PROGRESS IN AUTOMATION OF OVERSET STRUCTURED SURFACE GRID GENERATION

William M. Chan

NASA Ames Research Center

13th Symposium on Overset Composite Grids and Solution Technology, Mukilteo, Washington, October 17- 20, 2016

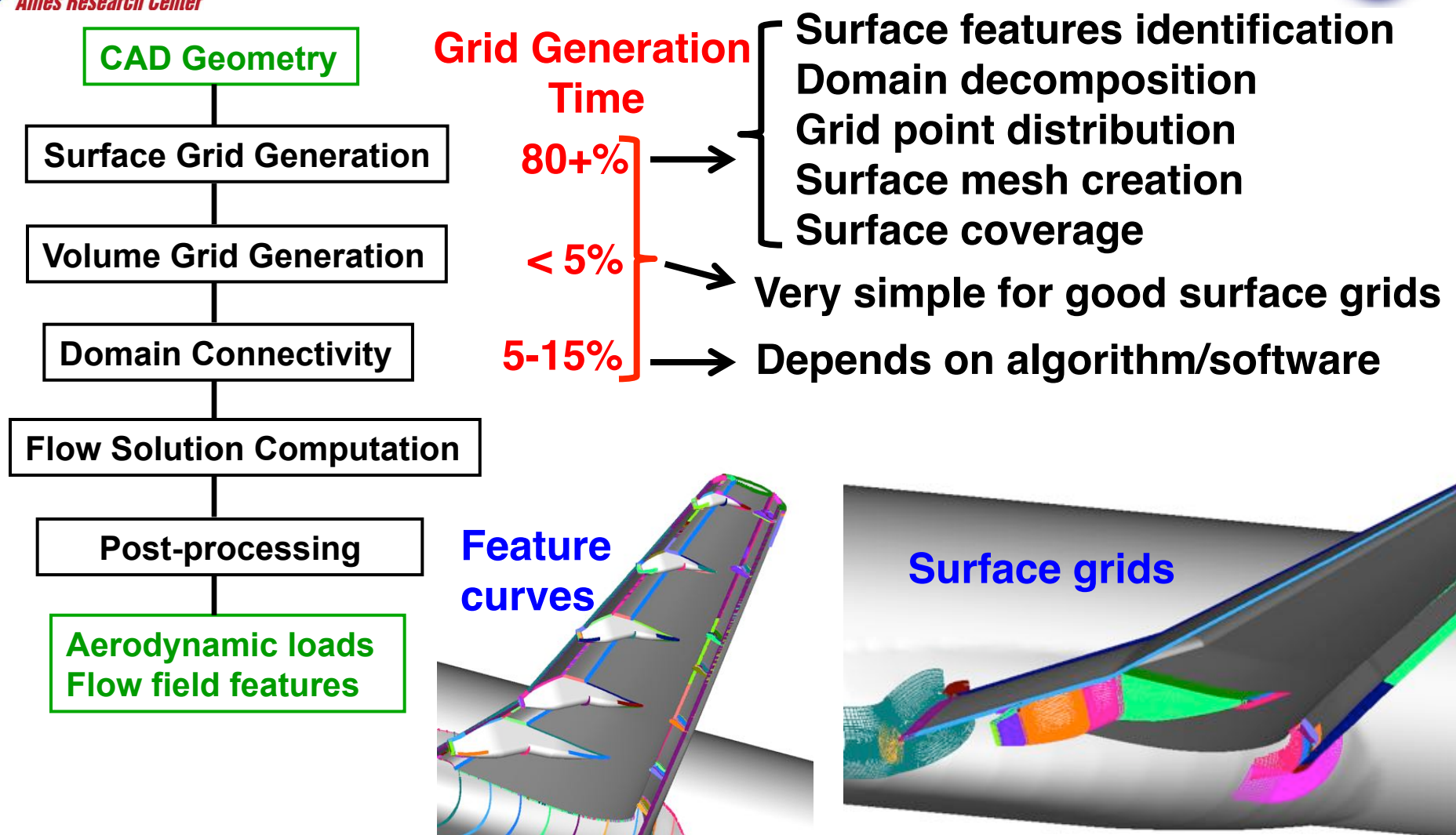# OVERVIEW

- A brief history of current technology

- Starting point and objectives

- Overview of three main steps

- Algebraic step details

- Future plans for subsequent steps

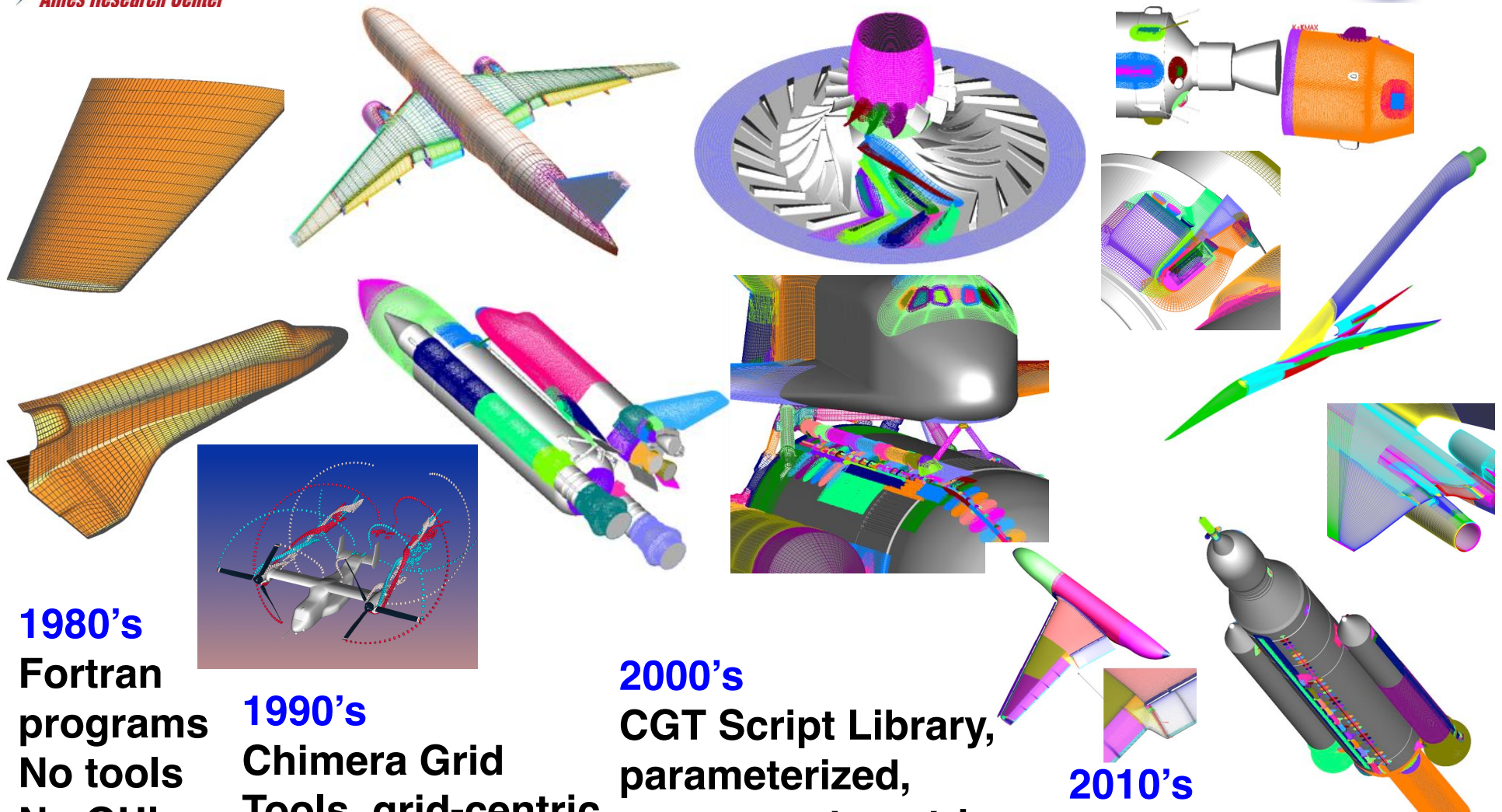- Summary and conclusions

STRUCTURED OVERSET GRID CFD PROCESS

CAD Geometry

Surface Grid Generation

Volume Grid Generation

Domain Connectivity

Flow Solution Computation

Post-processing

Aerodynamic loads
Flow field features

Grid Generation Time

80+% → Surface features identification
Domain decomposition
Grid point distribution
Surface mesh creation
Surface coverage

< 5% → Very simple for good surface grids

5-15% → Depends on algorithm/software

Feature curves

Surface grids

C = number of components
G = number of grids
P = number of grid points

A BRIEF HISTORY

1980's
Fortran
programs
No tools
No GUIs
C < 10
G < 10
P < 10M

1990's
Chimera Grid
Tools, grid-centric
scripts, Gridgen,
Overture, C ~ 50
G ~100, P ~ 50M

2000's
CGT Script Library,
parameterized,
component-centric
scripts, Glyph,
C ~ 100, G ~ 200,
P ~ 100M

2010's
Parallelized using
team of workers,
C ~ 100's, G ~ 1000,
P ~ 100's M
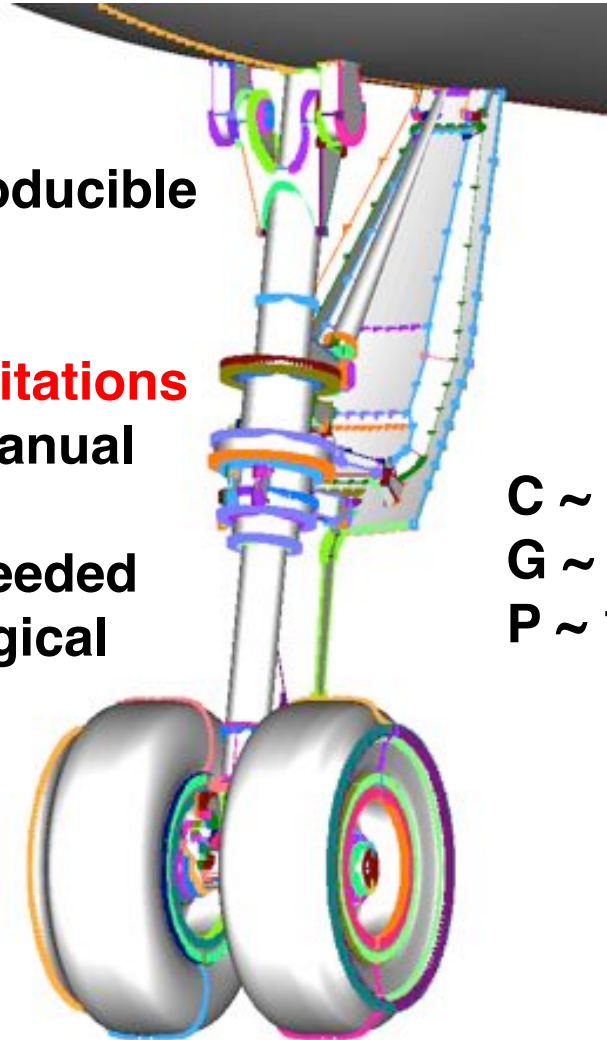
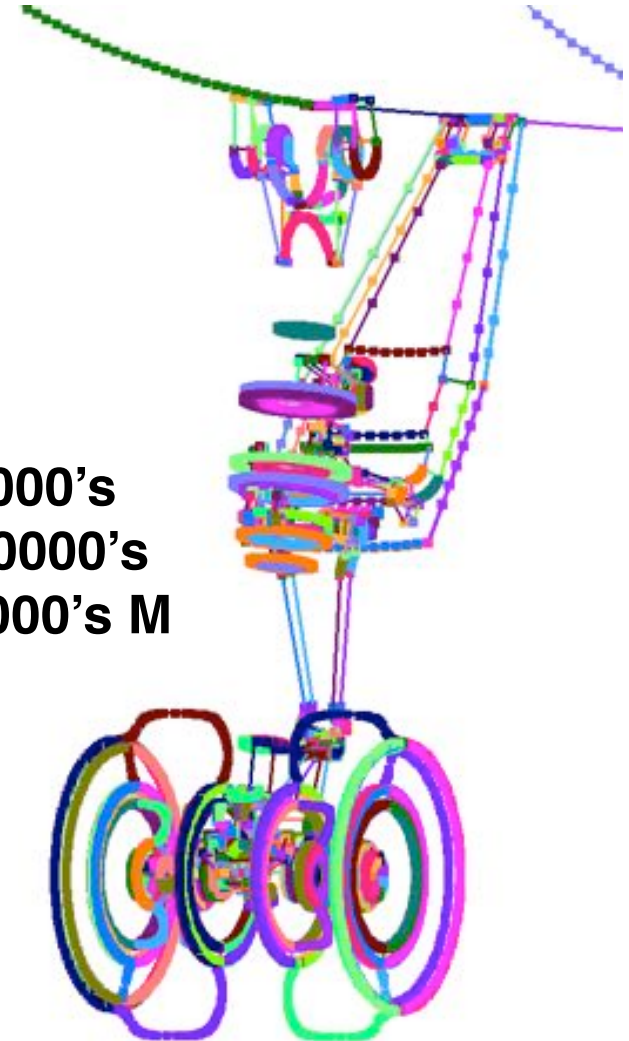# CURRENT TECHNOLOGY LIMITATIONS AND FUTURE COMPUTATIONS

GUI limitations
- **Steps not easily reproducible**
- **Not parameterized**

Scripting approach limitations
- **Require significant manual effort to build**
- **Script modification needed even for small topological changes**

C ~ 1000's
G ~ 10000's
P ~ 1000's M

- **Manual scripting approach (even with team) is no longer practical**
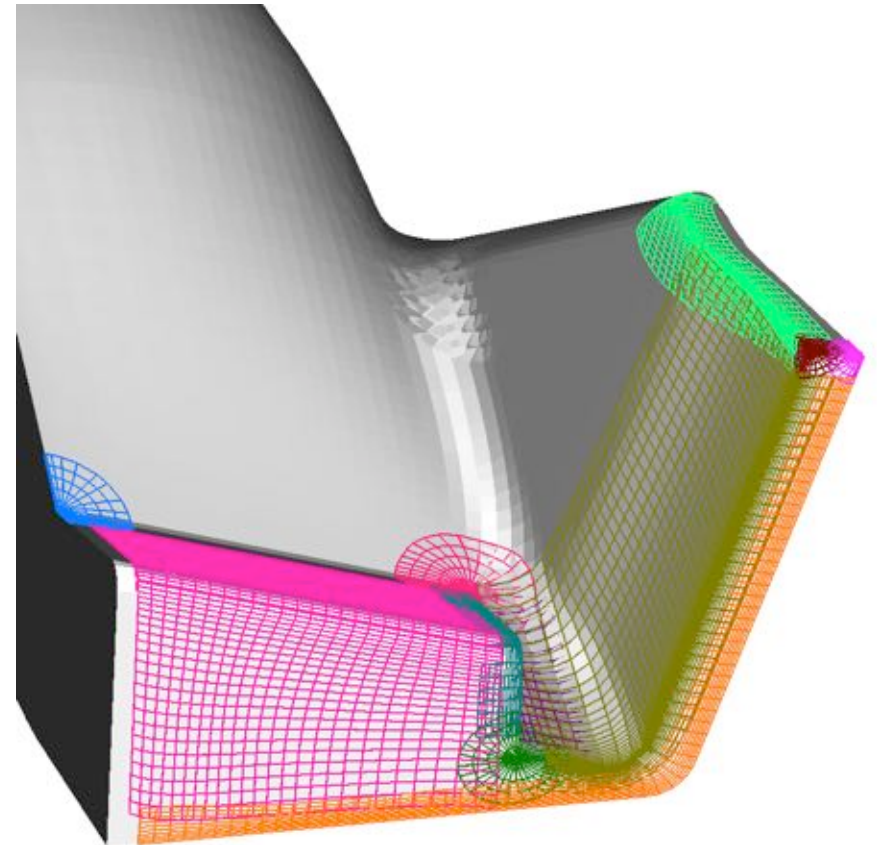- **Difficult to achieve higher level of abstraction beyond component level**

PREVIOUS WORK ON SURFACE GRID GENERATION AUTOMATION

**Chan and Gomez
(AIAA Paper 99-3303)**
Wrap around surface features using hyperbolic grids and spider web grids
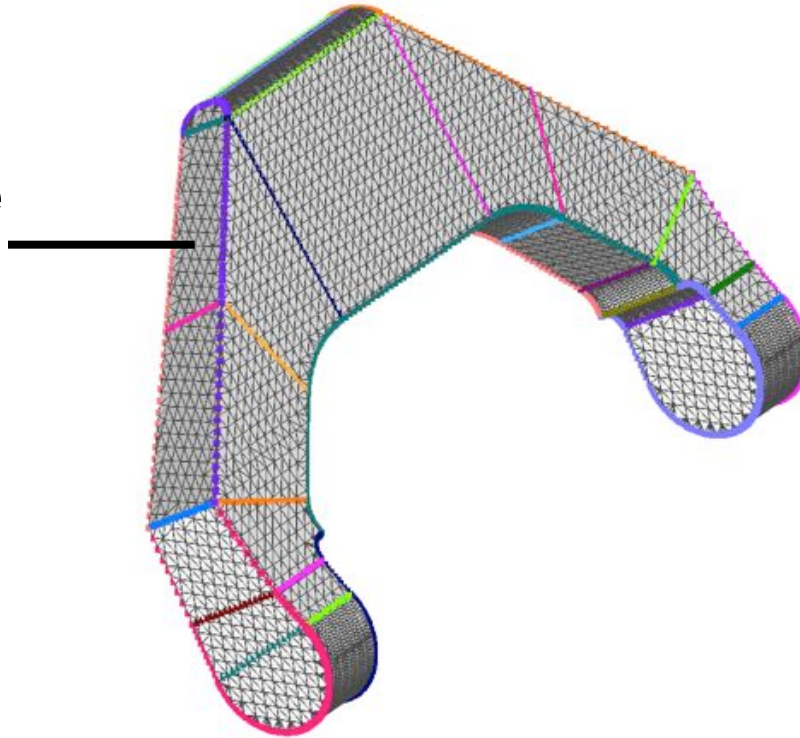
**Dannenhoffer and Haimes
(AIAA Paper 2011-3540)**
**Use feature trees from CAD solid models to build up from basic solid shapes**

STARTING POINT

Reference surface ————

- **Unstructured surface triangulation geometry representation derived from native CAD, STEP or IGES file with proper resolution of high curvature regions**

- **Discrete surface curves (subset of triangle edges)**
    - **Surface features (hard edges to be retained in computational grid)**
    - **Max curvature curves (e.g., leading edges)**
    - **Open boundaries**

CURRENT MANUAL PROCEDURE
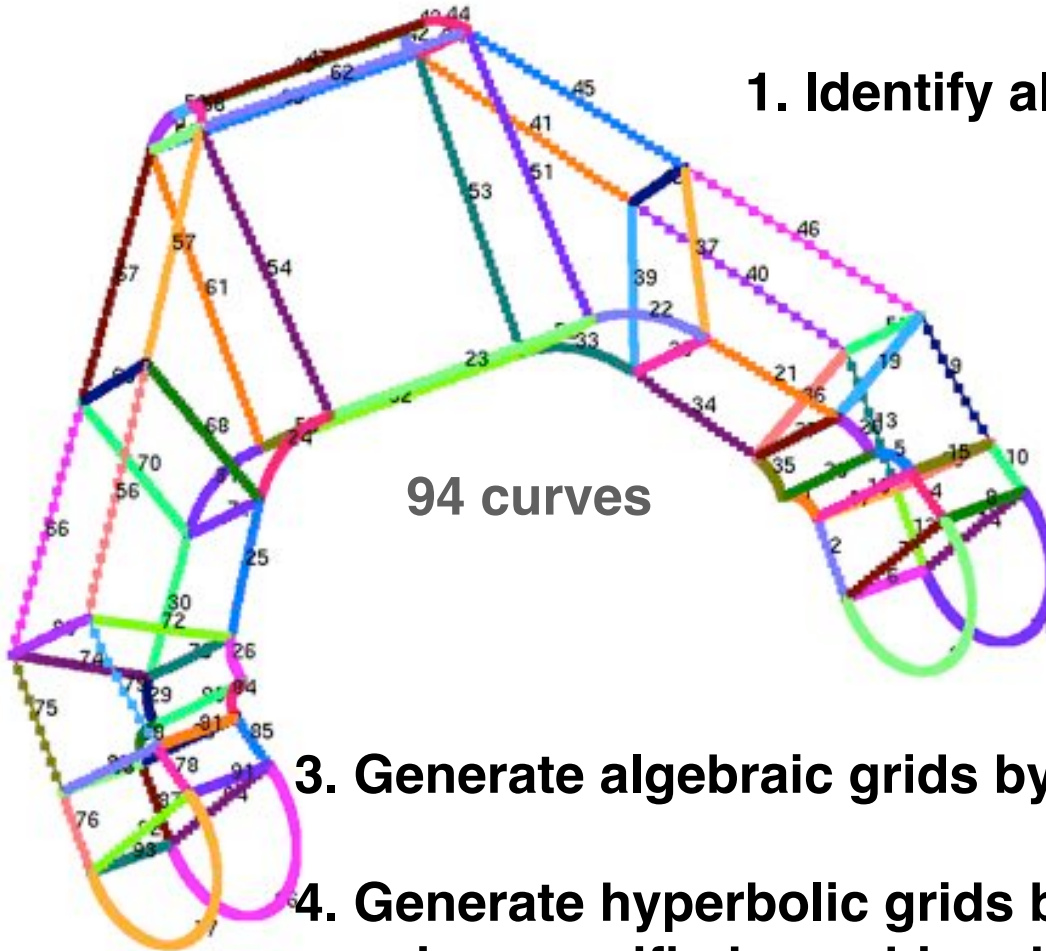
94 curves

1. Identify algebraic and hyperbolic domains

2. Determine grid point distribution on curves:
- uniform/stretched
- end point and max spacing
- match grid points on opposite boundaries

3. Generate algebraic grids by specifying bounding curves set

4. Generate hyperbolic grids by marching from initial curves using specified marching distance, initial and end spacing, max stretching ratio

5. Concatenate algebraic and hyperbolic grids where appropriate

OBJECTIVES

Develop algorithm/software for overset structured surface grid generation
- **Significant manual effort reduction (near term)**
- **Full automation (long term)**

REQUIREMENTS

- **Capture essential surface features (sharp edges, max curvature lines such as leading edges) so that they lie on grid interior**
- **Distribute grid points to**
  - **resolve surface and flow features accurately**
  - **maintain smooth variation with low stretching ratio (trunc. error)**
- **Maintain NFRINGE points overlap between neighboring grids where NFRINGE is determined by flow solver differencing stencil**

GOOD TO HAVE

- **Compatibility of cell attributes in overlap regions**
- **As few domains as possible, but not critical since flow solver will break grids up for MPI runs**

TOP LEVEL DEVELOPMENT PLAN
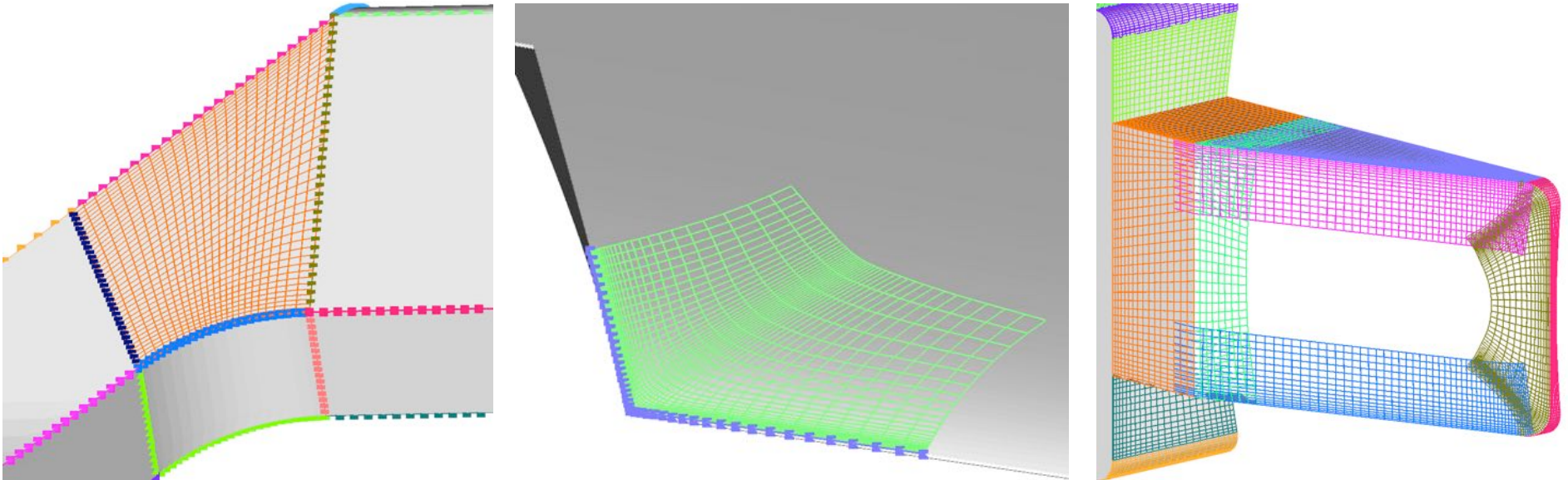
Build grids around surface features using
   1. Algebraic methods **(domains bounded by 4 curves)**
   2. Hyperbolic methods **(domains bounded by 1 curve)**

Fill remaining interior gaps on smooth regions of geometry with algebraic or hyperbolic grids

ALGEBRAIC STEP
Control Parameters

Sharp edge ——
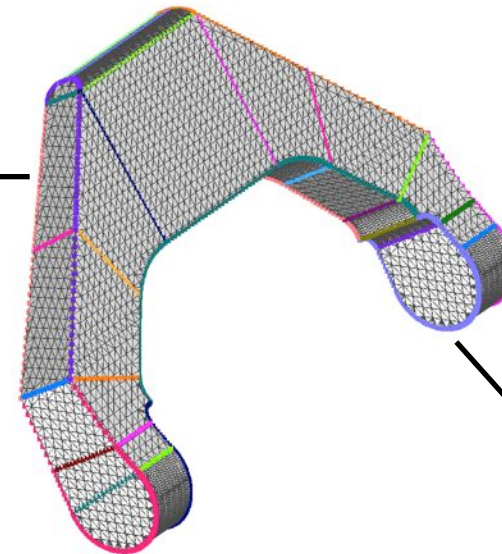
Turning curve

Default values in () below

Classification Parameters

θ_{sharp} : triangle edge is a sharp edge if angle between neighboring
triangles unit normal is larger than θ_{sharp} (20 deg)

θ_{tot} : curve is turning curve if the total turning angle > θ_{tot} (30 deg)

Grid Point Distribution Parameters

Np_{min} = minimium number of points on a curve (5)

SR_{max} = max stretching ratio (1.2)

Δs_{max} = max grid spacing (0.5% of global bounding box diagonal)

θ_{ml} = max local turning angle on turning curves (5 deg)

ALGEBRAIC STEP
1. Determine Curves Network Connectivity

4-sided domain

5-sided domain

degree 4 junction

degree 3 junction
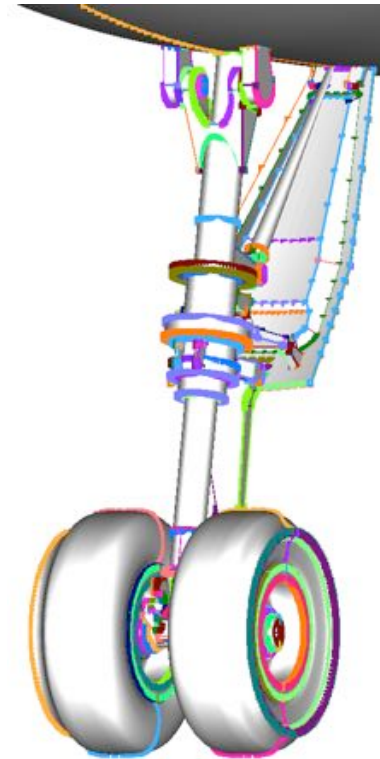
Feature curves
Unsplit : 688
Split   : 1272

Junctions
Deg 3 : 610
Deg 4 : 149
Deg 5 :   8
> Deg 6 : 0

Four-sided domains : 311

- Split curve at interior point if necessary so that all curves only meet at end points
- Identify all junction points (where 3 or more curves meet)
- Store curve ID and end point for all curves meeting at junction

ALGEBRAIC STEP
2. Determine Curve Attributes

Not turning curve

On open boundary

Turning curve

Short curve (blunt t.e.)

Not on sharp feature

Not on open boundary
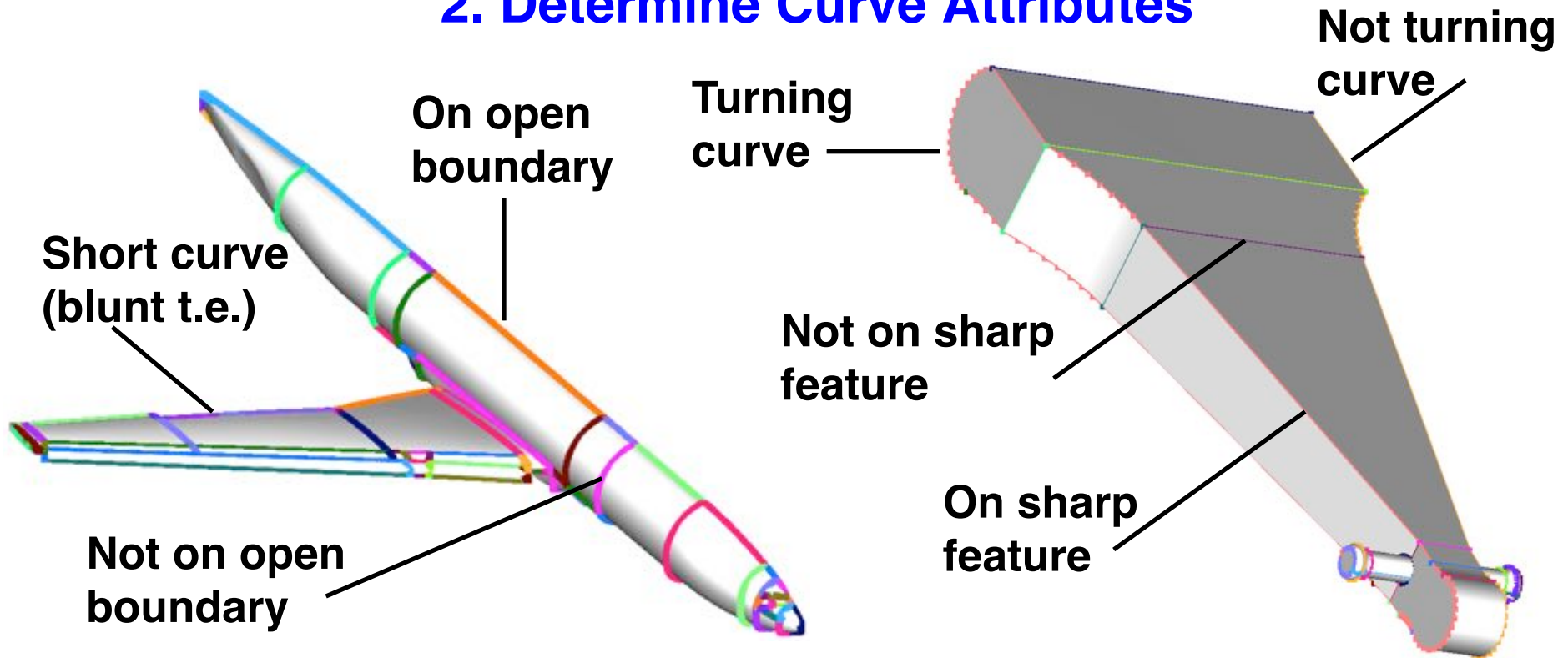
On sharp feature

- On sharp feature (yes – must preserve, control parameter θ_{sharp})
- On open boundary (yes – can only go one side for grid concatenation)
- Is a "short" curve (total arc length <= $(Np_{min} - 1) * \Delta s_{max}$)
- Is a "turning" curve (total turning angle > θ_{tot})

ALGEBRAIC STEP
3. Distribute Grid Points on Curves
3.1. Determine Distribution Type and Spacings

Turning curves or short non-turning curves:
 Uniform spacing Δs
 determined by Np_{min} or θ_{ml}
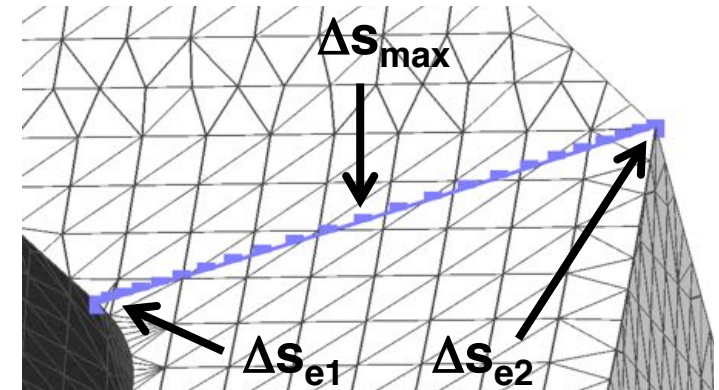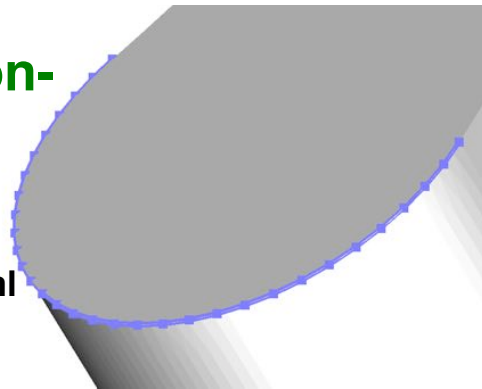
Long non-turning curves:
 Non-uniform spacing hyperbolic tangent distribution with specified
  - max stretching ratio SR_{max}
  - max interior spacing Δs_{max}
  - spacing at end points Δs_{e1} , Δs_{e2}
   - on geometry open boundary $\Delta s_e = 0.5\ \Delta s_{max}$
   - on geometry sharp edge with dihedral angle θ
   - use scale factor $\kappa(\theta)$ such that $\Delta s_e = \kappa(\theta)\ \Delta s_{max}$
    - concave corner ($\kappa > 1$, increases linearly with increasing concavity)
    - convex corner ($\kappa < 1$, decreases negative exponentially with
        increasing convexity)

ALGEBRAIC STEP
3. Distribute Grid Points on Curves
3.2. Match Grid Spacing at Junctions

Impose grid spacing continuity for matching curve pairs by taking minimum of predicted grid spacings
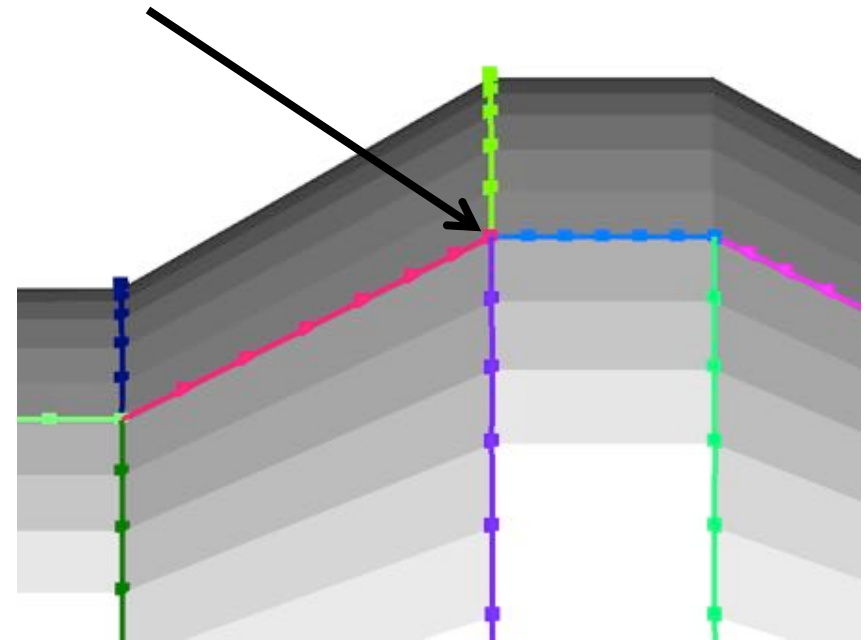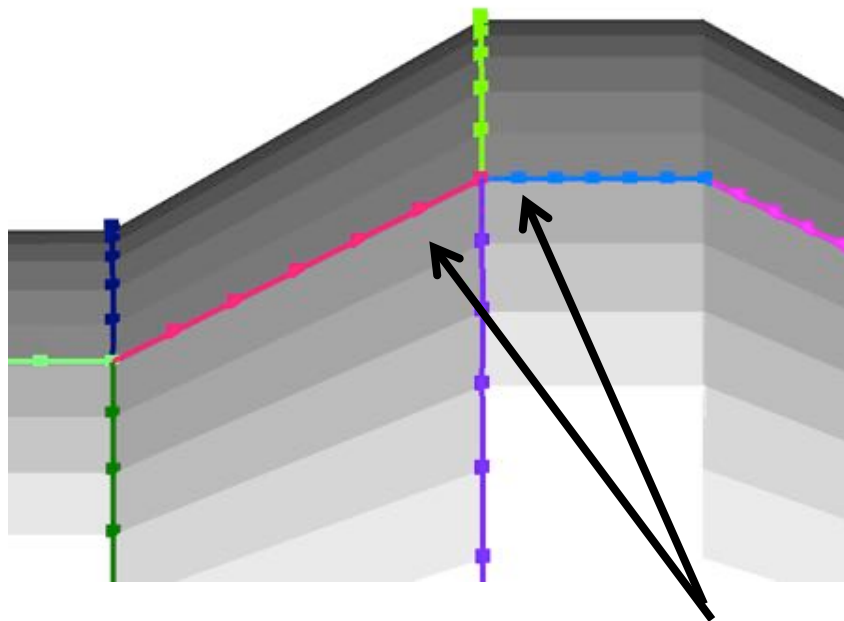
Matching pair at junction
(compare curve end point unit vectors at junction)

ALGEBRAIC STEP
4. Identify Four-Sided Domains

Geometry surface and feature curves
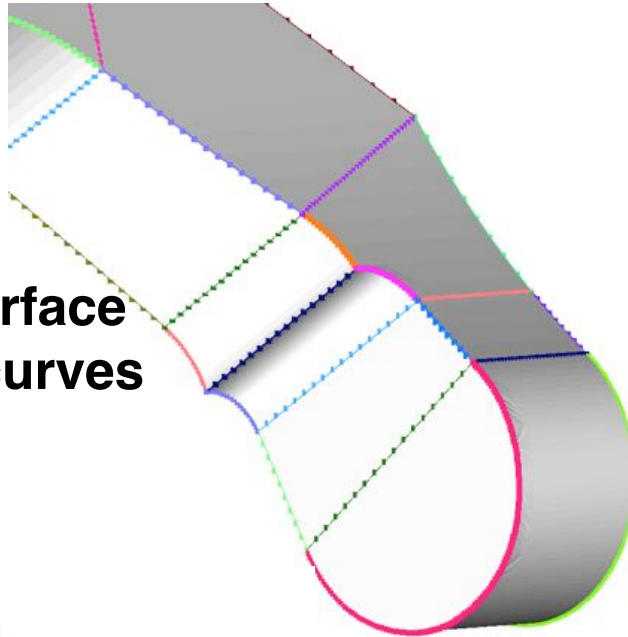
Valid choice on surface
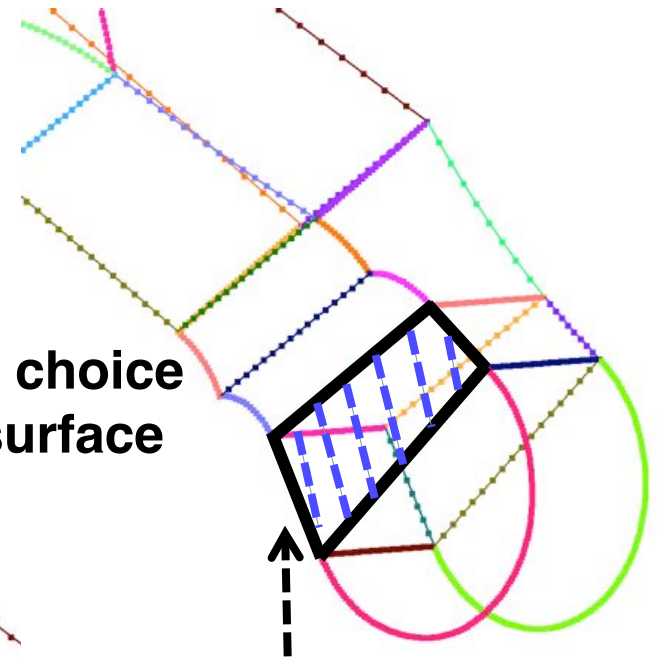
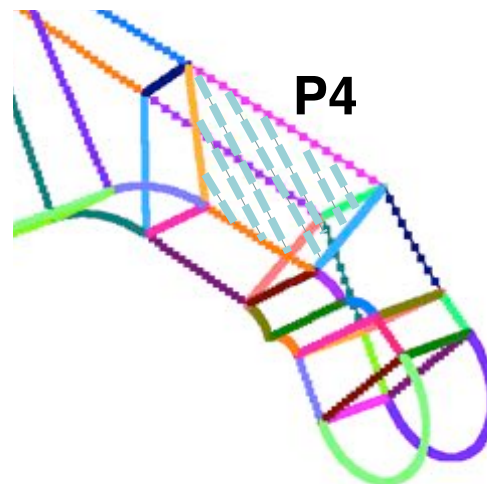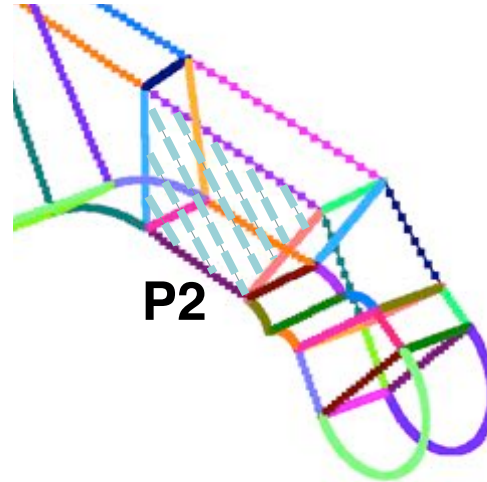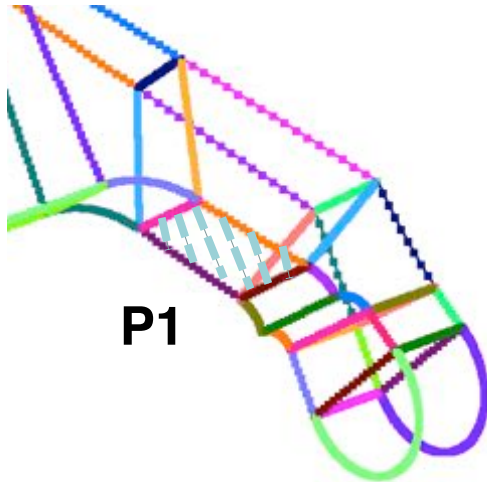**Invalid choice not on surface**
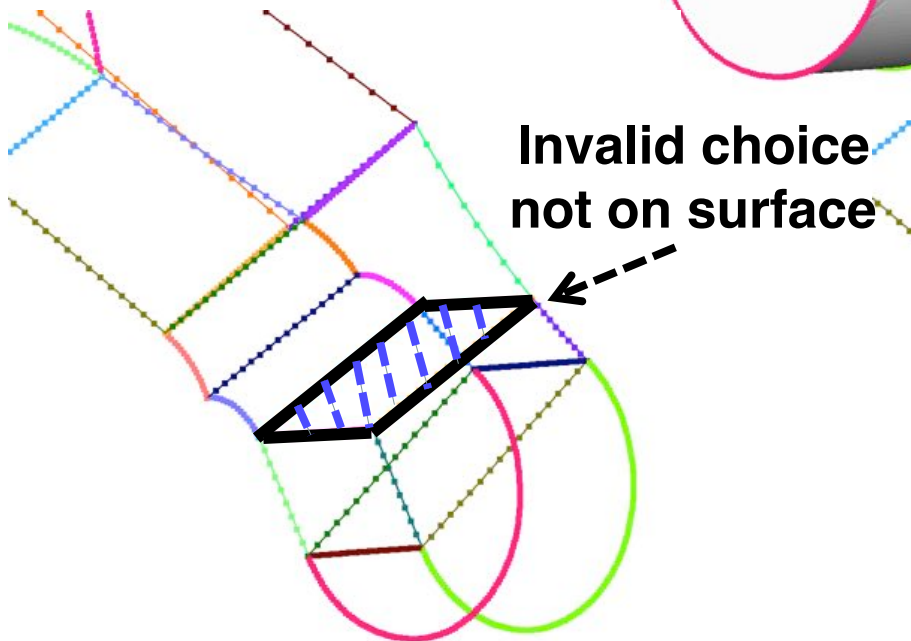
Prefer more orthogonal option at corners
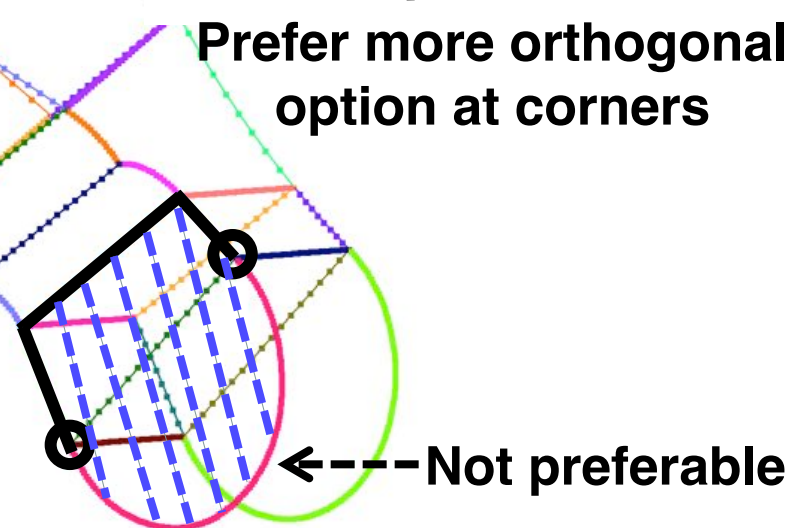
Not preferable

ALGEBRAIC STEP
5. Identify Adjacent Four-Sided Domains for Subsequent Concatenation

P1

P2

P3

P4
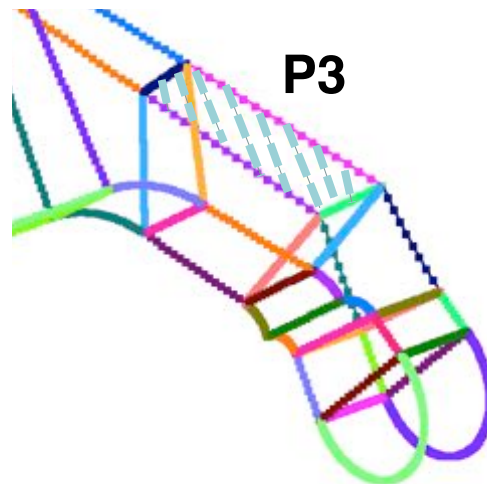
- **Start with initial curve on open boundary or sharp edge bounding a four-sided domain**

- **Look for other four-sided domains that can be concatenated to initial or opposite curve**

- **Single direction sweep only**

P1, P2, P3, P4 can be concatenated – impose same number of grid points on curves in concatenation direction

ALGEBRAIC STEP
6. Generate Surface Mesh Using TFI and Concatenate
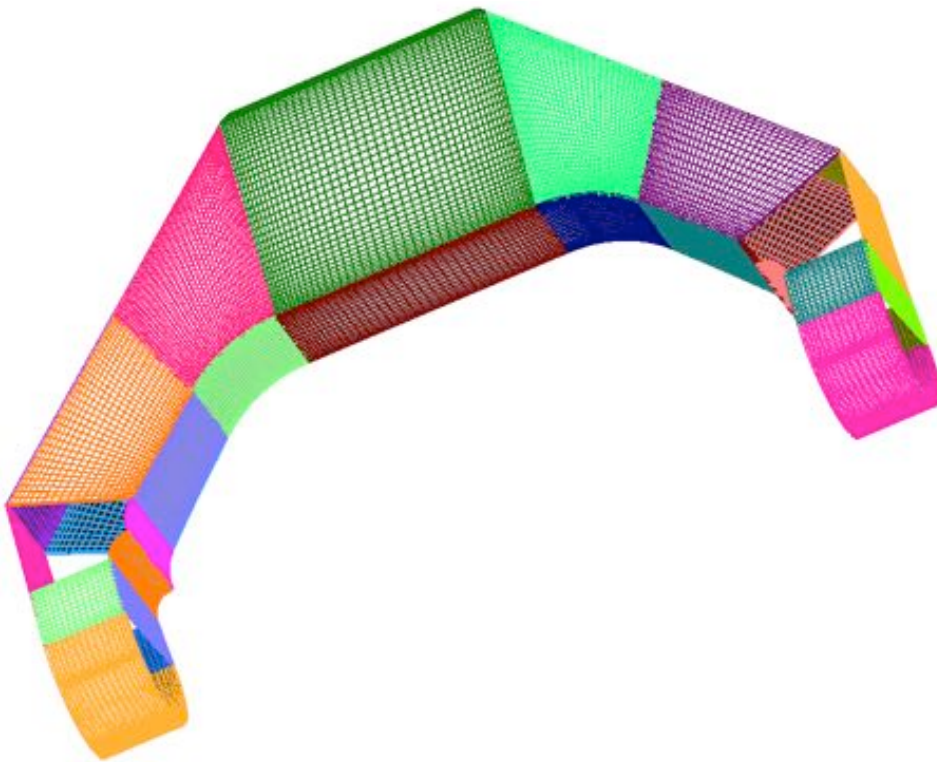
Before concatenation: 38 patches
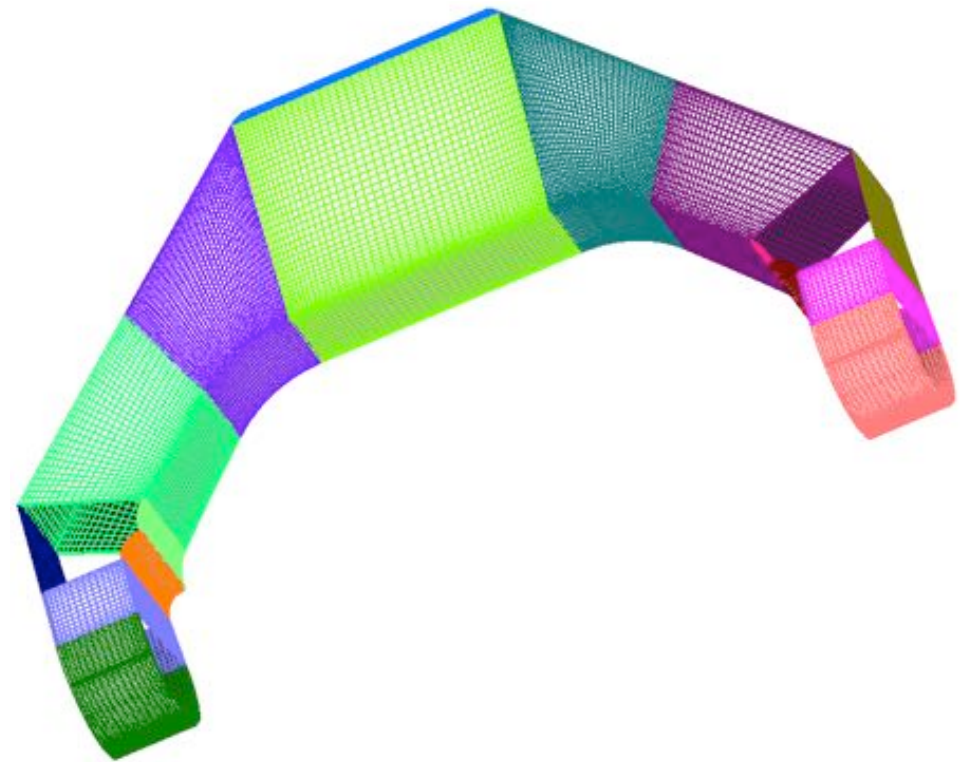
After concatenation: 16 patches

ALGEBRAIC STEP
7. Perform TFI Extension at Boundaries and Identify Initial Curves for Hyperbolic Marching (in progress)
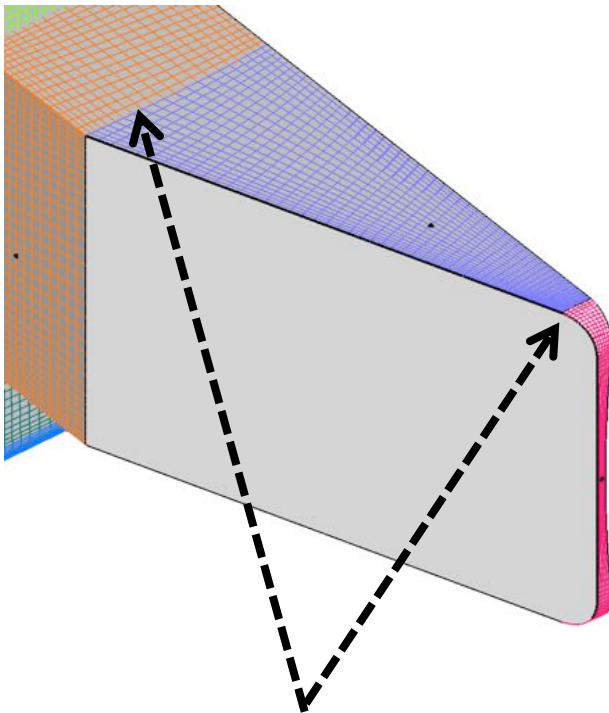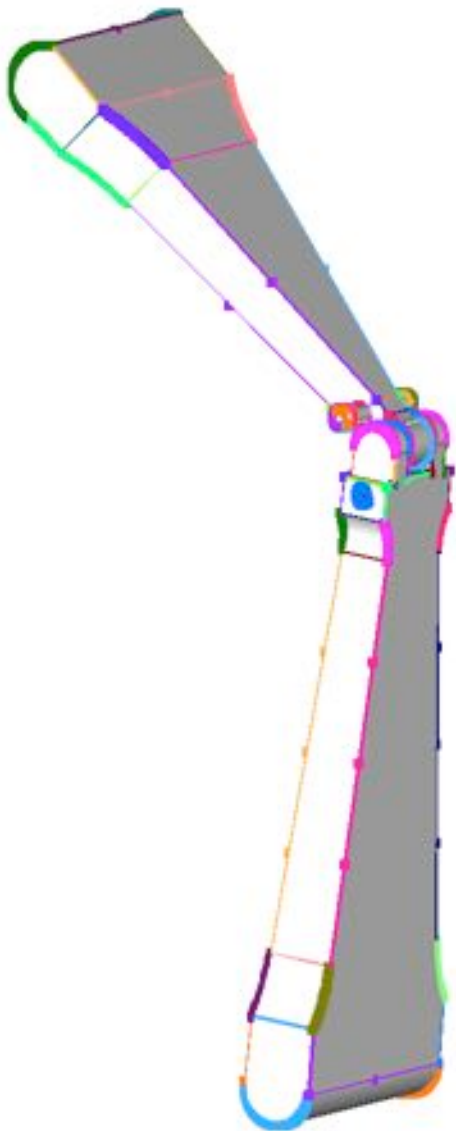

Abutting patch boundaries after TFI
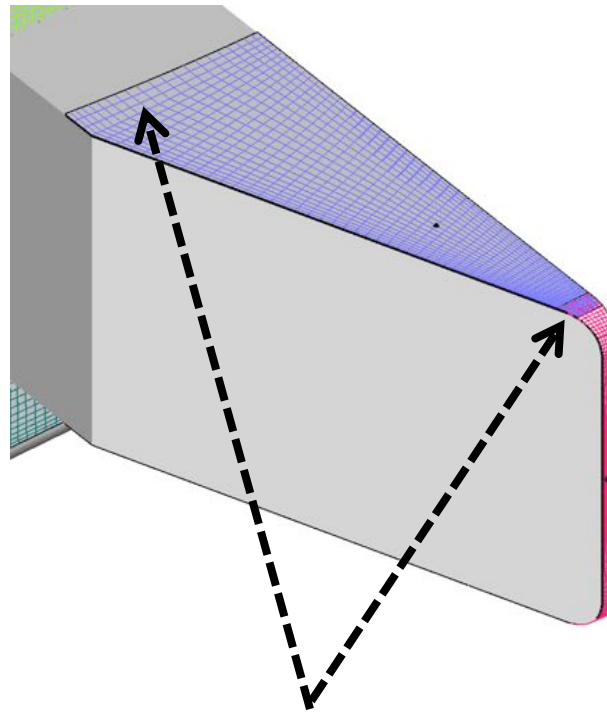
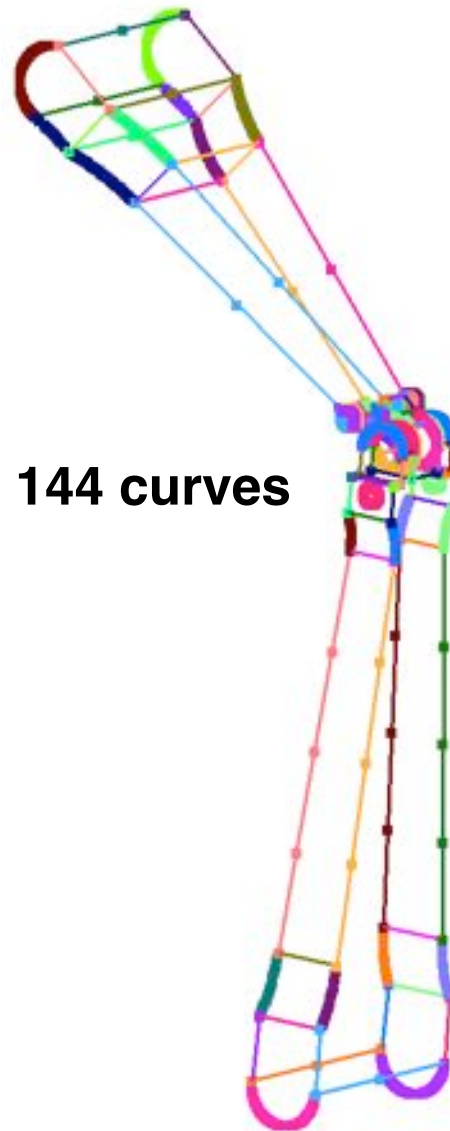**TFI extension at boundaries not on sharp edge**
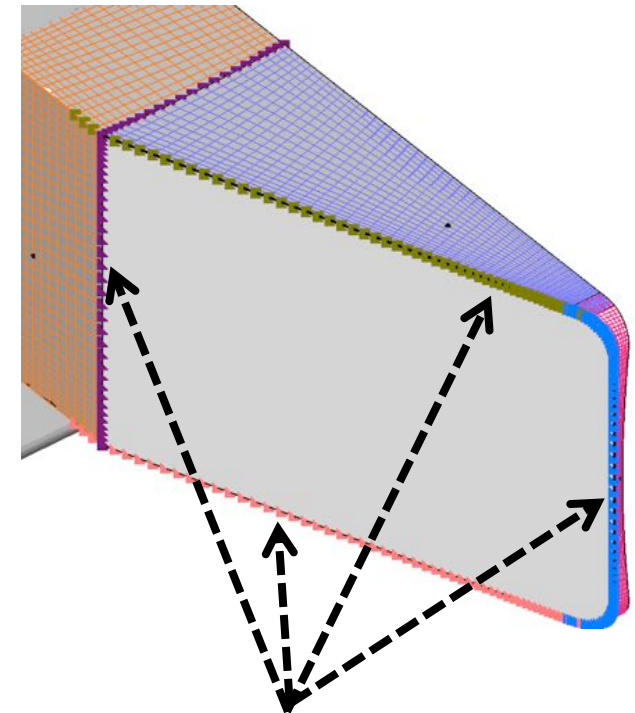
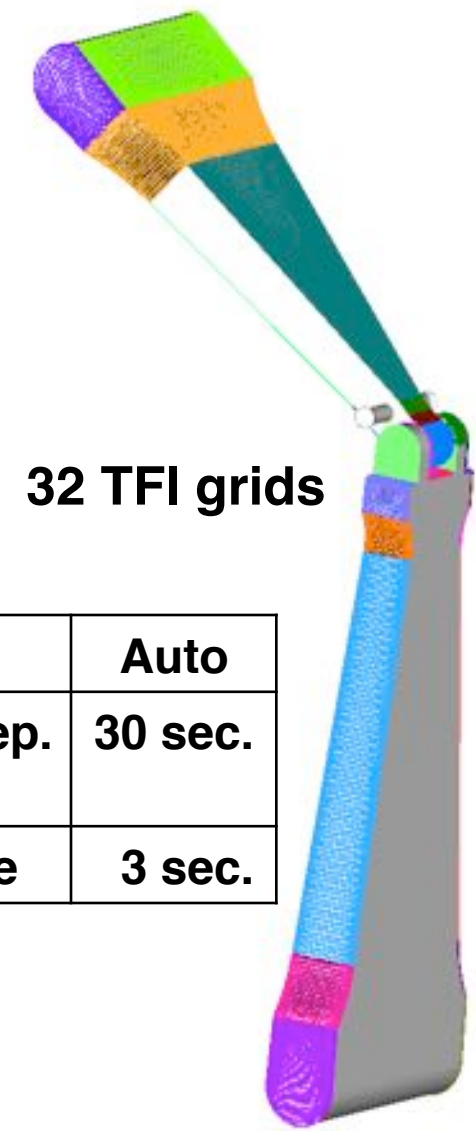**Initial curves on exposed boundaries for hyperbolic marching**

TEST CASES
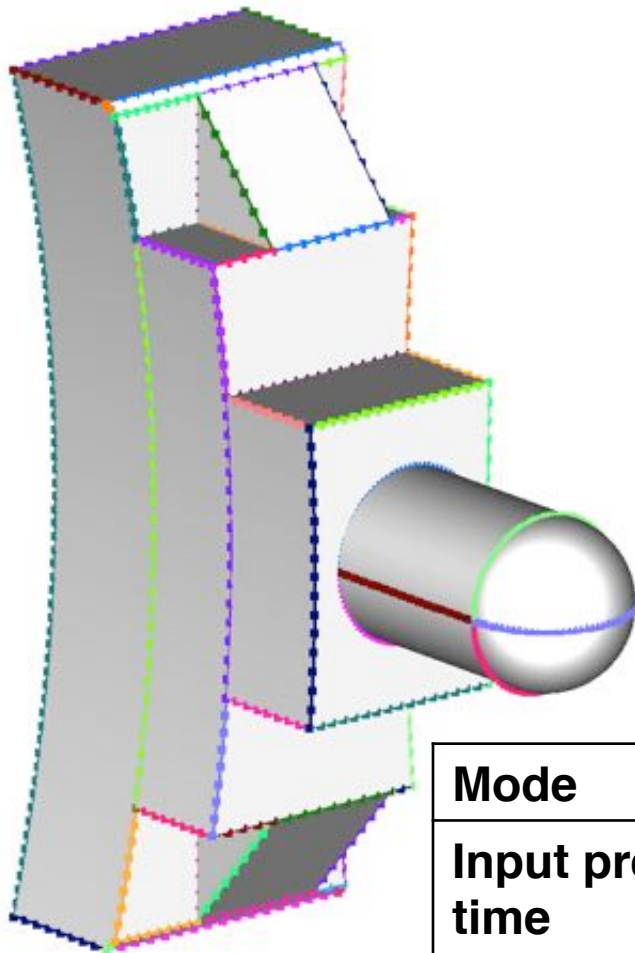# Landing Gear Strut

144 curves

32 TFI grids

Mode	Auto
Input prep. time	30 sec.
CPU time	3 sec.

TEST CASES
Attach Hardware

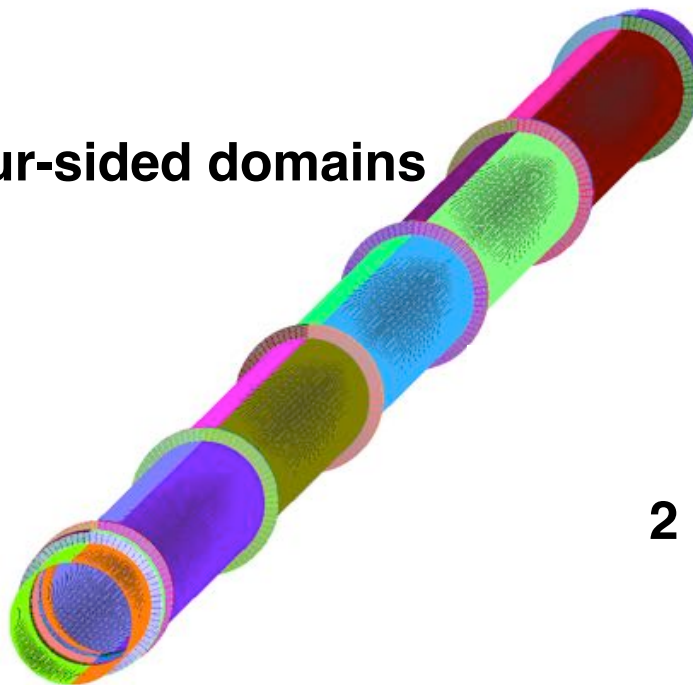| Mode	Auto
Input prep. time	30 sec.
CPU time	2.5 sec.
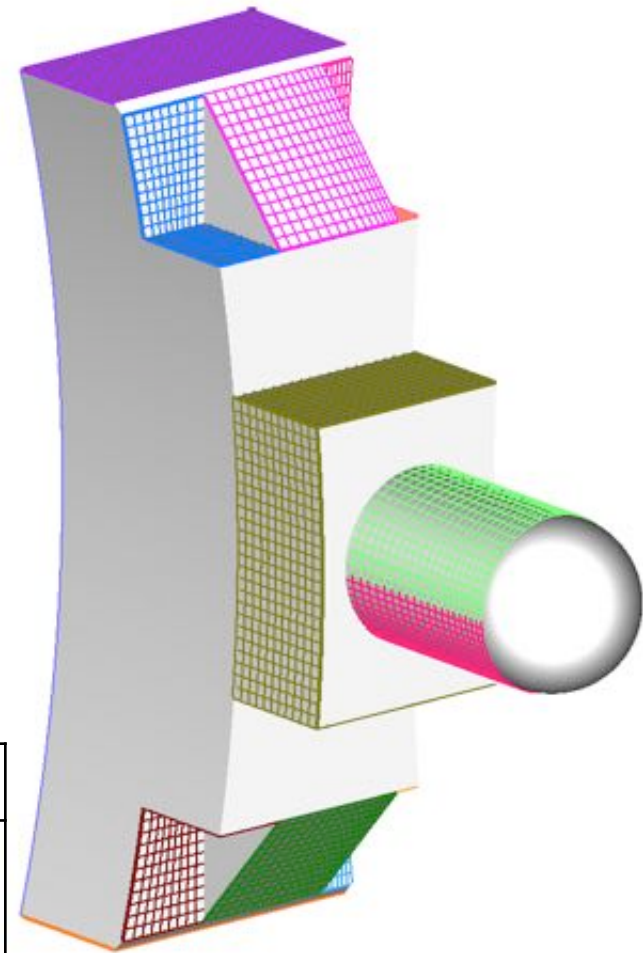
73 curves

14 TFI grids

TEST CASES
Feedline

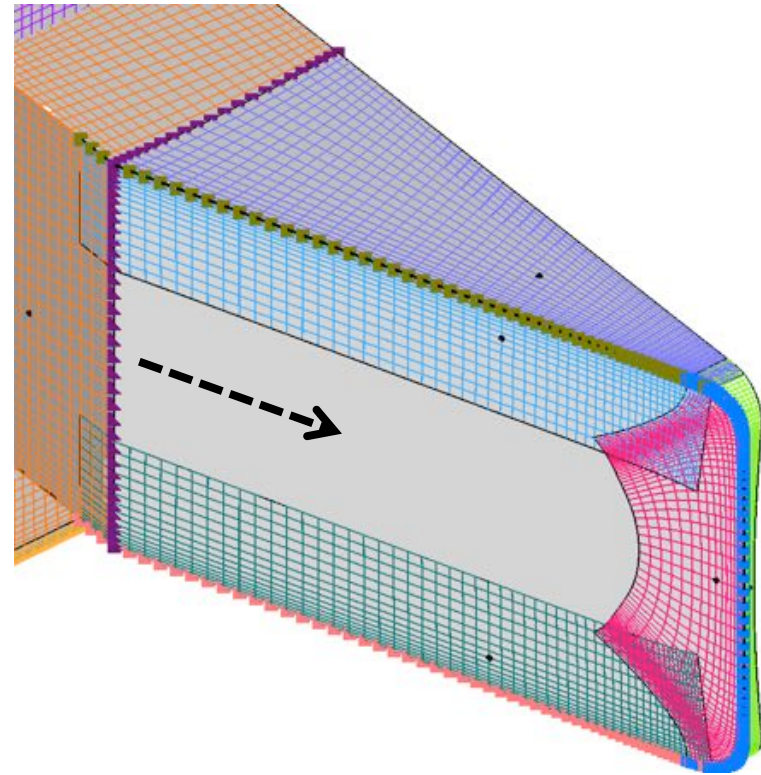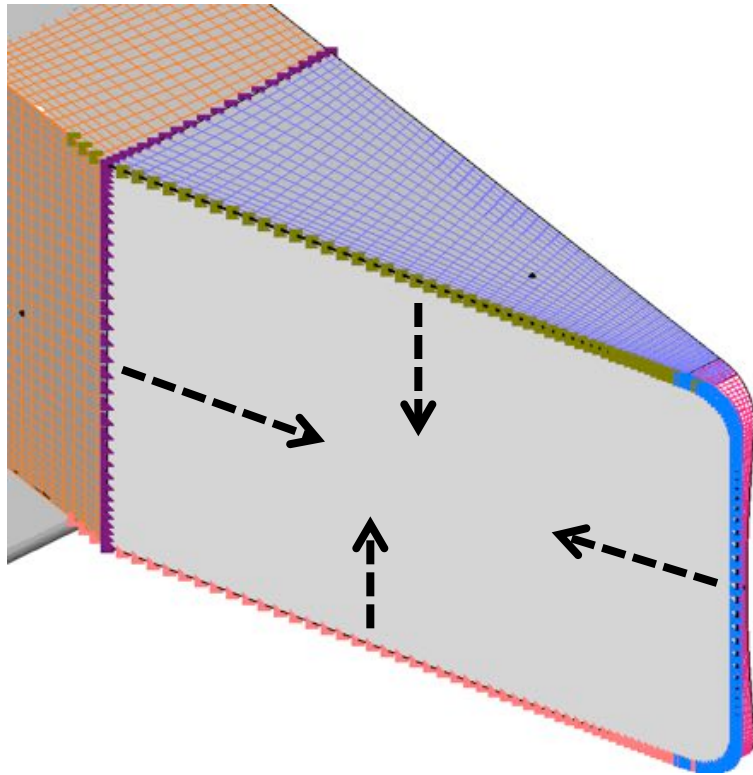**126 curves**
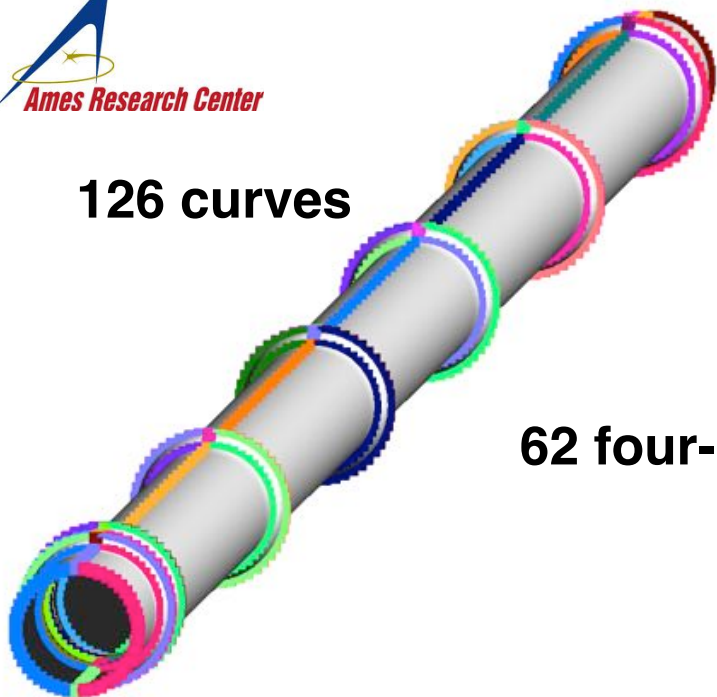
62 four-sided domains

2 TFI grids
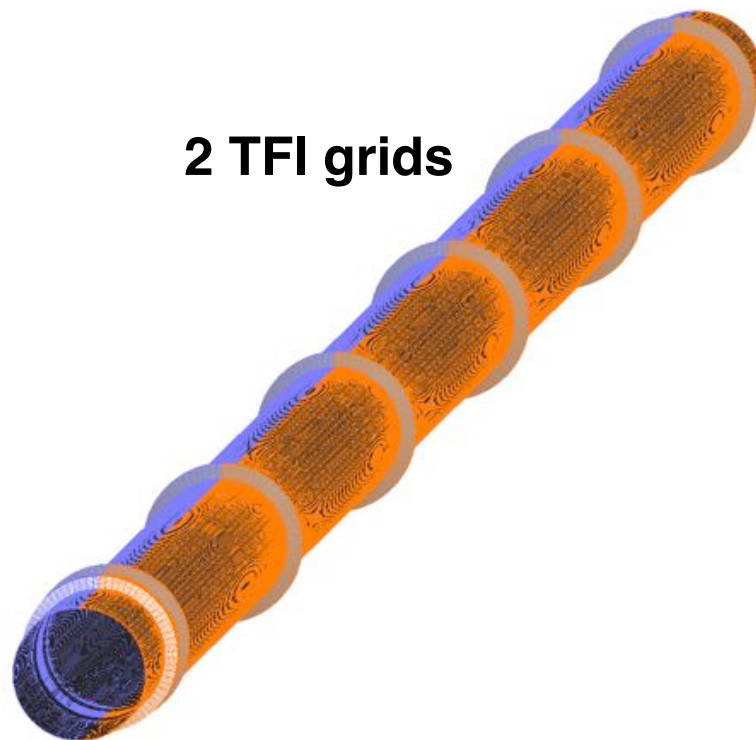
Mode	Manual	Auto
Input prep. time	30 – 60 min.	30 sec.
CPU time	2 sec.	2.5 sec.

FUTURE PLANS FOR HYPERBOLIC STEP

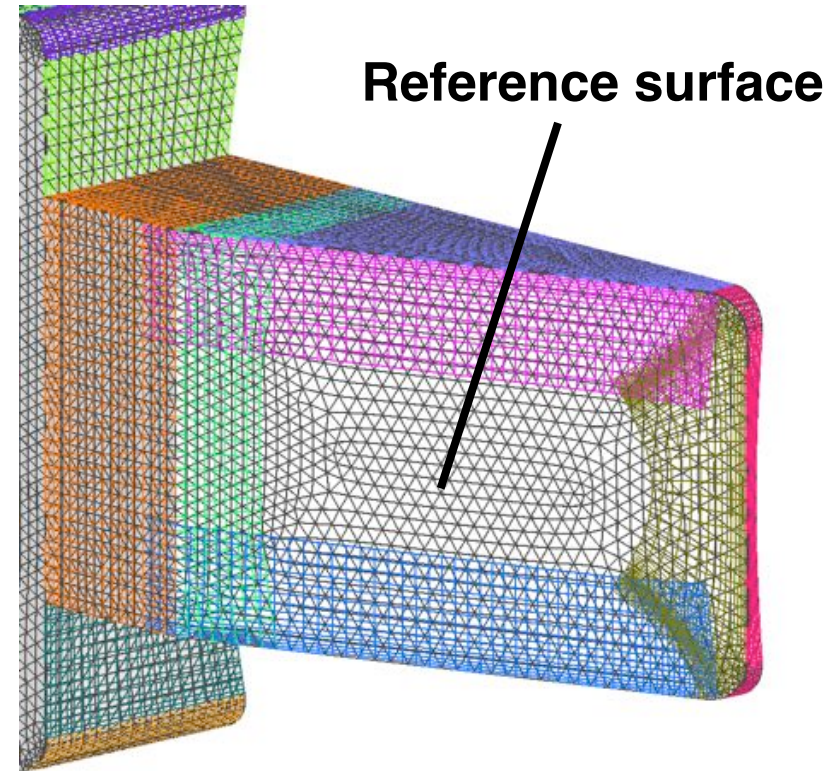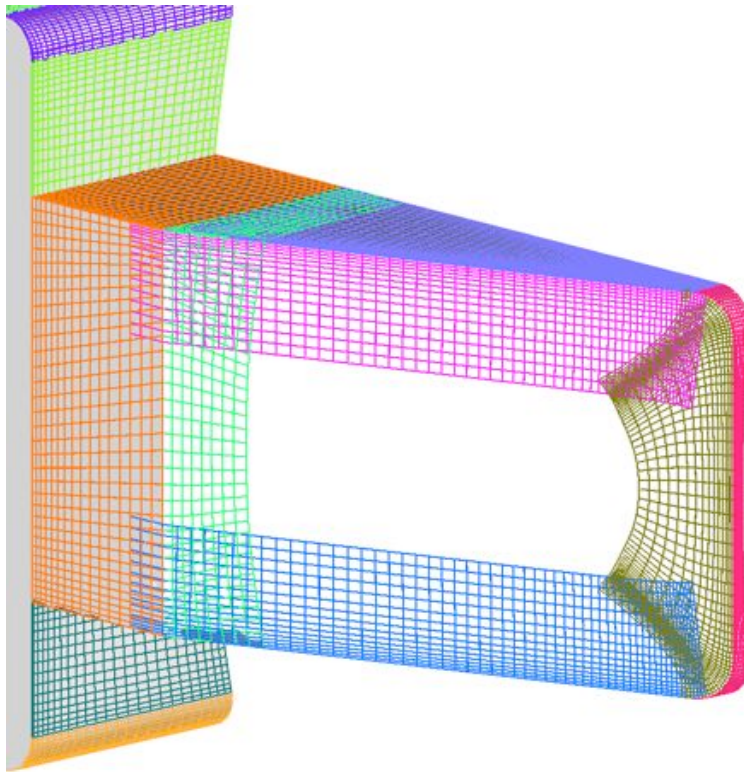**Auto extract initial curves for hyperbolic marching**
 - Exposed boundary curves from algebraic step
 - Unused original curves after algebraic step

Auto determine spatially variable marching
 - distance
 - direction

FUTURE PLANS FOR GAP FILLING STEP

Reference surface

- Gap regions (if there are any) are not bounded by surface features
- Project reference surface vertices onto surface grids from algebraic and hyperbolic steps
- Identify vertices that fail to find donor stencil (orphan vertices)
- Identify orphan cells from orphan vertices on reference surface
- Build algebraic/hyperbolic grids over orphan cells

SUMMARY AND CONCLUSIONS

- A high level procedure has been designed for automation of overset structured surface grid generation
- Geometry starting point
  - unstructured surface triangulation representation
  - a set of surface feature curves that tesselates surface
- Approach: algebraic, hyperbolic, gap-filling steps
- Algebraic step
  - Auto curves network connectivity
  - Auto grid point distribution
  - Auto grid generation on four-sided domains
  - Auto concatenation sweep
  - Auto boundary extension (work in progress)
  - Significant time saving over current manual practice
- Future plan: automation of hyperbolic and gap filling steps