CARTESIAN-CELL BASED GRID GENERATION AND ADAPTIVE MESH REFINEMENT

WILLIAM J. COIRIER
NASA LEWIS RESEARCH CENTER

KENNETH G. POWELL
UNIVERSITY OF MICHIGAN

MOTIVATION

Wouldn't it be nice to just define the geometry and the free-stream conditions, and let the grid generation/adaptive refinement do the rest?

Objectives

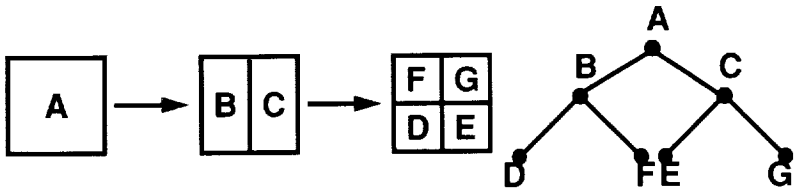
- Automated Grid Generation for Complex Bodies
- Automated Grid Refinement (Convergence?)
- Alternative to Triangular/Tetrahedral Meshes

A Cartesian-Mesh Approach

- Use Cartesian Cells of Unit Aspect Ratio to Create Background Mesh
- "Cut" Bodies Out of Background Mesh, Creating Irregularly Shaped Boundary Cells
- Arbitrary Numbers of Arbitrarily Shaped Bodies Are Allowed
- Geometry Defined With Sets of General Basis Functions Along Surfaces
- Background Mesh Created By Recursively Refining Cartesian Cell Into Four Cells

GRID GENERATION

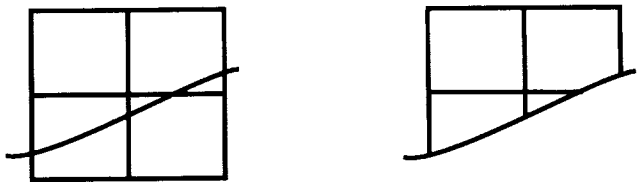
- Grid Generation Process Creates Binary Tree



- Binary Tree Allows Quad and Binary Refinement
- Connectivity/Tree Hierarchy Closely Related

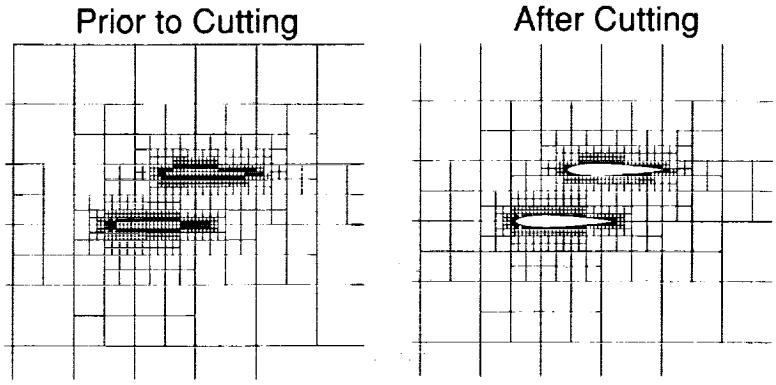
GRID GENERATION

- Recur to Leaves of Tree and Determine Intersections (if any) with Bodies
- Use Simple Set of Rules to Determine If It is Legal to Cut Leaf into Cell: Recursively Refine if Illegal
- Vertex Locality Used to Determine Cut Cell Geometry



CELL CUTTING

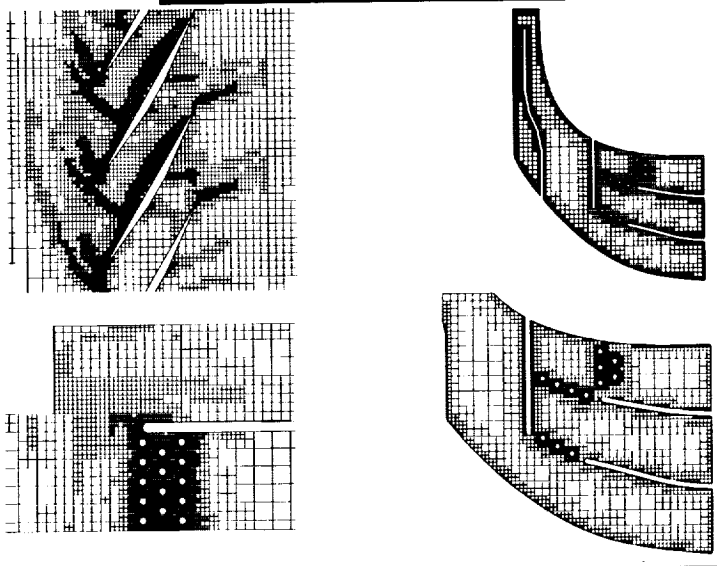
- Example: Staggered Biplane Configuration of Clarke, Salas and Hassan (AIAA J. 1986)



DATA STRUCTURE(S)

- Cartesian Cell Geometric Data Inferred From Tree
- Cut Cell Geometric Data From (Local) Ordered List of Pointers to (Global) List of Vertices
- Connectivity Is Inferred Directly From Tree By Logical Tree Traversals (Centroid Compares, Face Matching)
- Code Written in ANSI C: Dynamic Memory Allocation/Deallocation, Self-Referential Data Structures

SAMPLE GRIDS



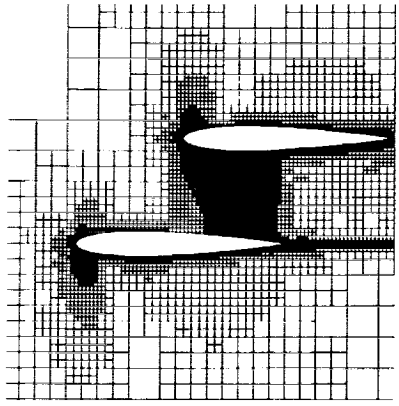
FLOW SOLVER FORMULATION

- Cell Centered, Finite Volume, Upwind Based Scheme
- Linear Reconstruction (Minimum-Energy) of Primitives Used to Compute Left/Right Interface States as Input to Approximate Riemann Solver
- Adaptive Mesh Refinement Using Cell Size Weighted Criterion Based on Velocity Divergence and Curl (Compressibility and Rotation)
- Perform Flow Solve/Adaptation Set Number of Times

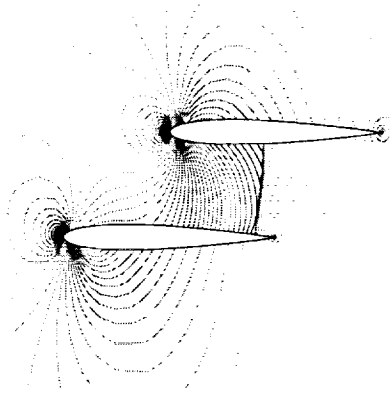
ADAPTIVE MESH REFINEMENT

- Staggered Biplane Case

Grid



Pressure Contours

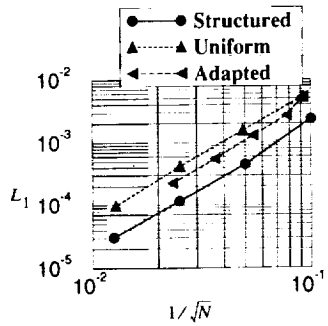


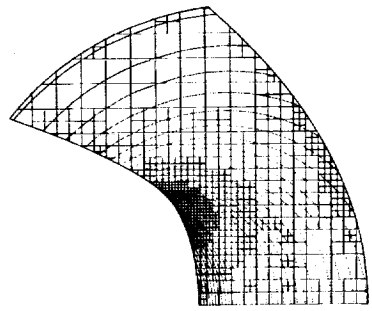
ACCURACY ASSESSMENT

- Use Exact, Analytic Solution (Ringleb's Flow)
- Infer Order of Error From Uniform and Adaptive Refinement
- Infer Magnitude of Error by Comparing to Structured Solver
- Asks Question:

Can Adaptive Mesh Refinement Beat Uniform Refinement and/or Structured Uniform Refinement?

ACCURACY ASSESSMENT

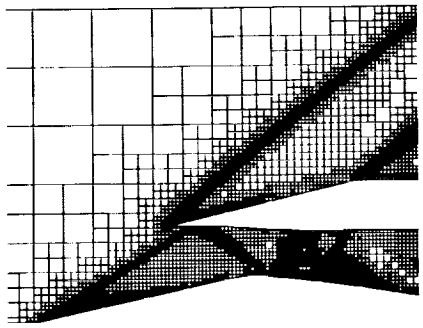


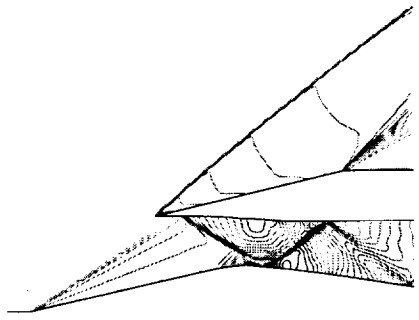


- Approach is 2nd Order (Global), Better than 1st (Local)
- **Smooth Flow:** Can't Beat Uniform Refinement or Structured

ADAPTIVE MESH REFINEMENT

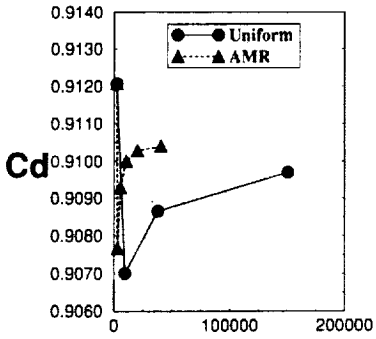
- What About Non-Smooth Flows?
- Grid Convergence Study on Supersonic, Axi-Symmetric, Mixed-Compression Inlet





ADAPTIVE MESH REFINEMENT

- Compare Uniform and Adapted Drag Coefficients

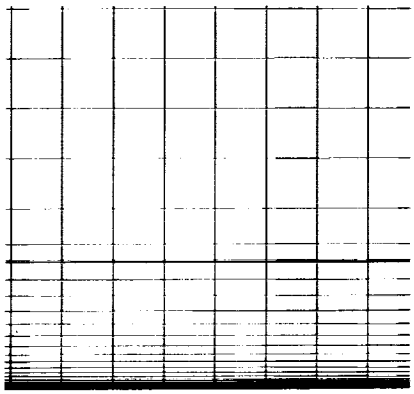


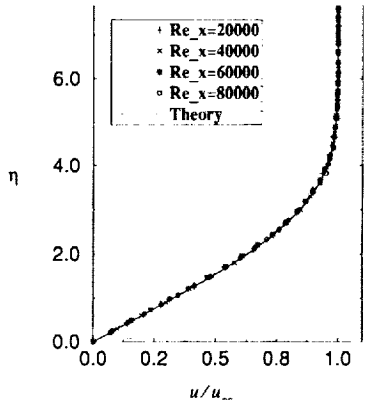
Conclusion

- AMR Grid Converged
- Uniform Not Converged (150,683 Cells!)
- Adaptive Mesh Refinement Best For Non-Smooth Flows With Multiple Length Scales

VISCOUS FLOWS

- Presently Extending to Viscous Flows
- "Cut" Level Distance Lines From Bodies





CONCLUDING REMARKS

- Proven to be an Accurate Alternative to Triangular/Tetrahedral and Structured Grids
- Adaptive Refinement Best on Flows With Widely Varying Length Scales

FUTURE DIRECTIONS

- Can This Approach Work Well For Viscous Flows? (Grid Smooth Enough With Distance Cutting?)
- What About 3D?
- WYSIWYG Front End?