Unsteady Flow Simulation With Moving Boundaries

O. Hassan, K. Morgan and N. P. Weatherill

School of Engineering, University of Wales

Swansea, United Kingdom

CENTRES OF EXCELLENCE
CANOLFANNAU RHAGORIAETH

PRIFYSGOL CYMRU ABERTAWE
UNIVERSITY OF WALES SWANSEA

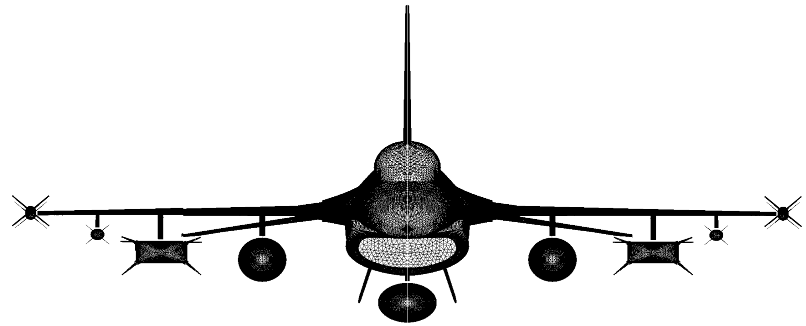
Unstructured Mesh Generation Techniques

Solution Algorithm

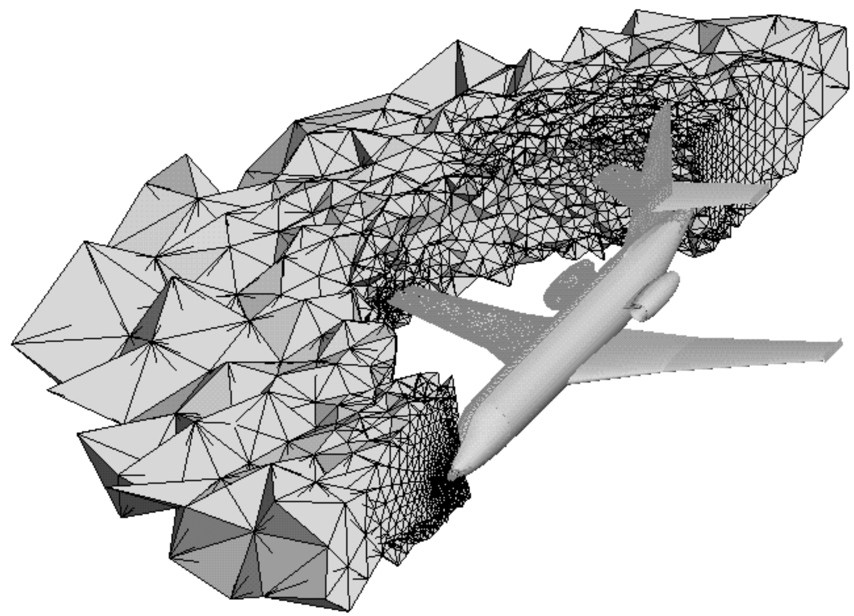
Mesh Adaptation for Unsteady Flow Problem with Moving Boundary

Parallel Implementation

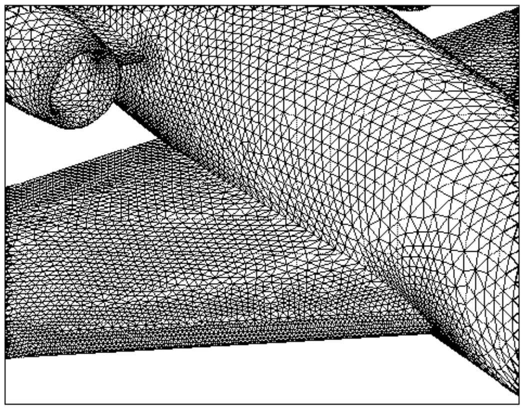
Conclusion

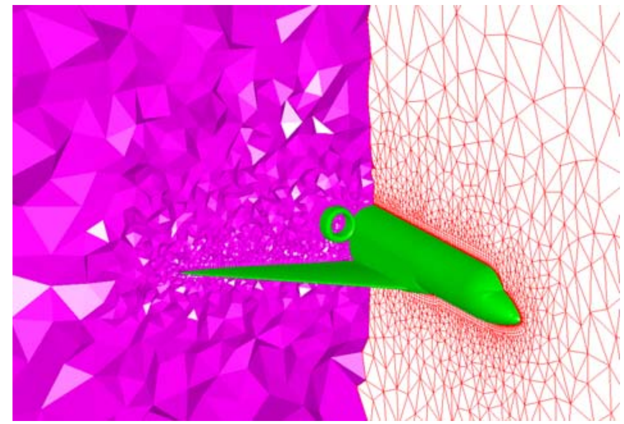


Unstructured grid technology provides the required flexibility for these (and other) applications

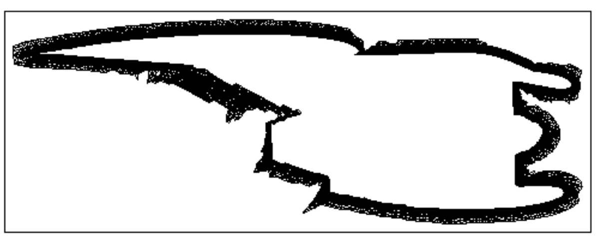


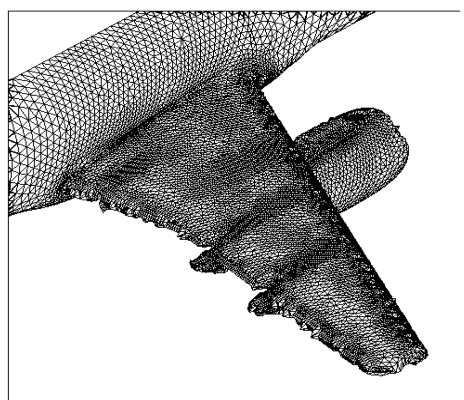
- Surface mesh generation: Advancing Front
- Volume mesh generation: Delaunay Triangulation with automatic point insertion (**Requires 100Mb / 10^6 elements**)

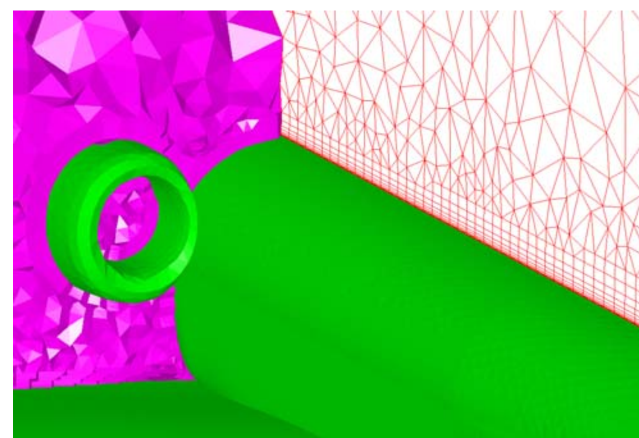




- Boundary layer generation: Hybrid meshes by the Advancing Layer method







The Favre Averaged Navier Stokes Equations

$$\frac{d}{dt}\int_{\Omega(t)} \mathbf{U}d\mathbf{x} + \int_{\Gamma(t)} \left(\mathbf{F}_j - v_j\mathbf{U}\right)n_j\, d\mathbf{x} = \int_{\Gamma(t)} \mathbf{G}_j n_j d\mathbf{x}$$

Where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho\varepsilon \end{bmatrix} \qquad \mathbf{F}_j = \begin{bmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ u_j(\rho\varepsilon + p) \end{bmatrix} \qquad \mathbf{G}_j = \begin{bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \\ u_k\tau_{jk} - q_j \end{bmatrix}$$

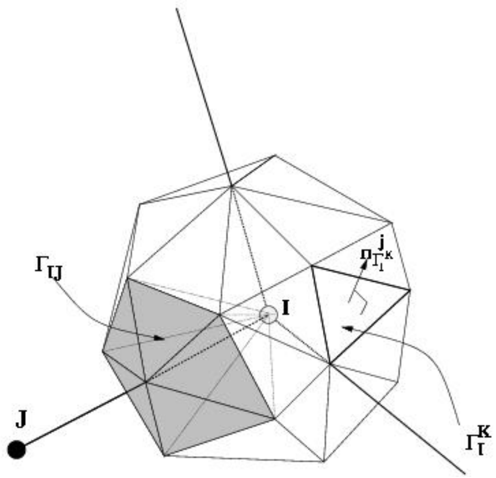
and

$$\varepsilon = c_v T + \frac{1}{2}u_k u_k \qquad p = \rho RT$$

- Turbulent is modelled by adding the one equation model of Spalart and Allmaras

- Edge Based Data Structure is used for the evaluation of the fluxes
- The **ALE** term is evaluated ensuring numerical fluxes that satisfies **GCL**
- Resulting Equation

$$\frac{d}{dt}[V^I \mathbf{U}^I] = R^I$$



- For unsteady problems, the second order approximation

$$\frac{d}{dt}\int_{\Omega(t)} U_i d\Omega \Bigg|_{t=t_{n+1}} = \frac{1}{\Delta t}\left(\frac{3}{2}V_I^{n+1}\mathbf{U}_I^{n+1} - 2V_I^n\mathbf{U}_I^n + \frac{1}{2}V_I^{n-1}\mathbf{U}_I^{n-1}\right)$$

is adopted

  - An implicit formulation is employed and this removes the stability constraints associated with explicit schemes
  - At each time step, the equation is solved by explicit relaxation with multi-grid acceleration
  - This approach can be thought of as converging the set of steady state equations with the addition of the time source for every physical timestep
  - No significant memory penalties compared to explicit procedures

- The turbulent viscosity equation is discretised in a similar fashion

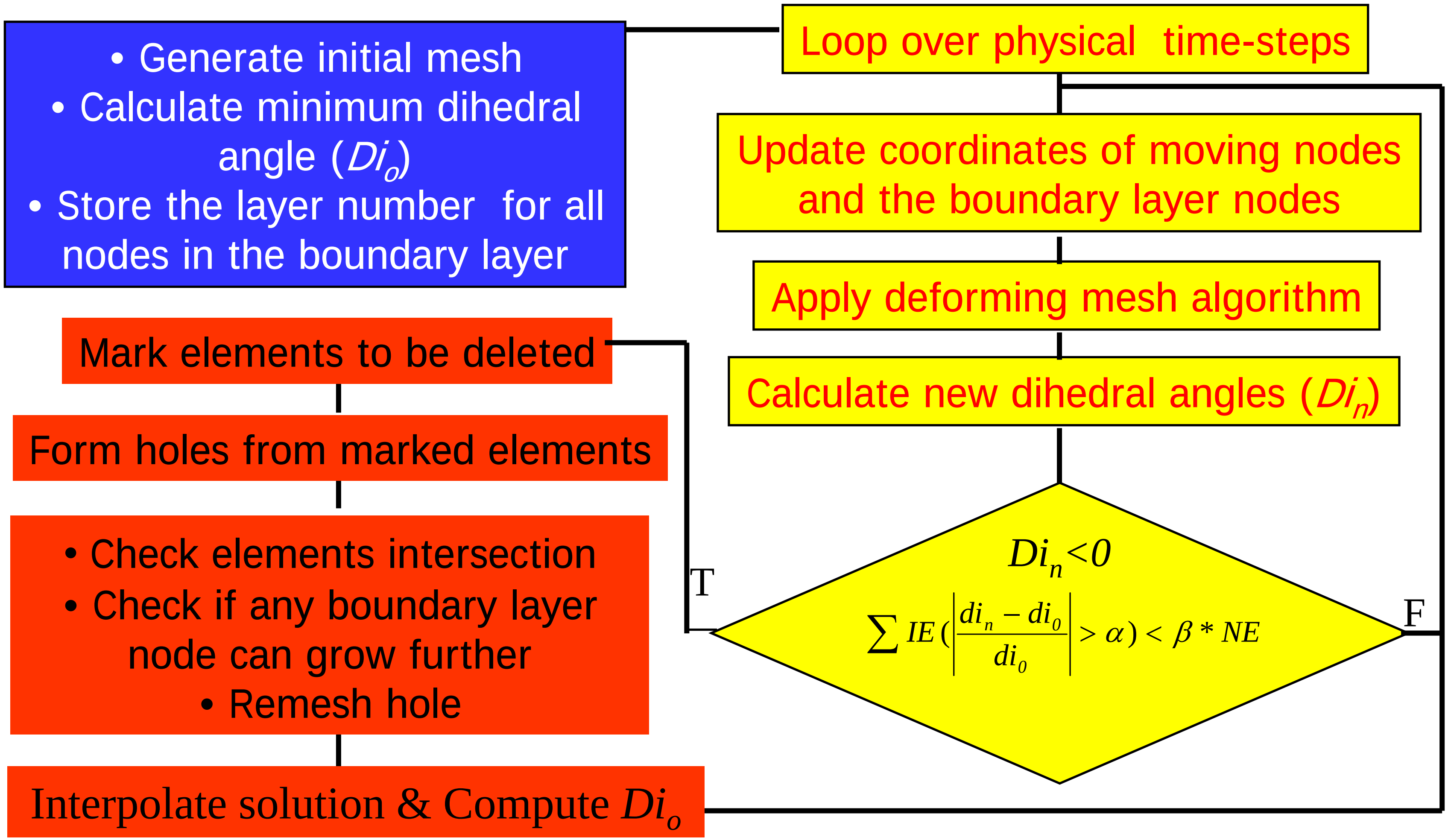
- Stabilisation achieved by replacing the actual flux function by JST flux function

- Discontinuity capturing achieved by the addition of a switched artificial diffusion

- For steady state Runge-Kutta relaxation and local timestepping is utilised

- Convergence acceleration is achieved by using the **Full Approximation Storage (FAS) Multigrid scheme**
  - Coarse grids are achieved by agglomeration
  - Volume weighted operator is used for restriction
  - Injection is used for prolongation

- Parallel implementation using **MPI** allows agglomeration across partitions



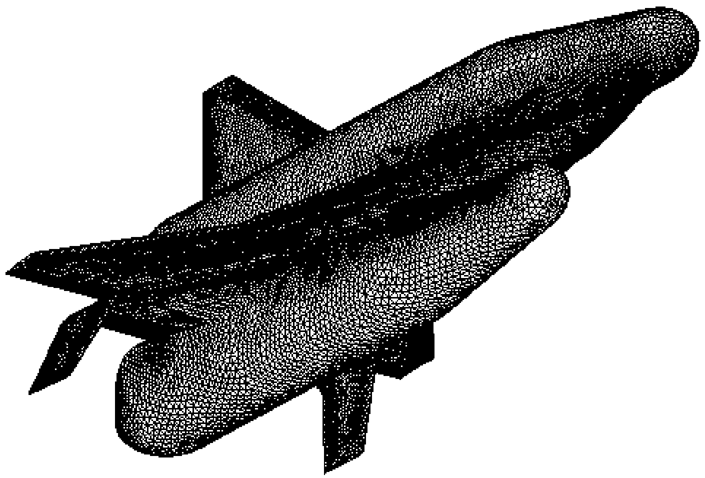
• Generate initial mesh
• Calculate minimum dihedral angle (Di_o)
• Store the layer number for all nodes in the boundary layer
Loop over physical time-steps
Update coordinates of moving nodes and the boundary layer nodes
Apply deforming mesh algorithm
Calculate new dihedral angles (Di_n)
$Di_n<0$
$\sum IE\left(\left|\frac{di_n - di_0}{di_0}\right| > \alpha\right) < \beta * NE$
T
F
Mark elements to be deleted
Form holes from marked elements
• Check elements intersection
• Check if any boundary layer node can grow further
• Remesh hole
Interpolate solution & Compute Di_o

Shuttle Booster Separation Simulation

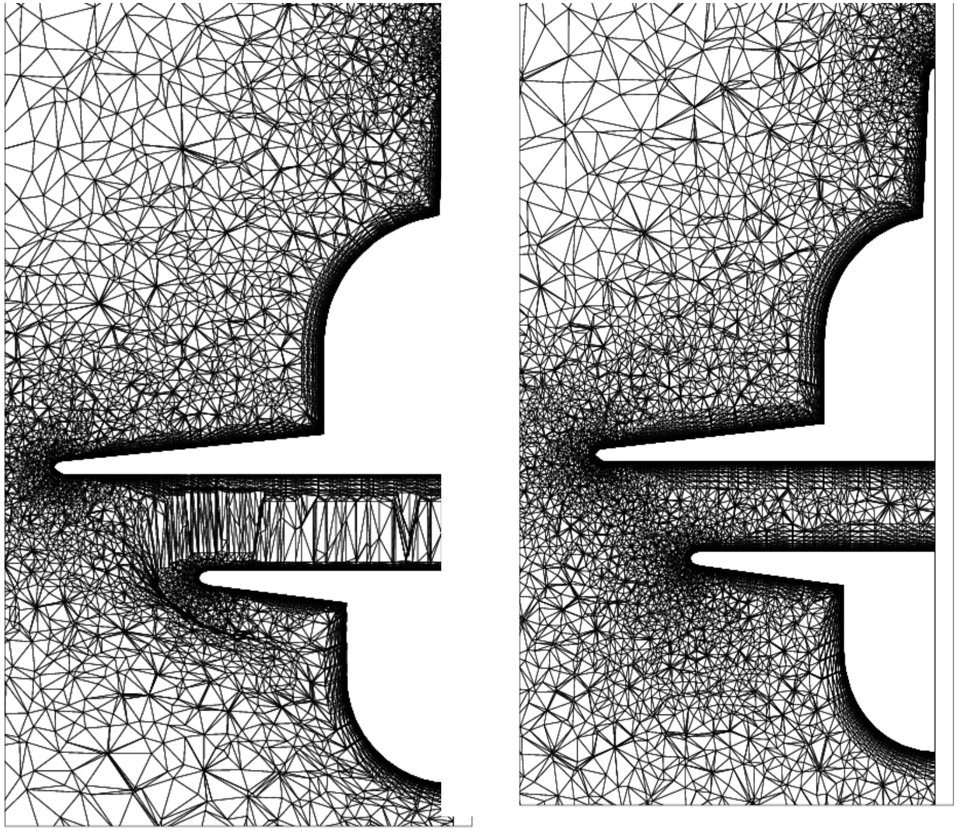
α_{init} = zero degrees
M= 0.85 Degree
Re = 3 * 10 ^{6}
Prescribed Shuttle movement

Initial mesh: 2.9 million elements
Final mesh: 3.3 million elements

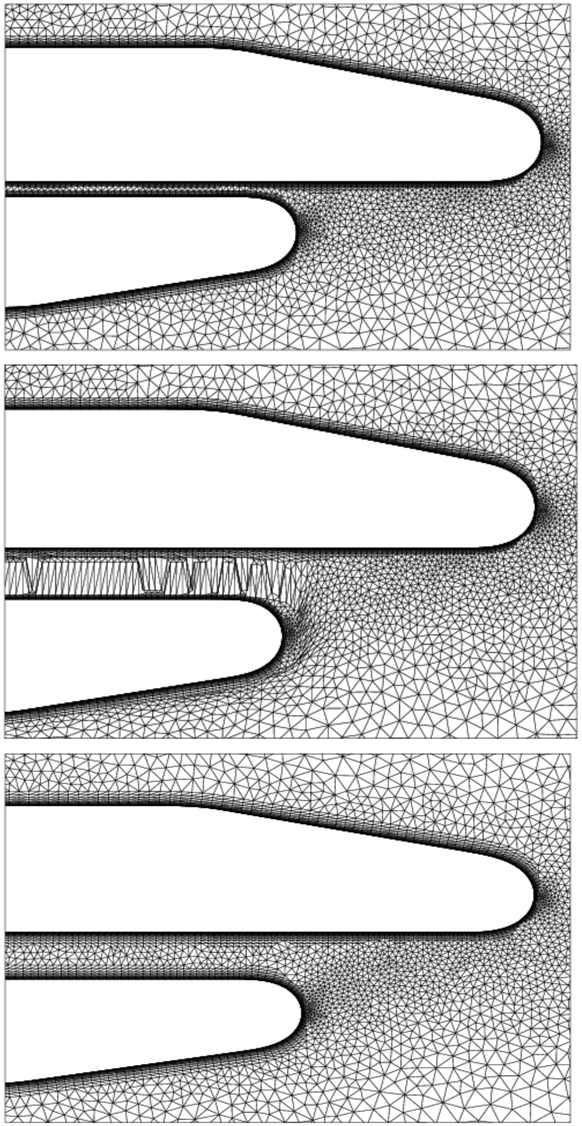
20 time steps
300 multigrid cycle per time step



Shuttle Booster Separation Simulation

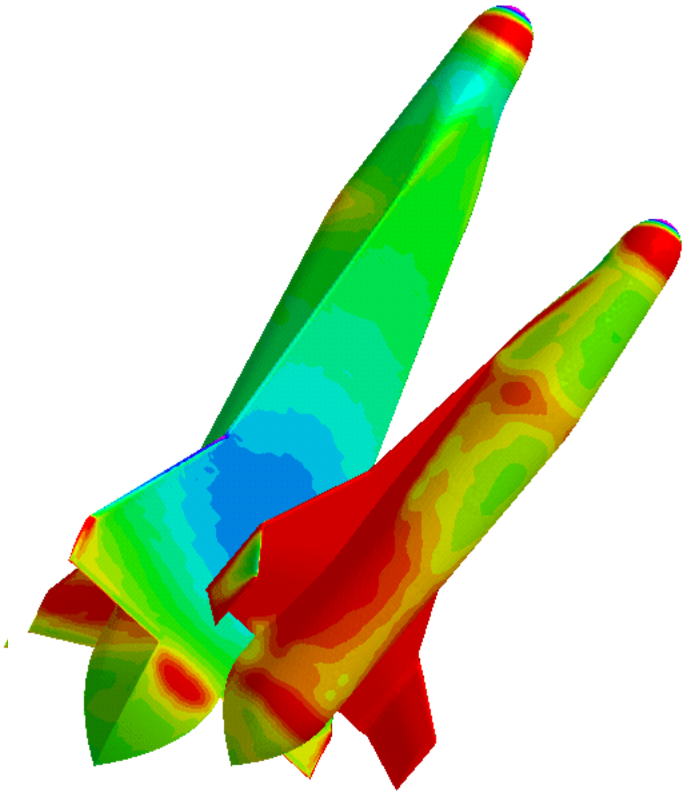


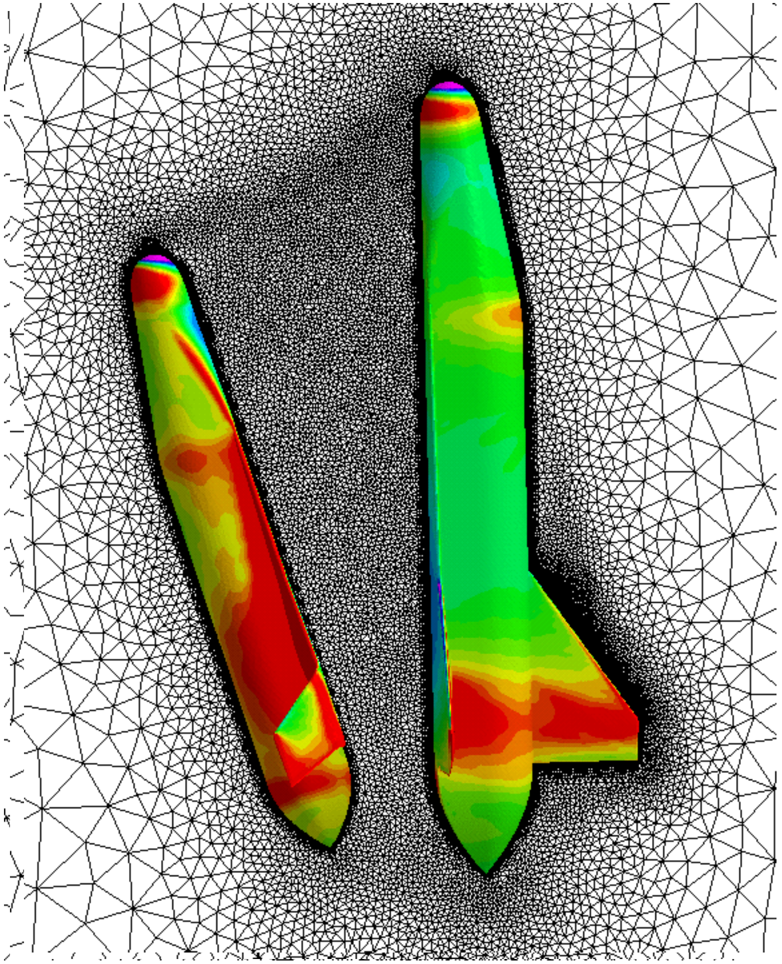
Cut through the volume mesh



Meshes on the symmetry plane

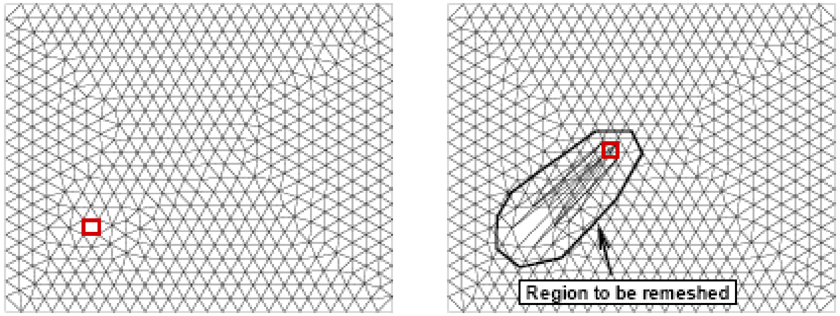
Shuttle Booster Separation Simulation





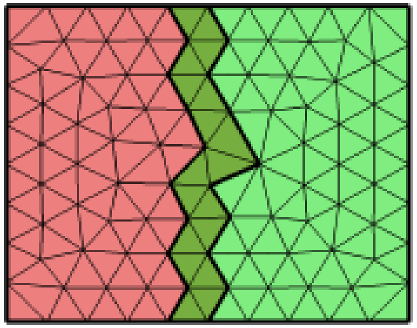
➢ **MPI** was employed to allows the parallel solution procedure to be imported to the **IBM**

➢ Elements are select to be remeshed in each domain separately
- Selection Based on Deviation from Prescribed Spacing
- Selection Based on Element Quality
- Selection Based on Intersection Tests

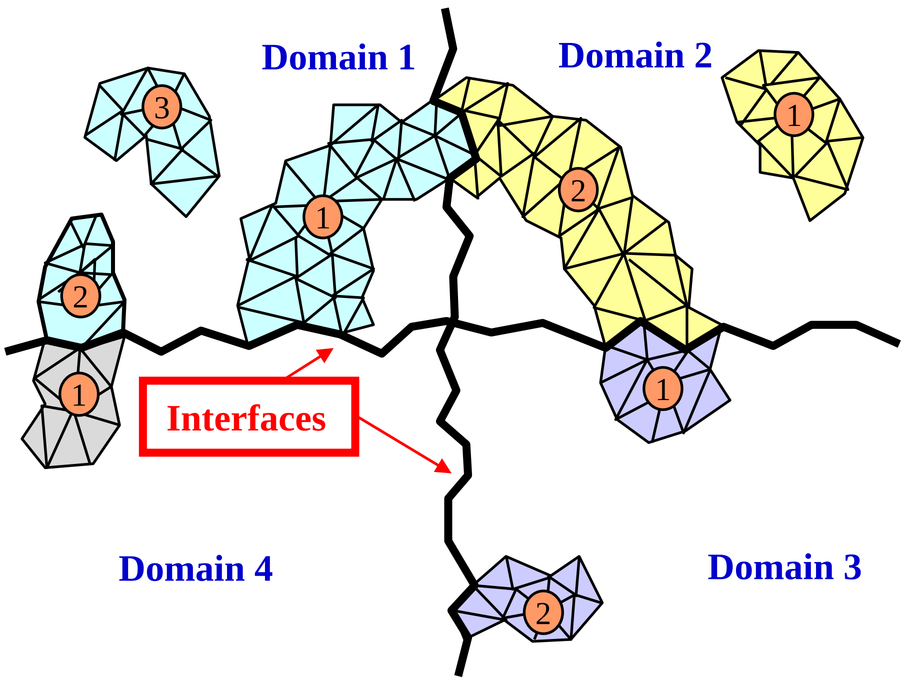


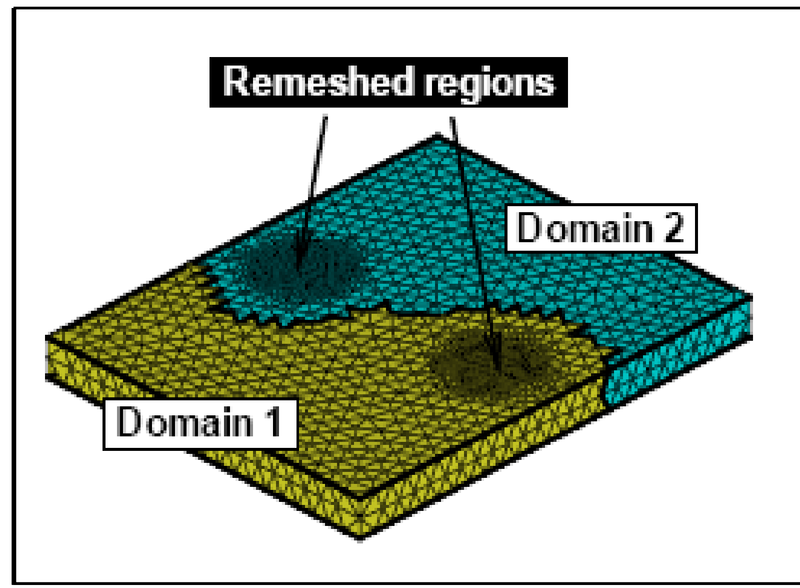
➢ To determine intersection of elements due to moving geometries, one overlapping ghost layer of elements is used.

➢ If intersection with the ghost element has occurred, the search will also take place in to the domain which own the ghost cell.

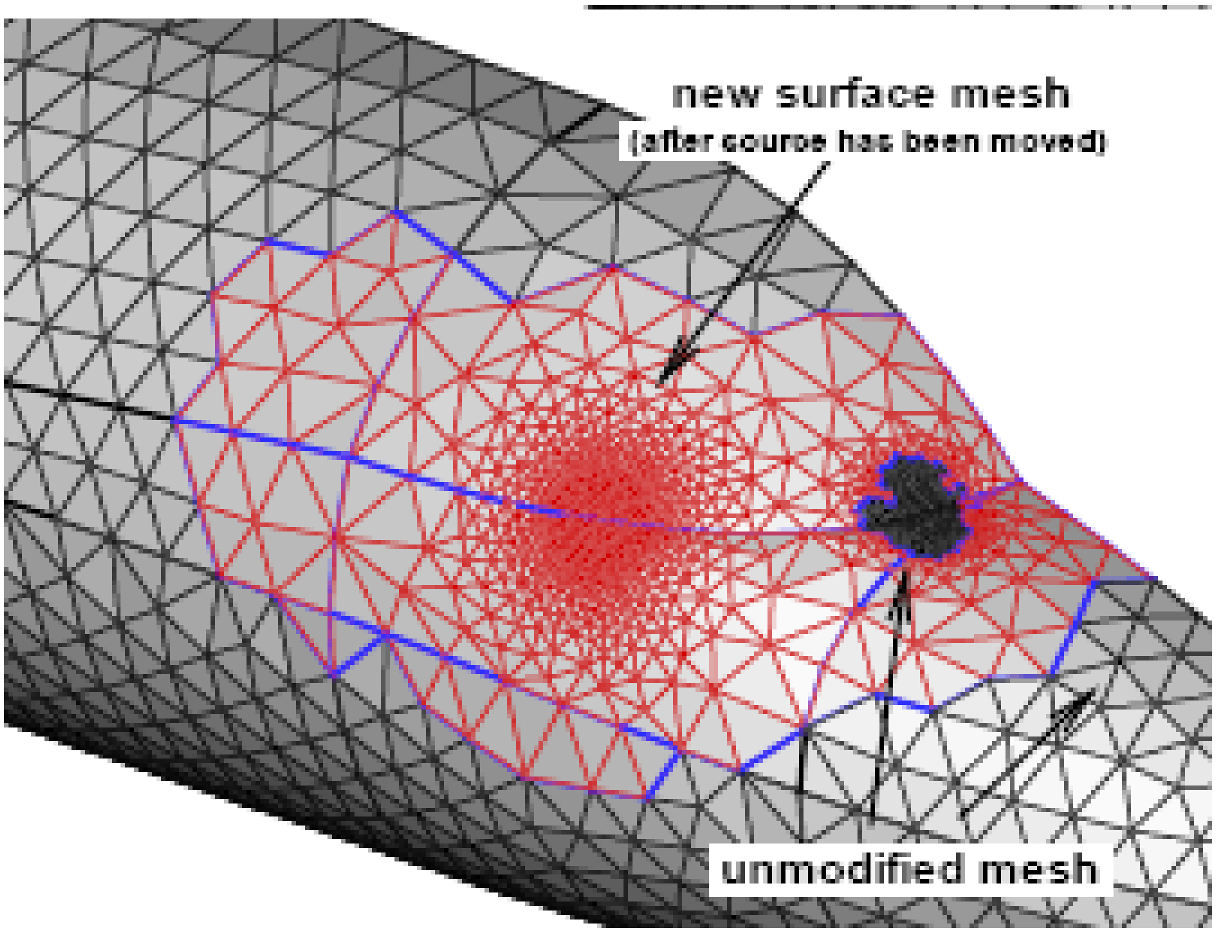


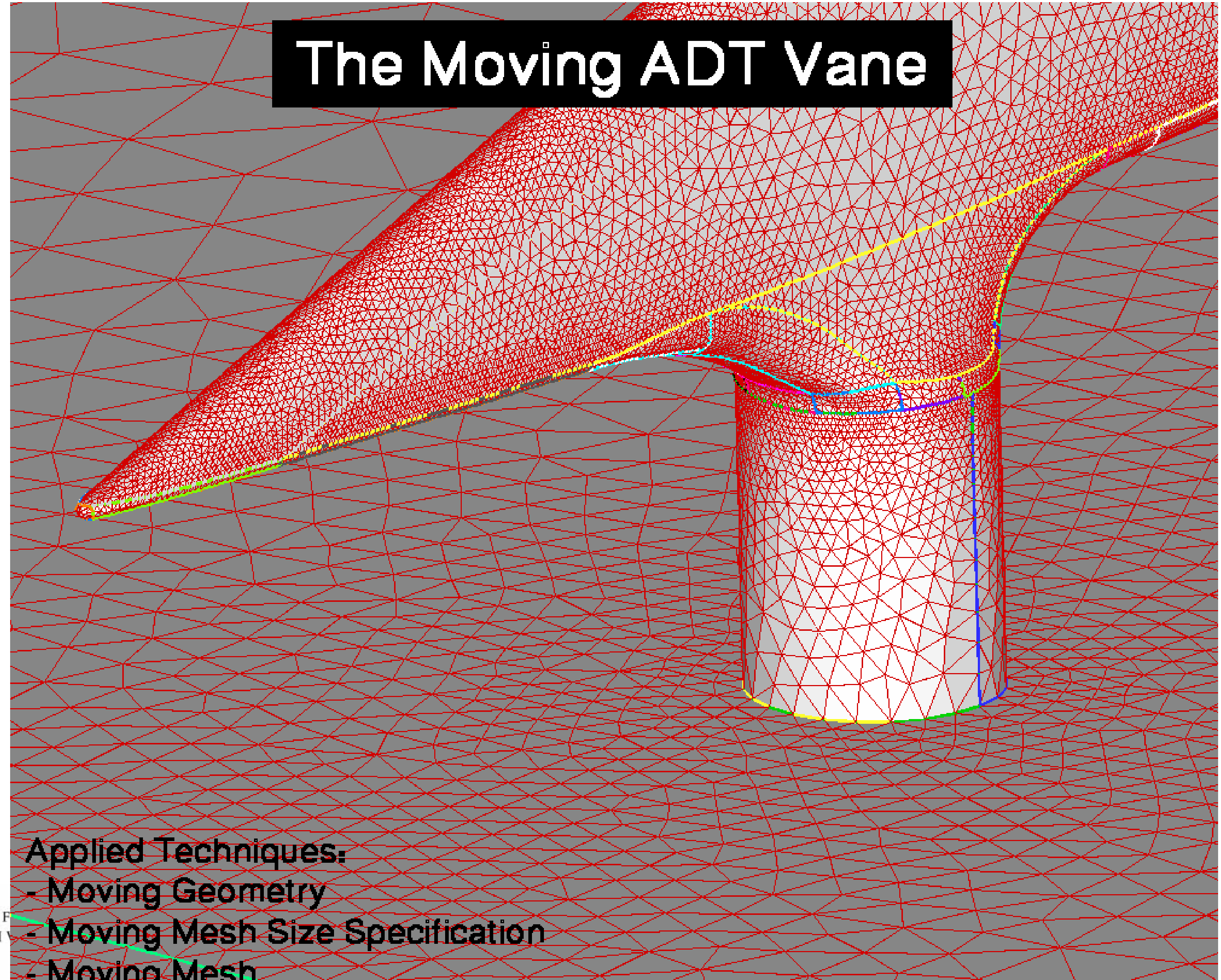
- Constraint Repartitioning is employed to ensure that each region to be remeshed will be contained completely on one process





The Geometry definition is utilised for the regeneration of the surface portion of the hole




The Moving ADT Vane
Applied Techniques:
- Moving Geometry
- Moving Mesh Size Specification
- Moving Mesh

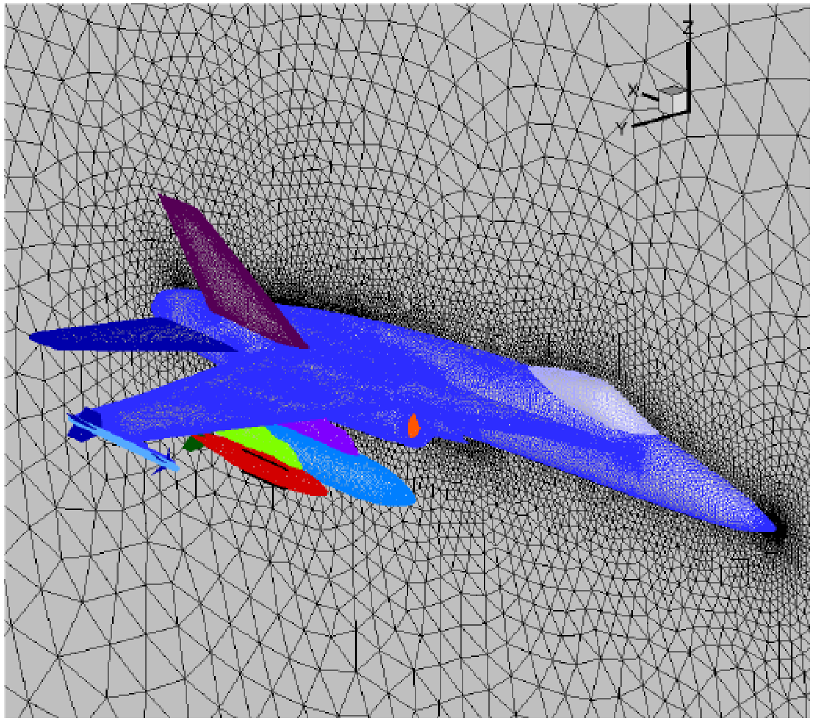
Geometry for a complete F18 Configuration

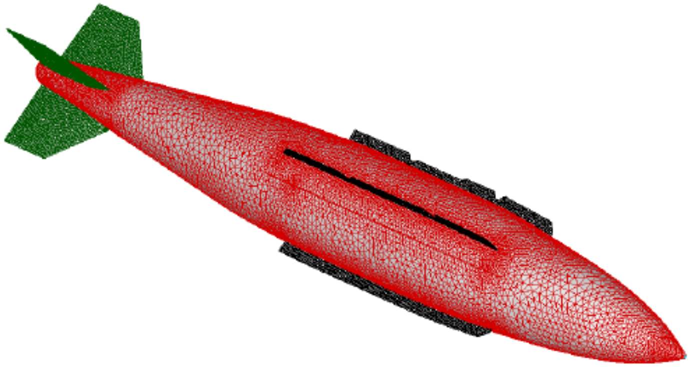
Store Separation Simulation

α_{init} = 0.46 degrees
M= 0.96
Container motion computed

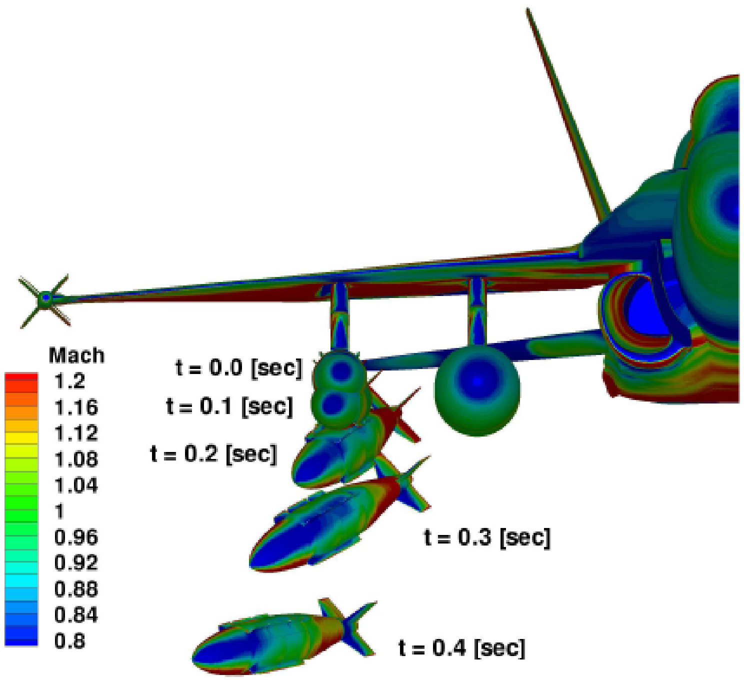
2.1-2.3 million Nodes
12.1-13.4 million Elements

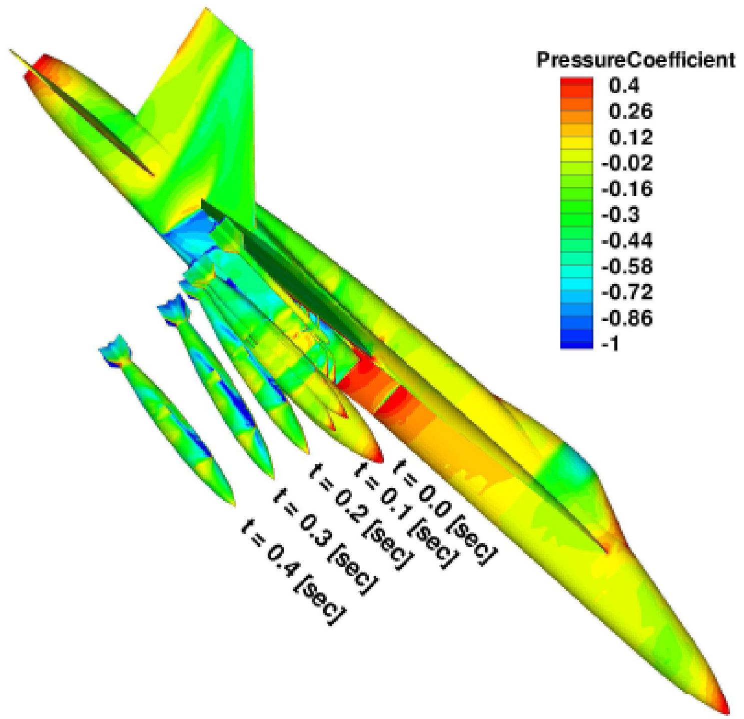
40 Physical timesteps with sub-cycling
10.4 h on 24 Processors



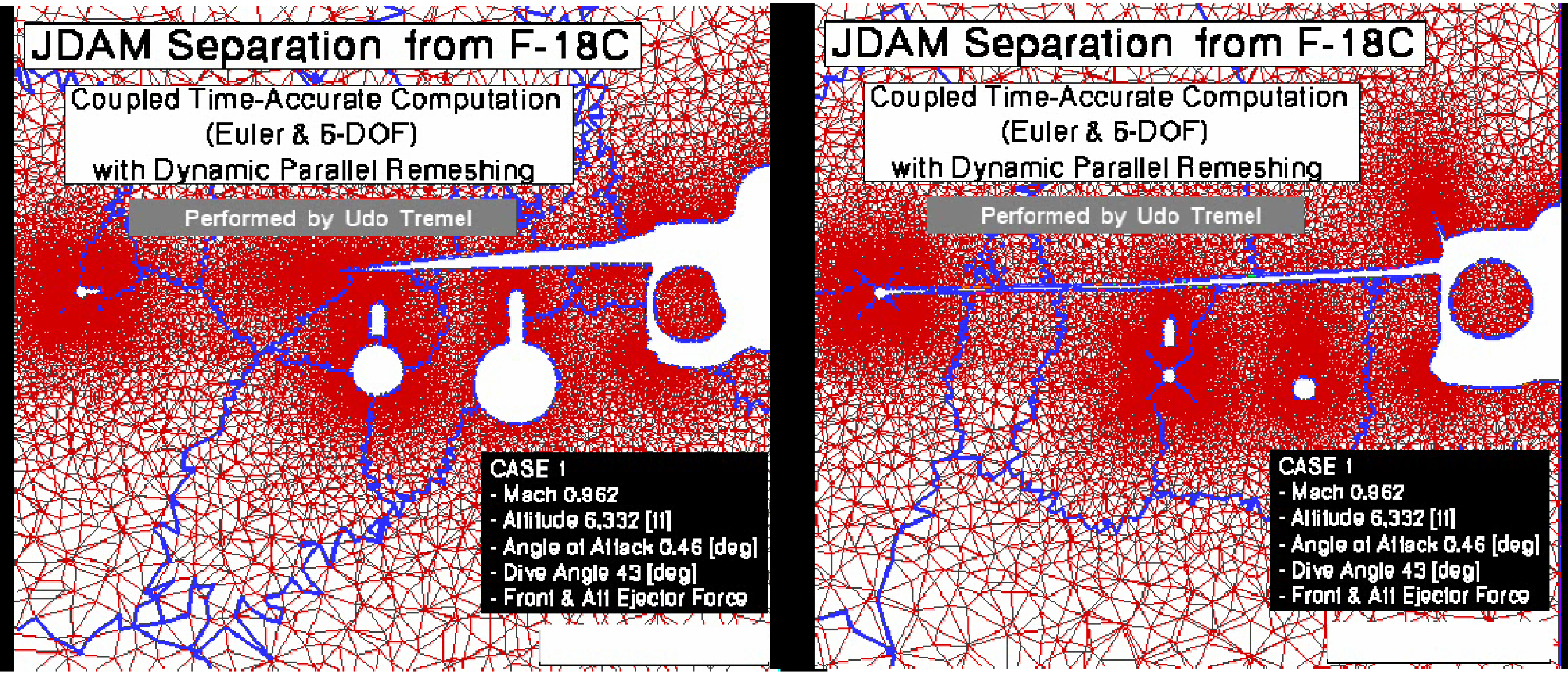


Store Separation Simulation





Store Separation Simulation



- **CFD Solution 40%**
- **Motion Application 57%**
  - **Mesh Deformation 10.3%**
  - **Volume mesh Analysis 3.7%**
  - **Volume remeshing 37%**
  - **Re-partitioning 6%**
- **I/O 3%**

- The implementation of the parallel solution procedure on the IBM was successfully completed
- Parallel implementation of the adapted remeshing has been completed
- A challenging problem has been simulated and the agreement with available experimental observations is good.
- Parallel remeshing of the boundary layer to be implemented
- Use of high order elements to minimize the number of points in the boundary layer
- Use of zonal LES / DES to capture wake flow and its influence on the separation