Source of Acquisition
NASA Johnson Space Center

Best Practices in Overset Grid Generation

Reynaldo J. Gomez III
NASA Johnson Space Center
Houston, Texas

6th Symposium on Overset Grids & Solution Technology
October 8-10, 2002

William Chan/ARC presented these charts at an AIAA Meeting in June 2002.

Overview

Acknowledgements

"Best Practices In Overset Grid Generation" AIAA 2002-3191

- Background
- Geometry
- Surface Grid Generation
- Volume Grid Generation
- Domain Connectivity
- Scripting

Supersonic/hypersonic flows

Issues

Future Plans

Acknowledgements

Paper co-authors

- William M. Chan, Stuart E. Rogers
- NASA Ames Research Center

- Pieter G. Buning
- NASA Langley Research Center

James S. Greathouse, Phillip C. Stuart,
Darby Vicker,Randy Lillard

NASA Johnson Space Center

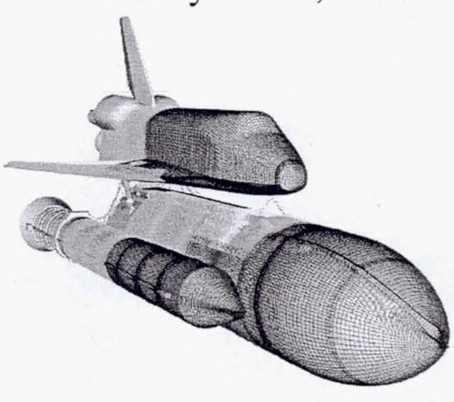
Background - How did we get here?

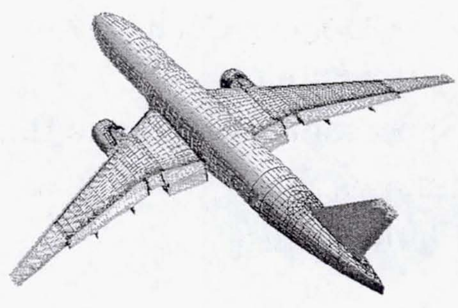
SSLV 1987-1993

- Initial OVERFLOW development, collar grids, complex geometry issues, plumes

NASA Advanced Subsonic Technology/Integrated Wing Design Program 1996 - 1999

- Chimera Grid Tools, tcl scripting, turbulence modeling, accuracy issues, PEGASUS 5.0





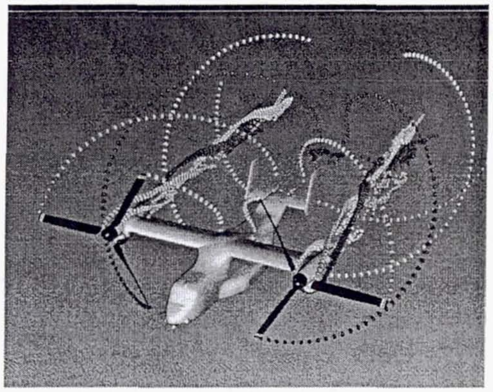
Background - continued

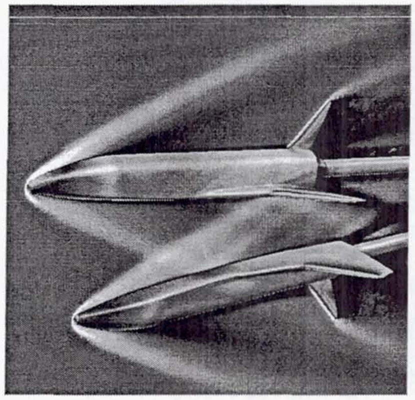
DoD High Performance Computer Modernization Project and HPCC 1995-present

- OVERGRID, OVERFLOW-D, rotorcraft applications

CICT/SLI 2000-present

- OVERFLOW 2.0, Moving body tools





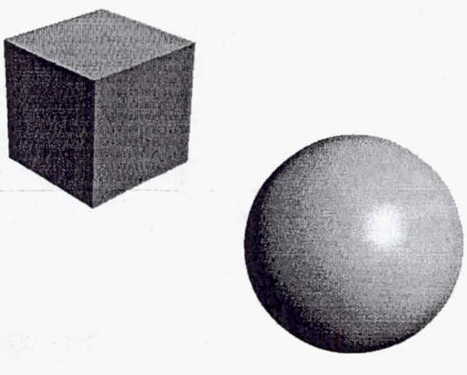
Geometry

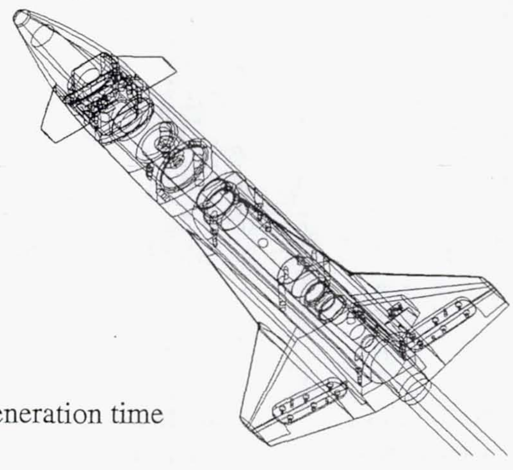
Ideal

- Accurate, manifold geometry in a format compatible with your grid generation tools

Reality

- Missing geometry, duplicate surfaces, poor parameterizations, as built geometry, wide range of formats, extraneous geometry . . .



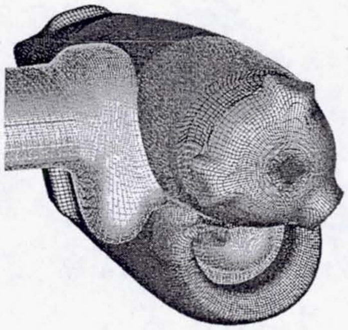


Geometry repair can take up to 80% of the total grid generation time
For most of us this is still an ad hoc process

CAD versus Discrete geometries

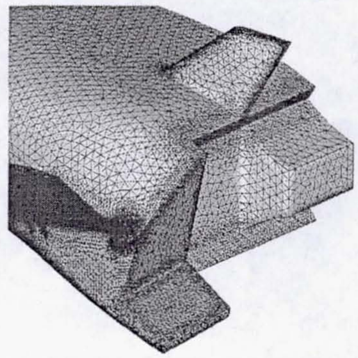
CAD

- Direct interface to CAD data preferred
  - Avoids IGES flavoring issues
- Interchange standards
  - IGES, various flavors
  - STEP

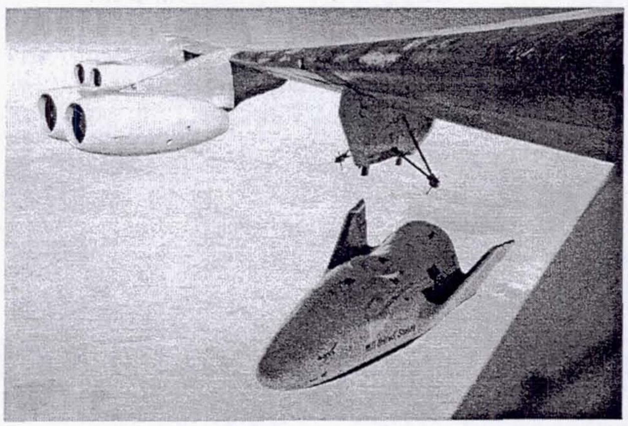


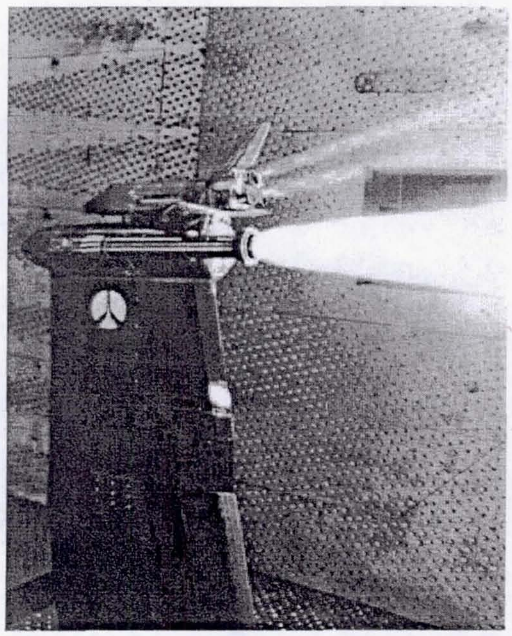
Discrete databases

- High curvature regions should have higher resolution than final grid
- Panel networks vs. Surface triangulations
  - Creation time vs. automation and memory
- May simplify automation
  - Geometric feature detection, scripting



As-Built/Tested Geometry



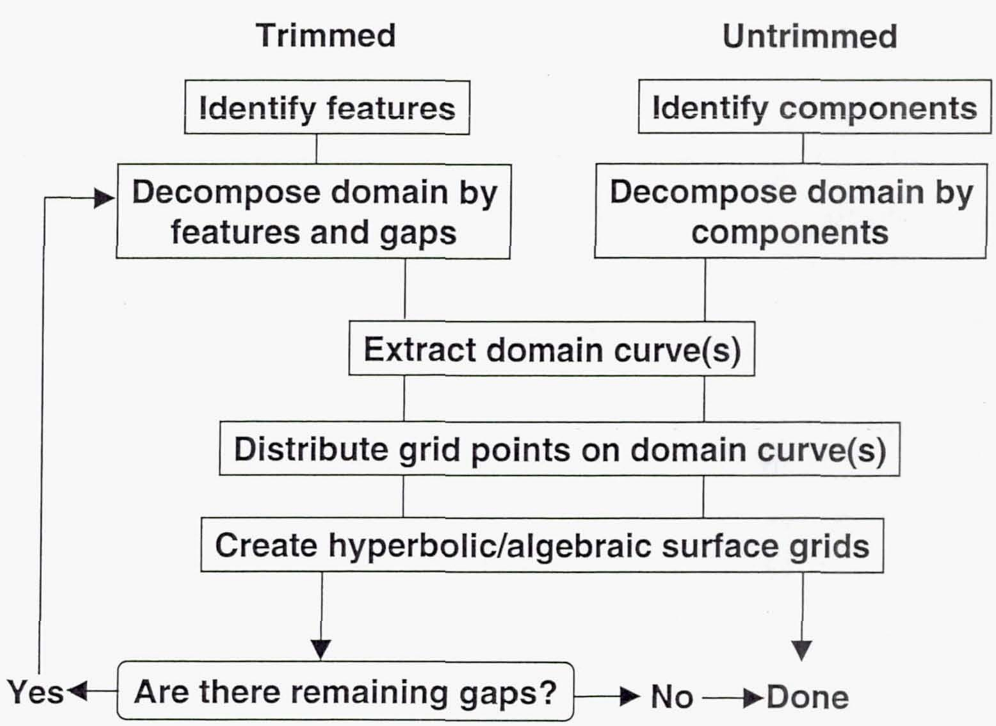


Simulating/reproducing test conditions may be more complex than operational conditions
Use the most realistic geometry that you can

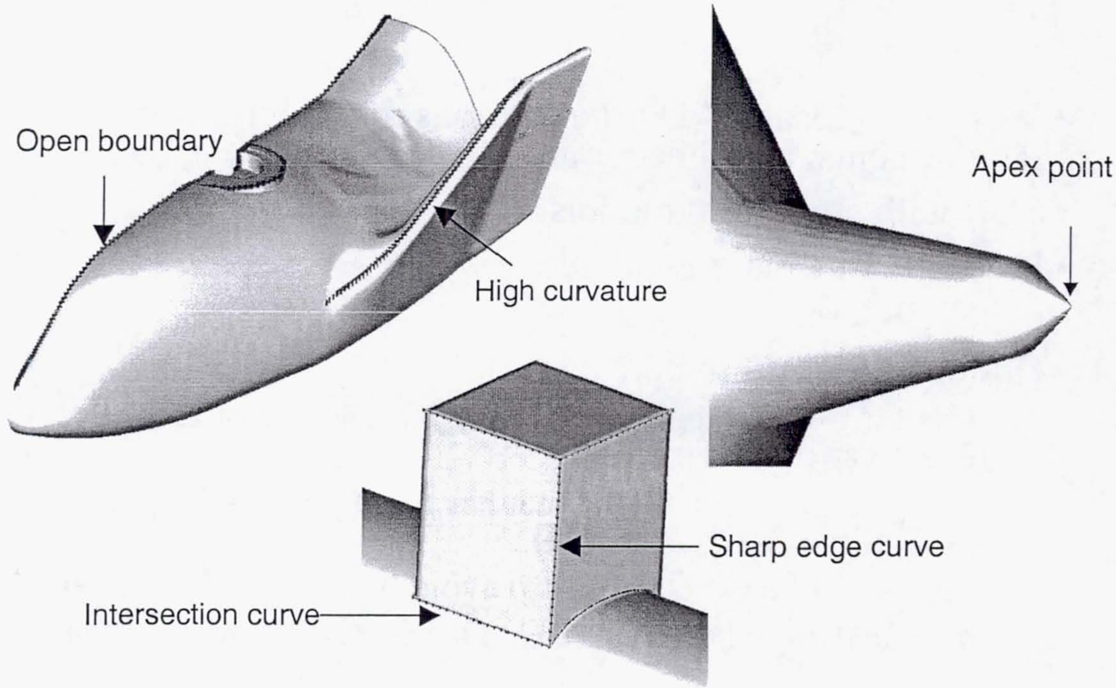
CAD Geometry Recommendations

- Work with local CAD organizations to establish specific requirements, avoid becoming a CAD expert
- Start with simplified models
- Use solid models if available
  - STEP format
- Options for CAD import
  - Use third-party applications to repair and flavor CAD geometry (CADFix)
  - Use grid generation code that has a robust CAD import capability (Gridgen, ICEM)
  - Import native CAD format to avoid IGES issues (Gridgen)
  - Interface directly to native CAD system (CAPRI, ICEM)

Surface Grid Generation Process



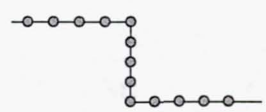
Surface Feature/Component Identification



Grid point distributions

Choose a maximum grid spacing for the near field flow, Δs_g
- Spacing on smooth regions of the surface
- Geometric fidelity versus computational cost

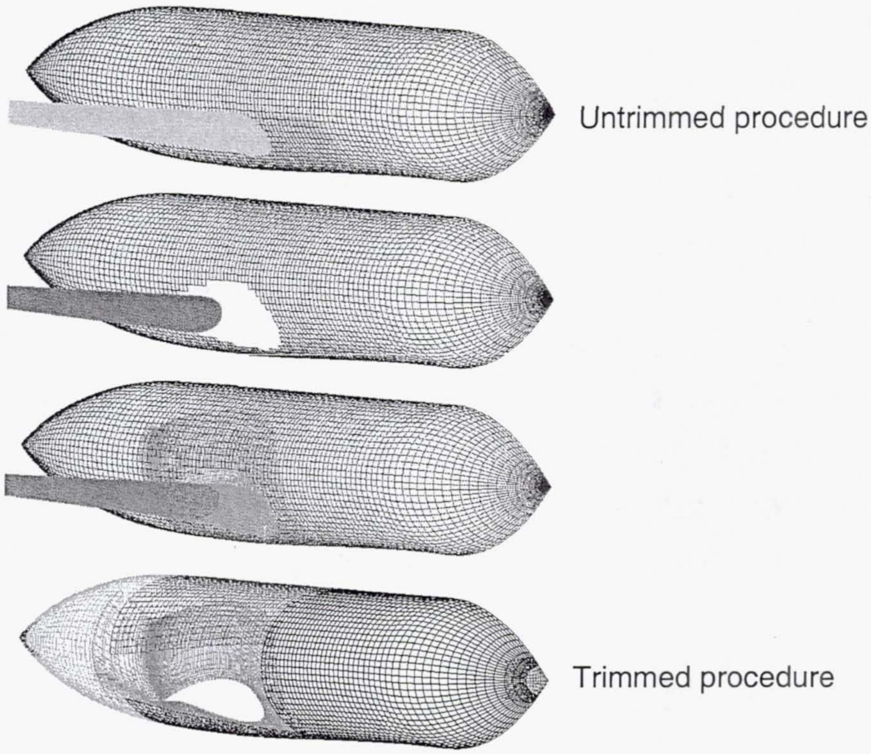
Resolve relevant geometry features with at least 5 points



Use small stretching ratios
- ≤ 1.2 for surface grids
- ≤ 1.3 for volume grids

Use multigridable number of points if applicable

Trimmed and Untrimmed Example



Comparison of Trimmed and Untrimmed Approaches

Untrimmed

- Follows components
- Hole cutting on surface where components intersect
- No gaps left on surface
- Simplifies modification of components

Trimmed

- Follows surface features
- No hole cutting needed on surface
- May leave gaps on surface domain that need to be filled
- Need to repartition surface domain when adding/removing components

Use trimmed approach for all required components, use untrimmed on all optional components

Surface Domain Decomposition

Smallest number of topologically simple domains
- Add or split domains to simplify volume gridding
- Load balance after grids are completed

Capture high flow gradient regions within a single domain

Avoid highly skewed domains

Avoid domains with singularities
- Unless the geometry has a natural singularity
- May limit time step in flow solver

Airfoils – Special Treatment

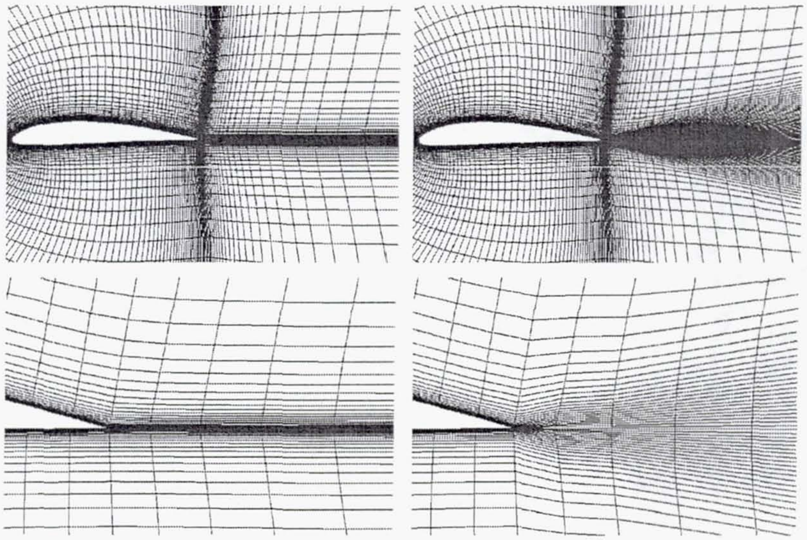
Airfoils appear in a wide range of applications
- Aerospace vehicles, turbomachinery, rotorcraft, missiles, submarines
- C versus O topology in streamwise direction
  - Use C-grid if wake resolution is important
    - Subsonic aircraft
  - Use O-grid otherwise (simpler grids)
- Areas requiring special treatments
  - Collar grid at wing root
  - Cap grid on wing tips

C-Grid Wake smoothing

Avoid small cells in wake which slow convergence

Provides better wake capturing at different angles of attack

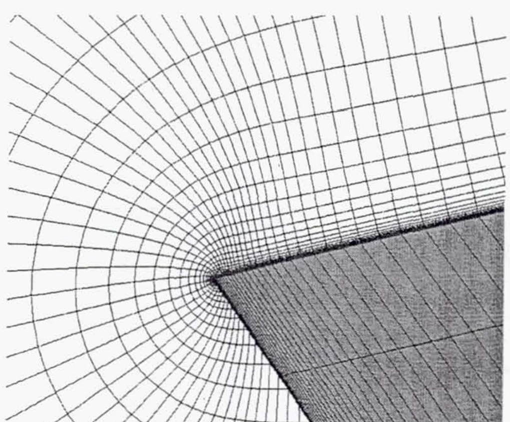
Improves inter-grid communication with neighboring grids

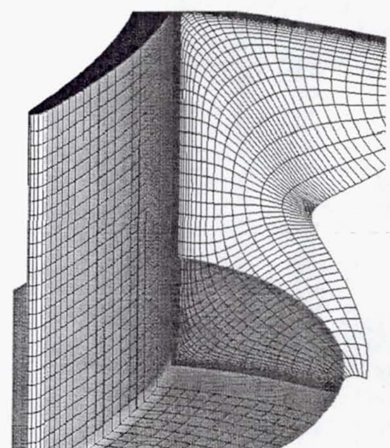


Grid Point Distributions @ Corners

Cluster towards convex corners, uniform spacing for concave regions

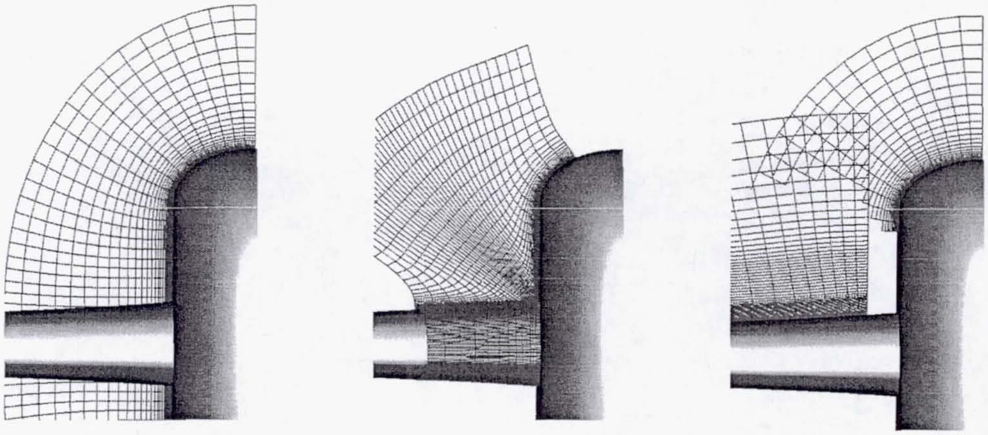
Use equal spacing and stretching ratios on each side of a corner



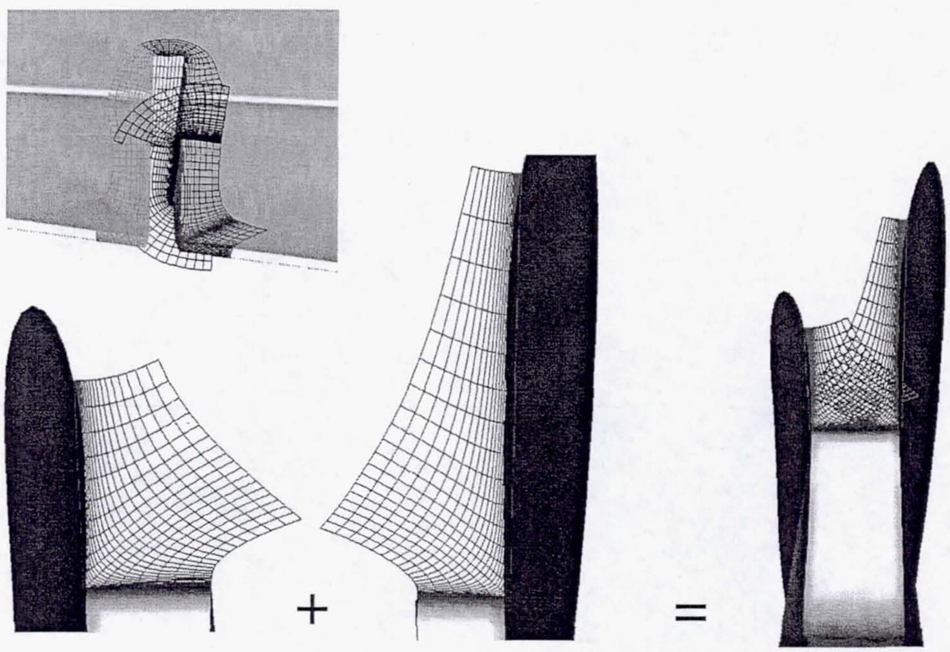


Collar grids

Provide communication between intersecting components



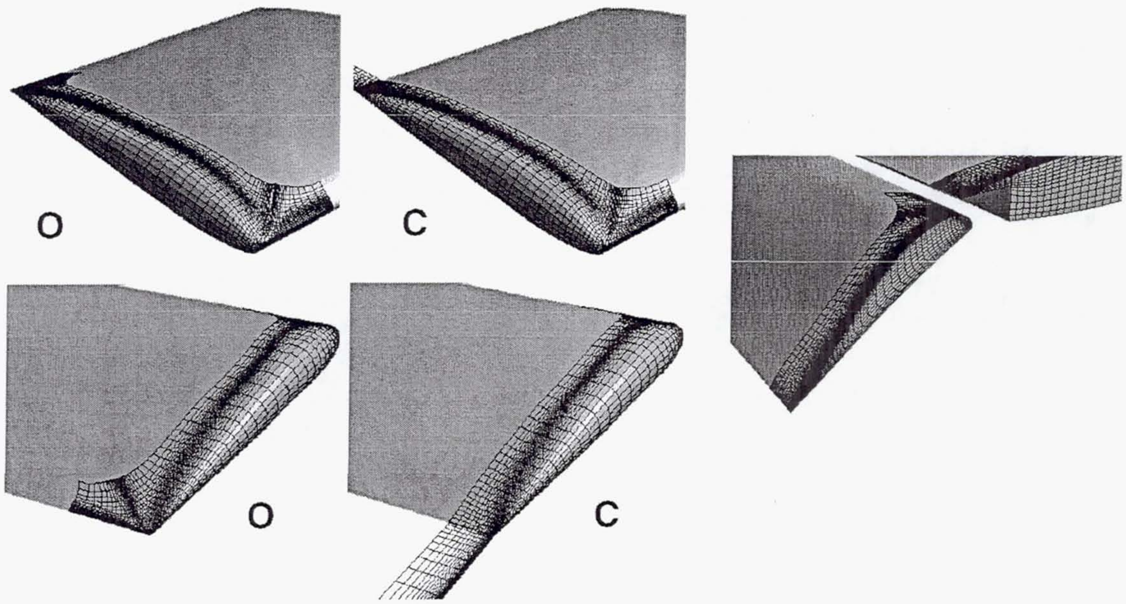
Collar Grids - Splitting



Break difficult concave corners into two collar grids

Cap grids

Avoids introducing a singularity at the wing tip



Grid Spacing Compatibility Between Grids

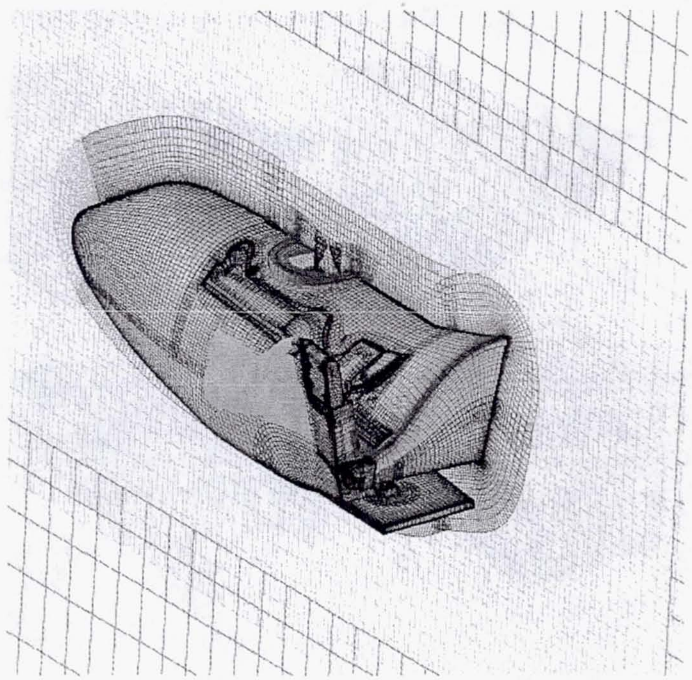
Similar resolution in overlap regions

- Similar flow feature resolution
- Simplifies MIXSUR processing
- Fewer interpolation error issues
- Less important in low flow gradient regions

At least one non-interpolated/field point between interpolated points

Volume Grid Generation

Near-body
•Body conforming grids
Off-body
•Stretched Cartesian grids

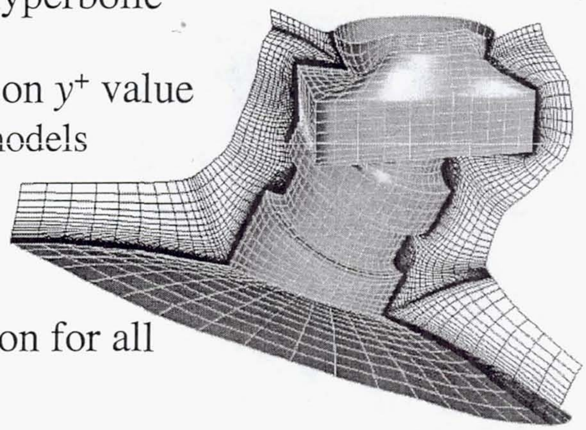


Volume Grid Generation Strategy

Use body conforming grids to resolve near-field
Grow to distance = max(D_m,D_v)
- D_m = distance where stretched normal spacing reaches Δs_g
- D_v = distance at which wall viscous effects are contained

Use Cartesian grids in off-body region
- Core Cartesian mesh should completely enclose near-body grids
  - Consider using multiple box, one for each component
- Constant spacing (Δs_g) in core grid
- Stretch to far field based on
  - Freestream Mach number
    - $M_\infty > 2.0$, 1-2 body lengths depending on angle of attack
  - Farfield boundary conditions
    - Subsonic, 20 body lengths
  - 2-D vs. 3-D
    - 2-D up to 60 chord lengths

Near-body Volume Grid generation



Most efficiently generated using hyperbolic methods

Viscous grid initial spacing based on y^+ value

- < 1 for 2-equation turbulence models
- ≈ 1 for 1-equation models
- 35-100 for wall functions

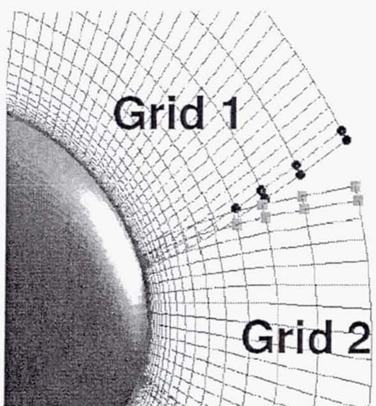
Use same normal stretching function for all viscous grids

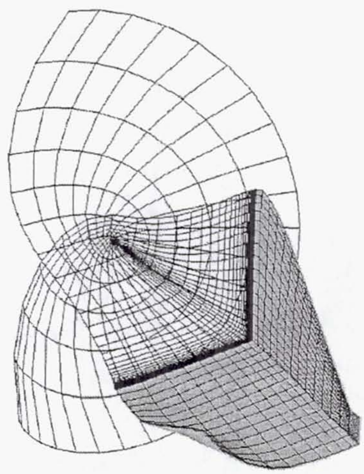
Near-body volume issues

Use "splay" boundary condition option to maintain off-body overlap

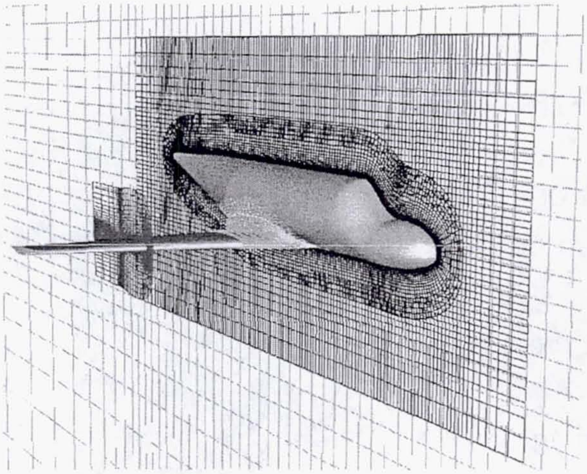
Positive Jacobian for each cell doesn't always guaranteed a valid grid

- Visual checks can be helpful

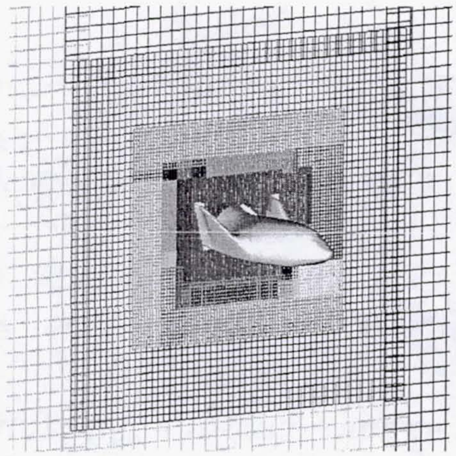




Off-body Cartesian Grid Generation



Small number of grids with uniform core and stretched outer layers



Many grids with successive levels of refinement

Solution adaptive

More communication overhead

Domain Connectivity

Multiple options

- PEGASUS 5.x/NASA, BEGGAR/AFSEO, OVERTURE/LLNL, CFD-FASTRAN/CFDRC, GASP/AeroSoft

PEGASUS 5.x

- Major improvement over PEGASUS 4.x
- Nearly automated input, based on solver b.c.s
- Automated viscous surface projection
- Overlap optimization
  - More CPU, offset by parallel performance
- Makefile-like restart capability

Use double fringes

- Flow solver maintains differencing stencil for all interior points ⇒ maintains accuracy
- Requires more overlap ⇒ more grid points

Scripting

Overall process
- Initially use a GUI to set up input files
  - Record steps to a script file
- Subsequent analyses are scripted
  - Changes in geometry and grid parameters

Scripting (pros/cons)
- Introduces some process overhead
- Allows rapid rerun of the entire process
- Simplifies grid refinement and parameter studies
- Documents grid generation procedures

Scripting Practices

Use a high level language
- tcl, Perl, Python, Ruby, Lua, etc.
- Unix shells
  - Commonly used lack floating point arithmetic and subroutine capabilities

Parameterize important variables
- Geometry – deflection angles, locations of fins, etc.
- Surface and volume grids
  - Grid spacings (Δs_g, leading edge, trailing edge, etc.)
  - Stretching ratios
  - Marching distances

Use small number of independent parameters and build rules for other parameters

Define grid boundary conditions once; shared with other modules

Grid Refinement

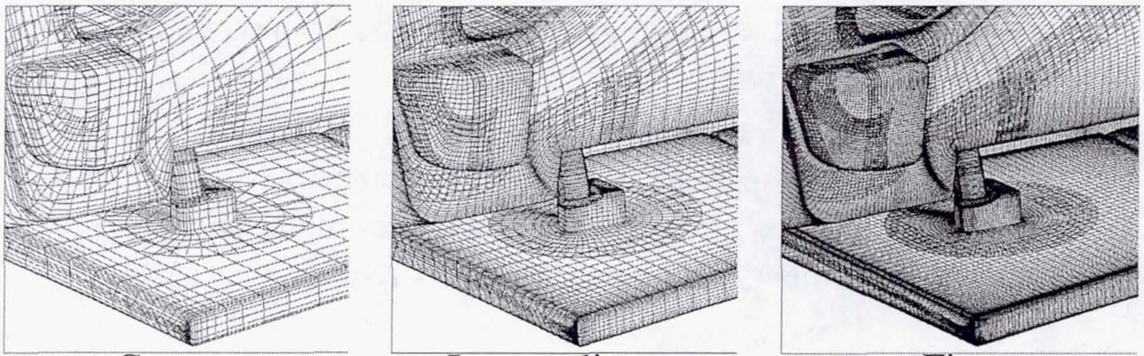
Grid refinement studies, numerical sensitivity studies, comparisons with exact results can be very time consuming.

Scripting can help automate this process and make it more commonplace.

Simple refinement and decimation are not ideal

- Stretching ratio changes

Richardson Extrapolation δC_m = -0.024, -0.0008, -0.00003 at $\alpha = 16^{\circ}$



Coarse	Intermediate	Fine
0.5 million points	3.8 million points	27.2 million points

Force and Moment Integration

Options

- Block zonal near-body grids
  - BEGGAR/AFSEO, CFD-FASTRAN/CFDRC
- Convert to triangulated surface
  - TESS/Dietz
- Retract and connect with triangles
  - MIXSUR/Chan

MIXSUR

- Results dependent on quality of overlapping grids
  - Similar size cells with sufficient overlap
    - Nearly automatic
  - Otherwise iterative procedure involving prioritization and manual subsetting

Supersonic/hypersonic flows

Some flow solver do not accurately compute shock strength on a stretched grid

One solution is to use a shock aligned grid with an equispaced region around the shock

- Blottner, F.G., "Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip," Journal of Spacecraft and Rockets, Vol 27, No. 2, March-April 1990.
- LAURA/Gnoffo uses this technique

Does not require farfield box grids

Bottom line

Accurate geometry + high quality grids

- Necessary for an accurate solution

Other requirements

- Verified/validated solver with appropriate physics
- Convergence criteria consistent with application
  - Aerodynamics - forces and *moments*
  - Heat transfer - maximum and minimum heat transfer coefficients

Issues

CFD-ready geometry
- Common problem for most CFD methods

Surface grid generation and MIXSUR
- Most time consuming steps
- Automated surface coverage techniques should help

Automated control surface motion
- Static solutions
- Dynamic solutions + control surface motion

Future Plans

Add CAD database capability to Chimera Grid Tools
- CAPRI or other CAD library

Test OVERTURE CAD import capability

Automated surface feature detection improvements

Automated surface coverage using hyperbolic/algebraic grids

Develop tools for rapid script creation