An Adaptive 3D Cartesian Approach for the Parallel Computation of Inviscid Flow About Static and Dynamic Configurations

Jason D. Hunt, Kenneth G. Powell

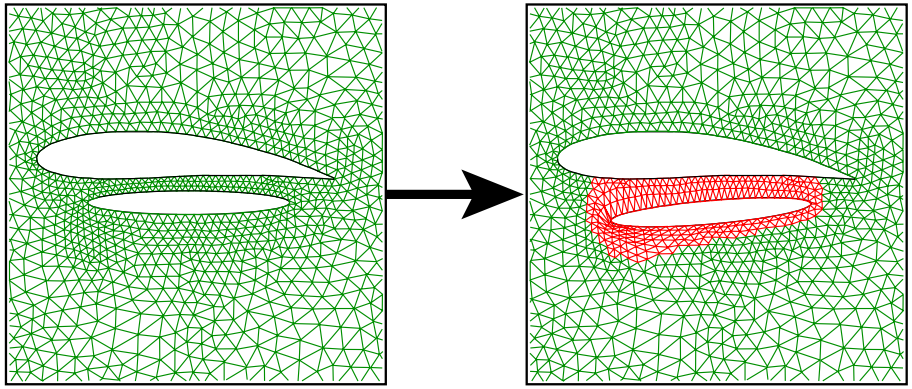
W. M. Keck Foundation Laboratory for Computational Fluid Dynamics
Department of Aerospace Engineering
The University of Michigan

Funded by: DOE CSGF, NASA GSRP

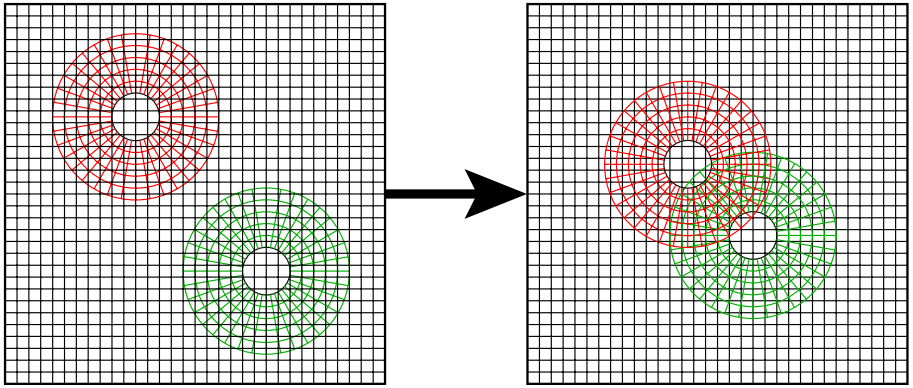
Motivation & Background

- Computational Fluid Dynamics (CFD)
  - Becoming a mature field
  - Problems involving moving or deforming objects remain a challenge
  - Unstructured grid approach
    - Grid shearing in regions of large relative motion
    - Grid generation is sensitive to body definition
  - Chimera approach
    - Data interpolation is difficult in regions between close objects
    - Body-fitted grids are sensitive to body definition

Unstructured Grid Approach



Overlapping Grids Approach



Approach Foundation

- Cartesian grid approach
  - Independent to body definition
  - Very little user interaction
- Approach requirements
  - Inviscid compressible flow
  - Three-dimensional Cartesian framework
  - Arbitrary geometric configurations
  - Adaptive mesh refinement
  - Parallel computation

- Inviscid compressible flow
  - Euler Equations

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{U}\, d\Omega + \int_{S} \boldsymbol{\mathcal{F}}_r \cdot \mathbf{n}\, dS = 0$$

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$$

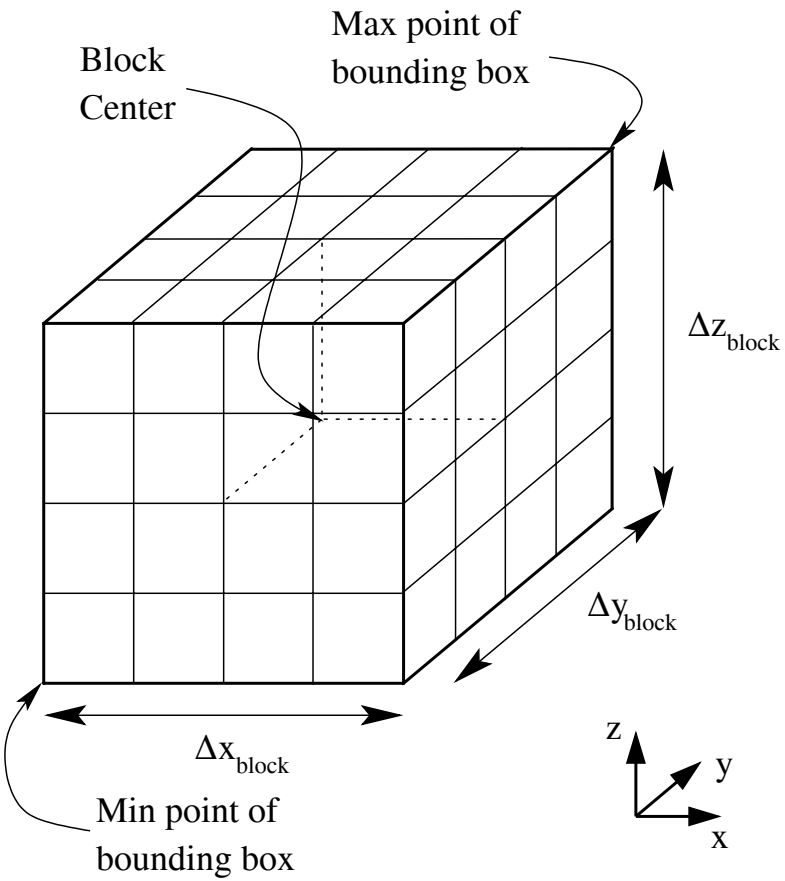
$$\boldsymbol{\mathcal{F}}_r = \begin{pmatrix} \rho(\mathbf{v} - \mathbf{v}_s) \\ \rho u(\mathbf{v} - \mathbf{v}_s) + p\hat{\mathbf{i}} \\ \rho v(\mathbf{v} - \mathbf{v}_s) + p\hat{\mathbf{j}} \\ \rho w(\mathbf{v} - \mathbf{v}_s) + p\hat{\mathbf{k}} \\ (\rho E + p)(\mathbf{v} - \mathbf{v}_s) + p\mathbf{v}_s \end{pmatrix}$$

  - Equation of state

$$p = \rho e(\gamma - 1)$$

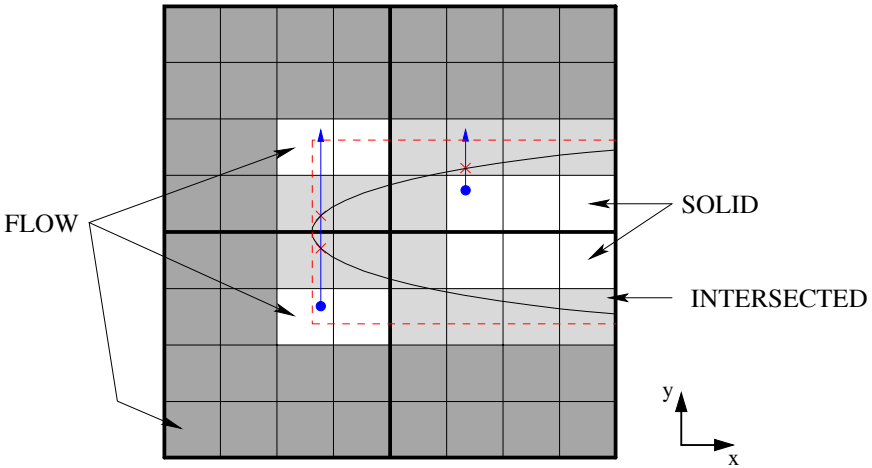
Approach Foundation

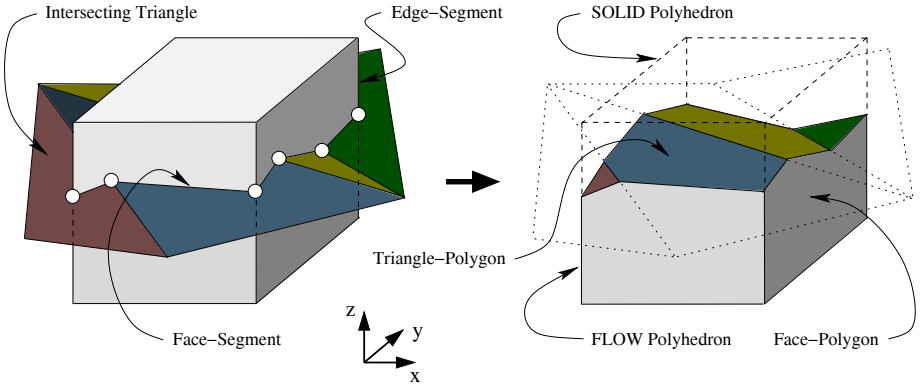
- Three-dimensional Cartesian framework
  - A spatial region is represented by a block of N x N x N cells
  - The cells constitute a structured Cartesian grid
  - A finite-volume flow solver is applied to each cell
    - The MUSCL approach is used to achieve higher order spatial accuracy
    - Two-stage Runge-Kutta time integration utilized
    - Fluxes obtained via Roe's approximate Riemann solver



Approach Foundation

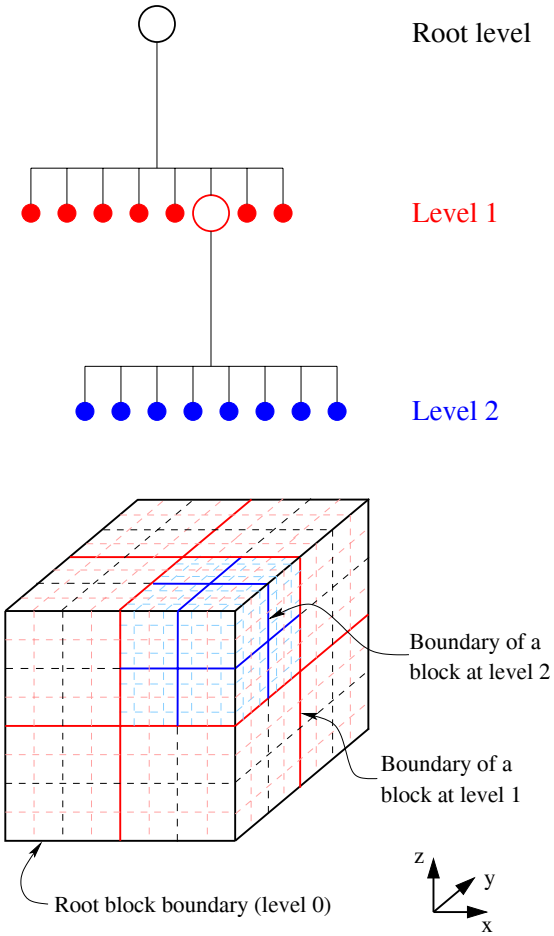
- Arbitrary Geometric Configurations
  - Configuration components are defined by closed triangulated surfaces
  - Cells must be identified as one of three types: ***flow***; ***solid***; or ***intersected***
  - Intersected cells must be ***cut***
    - Cart3D – Aftosmis, NASA Ames
  - Introduced issues
    - Prohibitively small time steps may result from cut-cells with a very small volume
    - ***Split*** cells can also be produced





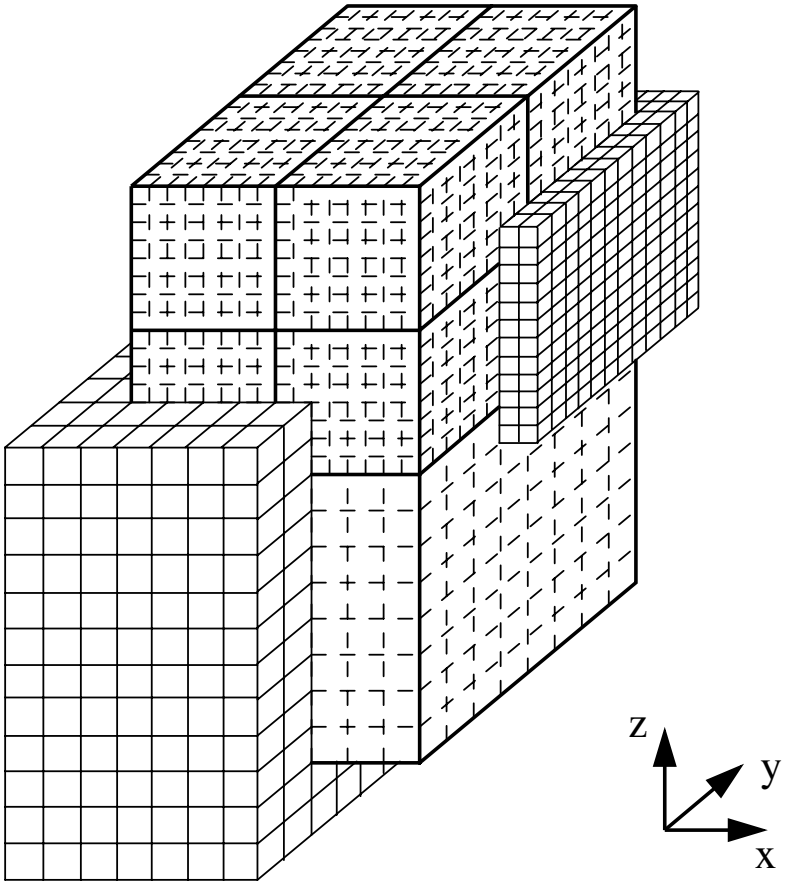
Approach Foundation

- Adaptive Mesh Refinement
  - Utilizes a block-octtree data structure
    - A block of cells is stored in each node of an octtree
    - Blocks deeper in the tree represent smaller sub-regions within the domain
  - Flow-based adaptation
    - Adapts to flow features such as shock and expansion waves
  - Geometry-based adaptation
    - Adapts to geometric features such as local surface curvature



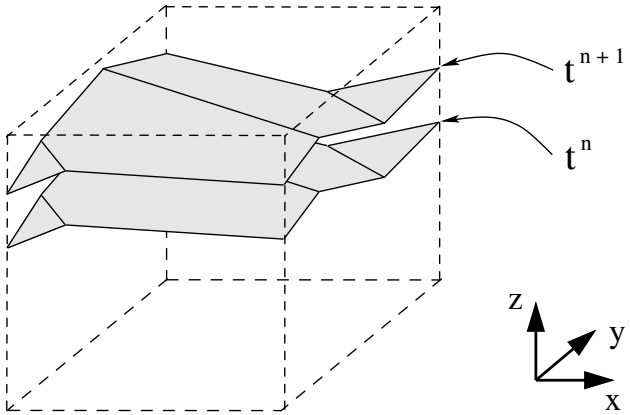
Approach Foundation

- Parallel Computation
  - PARAMESH – MacNeice, NASA Goddard
  - Blocks are distributed across processors to balance work
  - Each processor maintains a copy of the geometric configuration
  - Layers of ghost cells are used to facilitate processing each structured block independently



Dynamic Configurations

- Component motion
  - Components can be moved independently
  - Motion restricted to prescribed rigid-body motion
- Necessary considerations
  - Runge-Kutta formulation permits a varying control volume
  - Cell geometry needed at three instances during an update
  - Time step calculation and flux computation must now include facial velocities



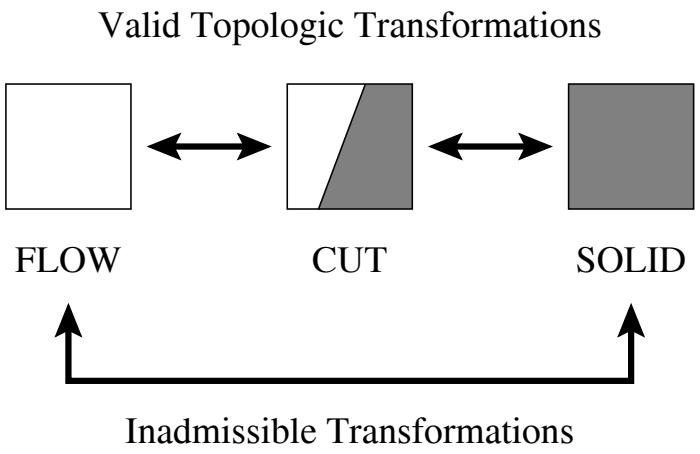
$$\mathbf{Res}\left(\overline{\mathbf{U}}V\right) \equiv -\sum_{i=1}^{nFaces} \mathbf{\Phi}_i\, A_i$$

$$(\overline{\mathbf{U}}V)^{(1)} = (\overline{\mathbf{U}}V)^{(0)} + \frac{\Delta t}{2}\,\mathbf{Res}\left((\overline{\mathbf{U}}V)^{(0)}\right)$$

$$(\overline{\mathbf{U}}V)^{(2)} = (\overline{\mathbf{U}}V)^{(0)} + \Delta t\,\mathbf{Res}\left((\overline{\mathbf{U}}V)^{(1)}\right)$$

Dynamic Configurations

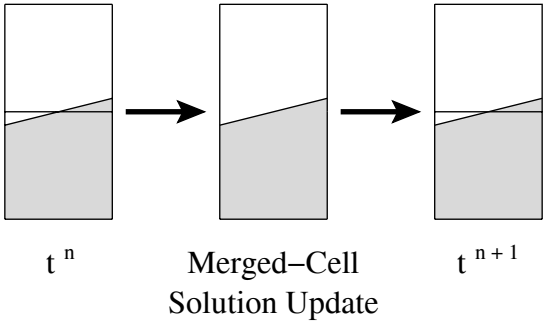
- Topologic Transformations
  - Cell volumes can appear or disappear during a solution update
  - Results from a cell transforming from solid to cut or vice versa
  - The formulation of the time step calculation does not admit transformations between flow and solid cells
  - Runge-Kutta time integration can not tolerate this



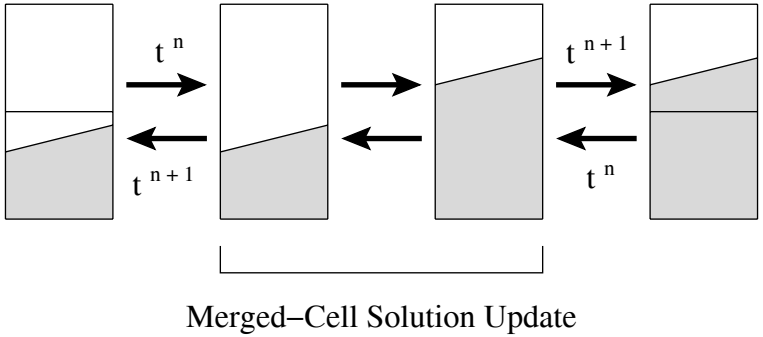
Cell Merging

- Motivation for cell merging
  - Prohibitively small time steps
  - Cell-type transformations during a time step
- Concept of cell merging
  - Multiple simply connected cells are grouped together to avoid flow solving issues
  - Each group of cells is treated as an individual composite cell during a solution update
  - At the end of the update each member cell receives its appropriate portion of the updated solution

Static Configuration

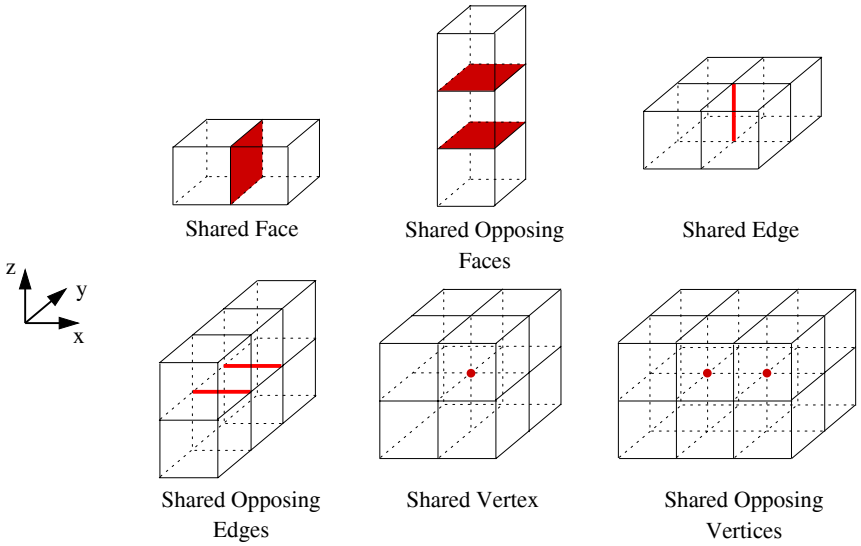


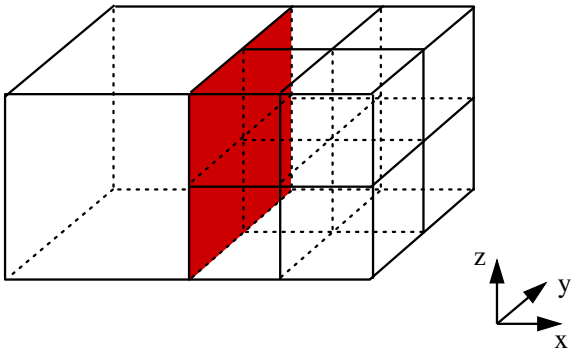
Dynamic Configuration



Cell Merging

- Implementation requirements
  - Time complexity similar to that of a solution update
  - Accommodate adaptive mesh refinement
  - Parallelizable
- Cell-merging algorithm core
  - Identify problematic cells
  - Generate and score valid merging choices for each problem cell
  - Choose a merged-cell cover





Cell Merging

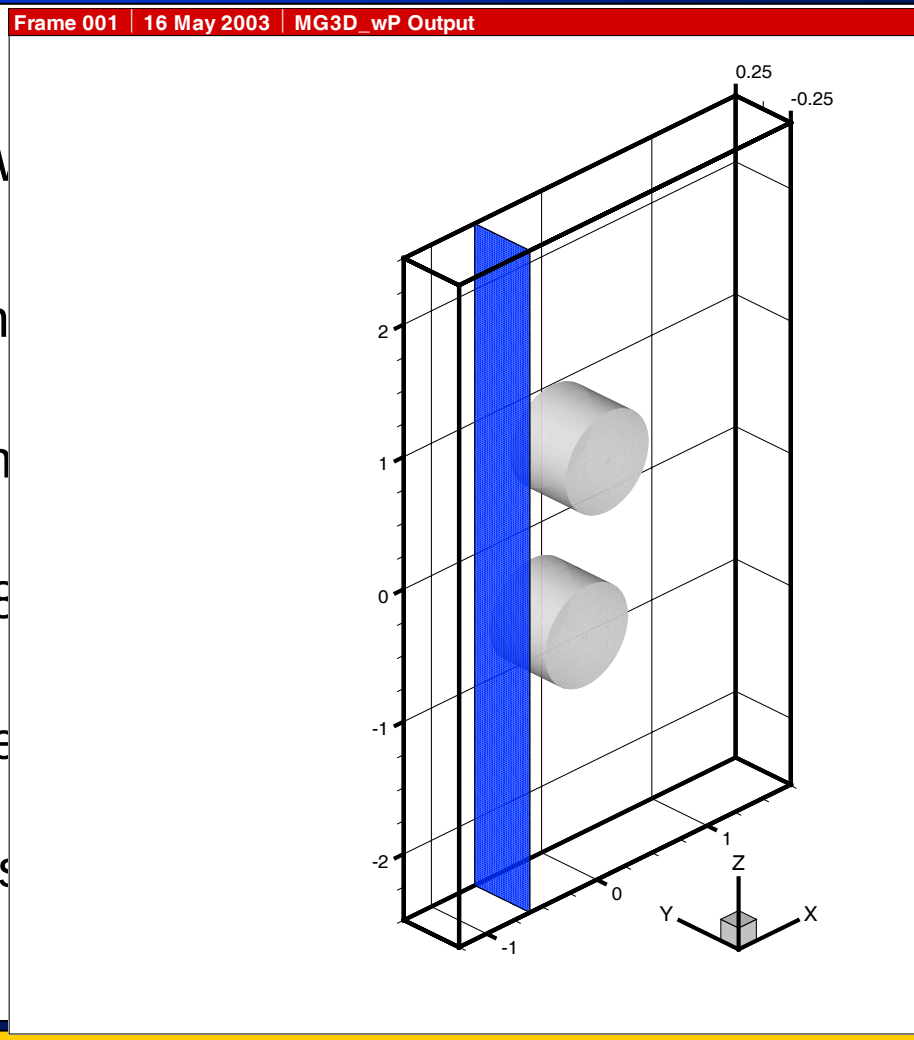
- Choosing a merged-cell cover
  - A merged-cell cover is a set of merged-cells that satisfies the following conditions:
    - Every problem cell is part of a merged cell or is ***covered***
    - No merged cells overlap
  - For each problem cell within the considered region, choose the best choice that does not introduce an overlap
  - If all the problem cells were not covered, make a second pass and try to choose the best choice that covers an uncovered cell while not introducing an overlap
- Parallelization
  - Cell merge each block independently
  - Cell merge within larger regions, as necessary, by traversing back through the octtree
  - As a last resort merge the grid as a whole

Cell-Merging Usage

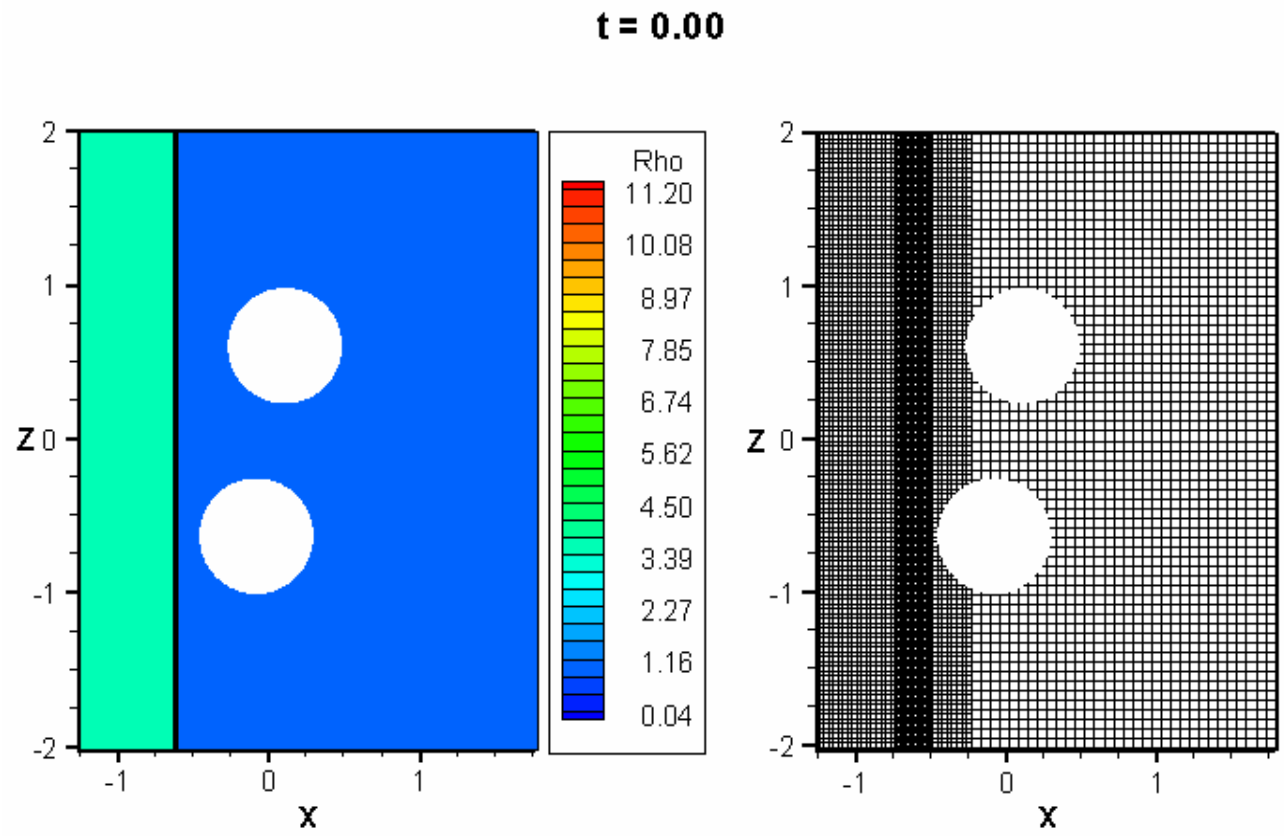
- With a static configuration:
  - Cell merge once at the beginning of the simulation
  - Cell merge again only if flow adaptation alters the grid
- With a dynamic configuration:
  - Cell merge every time step
  - Circular dependency exists between the global time step and the computed merged-cell cover
    - Sometimes a viable merged-cell cover can be computed by assuming a stationary configuration
    - Starting from this assumption, the dependency is resolved by iterating to find a viable merged-cell cover for the associated time step prior to performing solution update

Computational Results

- Shock-Wave Interaction with Tv
  Cylinders
  - Recreated from an example given
    LeVeque in AIAA 89-1930-CP
  - Two cylinders are positioned such
    slightly ahead of the other
  - A shock wave moving at Mach 2.8
    with the cylinders
  - An animation of the normalized de
    contours is presented on the next
    through a simulation time of 0.06 s

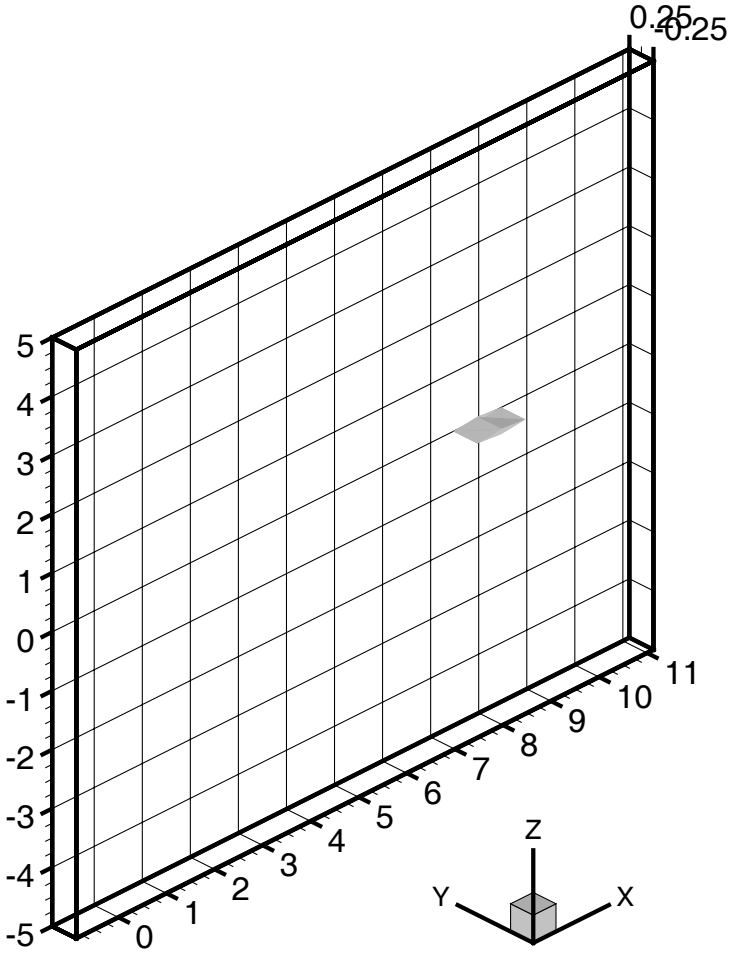


Computational Results

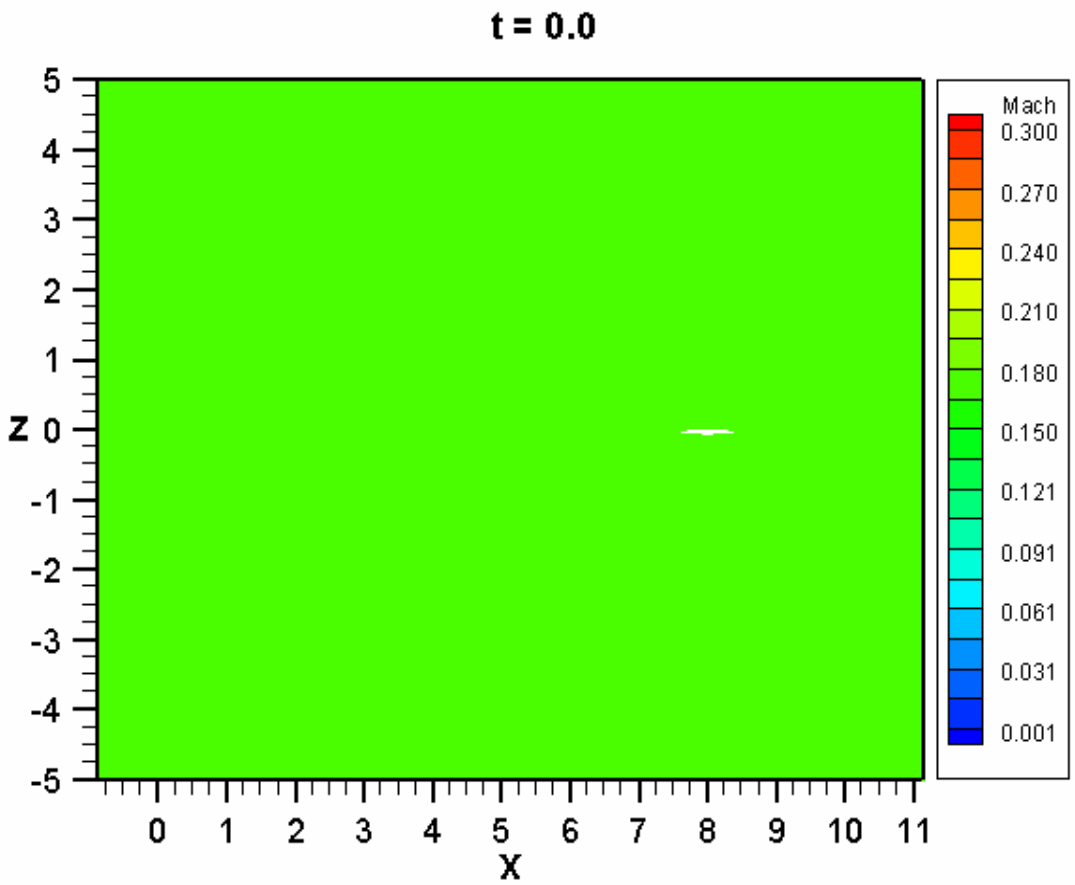


Computational Results

- Moving Symmetric Diamond Airfoil
  - Diamond profile has a 5° half-angle
  - Airfoil is impulsively given a velocity
  - Equivalent flow relative to the airfoil: Mach 2 at a 5° angle-of-attack
    - Airfoil moves horizontally
    - Ambient air has a non-zero vertical component
  - An animation of the Mach contours is presented on the next slide through a simulation time of 4.0 sec.



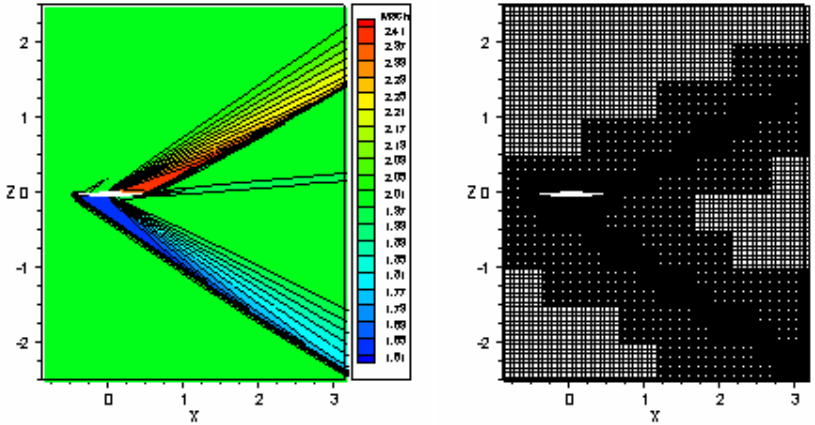
Computational Results



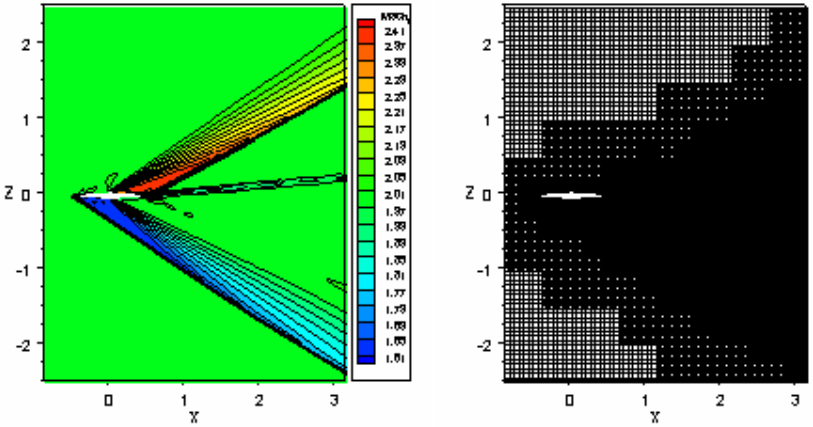
Computational Results

- Comparison to an equivalent steady-state simulation
  - Relative Mach numbers are plotted
  - Results for the fully developed regions above and below the wake are in good agreement
  - The wake region results also show similarities, but the wake region has not become fully developed yet
  - Much more refinement is evident in the wake region of the moving case because the flow is not fully developed

Steady-State Results

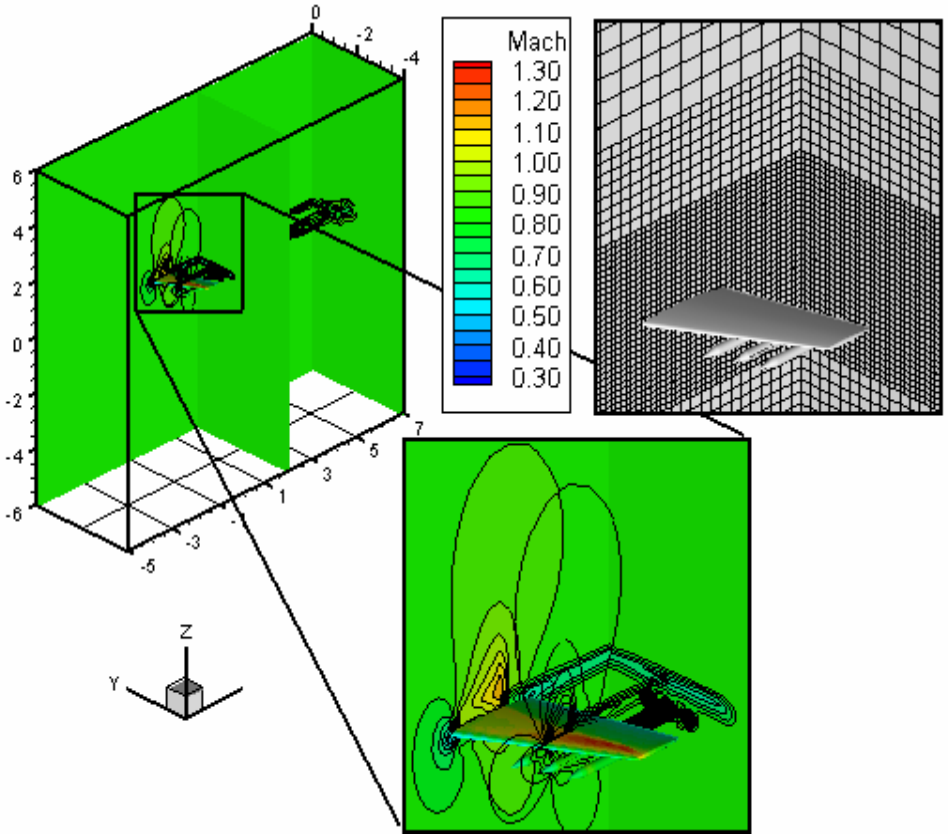


Moving Airfoil, t = 4.0

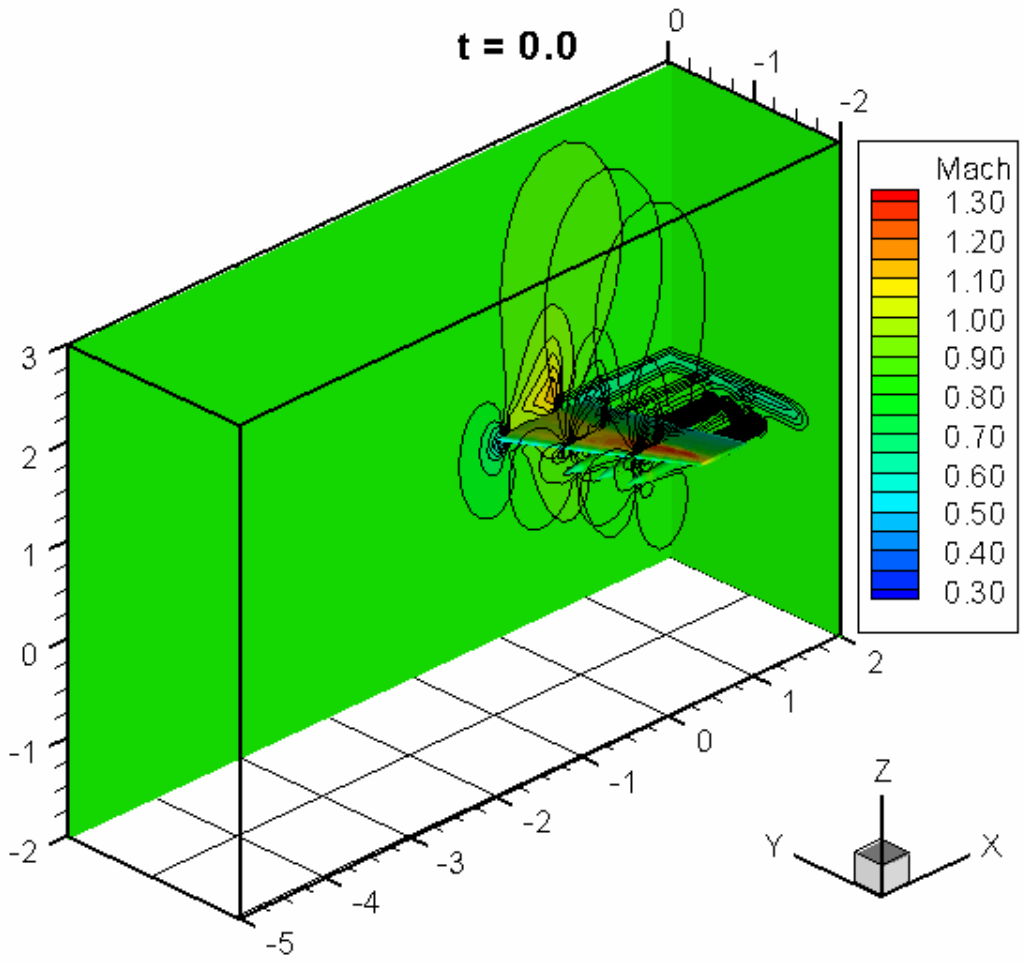


Computational Results

- Ordinance Firing
  - Transonic flow over an Onera M6 wing with three under-the-wing ordinances
  - Two ordinances fired at separate times
    - The outermost ordinance is fired at t = 0.0
    - The innermost ordinance is fired at t = 3.0
  - An animation of the Mach contours is presented on the next slide through a simulation time of 6.0 sec.



Computational Results



Conclusions & Future Work

- Conclusions
  - Developed a parallel block-adaptive Cartesian code to compute compressible flow about static configurations
  - Implemented dynamic configurations
  - Developed a cell-merging algorithm within the parallel block-adaptive Cartesian framework
  - Demonstrated the capabilities of the approach
- Future Work
  - Incorporate split cells
  - Sophisticate permissible component motion
  - Implement the use of hybrid prismatic-Cartesian grids to solve viscous flow problems