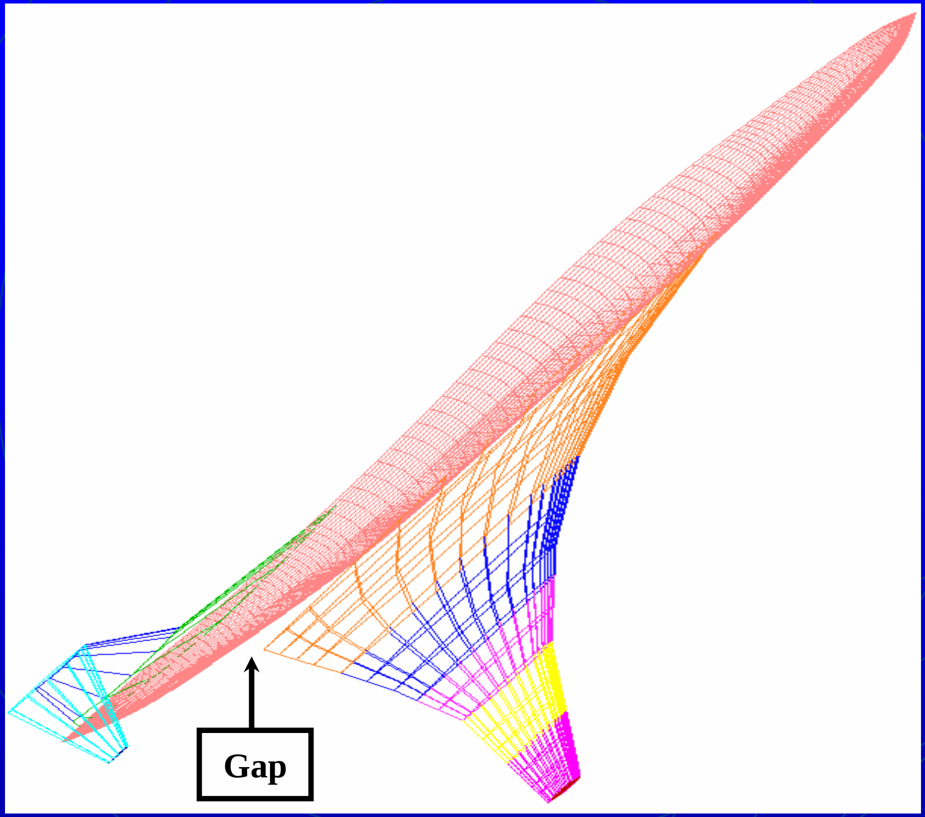
  if component #k intersects component #s.
- iPatch uses a Gaussian-Newton method to find the intersection points with option to extrapolate the first B-spline surface to assure intersection of two intersecting components.
- iPatch identifies the trimmed surface of each component after finding the intersection points and generates discrete definitions of its boundary curves in the right-hand orientation.
- iPatch generates the underlying surface in NURBS format.
- AutoSrc generates the sourcing terms to control the resolutions of CFD grids.
- VGRID (a CFD grid generation code) uses the discrete boundary curves of the watertight geometry, the associated NURBS surfaces, and sourcing terms to generate CFD grids.

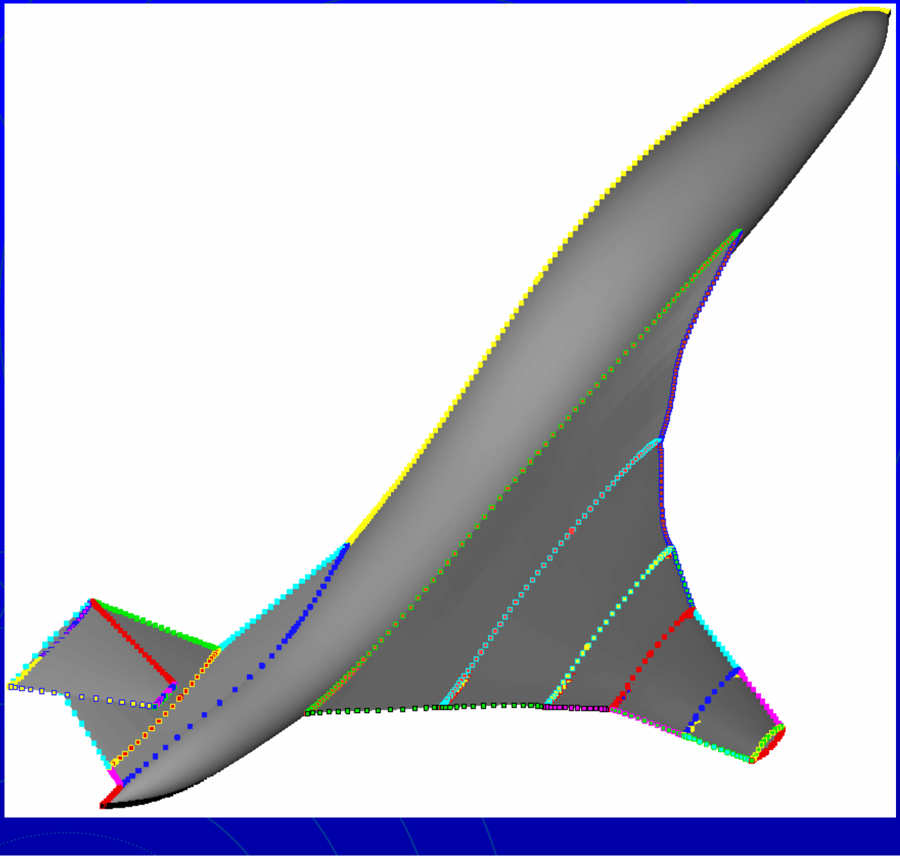
Example 1: Trimming Approximate Overlap

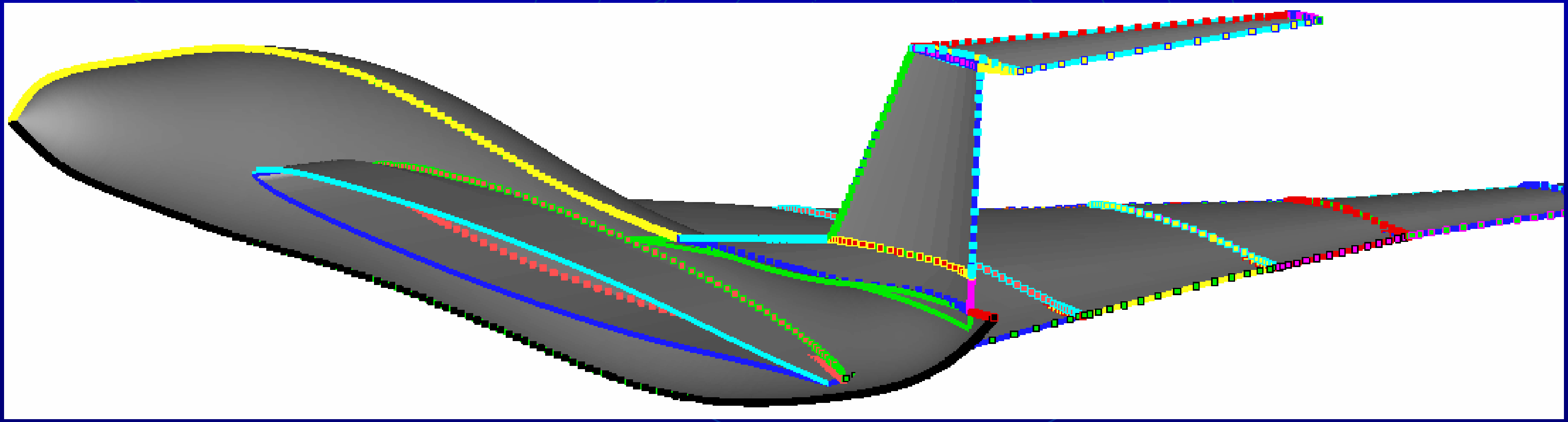
Use Hausdorff distance to find the two closest grid lines and extend grid lines on one grid to the other grid to form a merged grid. The gaps between the two closest grid lines range from 0.0001 to 0.1.



Example 2: Extrapolation for Intersection

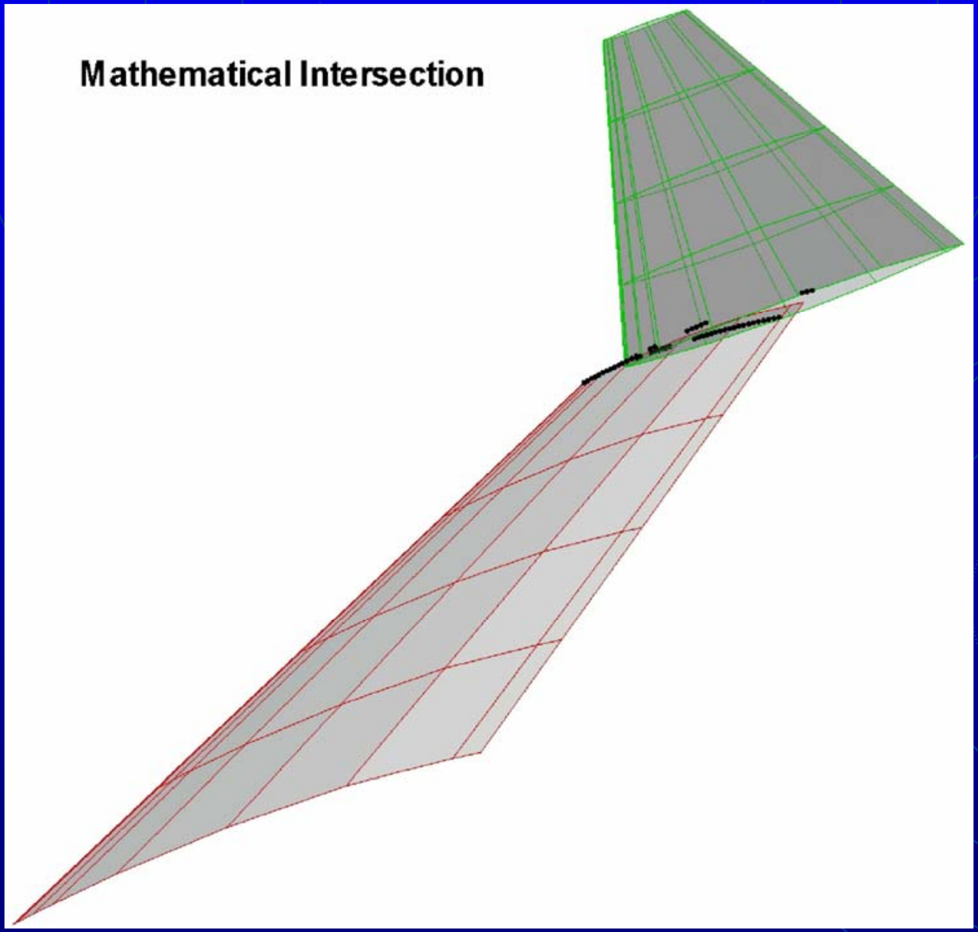


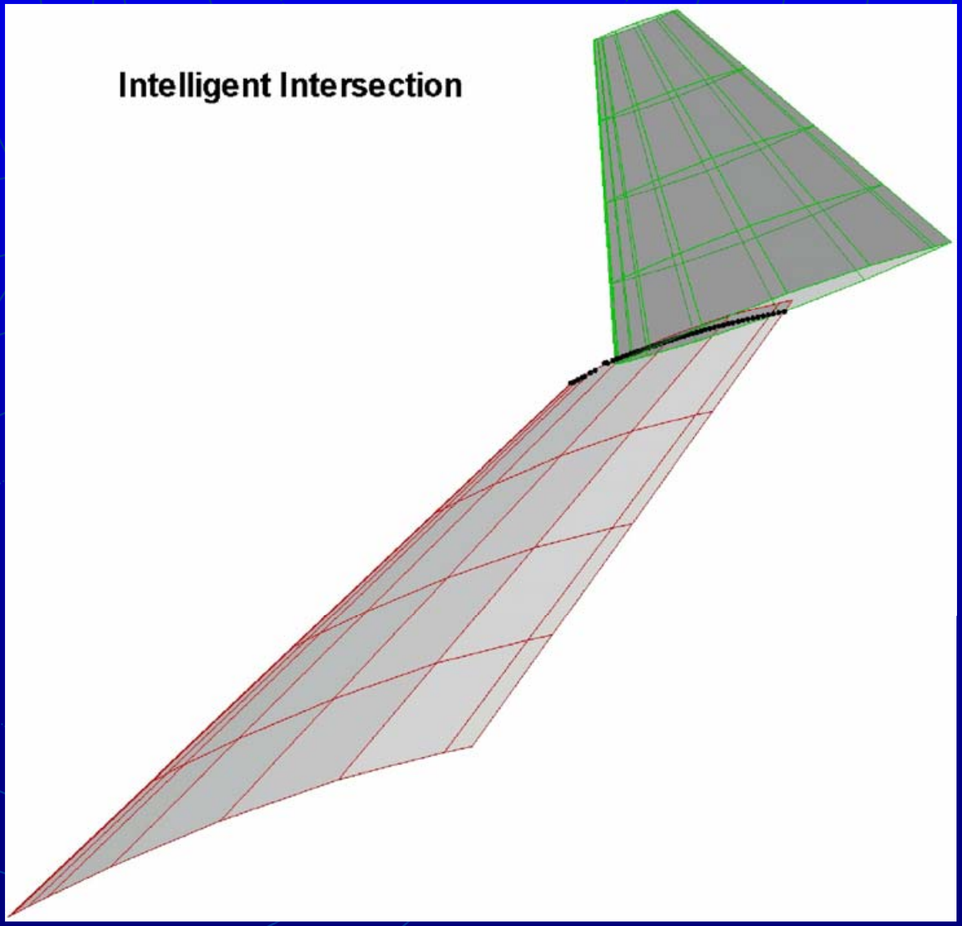




Example 3: Intelligent Intersection

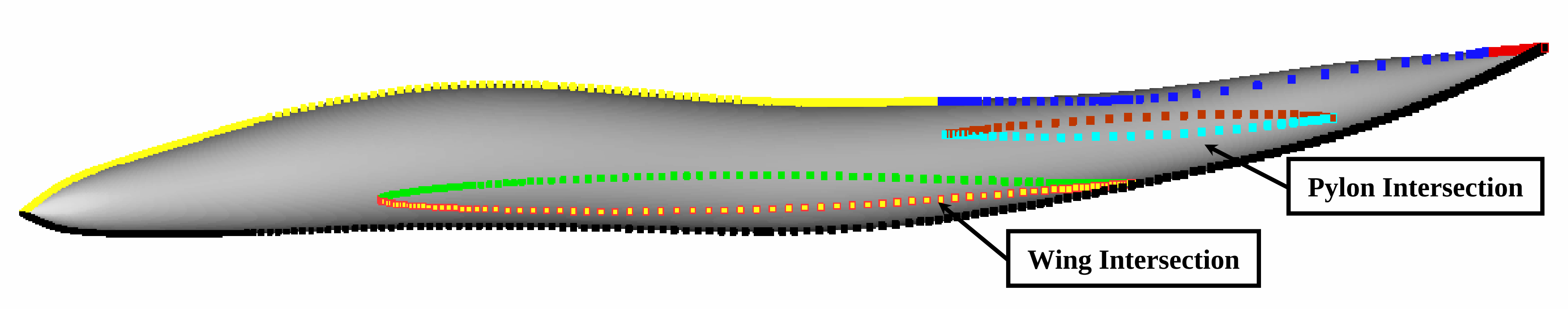
The vertical tail should intersect the lower surface of the horizontal tail. This becomes extremely difficult due to thinness of the horizontal tail and numerical intersection errors must be treated properly.

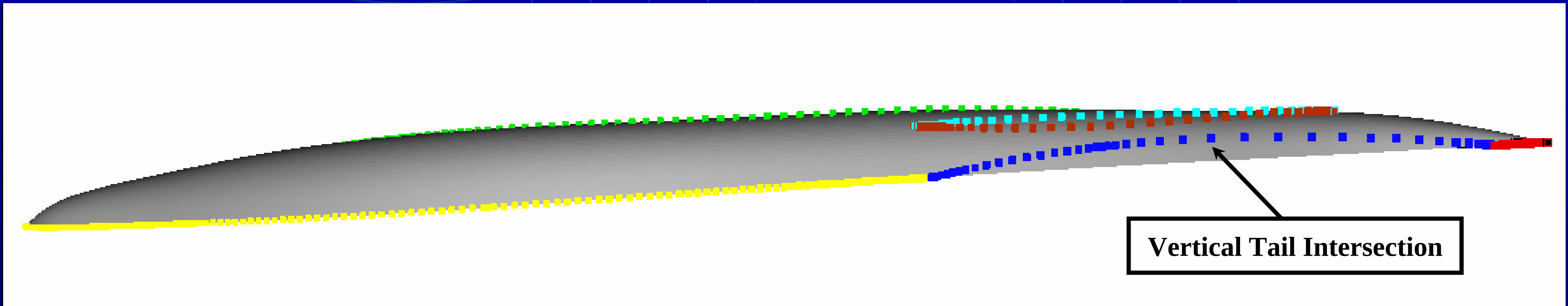




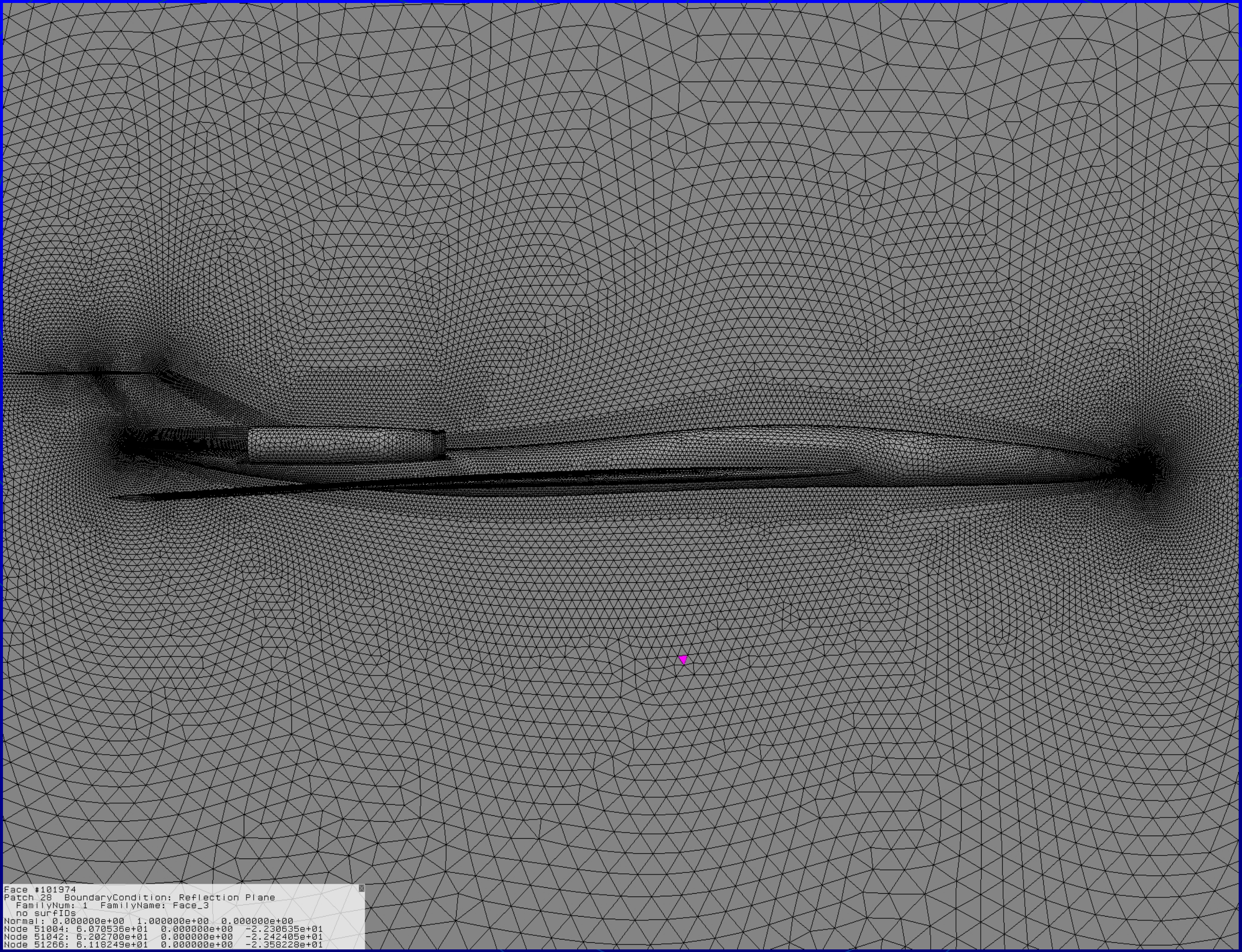
Example 4: Eight Fuselage Boundary Curves

Top boundary: 3 curves by V-tail intersection
Bottom boundary: 1 natural boundary curve
Pylon inner loop: Pylon intersection
Wing inner loop: Wing intersection





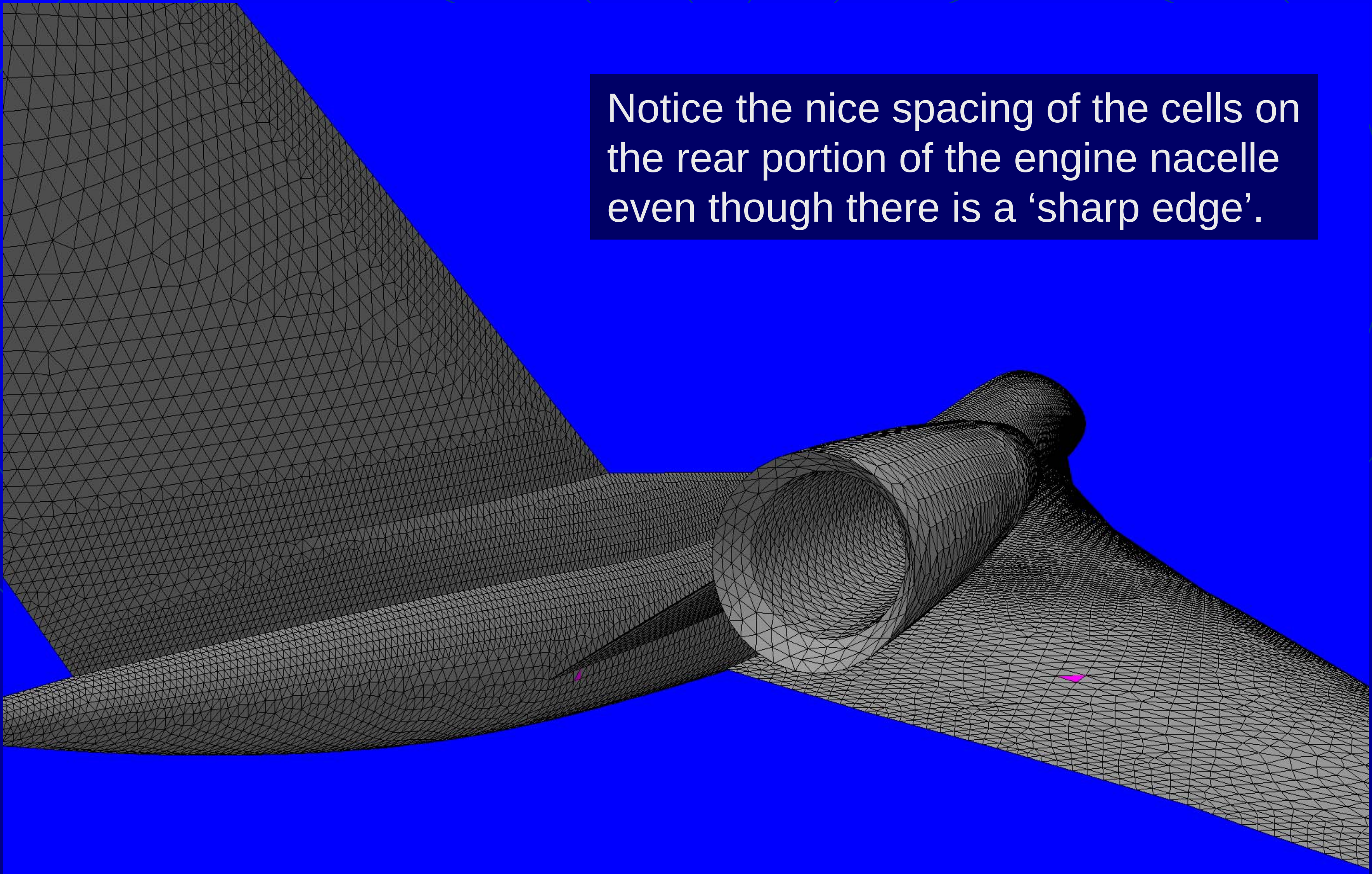
Example 5: CFD Grid on Symmetry Plane



```
Face #101974
Patch 28  BoundaryCondition: Reflection Plane
  FamilyNum: 1  FamilyName: Face_3
  no surfIDs
Normal: 0.000000e+00  1.000000e+00  0.000000e+00
Node 51004: 6.070536e+01  0.000000e+00  -2.230635e+01
Node 51042: 6.202700e+01  0.000000e+00  -2.242405e+01
Node 51266: 6.118249e+01  0.000000e+00  -2.358228e+01
```

Courtesy of Michael Arndt (National Institute of Aerospace)

Example 6: Surface Grid



Notice the nice spacing of the cells on the rear portion of the engine nacelle even though there is a ‘sharp edge’.

Courtesy of Michael Arndt (National Institute of Aerospace)

iPatch-Supported Configurations

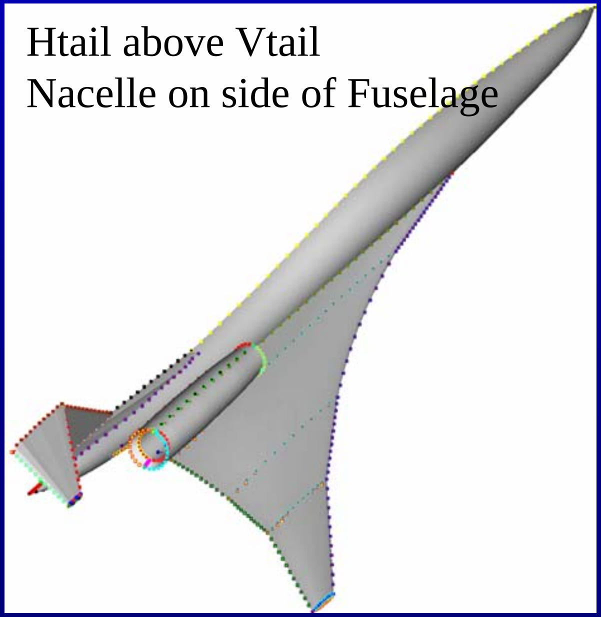
(Automatic Detection of 150+ Different Types of Configurations)

- Designed Components:
  - Wing and Fuselage
  - Canard (None, or Side of Fuselage)
  - Htail (None, Above Vtail, Side of Vtail, or Side of Fuselage)
  - Nacelle and Pylon (None, Side of Fuselage, Side of Vtail, Above Wing, or Below Wing)
  - Vail (None, or Above Fuselage)
- Optional Components:
  - Circular or flat wing tip
  - Circular or flat canard tip
  - Circular or flat tail tip
  - Inner surface of nacelle

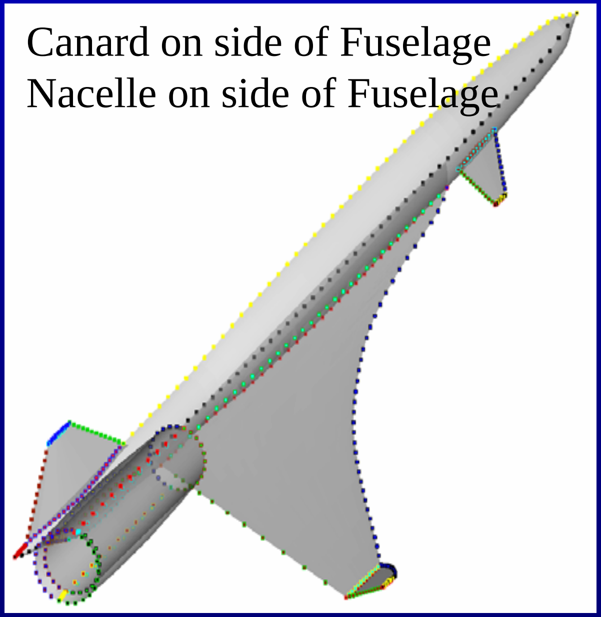
Limitation: No three components with a common intersection point and no partial intersection.

User Inputs: awave or plot3d geometry file, indices of wing section boundaries, density of boundary points.

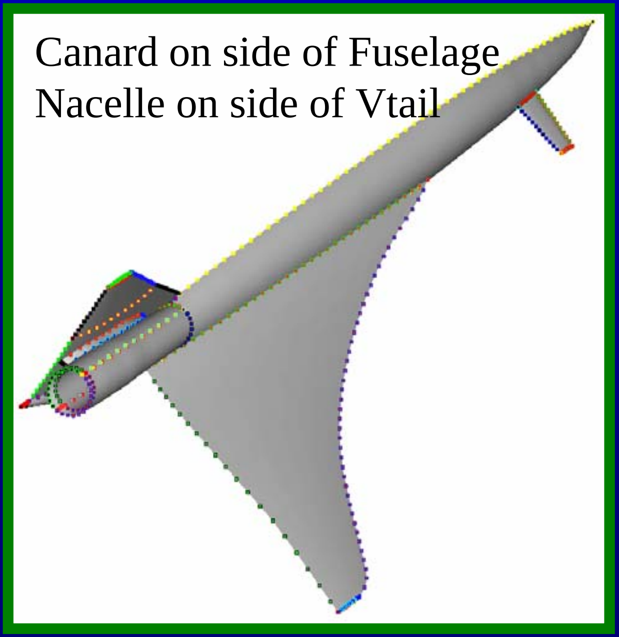
V&V: gcc and cc on SGI and Linux, gcc on PC



Htail above Vtail
Nacelle on side of Fuselage



Canard on side of Fuselage
Nacelle on side of Fuselage

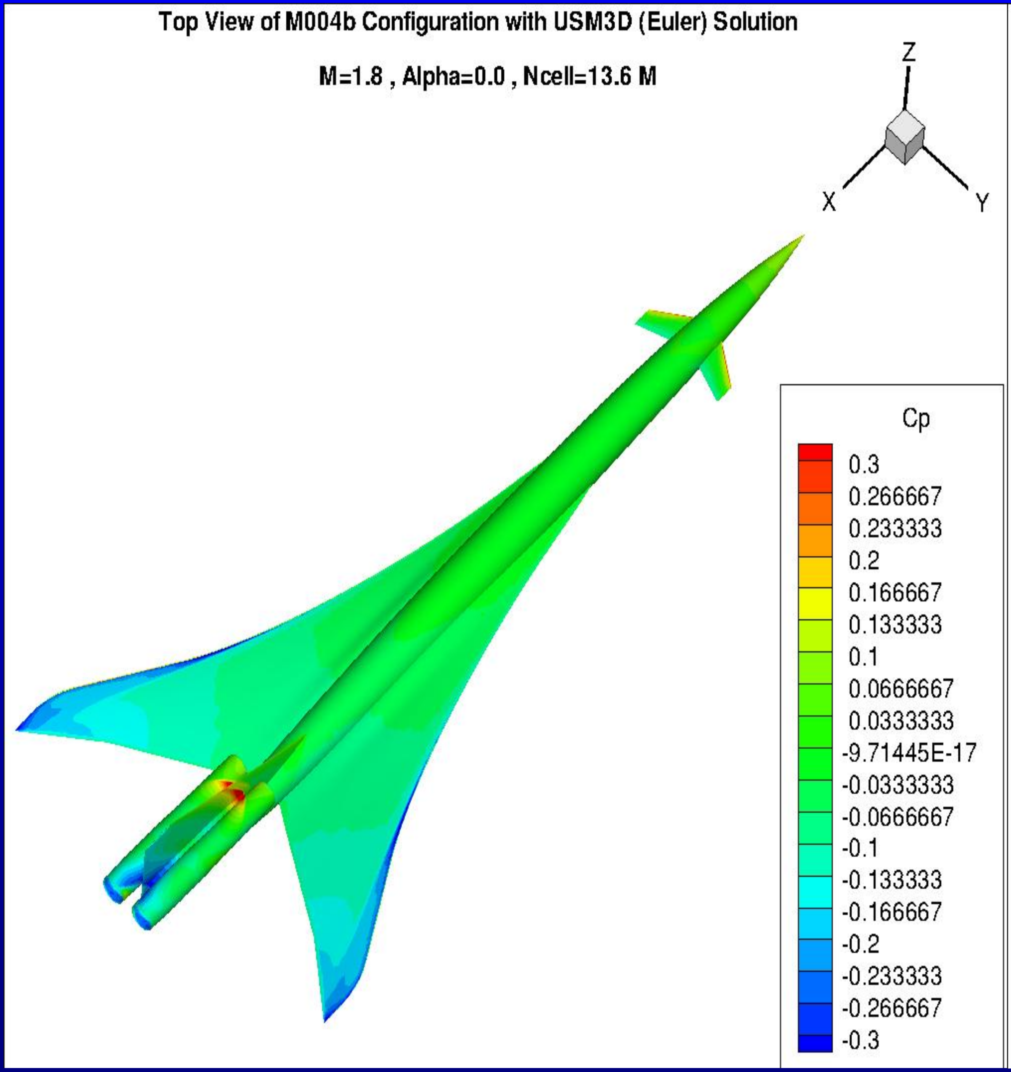


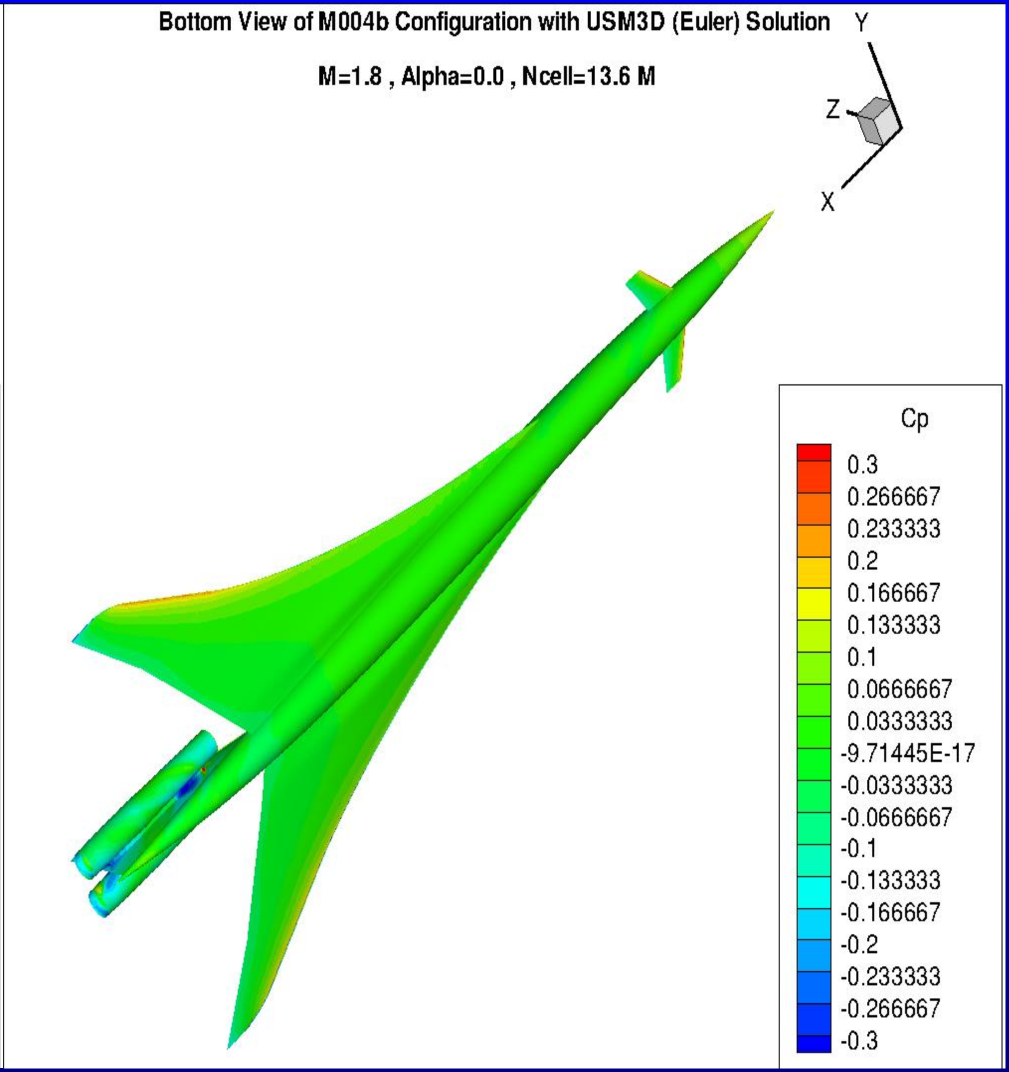
Canard on side of Fuselage
Nacelle on side of Vtail

Euler Solution With Manual Sourcing Process

Actual Computation Time: about 24 hours (5000 Iterations)

Turn-around Time: 4 Weeks (Including Test of Sources)

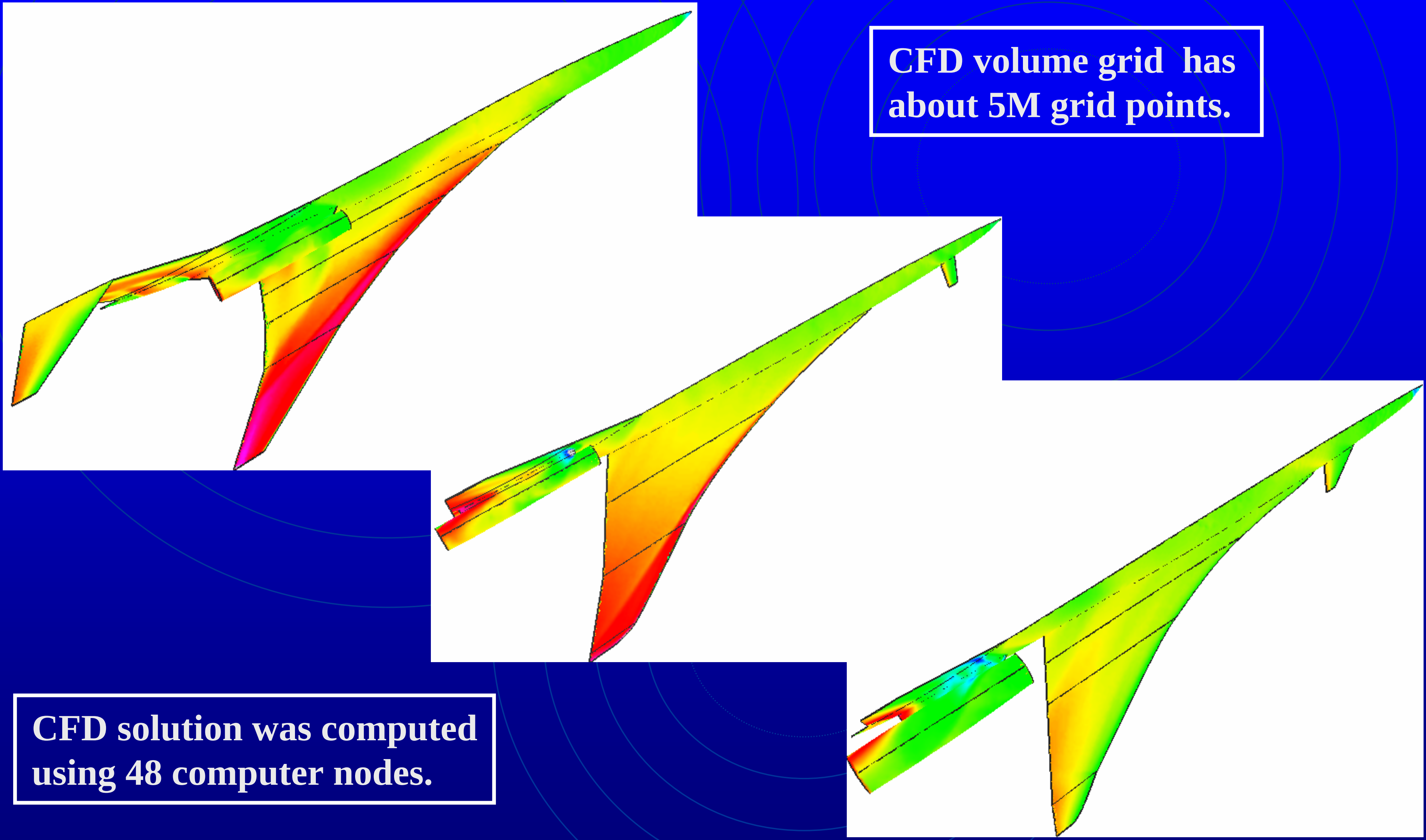




Courtesy of Mamad Takallu (Lockheed Martin)

Euler Solution With Automatic Sourcing Process

Computation Time ≅ 20 min (grid generation) + 15 min (CFD solution) + 25 min (IO)



CFD volume grid has about 5M grid points.

CFD solution was computed using 48 computer nodes.

Courtesy of Richard Campbell (NASA Langley Research Center)

Conclusions and Future Work

1. By using iPatch and AutoSrc, one can start with a grid definition of conceptual geometry and get Euler CFD solutions in a few hours.
2. The automatic CFD analysis process only works on configurations that satisfy the following two conditions:
   - No three components intersect at a common point.
   - There is no partial intersection between any two components.
3. For more realistic definitions of aircraft configurations, we need to incorporate surface-to-surface blending capability in iPatch.
4. *It might take some trial-and-error runs to get the desired size and resolution of volume grids by using AutoSrc and VGRID. The automatic grid generation process is still under further development.*

Backup Slides

Simulation-Driven Design Benchmark Report

Getting It Right the First Time

October 2006, Aberdeen Group

- **Findings**
  - All best in class manufacturers use simulation in the design phase compared to only 75% of laggards.
  - Best in class manufacturers are 63% more likely to provide CAD-embedded simulation to their engineers.
  - ***Best in class manufacturers are 48% more likely to provide technologies to transfer models from CAD to independent preprocessors for their analysts.***
  - Best in class performers are 42% more likely than all others to provide specific examples to users for training.
- **Recommendations for Action**
  - Perform more simulation of product performance in the design phase.
  - Provide CAD-embedded or CAD-driven simulation capabilities to engineers.
  - Use training materials and specific examples to get new users up to speed.
  - ***Employ technologies that transfer geometry from CAD to independent preprocessors for analysts.***
  - Track requirements and regulatory product compliance prior to design release.

Multi-CAD Design Chain Benchmark Report

Engineering from Today's Multi-CAD Environment

December 2006, Aberdeen Group

- **Findings**
  - Top performers are 63% and 90% more likely to use an insulated product development process with customers and suppliers respectively, delivering native CAD formatted design data while designing on different internal, standardized CAD tools.
  - Top performers are ten times more likely to use third-party translation applications.
  - Top performers are 2.5 times more likely to use engineering visualization to assemble mixed CAD design data.
  - Top performers are 23% more likely to use a single data management tool and 50% more likely to use that technology to track and manage the associations that occur when designs exist in multiple formats.
- **Recommendations for Action**
  - Employ an insulated product development process with customers and suppliers.
  - OEMs should outsource the translation or re-creation of design data.
  - ***Deploy third-party translation applications to convert design data to different formats.***
  - Utilize engineering visualization tools to assemble multi-formatted design data.
  - Implement a single data management system to manage design data of all formats.
  - Use data management to associate designs in different formats to one another.