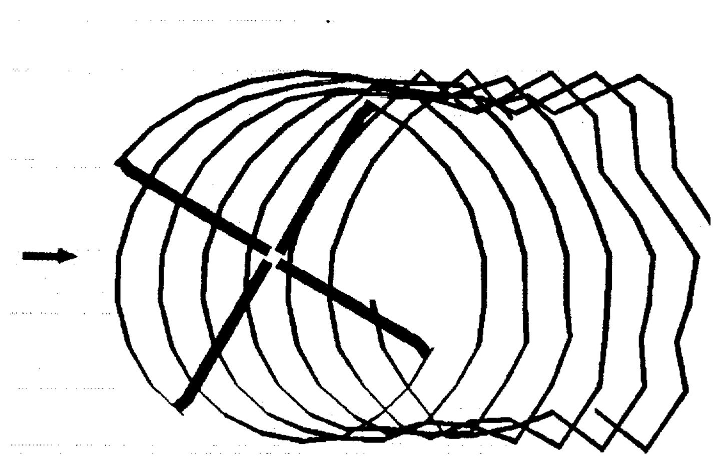
N94-22363

DYNAMIC MESH ADAPTION FOR TRIANGULAR AND TETRAHEDRAL GRIDS

RUPAK BISWAS
RIACS-NASA AMES RESEARCH CENTER

ROGER STRAWN
US ARMY AFDD-NASA AMES RESEARCH CENTER

ROTOR WAKE CAPTURING WITH A CFD METHOD

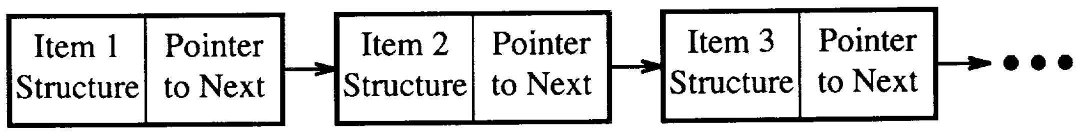


Requirements for Dynamic Mesh Adaption

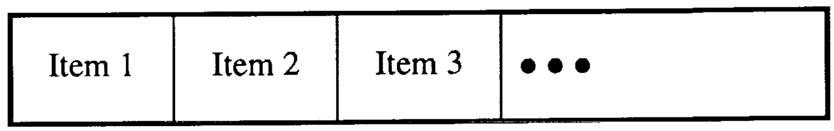
- Anisotropic refinement capability in order to efficiently resolve directional flow features
- Coarsening required for both steady and unsteady applications
- Algorithm scaling important
- Low memory overhead using dynamic memory allocation
- CPU time comparable to a time step of the flow solver

Linked-List Data Structure

Linked List



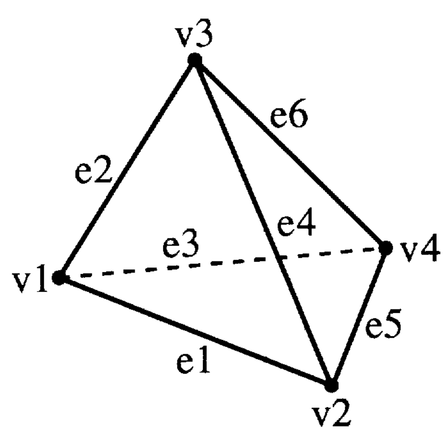
Static Array



- Facilitates quick insertion and deletion of items
- Dynamically allocates and frees memory
- No need for compaction and garbage collection

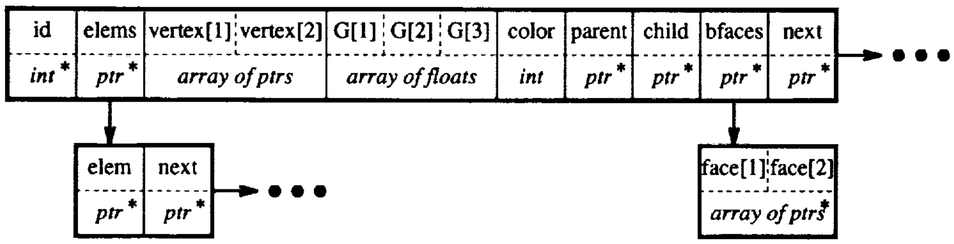
Edge-Based Data Structure

- An edge is a line segment that connects two vertices
- A tetrahedron can be uniquely defined by its six edges: e1, e2, e3, e4, e5, e6

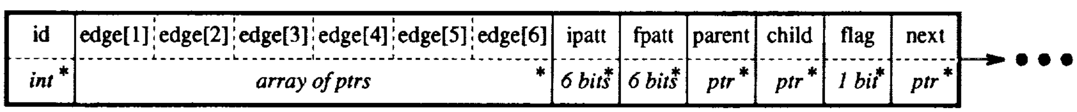


Adaptive-Grid Data Structure

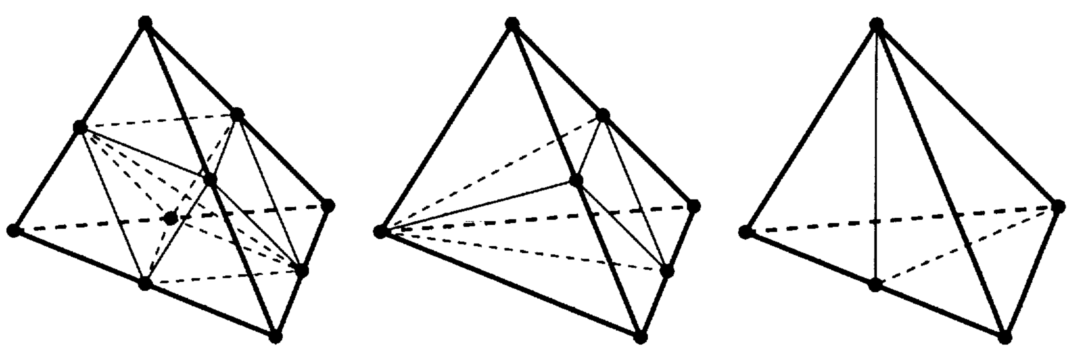
Edge List



Element List



Three Types of Element Subdivision

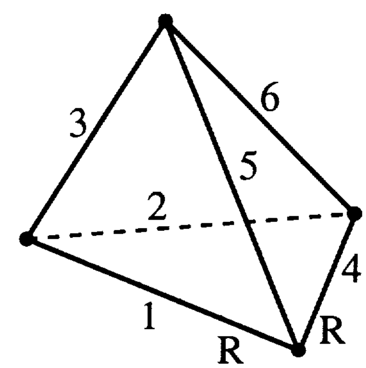


1:8 1:4 1:2

- The 1:4 and 1:2 elements are the result of anisotropic refinement or act as buffers between the 1:8 elements and the surrounding unrefined mesh

Mesh Refinement

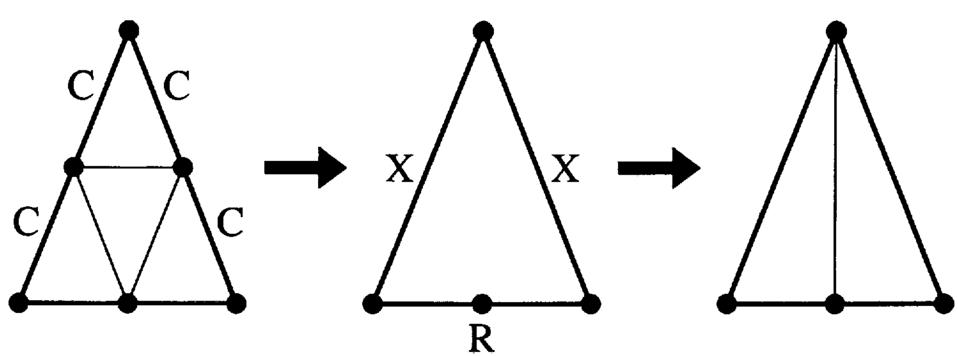
- Individual edges marked for refinement
- Marked edges combined to form binary pattern (ipatt) for each element
- Element patterns upgraded to form valid 1:8, 1:4, or 1:2 subdivisions (fpatt)



6 5 4 3 2 1	Edge #
0 0 1 0 0 1	ipatt = 9
0 0 1 0 1 1	fpatt = 11

Mesh Coarsening

- Elements with edges to be coarsened immediately revert back to their parents
- Parent elements have their ipatt values modified to reflect the fact that some edges have coarsened
- Parent elements then appropriately refined

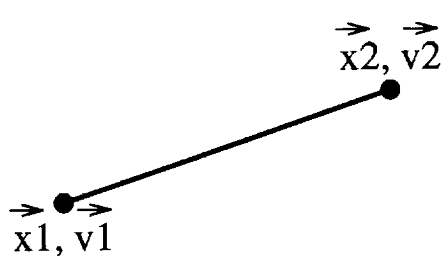


Additional Constraints for Coarsening

- In general, edges and elements must be coarsened in an order reversed from the one by which they were refined
- An edge can coarsen if and only if its sibling also marked for coarsening
- Edges of non-leaf elements or of their siblings cannot be coarsened

Anisotropic Error Indicator for Edges

- Adaption based on an error indicator computed for every edge of the mesh
- Flow gradients must be aligned with the edges for them to be marked for refinement
- Relative number of edges marked for coarsening and refinement adjusted to maintain a user-specified upper limit on problem size



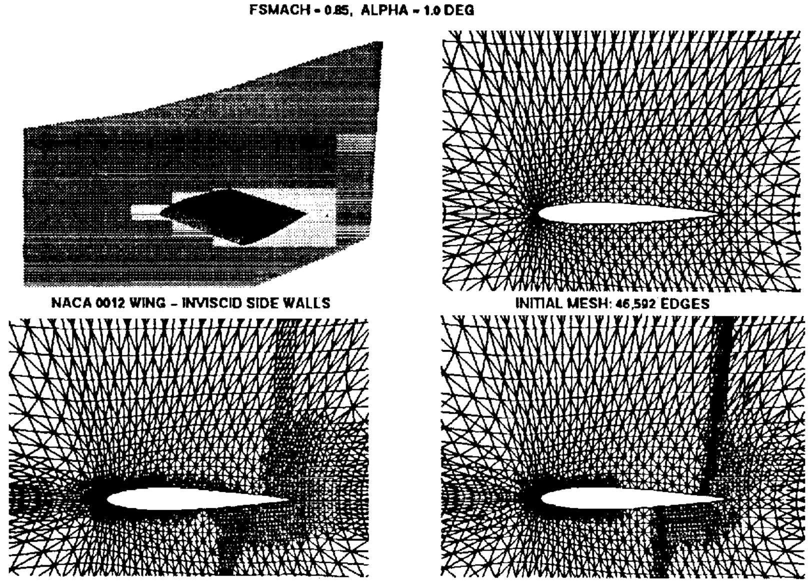
$$|E_e| = \|\Delta\vec{x} \cdot \Delta\vec{v}\|$$

Unstructured-Grid Euler Solver

- Basic code written by Barth; rotary-wing version developed by Strawn and Barth
- Finite-volume method with upwind differencing
- Computational control volumes centered at cell vertices
- Edge data structure allows arbitrary polyhedra
- Solution advanced in time using conventional explicit procedures

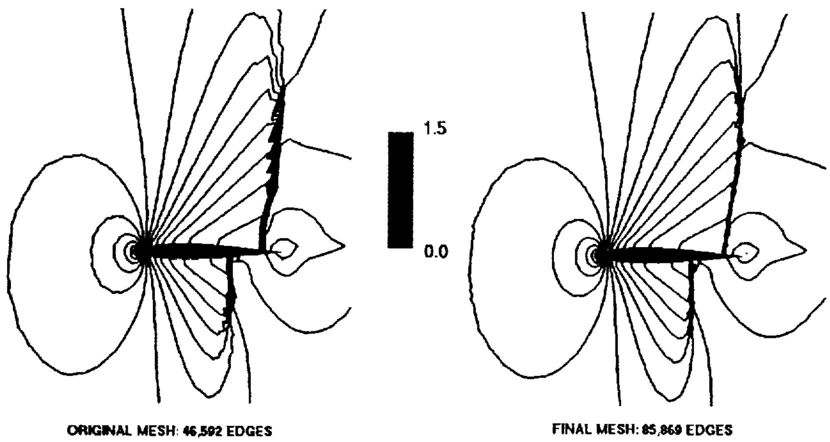
EXAMPLE: 3-D ADAPTIVE GRID REFINEMENT AND COARSENING

FSMACH = 0.85, ALPHA = 1.0 DEG

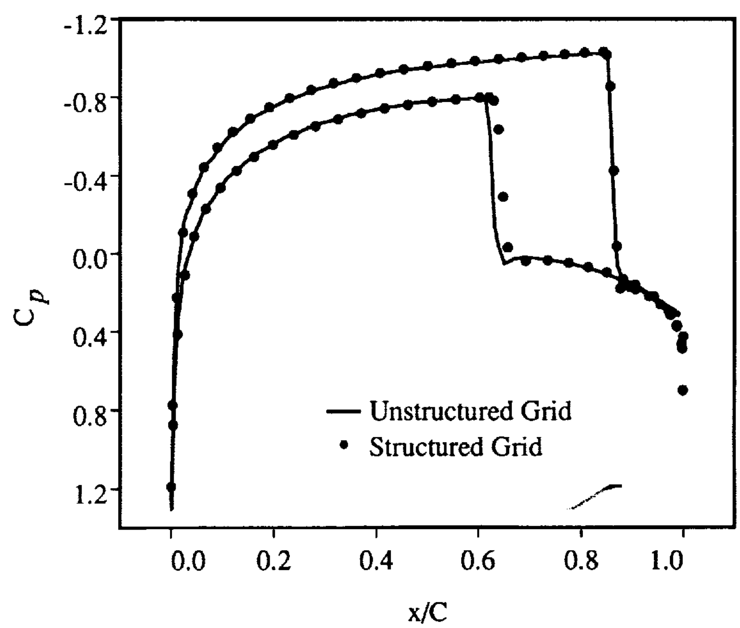


MACH NUMBER CONTOURS

FSMACH = 0.85, ALPHA = 1.0 DEG.

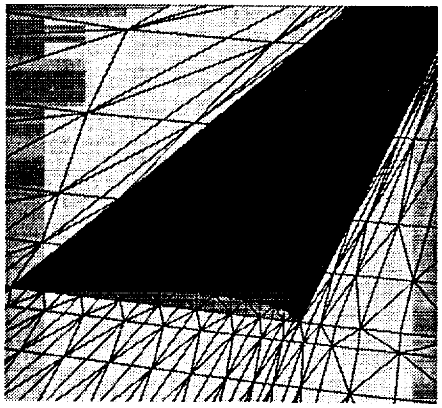


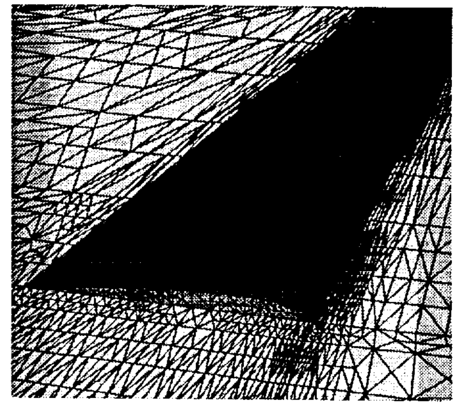
Example: Inviscid 3-D Wing



SOLUTION -ADAPTED MESH FOR A HOVERING ROTOR

Mtip - 0.90, AR - 13.7, NONLIFTING BLADE



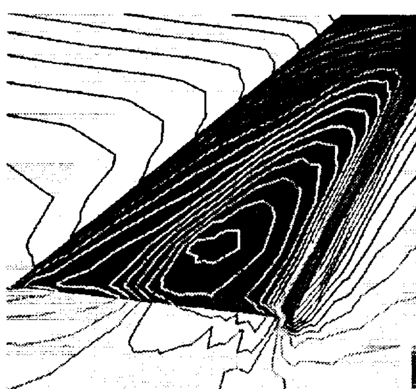


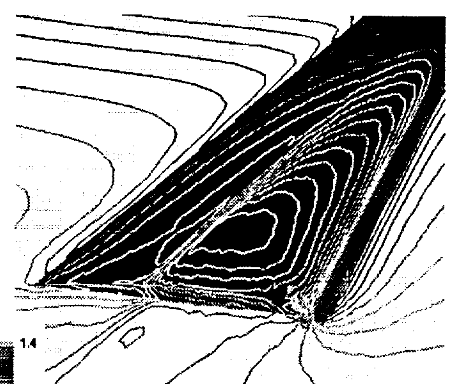
INITIAL MESH: 5,267 POINTS, 28,841 EDGES

FINAL GRID: 27,494 NODES, 172,974 EDGES
3 REFINEMENT LEVELS
2 COARSENING LEVELS

MACH CONTOURS FOR THE ROTOR BLADE

Mtip - 0.90, AR - 13.7, NONLIFTING BLADE





INITIAL MESH: 5,267 POINTS, 28,841 EDGES

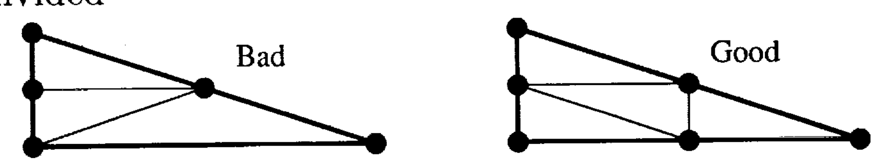
FINAL GRID: 27,494 NODES, 172,974 EDGES
3 REFINEMENT LEVELS
2 COARSENING LEVELS

Current Projects

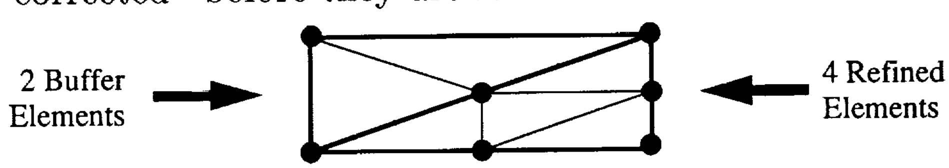
- Mesh quality for 2-D and 3-D adaptive schemes — Goal is to guarantee that mesh quality does not degrade
- Concurrent operation of flow solver and dynamic mesh adaption on CM-5
- Error estimates/indicators for unstructured-grid solutions

Mesh Quality for Solution-Adaptive Grids

- Elements are checked for quality before they are actually subdivided

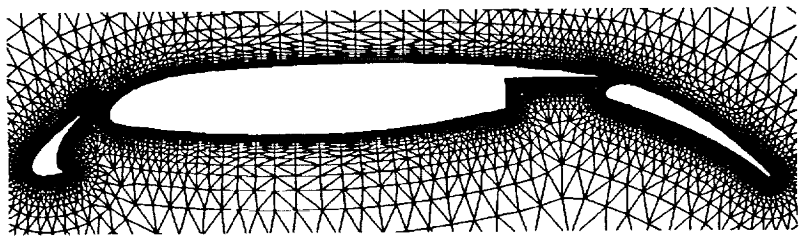


- Buffer elements with large angles that may result at boundaries between different refinement levels are "corrected" before they are further subdivided

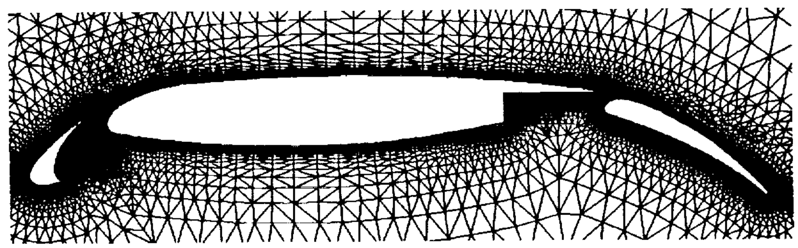


- Both techniques can be used in two and three dimensions

MESH ADAPTION FOR A 2-D VISCOUS GRID



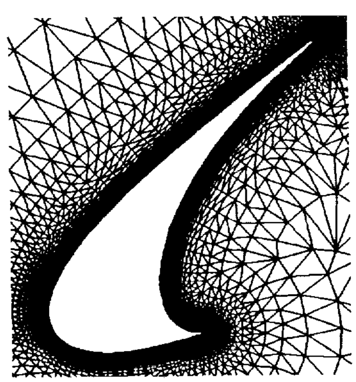
ORIGINAL GRID: 27,705 NODES, 54,725 TRIANGLES



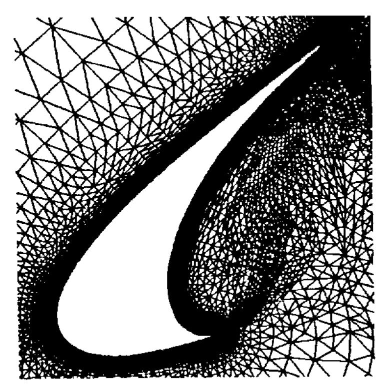
3 REFINEMENT LEVELS, 2 COARSENING LEVELS: 73,142 NODES, 144,270 TRIANGLES

MESH ADAPTION FOR A 2-D VISCOUS GRID

CLOSE-UP OF FIRST AIRFOIL ELEMENT



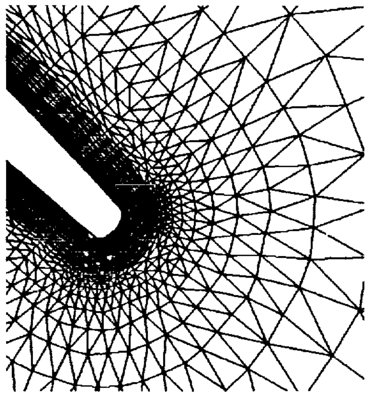
ORIGINAL GRID: 27,705 NODES, 54,725 TRIANGLES



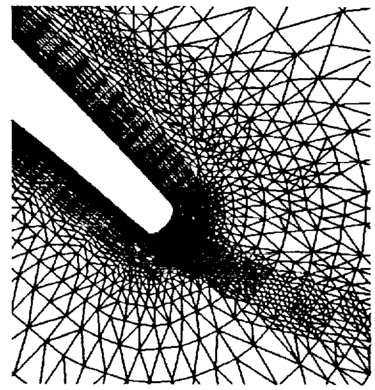
3 REFINEMENT LEVELS, 2 COARSENING LEVELS:
73,142 NODES, 144,270 TRIANGLES

MESH ADAPTION FOR A 2-D VISCOUS GRID

TRAILING EDGE OF THIRD AIRFOIL ELEMENT



ORIGINAL GRID:
27,705 NODES, 54,725 TRIANGLES



3 REFINEMENT LEVELS, 2 COARSENING LEVELS:
73,142 NODES, 144,270 TRIANGLES

Summary and Conclusions

- A new procedure has been developed for dynamic adaption of two- and three-dimensional unstructured grids
- An innovative new data structure combined with dynamic memory allocation results in fast coarsening and refinement
- Mesh quality can be "controlled" for arbitrary refinement levels
- Computed results using the solution-adaptive algorithm show excellent agreement with results for conventional structured-grid solvers