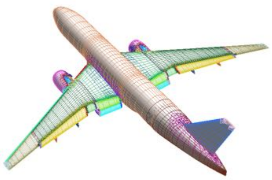
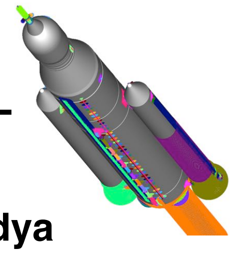
Ames Research Center

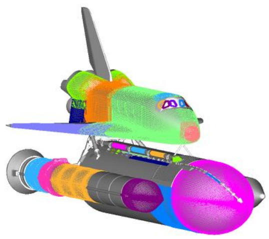


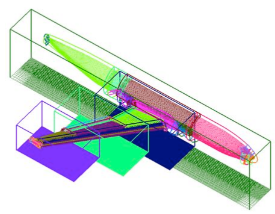
CHIMERA GRID TOOLS TUTORIAL

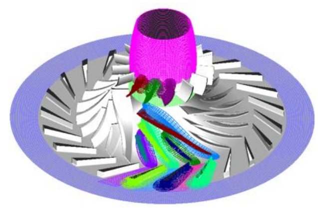


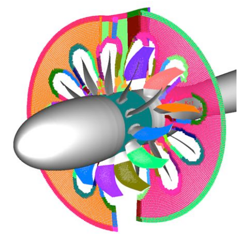
William M. Chan and Shishir A. Pandya

NASA Ames Research Center









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

TUTORIAL OVERVIEW

- Chimera Grid Tools (CGT)
  - Introduction
  - Pre-processing
  - Post-processing

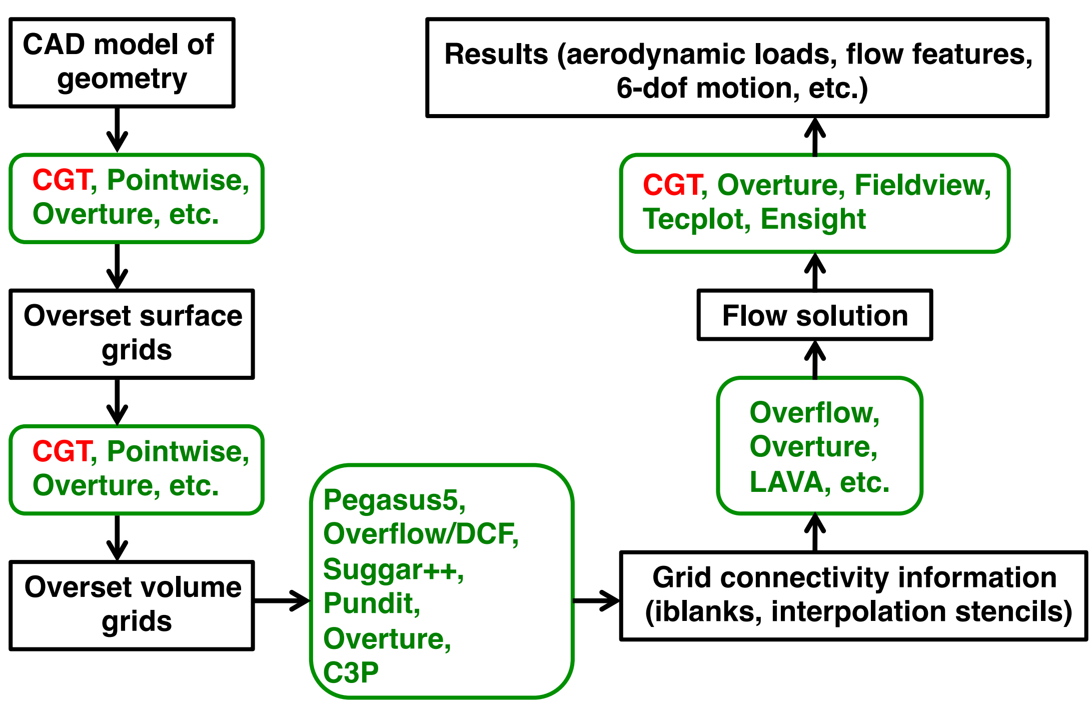
- Chimera Components Connectivity Program (C3P)

- Pre-Processing Script Creation (grid generation, input preparation for domain connectivity, flow solver, loads integration)
  - High Lift CRM (Component Centric Approach)
  - Generic Rocket (Grid Centric Approach)

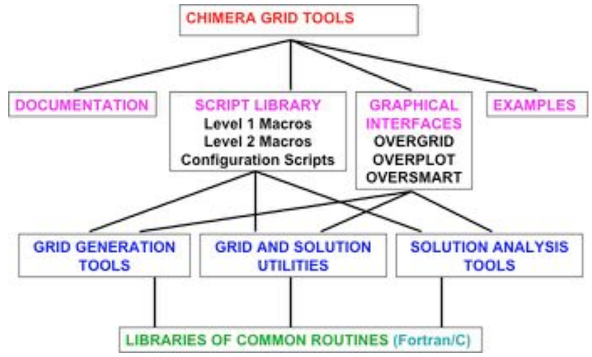
A more detailed OVERGRID demo is available at:
The OVERGRID Graphical User Interface in Chimera Grid Tools (Parts 1, 2, 3)
http://www.nas.nasa.gov/publications/ams/2014/05-13-14.html
http://www.nas.nasa.gov/publications/ams/2014/05-20-14.html
http://www.nas.nasa.gov/publications/ams/2014/05-29-14.html

OVERVIEW OF CHIMERA GRID TOOLS

TYPICAL MODELING AND SIMULATION PROCESS USING OVERSET GRIDS



CHIMERA GRID TOOLS (CGT) Version 2.1



What is CGT

- A collection of software tools for pre- and post-processing of overset grid CFD computations

Authors

- William Chan, Stuart Rogers, Shishir Pandya, David Kao, Pieter Buning, Robert Meakin, David Boger, Steve Nash

Availability (https://software.nasa.gov/software/ARC-16025-1A)

- U.S. citizens/permanent residents working under U.S. organization in the U.S.
- Fill out and return Software Usage Agreement form
- Source (Linux, Unix, Mac OS-X)
- Executables (Mac 10.5, Windows-XP)
- Version 2.1+ available for use by authors' associated projects

INSTALLATION, DOCUMENTATION, TUTORIALS

Installation software requirements

- Fortran 90 compiler (ifort, gfortran 4.4+, pgf90)
- C compiler (gcc, icc, pgcc)
- OpenGL, X11, Tcl/Tk libraries (OVERGRID)
  Tcl/Tk 8.5.8 or earlier for CGT 2.1
  Tcl/Tk 8.5.19 / 8.6.6 or earlier for CGT 2.1+
- Python, swig, matplotlib package or gnuplot (OVERSMART)
- Tcl wish, xmgrace or gnuplot (OVERPLOT)

Installation instructions

- chimera2.1/doc/{INSTALLATION.html, overgrid.html}

Documentation

- chimera2.1/doc/man.html

Recommended tutorials

- chimera2.1/gui/tutorial/* (OVERGRID)
- chimera2.1/examples/scriptlib/* (CGT script library)

EXECUTABLES

Run configure script to generate Makefiles

configure -- help **(get list of options)**

Executables
- **single precision**
- **double precision**

} ~ 60 independent pre/post processing tools

- **og (overgrid executable)**
- **smart.so (oversmart shared library)**

Big/Little Endian Unformatted File I/O
- **controlled by environment variable (ifort, gfortran)**
- **controlled by compiler flag (pgf90)**
- **conversion using p3dConvert or overConvert**

OVERGRID can auto-detect single/double precision, big/little endian

Script Library – about 230 Tcl procedures

PRE-PROCESSING STEPS AND BEST PRACTICE

Task: Given complex geometry definition, create grids and input files needed for overset grid CFD analysis
- **Grid file containing overset volume grids and iblanks**
- **Input file for**
  - **Domain connectivity program**
  - **Flow solver with boundary conditions for each grid**
  - **Forces and moments integration on components of interest**
  - **Coupled physics**
    - **Prescribed/6-DOF dynamics for relative motion problems**
    - **Species convection**
    - **Structural deformation**

Best practice:
- **Develop pre-processing script to create all input files needed above**
- **Use CGT's OVERGRID to check and visualize individual steps**
- **Use CGT's Script Library to record steps into script**
- **Use small number of independent parameters (e.g., pick Δs_{max}, express Δs at curve end points as $\kappa\ \Delta s_{max}$, $0 < \kappa < 1$)**
- **Check grid quality (individual and collection) using variety of tools**

PRE-PROCESSING USING CGT

Geometry Creation and Manipulation

Surface Grid Generation
- on triangulation or CAD
- algebraic, hyperbolic

Volume Grid Generation
- near-body curvilinear (hyperbolic)
- off-body Cartesian

Domain Connectivity Inputs
- Xray map creation and hole-cut instructions
- PEGASUS5
- C3P

Flow Solver Inputs (OVERFLOW)
- boundary conditions
- component hierarchy and prescribed/6-DOF dynamics
- prescribed dynamics animation (overgrid)

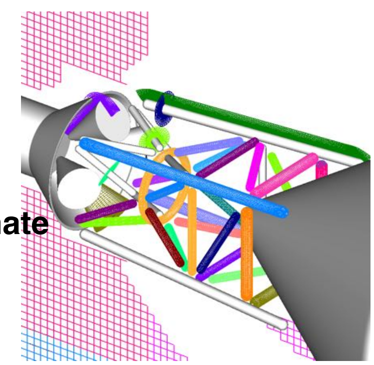
GEOMETRY CREATION

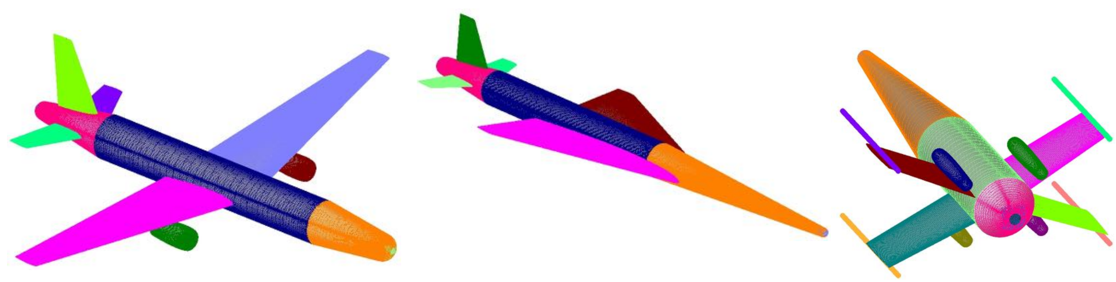
Script Library has macros to create

- Points
- Straight lines
- Analytic curves
- Cylinders
- Frustums
- Cartesian boxes
- Airfoil shapes
- > NACA 4 and 5 digit series
- > PARSEC (CGT 2.1+)

Combine with basic macros to generate more complex shapes

- Translate
- Scale
- Rotate
- Mirror
- Extract
- Concatenate
- Revolve
- Duplicate





GEOMETRY INPUT

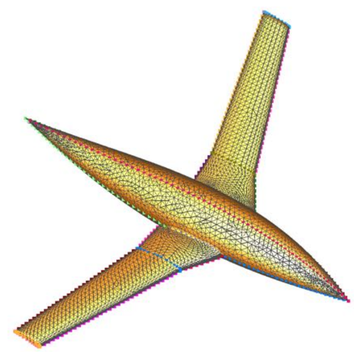
Native CAD (Pro-E, Catia V5, Parasolid, OpenCASCADE, SolidWorks, UniGraphics, FELISA, STEP, IGES)

- Create surface triangulation representation and surface curves using
  - Commercial software: ANSA, Pointwise
  - CAPRI library (CADNexus) as interface to convert native CAD parts, need CAD license and CAPRI users license
  - EGADS (open source from Bob Haimes) as interface
- CGT surface grid generator has option to project back to original CAD but usually a fine surface triangulation is sufficient

Surface Triangulation

- CART3D (.tri, .triq) (.trix in CGT 2.1+)
- UCD (.ucd)
- FAST (.fst)
- STL (.stl), FRO (.fro) (CGT 2.1+)

Structured Surface Grids (PLOT3D format)

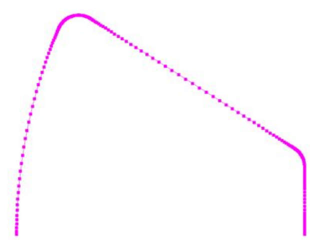


MOST COMMONLY USED GRID TOOLS

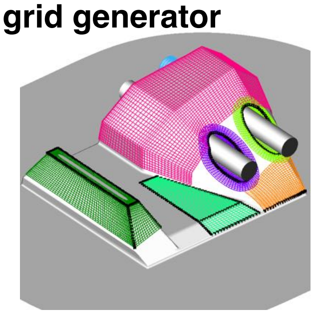
GRIDED – Structured grid editing tool with ~ 40 functions for structured grid manipulation and processing

TRIGED – Unstructured surface triangulation editing tool with ~ 30 functions for triangulation manipulation and processing

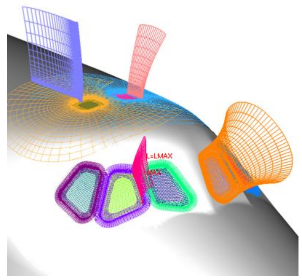
SRAP – Grid point redistribution



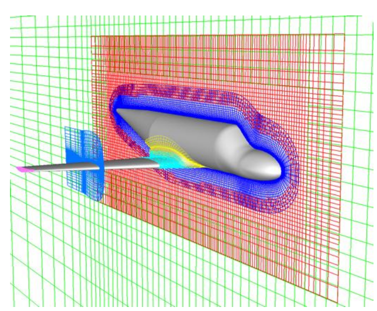
SURGRD – Hyperbolic and algebraic surface grid generator



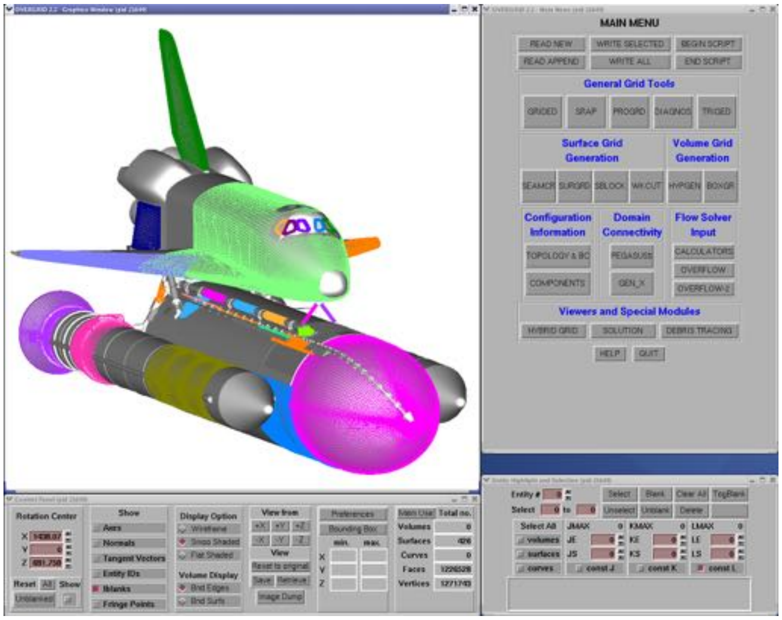
HYPGEN – Hyperbolic volume grid generator



BOXGR – Cartesian volume grid generator



OVERGRID



- CAD interface via CAPRI
- Geometry/grid processing (structured quads, triangulations)
- Grid processing, redistribution, projection
- Surface and volume grid generation (TFI, hyperbolic, Cartesian)
- Hole cutter generation
- Grid diagnostics
- Flow solver inputs and b.c. preparation
- Multi-component dynamics input/animation
- Standard atmosphere, mass properties, 6-dof input calculators
- Simple solution viewer
- Debris trajectory inputs
- Strand/AMR Cartesian grid viewer

Supported platforms – Linux, Mac OS-X,Windows-XP

CGT SCRIPT LIBRARY

~ 230 Tcl macros, 10x more compact, > 3x faster development time

Low – Mid Level

- File manipulation (e.g., combine files, format conversion,…)
- Geometry creation (e.g., points, lines, analytic curves, cylinders,...)
- Grid information (e.g., interrogate grid dimensions, coordinates, arc lengths, formats,…)
- Grid editing (e.g., extract, concatenate, split, duplicate, swap/ reverse indices, scale, translate, rotate, mirror, revolve, …)
- Grid redistribution
- Surface grid generation (TFI and hyperbolic)
- Volume grid generation (hyperbolic and Cartesian)
- X-ray hole cutter generation and hole cut instructions creation
- Pegasus5 and C3P input preparation
- Force/moments computation inputs
- OVERFLOW boundary conditions inputs and namelist i/o

Top Level

- Grid centric approach
- Component centric approach

PRE-PROCESSING STRATEGY USING SCRIPTS

Scripting approach
- Rapid replay of all steps
- Easy to parameterize inputs (e.g., grid stretching, spacings, etc.)
- Easy to make small changes
- Recommended even for one-of-a-kind cases
- Modification needed if surface topology changes

Surface Grid Generation
- Generate grids from surface triangulation geometry representation and surface curves derived from native CAD, STEP or IGES

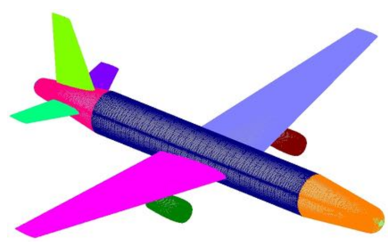
Volume Grid Generation
- Near-body: hyperbolic grids
- Off-body: curvilinear wake/plume grids, Cartesian box grids

Domain Connectivity, Force/Moments Computation, Flow Solver Inputs
- Construct and store common database in script (boundary conditions, component definitions, etc.)

SCRIPT DEVELOPMENT USING WORKING TEAM

- Identify components of a complex configuration where a component is a geometric part modeled by one or more grids

- Create stand-alone script for each component
  - Generation of surface and volume grids
  - Domain connectivity inputs (X-ray maps)
  - Solver boundary conditions
  - Forces and moments integration inputs



- Each team member responsible for one or more components, creates script and establishes grid connectivity within component

- Use file repository system to update script so that each team member can get most up-to-date version of each script

- Share global parameters file (e.g., wall spacing, stretching ratio,…)

- Create master script to call component scripts, assemble final grid system, generate input files for domain connectivity, force/moment integration, flow solver

POST-PROCESSING USING CGT

Forces and Moments Computation (mixsur/overint, usurp)

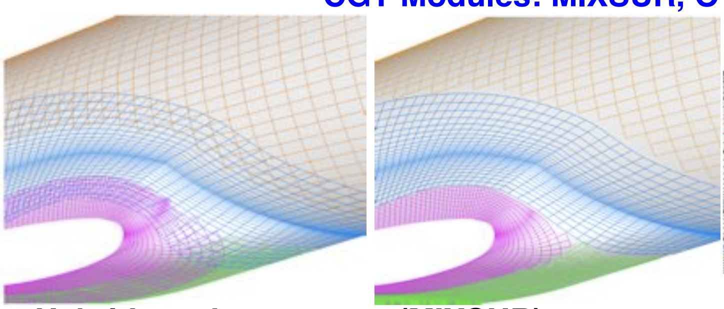
Solution Convergence Analysis
- Solution/turb. model residuals, forces/moments
- One page overview (oversmart)
- Individual plots (overplot)

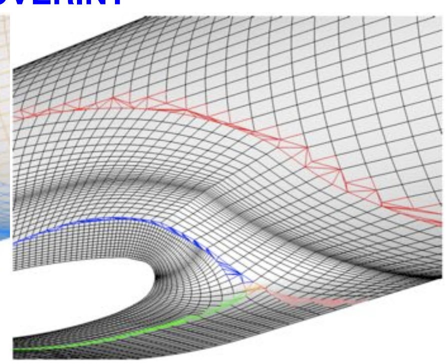
Flow Visualization (overgrid)
- Scalar and vector functions
- Turb. model dependent variables, species partial densities
- Unsteady 2-D flow and dynamics animation

Component Line Loads (triload)
- Cumulative line loads
- Sectional Cp

Dynamics Animation (overgrid)
- 6-DOF dynamics output from flow solver

FORCES/MOMENTS INTEGRATION APPROACH 1 – INTEGRATE ON HYBRID SURFACE MESH CGT Modules: MIXSUR, OVERINT





Hybrid mesh generator (MIXSUR)
- Automatically blank overlapping quads
- Automatically fill narrow gap with triangles
- Very fast but may sometimes contain a few bad triangles
  (690 surface grids, 3 million+ surface points, 39 sec., 1 proc.)

Integration tool (OVERINT)
- Integrates on non-overlapping quads and triangles
- Integrates linear function exactly

Chan, W. M., Enhancements to the Hybrid Mesh Approach to Surface Loads Integration On Overset Structured Grids, AIAA Paper 2009-3990

OVERINT LOCAL LOADS OUTPUT FILES

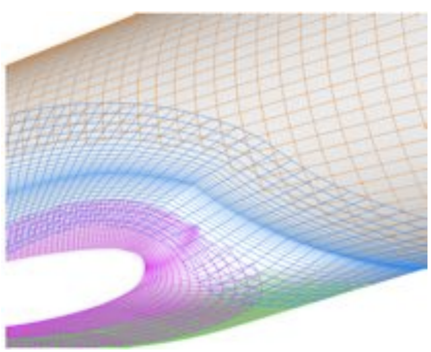
- Surface distributions of local forces and moments

- Four unstructured surface triangulation files, each with cell-centered scalar variables (extended CART3D .i.tri format)

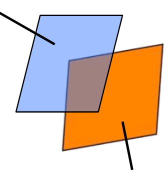
  (1) Cell ΔF
  (2) Cell ΔF / Cell area
  (3) Cell ΔM
  (4) Cell ΔM / Cell area

- Scalars: X, Y, Z components of forces/moments
  total magnitude, pressure, viscous, momentum contributions
  local cell area

FORCES/MOMENTS INTEGRATION APPROACH 2 – INTEGRATE ON WEIGHTED QUADS
CGT Module: USURP



$W_1 = 1$



Polygon subtraction in 3-D

$W_2 = (A_Q - A_{OV})/A_Q$

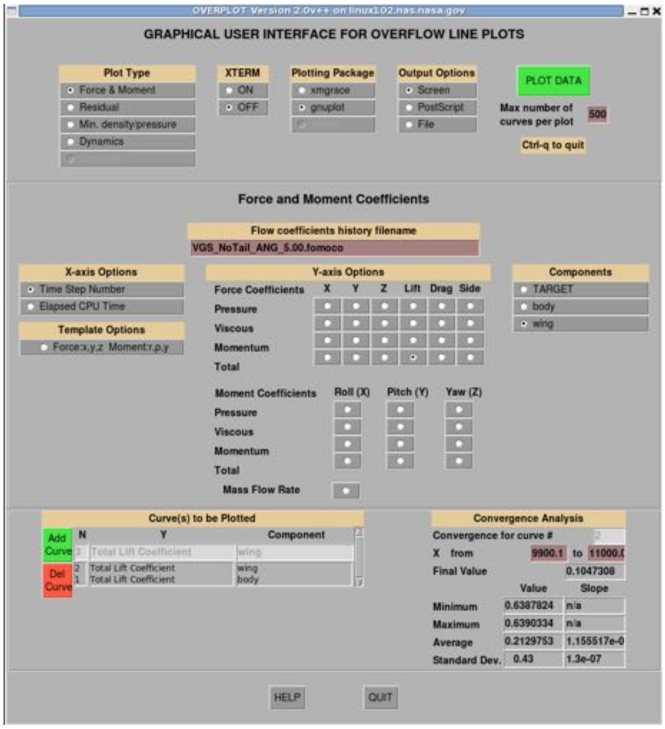
A_Q = Area of quadrilateral
A_{OV} = Area of overlap

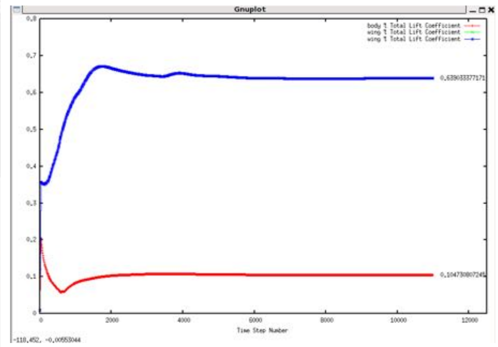
Quad panel weights calculator and integrator (USURP)
- Automatically computes panel weight for each quad
- Always returns a result by integrating over all quads
- No hybrid mesh => no visual checks
- Does not integrate linear function exactly
- Also has standalone and OVERFLOW modes

Boger, D. and Dreyer, J., Prediction of Hydrodynamic Forces and Moments for Underwater Vehicles Using Overset Grids, AIAA Paper 2006-1148

SOLUTION CONVERGENCE: OVERPLOT

Forces/Moments Panel (.fomoco)





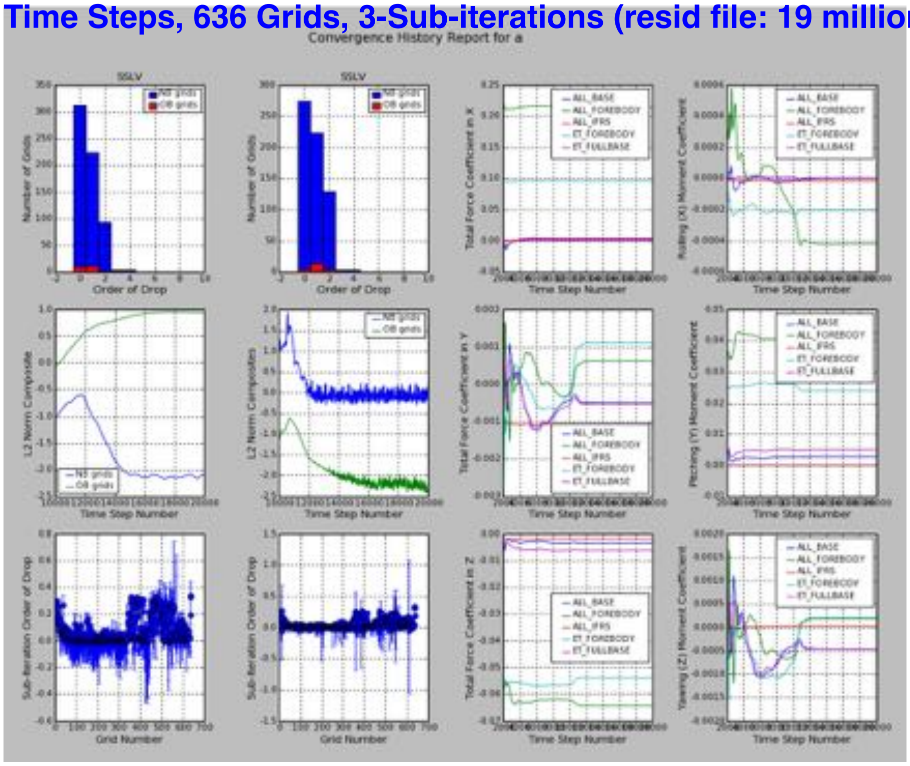
- Single coefficient plot with option to add more coefs.

- Six coefficients matrix plot (Fx, Fy, Fz, Mx, My, Mz)

SOLUTION CONVERGENCE: OVERSMART SUMMARY PAGE

Space Shuttle Launch Vehicle

10,000 Time Steps, 636 Grids, 3-Sub-iterations (resid file: 19 million lines)



SOLUTION VISUALIZATION

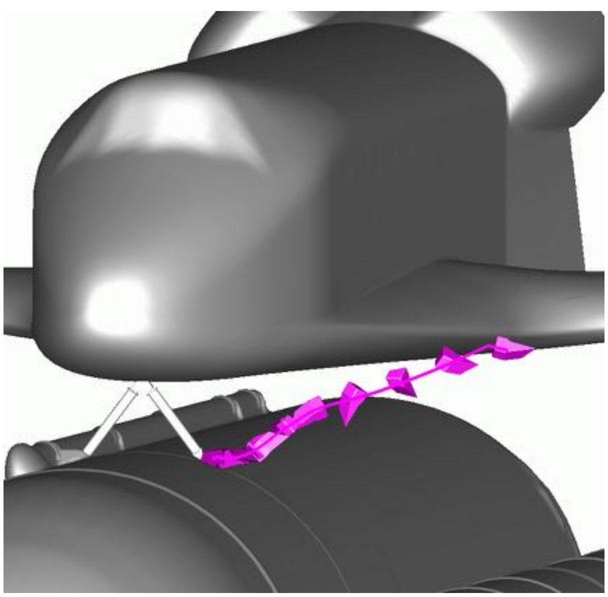
- 6-DOF component trajectories

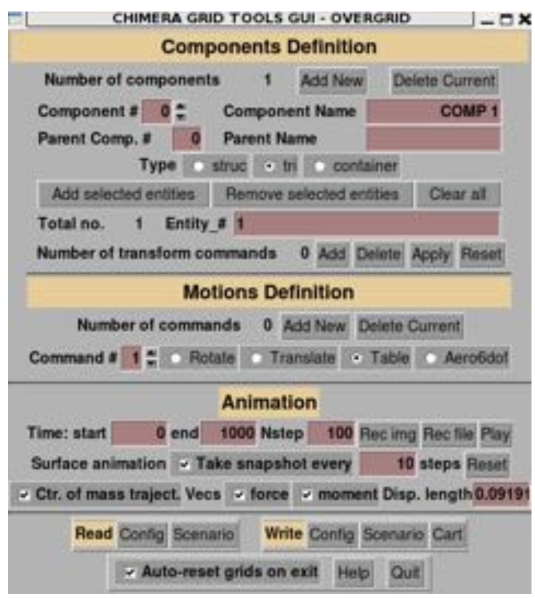
- Flow variables

  - Surface triangulations
    - vertex and cell-centered scalars

  - Overset structured surface and volume grids
    - steady (scalars and vectors)
    - unsteady (scalars)
    - 2-D moving body with adaptive grids (scalars)

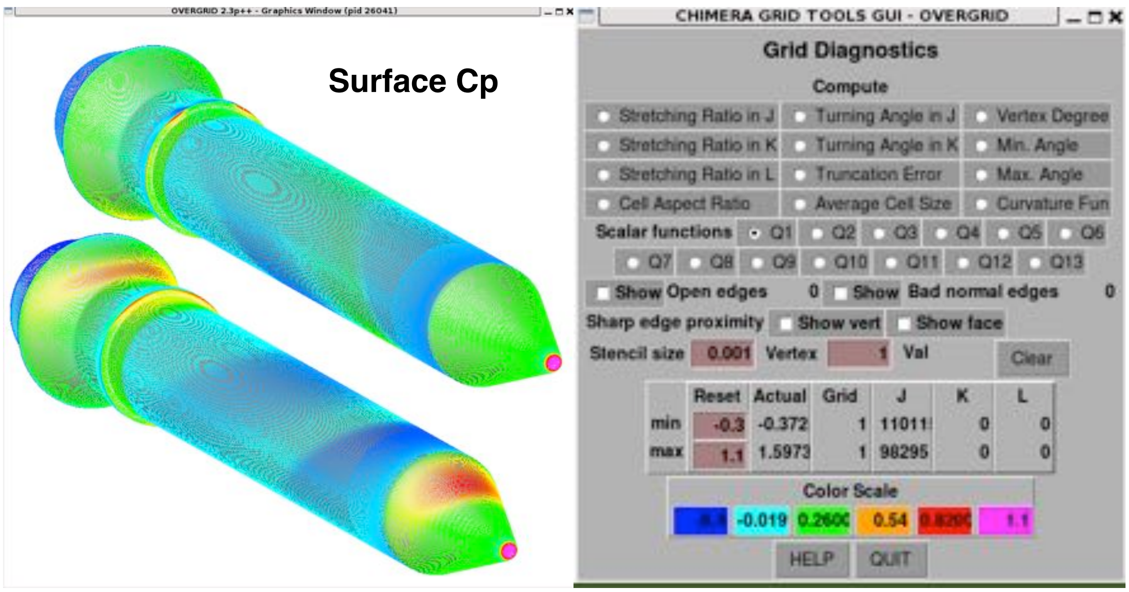
COMPONENT TRAJECTORIES VISUALIZATION FROM SIX-DOF COMPUTATIONS
(OVERGRID module)





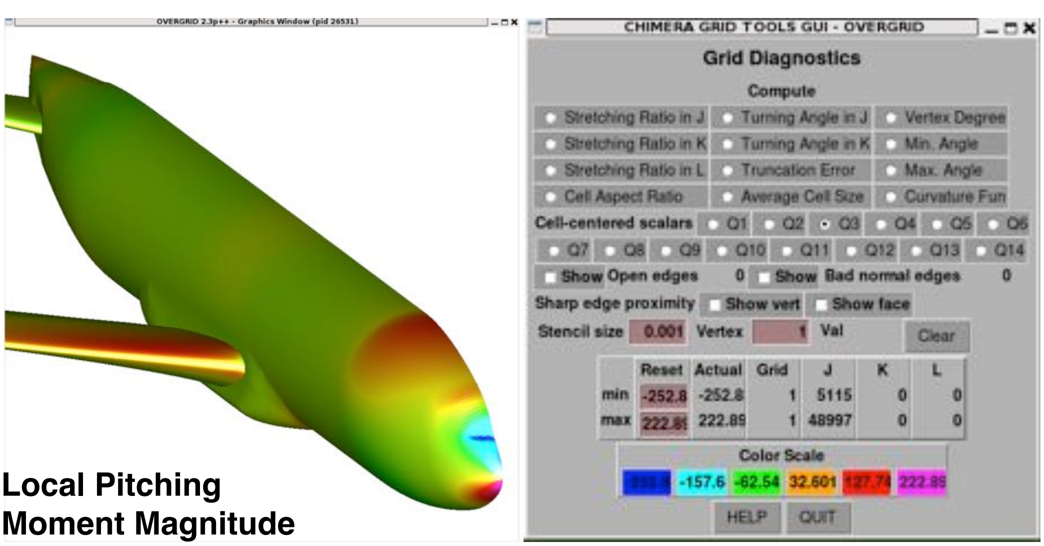
VISUALIZATION OF VERTEX-CENTERED DATA ON SURFACE TRIANGULATIONS

Standard CART3D triq file



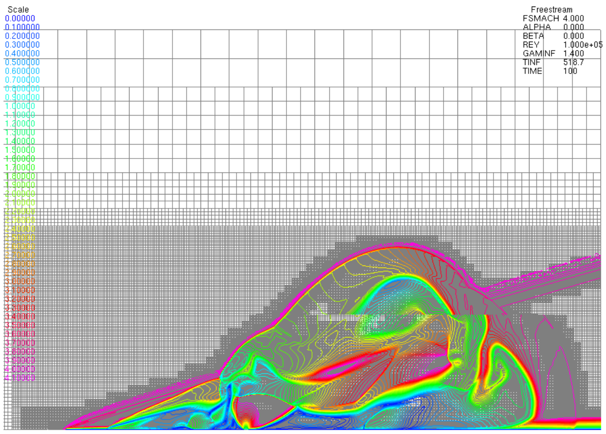
VISUALIZATION OF CELL-CENTERED DATA ON SURFACE TRIANGULATIONS

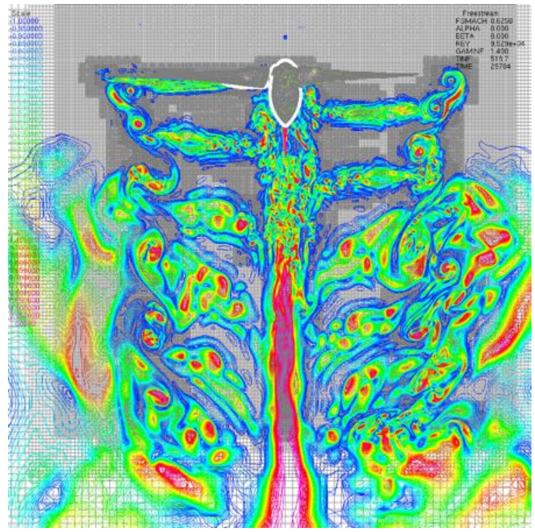
Recent addition: Extended CART3D tri file with cell-centered scalars
Local forces/moments tri file output from OVERINT



Local Pitching Moment Magnitude

SOLUTION VISUALIZATION



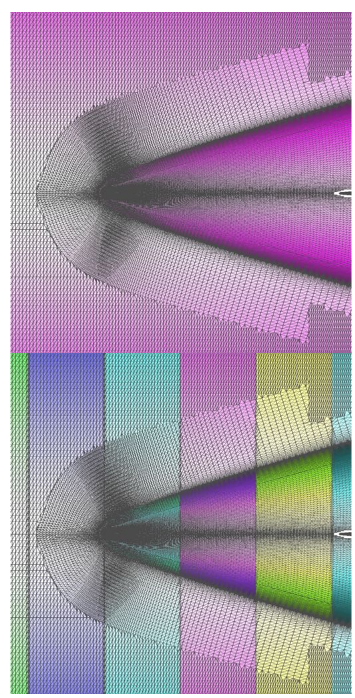


2-D with unsteady adaption

3-D with unsteady adaption

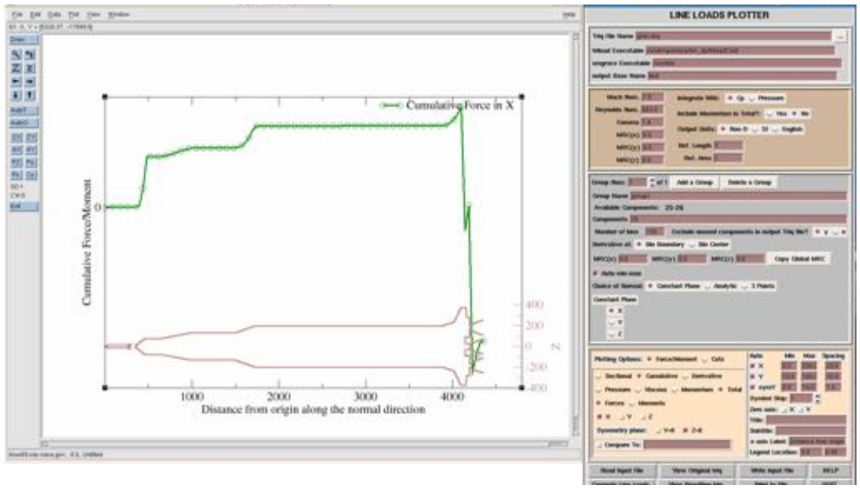
TRILOAD: LINE LOADS

- **Extract: surface triangulation (with solution) from structured, unstructured or overset to compute sectional loads.**
- **Slice-up: the input triangulation.**
- **Interpolate: solution data to newly introduced vertices.**
- **Integrate: forces/moments on each slice.**
- **Compute: Sectional loads, Cp vs. X along sectional boundaries.**



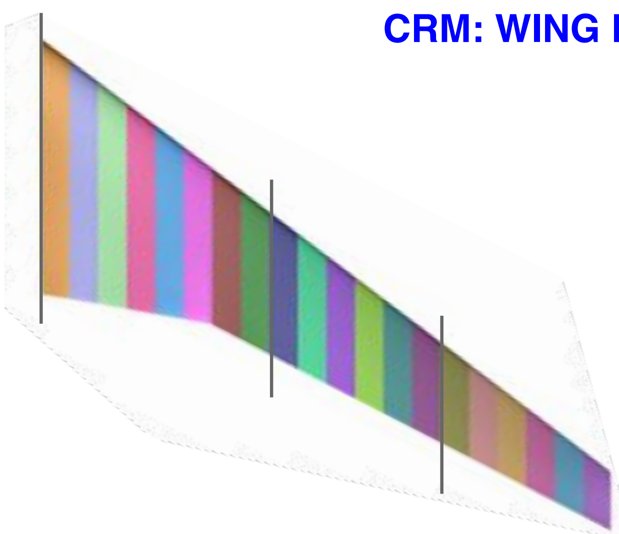
TRILOAD USE

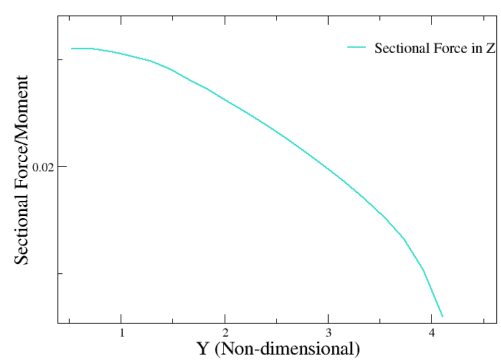
- **Command line: Development, debugging.**
- **GUI: For single, interactive run (setup).**

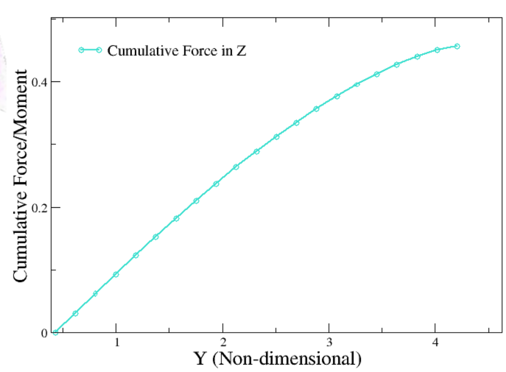


- **Script: Post-processing a database of solutions.**

CRM: WING LOADING

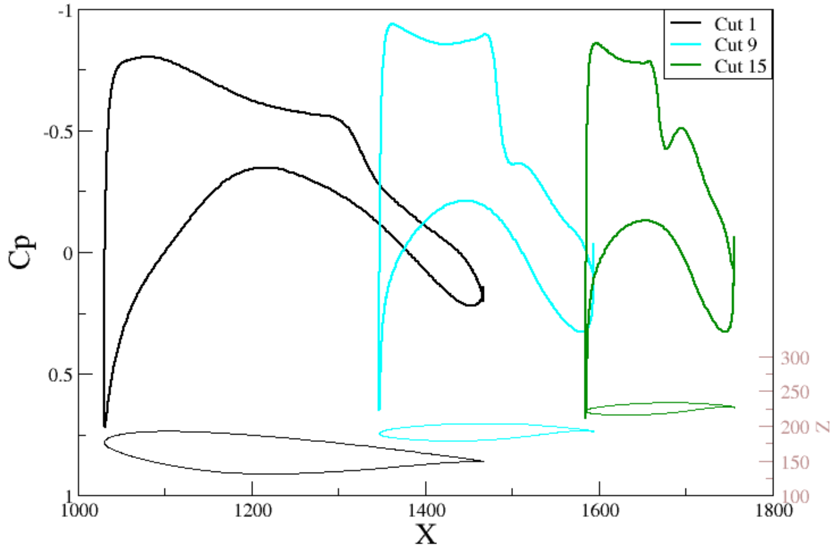






- **Common Research Model (CRM)**
  - **Drag Prediction Workshop**

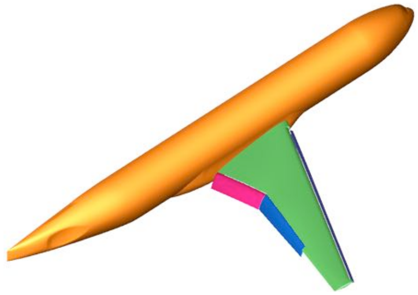
CRM WING: CP VS. X



OVERVIEW OF CHIMERA COMPONENTS

CONNECTIVITY PROGRAM (C3P)

CHIMERA COMPONENTS CONNECTIVITY PROGRAM (C3P)



- Perform domain connectivity for a set of structured overset surface or volume grids

- Components are defined from a set of grid subsets or input triangulations

- Each near-body grid wall subset is associated with a component

- Component hole cutters can cut
  - grid points associated with other components
  - grid points in any off-body grid
  - grid points away from the surface of its own component

C3P ALGORITHM

- Auto construct triangulation with auto closure of open boundaries if necessary for each defined component

- Construct minimum hole using standard X-ray map of each component with exact ray cast test for points near the surface

- Construct Cartesian map of component wall distance function using accurate computation near surface and fast sweeping scheme away from surface

- Use heuristic rules to perform hole boundary offset from minimum hole for near-body grids

- Use donor stencil map of near-body grids to offset hole boundaries for off-body grids

- Perform orphan points removal iterations by perturbing hole boundaries

C3P INPUT AND OUTPUT

Input

- Structured surface or volume grid file (PLOT3D)
- Input parameters file
  - Global (symmetry, number of fringe layers, …)
  - Boundary conditions for each grid (same as flow solver with additional tag to indicate component name for each solid wall)
  - Special inputs for components or grids to override defaults

Output

- PLOT3D grid file with iblanks (spatially variable hole cut)
- Interpolation stencil file (INTOUT or XINTOUT)
- Various diagnostic information (min hole cut, stencil quality table,..)

C3P FEATURES

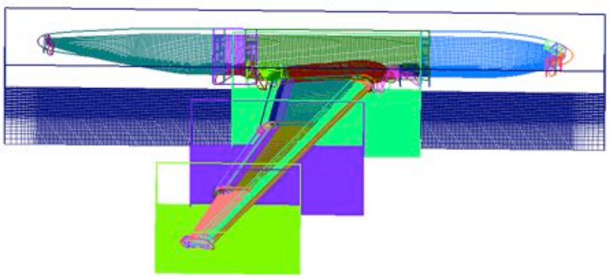
- Auto creation of standard X-rays file
- Auto creation of grid subsets cuttable by each X-ray (overridable)
- Auto detection of external versus internal cutters
- Auto blanking between off-body Cartesian meshes with appropriate overlap layers
- Domain connectivity check for surface grids
- Option to specify explicit hole cut regions in physical and index space
- OpenMP parallelization
- Auto output of standard X-rays, hole cutter instructions with constant offset distances in format to be used as first estimate for DCF inputs in OVERFLOW flow solver
- Low memory requirement
- Low I/O

Available under beta test agreement from NASA Ames for U.S. citizens or permanent residents working for U.S. organization in the U.S.

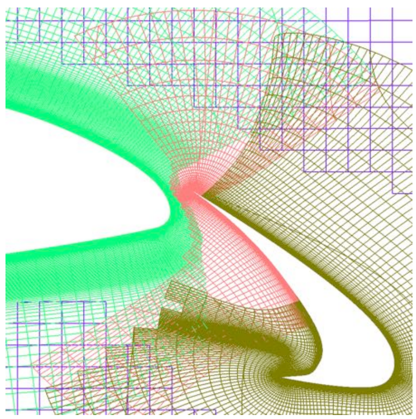
Publications: ICCFD7-1201 (2012), AIAA Papers 2013-3074, 2015-3425

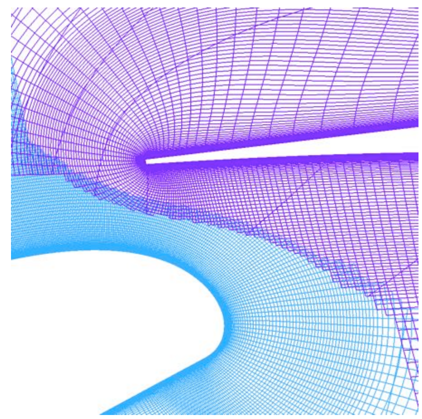
C3P RECENT TEST CASES

High Lift CRM



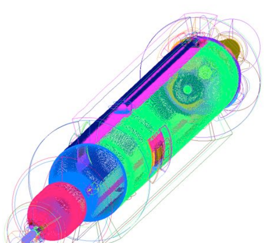
# Components	5
# Grids	72
# Grid points	73 million
# Fringe layers	2
# Orphan points	15
Wall clock time (wk station 24 threads)	3.3 minutes

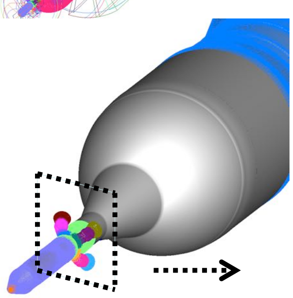


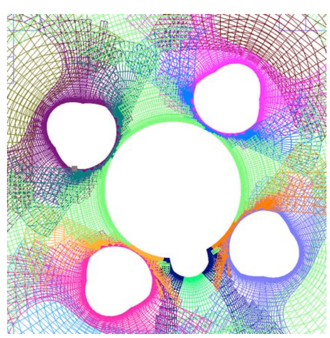


C3P RECENT TEST CASES

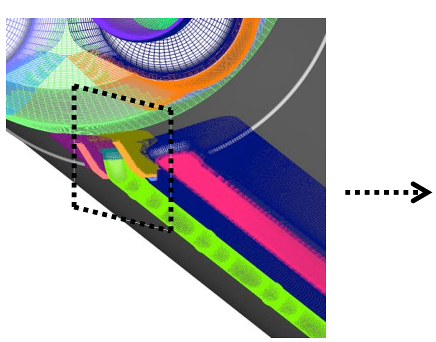
Rocket with Protuberances

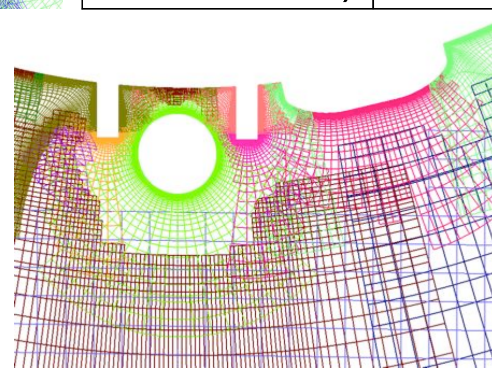






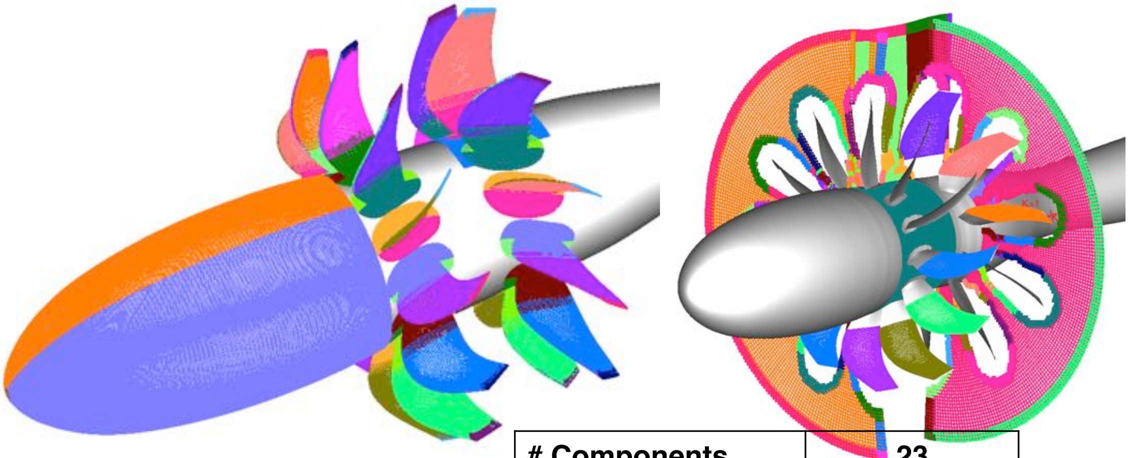
# Components	25
# Grids	122
# Grid points	80 million
# Fringe layers	2
# Orphan points	73
Wall clock time (wk station 24 threads)	3.4 minutes





C3P RECENT TEST CASES

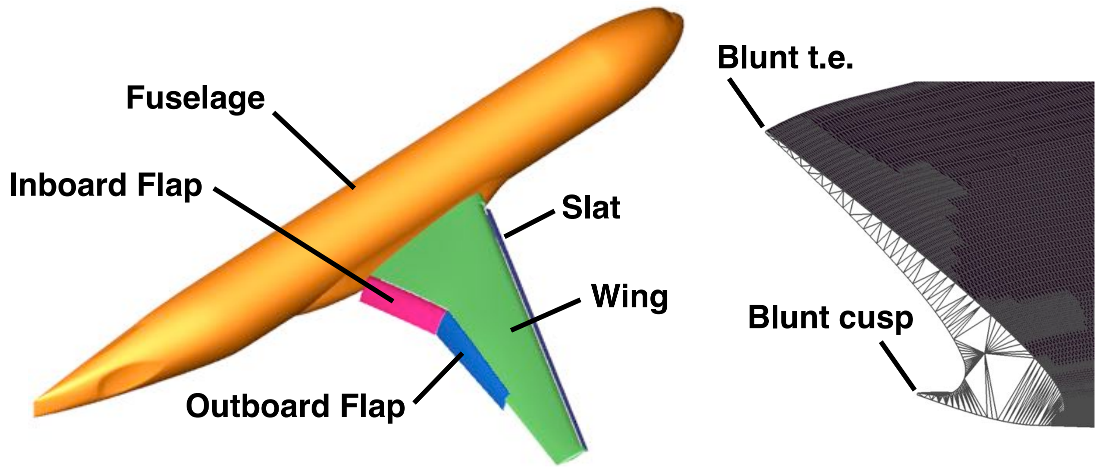
Open Rotor



# Components	23
# Grids	123
# Grid points	164.6 million
# Fringe layers	3
# Orphan points	0
Wall clock time (wk station 24 threads)	18.3 minutes

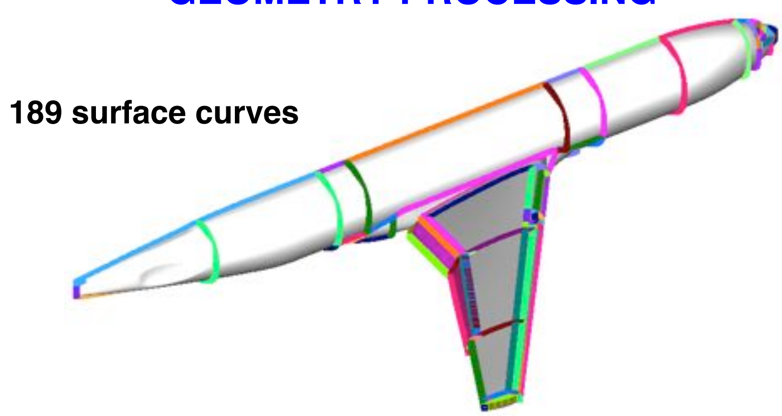
PRE-PROCESSING SCRIPT DEVELOPMENT FOR HIGH LIFT CRM

COMPONENTS AND FEATURE SIZE IDENTIFICATION



- Component characteristic lengths (slat, wing, flap chords)
- Small geometric feature sizes
    - slat, wing, flap trailing edge thickness
    - slat, wing cusp thickness
    - gap between: inboard flap and fuselage, slat and wing, wing and flaps, inboard and outboard flaps

GEOMETRY PROCESSING



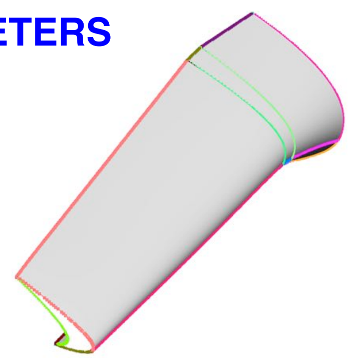
- Read STEP or IGES file into ANSA
- Generate unstructured surface triangulation (STL or advancing front)
    - control using max dihedral angle, cell size, chord deviation
- Paint triangles in each region separated by surface feature curves and CAD edges with different integer ID tags
- Write triangulation in STL format, convert to CART3D format (TRIGED)
- Use SEAMCRT module in Chimera Grid Tools to extract surface features curves in PLOT3D format
- Resulting curves are point matched with surface triangulation vertices

COMPONENT-CENTRIC MANAGEMENT SCRIPTS

- Develop script for each component
  - Create all surface and volume grids for component
  - All surface grids for component reside in one file
  - All volume grids for component reside in one file
  - Create X-ray maps for component and write to one file
  - Write flow solver boundary conditions for grids in component to one file
  - Write force/moment inputs for component to one file
  - Well suited for components with repeated parts
  - Well suited for X-ray map approach for connectivity
  - Executable individually or from a master script

- Develop master script to
  - Combine grids and X-maps to one file
  - Process component files and write input files for domain connectivity, force/moment computation, flow solver

GLOBAL CONTROL PARAMETERS



NFRINGE = number of fringe layers (2 for 5-point stencil in flow solver)

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

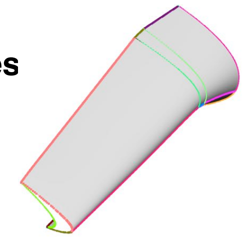
SR_{max} = max stretching ratio (1.2)

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

Δs at various curve end points expressed as scale factor times Δs_{max}

Δs_{wall} = normal grid spacing at wall (Reynolds number dependent)

TYPICAL COMPONENT SCRIPT PROCESSES



Given component surface triangulation and initial curves

- Split curves into individual files (t.1, t.2, …, t.n)
- Identify four-sided domains for TFI fill
- Identify initial curves for hyperbolic marching
- Redistribute grid points on curves with appropriate clustering at sharp edges, and matching point counts on opposite sides of four-sided domains
- Perform TFI surface grid generation on sets of redistributed curves
- Determine marching distances for hyperbolic grids and perform grid generation
- Concatenate appropriate algebraic and hyperbolic grid parts to form surface grids (root.sur)
- Create hyperbolic volume grids (root.vol)
- Create component X-ray maps (root.xry)
- Create and write boundary conditions and cuttee grid groups of component grids to root.bc.in
- Create and write force/moment component subsets to root.fomo.in

TYPICAL MASTER SCRIPT PROCESSES

Given component list with rootnames a, b, c, …

- Combine surface grids (a.sur, b.sur, …) to one file (grid.sur)
- Combine volume grids (a.vol, b.vol, …) to one file (grid.in)
- Combine X-ray maps (a.xry, b.xry, …) to one file (xrays.in)
- Process or specify hole cut instructions for each X-ray cutter cuttee grid list, and offset distance
- Process bc files (a.bc.in, b.bc.in, …)
- Process fomo files (a.fomo.in, b.fomo.in, …)
- Write input file for domain connectivity using C3P
- Write input file for force/moment computation using MIXSUR/OVERINT or USURP
- Write input file for OVERFLOW (includes domain connectivity inputs using X-rays, and all flow solver inputs)

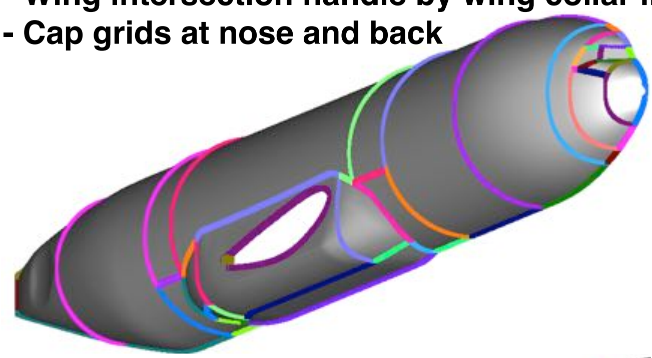
MASTER AND COMPONENT SCRIPTS FOR HIGH LIFT CRM

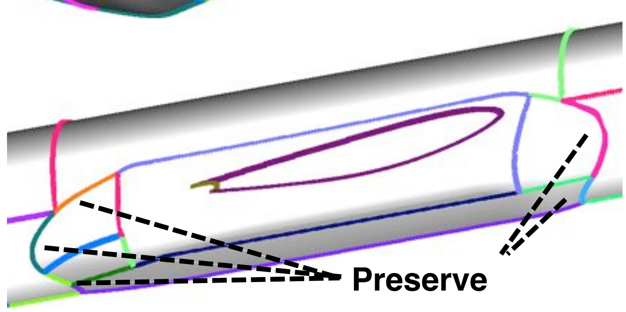
Master script → grid.in, xrays.in, flow solver input file, force/moment input file

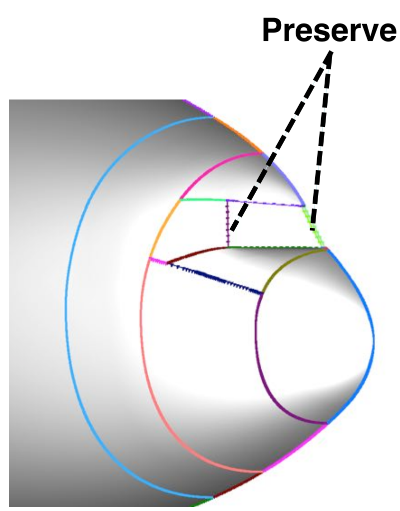
Fuselage script	Slat script	Wing script	Flap script	
↓	↓	↓	↓	
fuse.sur	slat.sur	wing.sur	flapi.sur	flapo.sur
fuse.vol	slat.vol	wing.vol	flapi.vol	flapo.vol
fuse.xry	slat.xry	wing.xry	flapi.xry	flapo.xry
fuse.bc.in	slat.bc.in	wing.bc.in	flapi.bc.in	flapo.bc.in
fuse.fomo.in	slat.fomo.in	wing.fomo.in	flapi.fomo.in	flapo.fomo.in

FUSELAGE COMPONENT SCRIPT

- Preserve open boundary on symmetry plane
- Preserve sharp crease lines around cockpit and wing bulge
- Wing intersection handle by wing collar in wing script
- Cap grids at nose and back

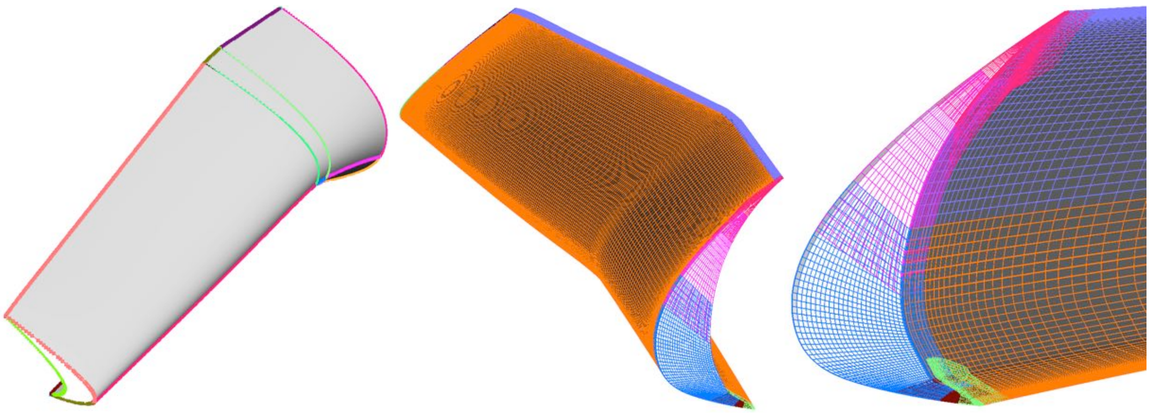






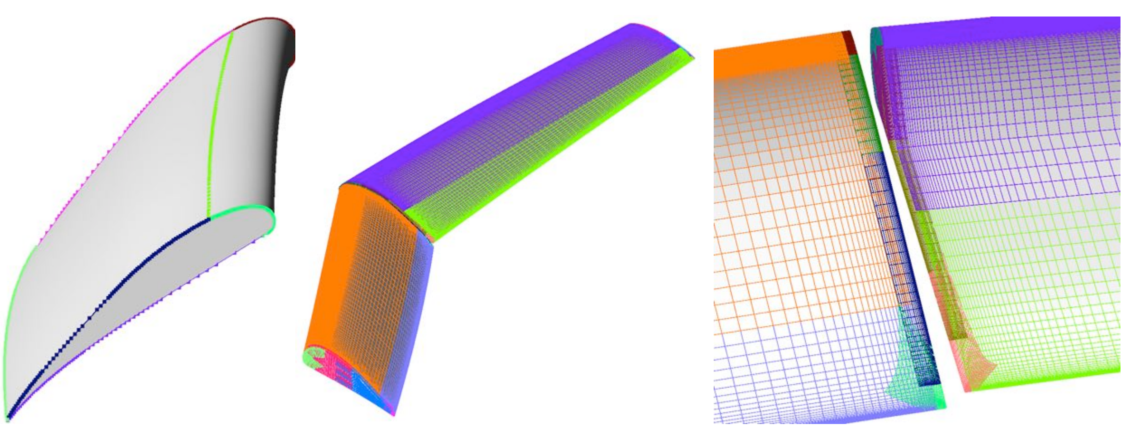
SLAT COMPONENT SCRIPT

- Finite thickness cusp and trailing edge
- Wrap around mesh on main element split into two
  (avoid periodic direction for OVERFLOW grid adaption later)
- Cap grids over root and tip
  (5 grids each to capture all sharp edges efficiently)



FLAP COMPONENT SCRIPT

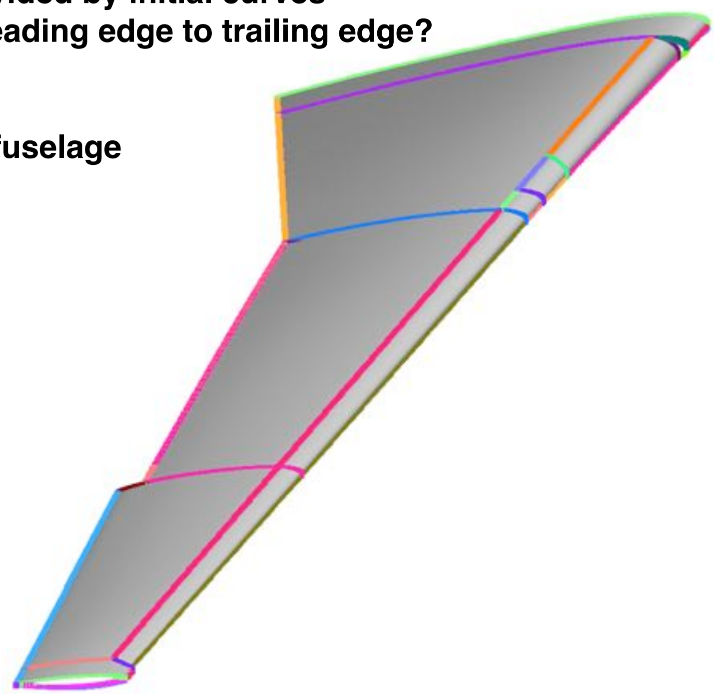
- Inboard and outboard flaps have same topology => one proc to handle both
- Finite thickness trailing edge
- No leading edge curve from geometry definition, use fine spacing all around since wake capture is important
- Tight gap between inboard and outboard flaps => use tight spanwise spacing adjacent to gap



WING COMPONENT SCRIPT
Regions

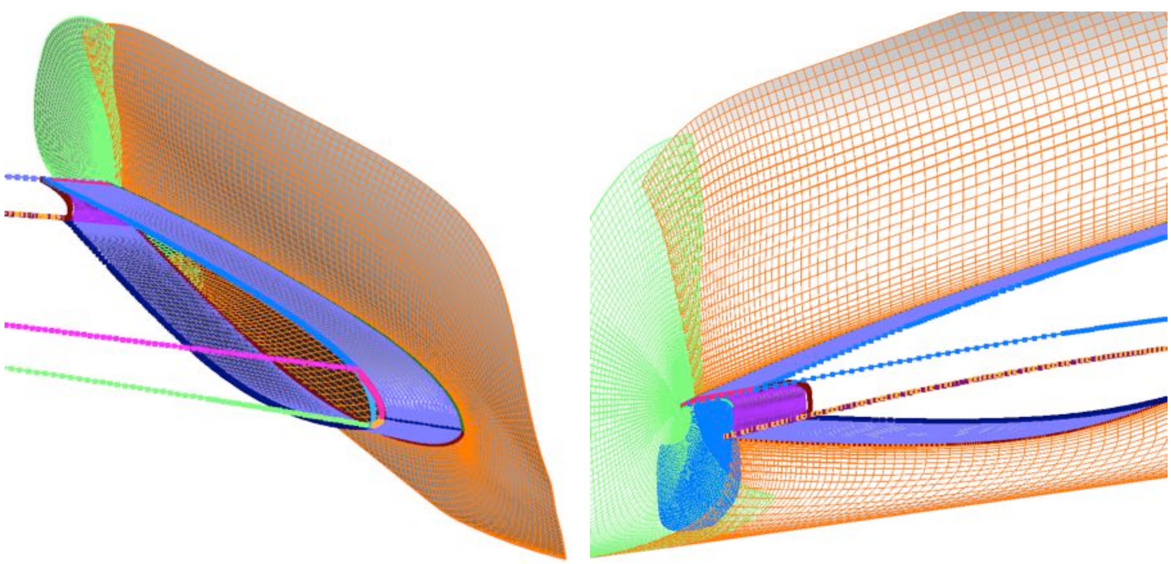
- Build grids by regions divided by initial curves
- Inboard to outboard, or leading edge to trailing edge?

Regions
- Collar grid connected to fuselage
- Slat cove
- Flap cove
- Cap grids over wing tip
- Upper middle
- Lower middle



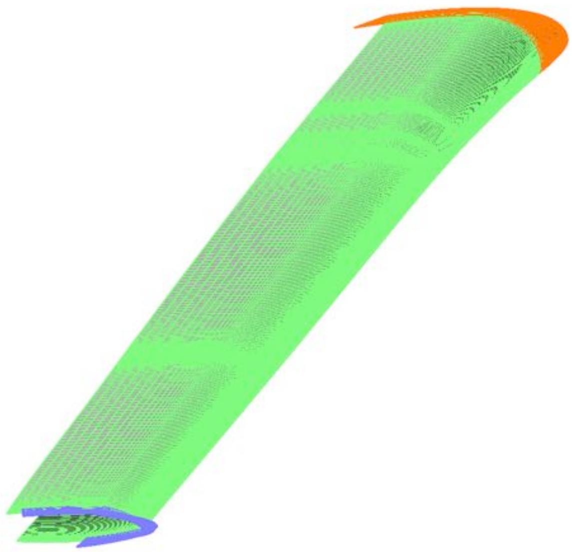
WING COMPONENT SCRIPT
Wing Fuselage Collar

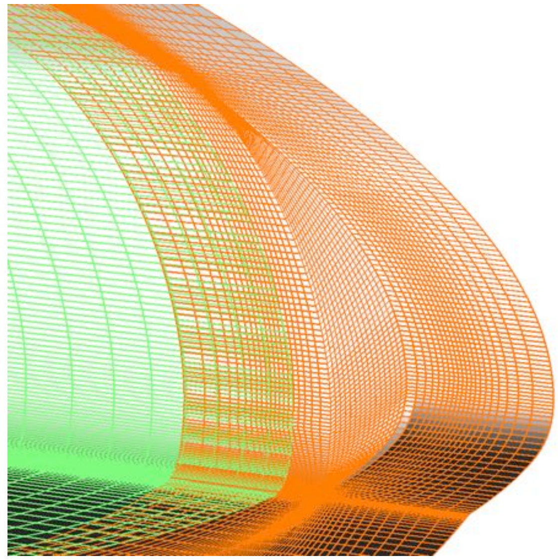
- Three collars: main, trailing edge and flap cove
- Retract from slat cove sharp edge



WING COMPONENT SCRIPT
Slat Cove

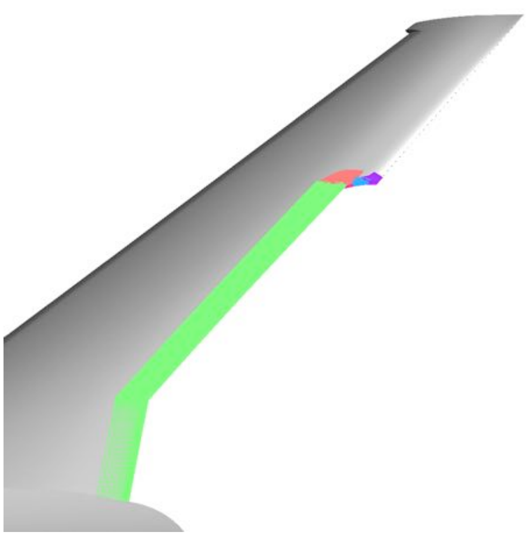
- Multiple algebraic and hyperbolic parts
- Some smoothing needed around cusp

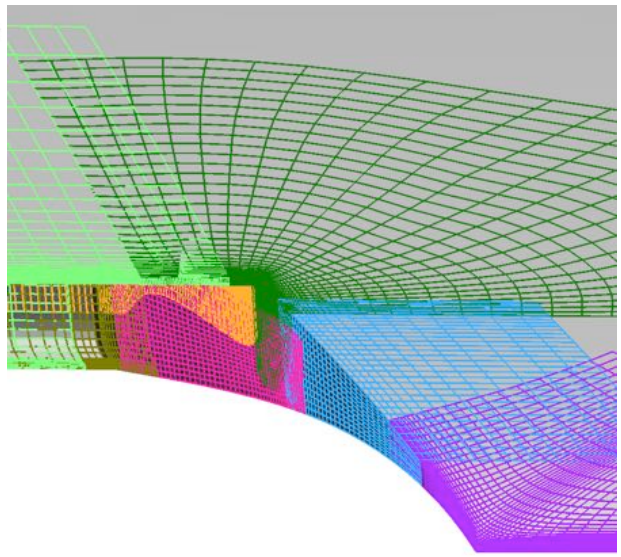




WING COMPONENT SCRIPT
Flap Cove

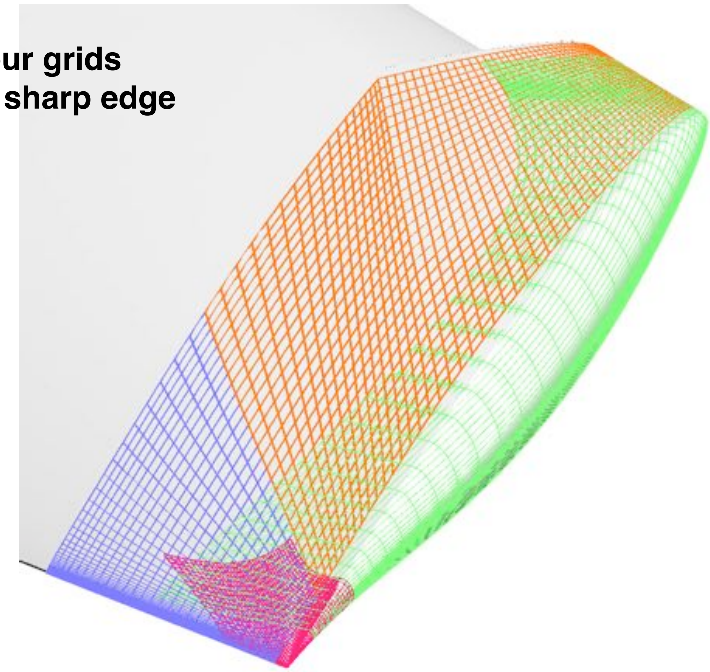
- Multiple algebraic and hyperbolic parts





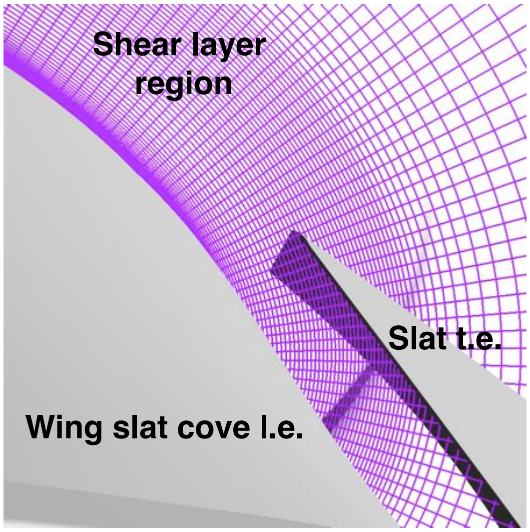
WING COMPONENT SCRIPT
Wing tip

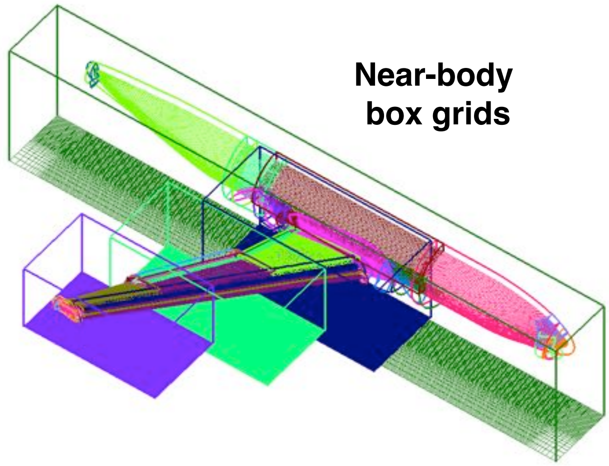
- Wing tip covered by four grids
- Retract from slat cove sharp edge



WING COMPONENT SCRIPT
Wing Surface Grids

VOLUME GRIDS

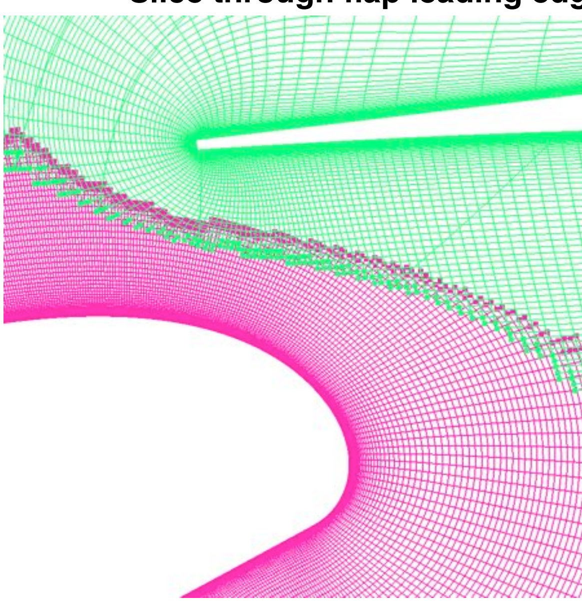


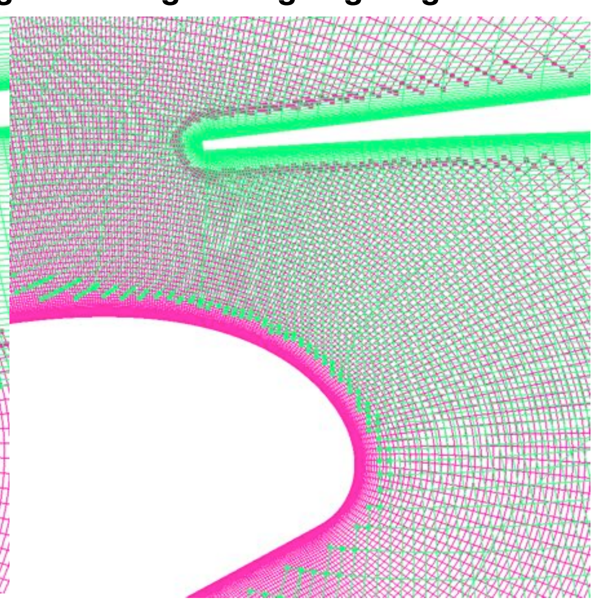


Wing and flap volume grids have 4 layers in normal directions
1. Three cells of constant spacing at wall
2. Stretched from wall spacing to shear layer spacing
3. Shear layer region with uniform spacing
4. Stretched from shear layer spacing to outer spacing

DOMAIN CONNECTIVITY

Slice through flap leading edge and wing trailing edge region





C3P hole boundary **DCF hole boundary**

PRE-PROCESSING SCRIPT DEVELOPMENT FOR GENERIC ROCKET

GRID VS. COMPONENT-CENTRIC

Configuration:
- Grid-centric
- Written for static geometries
- Rules must be followed
  - Set root names
  - Define input variables, defaults
  - Surface files: *.srf
  - Volume files: *.vol
  - Each file contains 1 grid
- Framework provided
- Short master script
  - BuildSurf
  - BuildVol
  - BuildPeg5i
- Peg5, X-rays supported

Component:
- Component-centric
- Written for repeated components and moving-body cases
- Rules must be developed by each user
  - Best practices:
    - Define input variables
    - Flexible/recommended filenames: *.sur, *.cut, *.vol, *.xry
    - Each file contains 1 component (Any number of grids)
- Framework contained in a master script
- Longer master script
  - Contains all calls for surface, volume, and connectivity according to user's choice.
- Supports Peg5, X-rays, C3P

GRID-CENTRIC SCRIPTING

Step 1: Setup directory structure
Surface Meshing Scripts, Makefile
Similar to CRM example,
but in sub-directories

Step 2: Config.tcl
Root names for all grids
Example:
set rootname=
fuse/nose \
fuse/mainfuse \
fuse/rear \
wing/wbcollar \
wing/mainwing \
wing/tip

GRID-CENTRIC SCRIPTING

Step 3: Inputs.tcl
Geometry parameters:
Size of a parametric part, location of a component, etc.

Shared surface meshing parameters:
Fuselage spacing so the wing collar can match it.

Volume meshing parameters:
Wall spacing, outer spacing, stretching ratio.
Component-specific spacings or boundary conditions.
Box grid sizing.
Hole cut instructions for X-rays

Step 3: Inputs.tcl (continued)

FOMOCO inputs:
Defaults for reference length, area, etc. and priority for specific surfaces

Overflow inputs:
Mach, α, Re, CFLmax, etc.

Step 4: Master Script
BuildSurf
BuildVol
BuildPlot -srf -vol
BuildPeg5i
run Pegasus
BuildMixsuri
run Mixsur