PARALLEL UNSTRUCTURED MESH ADAPTATION FOR TRANSIENT MOVING BODY AND AEROPROPULSIVE APPLICATIONS

Cavallo, P.A.†, Sinha, N.††, and Feldman, G.M.†††
Combustion Research and Flow Technology, Inc. (CRAFT Tech)
Pipersville, PA 18947
cavallo@craft-tech.com

ABSTRACT

Adaptive, unstructured grid methods, in which the mesh is allowed to deform, and grid quality subsequently restored through localized coarsening and refinement, offer the potential of a more rapid, straightforward approach to generalized moving body problems. Automation of this mesh movement and quality correction strategy requires a close coupling with the flow solution process. With parallel simulations now common, parallel coarsening and refinement methods for moving meshes are needed. In this work, parallel mesh adaptation strategies are developed to treat deforming, decomposed domains. Distortion of the moving mesh is assessed using a deformation matrix analysis. A two-pass approach is implemented in which cell migration shifts the interprocessor boundary, thereby accommodating coarsening and refinement of the interprocessor faces. The adapted grids are rebalanced among the processors using available techniques. Representative cases are presented to demonstrate the parallel approach and maintenance of cell quality for practical separation events.

1.0 INTRODUCTION

Three-dimensional numerical simulations on parallel computing platforms have become commonplace in recent years, and a number of Navier-Stokes solvers have been extended to operate in parallel environments. With this increase in computing capability, aerodynamic and propulsive simulations on large, unstructured grids comprised of millions of cells are now routine. Seeking to take full advantage of the unstructured grid approach, parallel mesh adaptation methods have been explored by several researchers [1-3]. For steady-state analyses, performing adaptation in parallel alleviates memory storage issues associated with multi-million cell models, and efficiency gains are realized through the use of multiple processors. Numerous advantages are to be found in transient flowfield analysis as well, where one may seamlessly invoke mesh adaptation from the parallel flow solver without the need to recompose the entire mesh and repartition after adaptation is complete. Such automated, "on the fly" modifications, to ensure adequate resolution and/or allow for the relative motion of surfaces, are an achievable objective.

While the adaptive, unstructured grid approach has shown considerable promise for moving body flows, the technique demands a "hands-off" framework to be fully tractable. Previous work in this area involved adaptation on the entire grid at once, on a single CPU, while the flow solution and grid movement process was performed in parallel [4-7]. Naturally this requires a man-in-the-loop process, in which the solution is halted, the decomposed domains recombined, adaptation performed, and the new grid repartitioned. Furthermore, these simulations were limited to inviscid, tetrahedral meshes. The development of parallel adaptation methods for viscous, multi-element grids improve upon this procedure, and represent a significant step towards "hands-off" adaptation for generalized moving body problems as well as solution-based coarsening and refinement for turbulent flows.

This paper presents recent developments in the adaptation of viscous, unstructured multi-element grids in parallel computing environments, where the mesh is partitioned by domain decomposition. A two-pass approach is employed to adapt the mesh in parallel, wherein the interprocessor boundaries are shifted using a cell migration method. This shift allows interprocessor faces to be readily coarsened and refined as interior faces. The adapted grids are rebalanced among the processors using available techniques. These advances in the mesh adaptation package, *CRISP CFD*®, are demonstrated on selected problems illustrating their utility in practical steady-state and transient applications in the areas of store separation, rocket staging, and missile launch from an aircraft with chemical interactions.

2.0 ANALYSIS APPROACH

2.1 Flow Solver

The unstructured flow solver used in this work is *CRUNCH CFD*® [8-10]. *CRUNCH CFD*® is a three-dimensional, edge-based, mixed-element unstructured Navier-Stokes solver for chemically reacting turbulent real gases and dynamic domains. The basic numerical framework of the *CRUNCH CFD*® code is a finite volume higher-order Roe/TVD scheme in which the flow

*AIAA-2004-1057, 42nd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan 5-8, 2004.

† Senior Research Scientist, Senior Member AIAA

†† Vice President & Technical Director, Associate Fellow AIAA

††† Research Scientist

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Parallel Unstructured Mesh Adaptation for Transient Moving Body and Aeropropulsive Applications	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Combustion Research and Flow Technology Inc (CRAFT Tech),6210 Keller's Church Road,Pipersville,PA,18947**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

variables are defined at the vertices of the mesh. An edge-based data structure is employed wherein a polyhedral control volume is constructed from the union of all cells incident to a given node, and the control volume faces are associated with each edge. The inviscid flux calculation proceeds by looping over all edges in the mesh, and is grid-transparent, while a cell-based method is employed to compute the flowfield gradients at the control volume faces for evaluating the viscous fluxes [8]. Turbulence modeling is provided by a k-ε model with various near-wall, compressibility, and EASM extensions.

In addition to turbulent, aerodynamic flows, *CRUNCH CFD*® has also been validated for propulsive simulations involving gas phase chemical reactions and solid particulates using Eulerian and Lagrangian two-phase flow models [9,10]. These extended capabilities in physical modeling permit the analysis of missile exhaust and control jet flows, in which combustion and particle-wall collisions may significantly alter the aerodynamic flowfield and the resulting net forces. For additional details on these models, the reader is referred to Refs. [9,10].

For moving body problems, the grid motion in *CRUNCH CFD*® is strongly coupled to the flow solution, as the additional fluxes generated by the moving control volume faces are taken into account [11]. For efficient computation of large 3-D problems, a parallel framework for distributed memory systems has been implemented along with an implicit sparse matrix solver, permitting large CFL numbers.

2.2 Grid Movement Procedure

The simulation of arbitrary dynamic, multi-body flows is accomplished by means of a generalized node movement scheme available in *CRUNCH CFD*®. Given a change in the boundary mesh, resulting from a rigid body motion, structural deformation, or design process, the interior nodes of the tetrahedral grid are redistributed by means of a generalized node movement solver. The method models the tetrahedral mesh as a deformable, elastic solid, subject to the equations of elasticity [11].

$$\frac{\partial \sigma_{ij}}{\partial x_i} = 0 \qquad \textbf{(1)}$$

In this model, the stress and strain tensors are related to the node displacements u_i

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \qquad \textbf{(2)}$$

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \qquad \textbf{(3)}$$

and the Lamé constants λ and μ are functions of cell volume. This permits small cells to resist further deformation, while larger cells are more elastic. When solving these equations in parallel on decomposed domains, the displacement of the interprocessor nodes are matched among all incident processors, and a subiteration is performed to ensure adequate propagation of the stress across interprocessor boundaries. Mesh movement with viscous, multi-element grids comprised of tetrahedra and near-wall prism cells is done by treating the prism cells as rigid, such that they move with the underlying solid boundary, and the tetrahedral region of the domain deforms to accommodate the motion.

This node movement scheme has proven to be much more robust than commonly used spring analogy or velocity diffusion techniques, as representing the full stress tensor allows mesh deformations to be propagated farther into the field, particularly for shearing motions. It should be noted that researchers have recently remedied this deficiency by including torsional springs [12,13]. Regardless of the method used, a reliable grid movement scheme permits a given mesh to be useful for a greater period of time before corrective measures need to be taken, reducing the frequency with which mesh adaptation is required.

2.3 Mesh Deformation Analysis

Mesh movement alone cannot tolerate large-scale boundary displacements, particularly for bodies in close proximity. For large degrees of motion the unstructured mesh will eventually become so distorted that the resulting poor cell quality will affect the computed flowfield, or worse, form inverted cells that terminate the simulation. While adaptive mesh techniques may be used to alleviate this distortion, a means of detecting and monitoring these mesh deformations is needed to inform the refinement and coarsening modules where and how to modify the grid.

A mesh deformation matrix concept [14,15] is used to detect distorted regions of the mesh in need of corrective coarsening and refinement. The analysis examines the transformation of a tetrahedral cell from its initial state at time t_0 to its current shape at time t_1. This transformation matrix [*A*] is formed from the edge matrix of the cell vertex coordinates at time t_0 and t_1.

$$[T] = \begin{bmatrix} \Delta x_{21} & \Delta x_{31} & \Delta x_{41} \\ \Delta y_{21} & \Delta y_{31} & \Delta y_{41} \\ \Delta z_{21} & \Delta z_{31} & \Delta z_{41} \end{bmatrix}_{t=t_0} \qquad [T'] = \begin{bmatrix} \Delta x_{21} & \Delta x_{31} & \Delta x_{41} \\ \Delta y_{21} & \Delta y_{31} & \Delta y_{41} \\ \Delta z_{21} & \Delta z_{31} & \Delta z_{41} \end{bmatrix}_{t=t_1} \qquad \textbf{(4)}$$

$$[A] = [T'][T]^{-1} \qquad \textbf{(5)}$$

The deformation matrix [*A*] is further decomposed into two matrices that represent dilatation and rigid body rotation.

$$[A] = [P][U] \qquad \textbf{(6)}$$

If the cell rotates and/or translates without deforming, then [*P*] is the identity matrix. Conversely, if the cell

deforms without rotating, [*U*] is the identity matrix. The eigenvalues of [*P*], σ_i, represent the dilatation of the tetrahedron in each of three principal directions, and are equivalent to the singular values of the transformation matrix [*A*]. In the limit, eigenvalues of one correspond to an undeformed cell, while an eigenvalue of zero indicates a cell that has collapsed to zero volume.

The reciprocal of the condition number based on the spectral norm of the transformation matrix [*A*] forms a deformation measure that indicates the extent of cell distortion from the reference state at t_0.

$$\tau = 1/\kappa_\infty(A) = \sigma_{\min}/\sigma_{\max} \qquad \textbf{(7)}$$

This deformation measure takes on values between 0 and 1 for each tetrahedral cell. Mesh coarsening is then driven by this deformation measure, in prescribing a larger point spacing to remove the poor quality elements. Mesh refinement is then invoked to restore an adequate point spacing based on the boundary spacing or current flowfield features.

3.0 ELEMENTS OF PARALLEL ADAPTATION

3.1 Mesh Modification Operators

The unstructured mesh adaptation package used in this work is *CRISP CFD*® [4,7], a mesh modification and quality improvement code for three-dimensional multi-element unstructured meshes. Meshes comprised of tetrahedral, prismatic, and hexahedral regions may be readily modified to generate more accurate flow solutions through local refinement and coarsening. In moving body applications, these coarsening and refinement methods are employed to accommodate boundary motion as well as refine evolving flow structures.

The tetrahedral region of the grid is locally refined by means of a constrained Delaunay refinement algorithm combined with a circumcenter point placement strategy. Any inconsistency between the circumradius of a tetrahedron and a desired point density triggers the point insertion procedure. This iterative cell refinement is repeated until the cell circumradii are consistent with a prescribed point spacing. Coarsening of the tetrahedral region is also permitted through an edge collapse procedure. In regions where the grid is distorted or where solution errors are negligible, edges may be selected for removal. All cells incident to the deleted edge are removed from the mesh, the adjacent cells are redefined, and the two nodes of the edge are collapsed to a single vertex.

The prismatic and hexahedral regions of the grid may be refined through cell subdivision procedures. As the boundary of the tetrahedral region is refined the adjacent prism layers are also modified. This is accomplished by splitting edges at the tet/prism interface, and propagating this subdivision down to the wall through all of the layers. In addition, a procedure is in place to refine entire layers of prisms if an improved boundary layer resolution is desired [7]. The refinement of the hexahedral region of the grid is accomplished using the pattern formation procedure of Biswas and Strawn [16], which employs a parent-child data structure to split the cells. Each hexahedral cell is then split according to a pattern, to generate 2:1, 4:1, or 8:1 subdivisons. This cell subdivision creates buffer cells, which are tetrahedral, pyramid, or prismatic elements used to transition between different levels of hexahedral refinement, thus ensuring a conforming mesh with no hanging nodes.

3.2 Prescribed Point Spacing

Given the above mesh modification procedures, adaptive refinement and coarsening are triggered by prescribing what the point spacing should be, based on the mesh deformation measure and/or solution error estimates. Larger spacings dictate that mesh coarsening will be performed, while a smaller point spacing will trigger an iterative mesh refinement.

An initial point spacing ρ_0 is defined for each vertex as the average edge length for all edges incident to the node. A larger grid spacing produces an iterative coarsening of the mesh. Using the mesh deformation measure defined in Equation (7), we modify the local grid spacing to be

$$\rho = \rho_0/\tau \qquad \textbf{(8)}$$

Note that the more deformed the cell, the larger the prescribed spacing, and hence an increased amount of coarsening will be performed. This improves the likelihood of the distorted cell being removed.

Conversely, the enrichment procedures are invoked by specifying a smaller spacing. After the coarsening phase, an appropriate gradation of cell size is restored by solving a Laplace equation for ρ, using the boundary mesh spacings as Dirichlet boundary conditions [17]. An approximate solution is obtained by summing the difference in the point spacing for all edges *N* incident to the node using a relaxation technique.

$$\rho^{n+1} = \rho^n + \frac{\varepsilon}{N}\sum_{k=1}^{N}\left(\rho_k^n - \rho^n\right) \qquad \textbf{(9)}$$

Prescribing new point spacings also drives solution-based coarsening and refinement. A variation on the solution error estimate developed in two dimensions by Ilinca et al. [18] has been implemented in three dimensions for arbitrary mesh topologies. The method is based on forming a higher order approximation of the solution at each mesh point using a least squares approach. The difference between the higher order reconstruction from incident nodes and the current solution forms the error measurement. If the current mesh is sufficiently fine to support the spatial variation in the solution, the estimated error will be low, allowing coarsening to take place. Conversely, a high degree of error indicates additional refinement is needed.

At any location (*x*,*y*,*z*), the piecewise quadratic solution is formed at node *j* from the solution at grid point *i* by Taylor series expansion using all first and second order derivatives. The solution itself is assumed to be piecewise linear, which is the case for most unstructured grid solution methods. The error at grid point *i* is the sum of the difference between the actual solution at node *j* and the reconstructed solution at *j* for each of *N* edges incident to node *i*.

$$e_i = \sum_{j=1}^{N} \left| U_j^* - U_j \right| \tag{10}$$

$$e_i = \sum_{j=1}^{N} \left| \begin{array}{l} \Delta U_{ij} + \Delta x \frac{\partial U}{\partial x} + \Delta y \frac{\partial U}{\partial y} + \Delta z \frac{\partial U}{\partial z} + \Delta x^2 \frac{\partial^2 U}{\partial x^2} + \Delta y^2 \frac{\partial^2 U}{\partial y^2} \\ + \Delta z^2 \frac{\partial^2 U}{\partial z^2} + \Delta x \Delta y \frac{\partial^2 U}{\partial x \partial y} + \Delta y \Delta z \frac{\partial^2 U}{\partial y \partial z} + \Delta x \Delta z \frac{\partial^2 U}{\partial x \partial z} \end{array} \right| \tag{11}$$

The error is scattered to each node by looping over all edges in the mesh, and then normalized to take on values between 0 and 1. In regions where the error is low, the new, local point spacing is increased as

$$\rho = CF\,\rho_0 \tag{12}$$

while in high error regions, smaller spacings are defined.

$$\rho = \frac{\rho_0}{1+(RF-1)e} \tag{13}$$

In Equations (12) and (13), *CF* and *RF* are user-defined coarsening and refinement factors, respectively. These factors permit varying degrees of coarsening and/or refinement within a given adaptation pass.

Computing the various derivatives requires solution of a least squares problem. Inversion of the 9x9 system requires a stencil of at least 9 points. Currently, all of the edges incident to a node are employed, as well as the cell and face centers for the elements incident to the node.

3.3 Parallel Implementation

The simultaneous alteration of the decomposed domains of an unstructured mesh presents a number of challenges. In parallel, each processor operates on its own partition, concurrent with and independent of the others. Previous work in parallel mesh refinement [1-3] demonstrated methods in which adaptation was performed on each processor, and patterns for cell subdivision were exchanged across interprocessor boundaries, ensuring a conforming mesh. Coarsening the interprocessor boundary was not a concern, nor was the possible motion of the mesh boundaries.

With moving meshes, however, and for optimal use of computational resources, there is a need to coarsen the original grid. The boundary mesh refinement and coarsening operations used in the current work do not employ pattern formation procedures, and the synchronization of these sequential operations quickly becomes complicated for arbitrary partitions where multiple processors may share a common edge. Therefore the first issue that arises in parallel adaptation is how to treat the interprocessor boundaries. Rather than modify these faces, the interprocessor boundaries are shifted using a cell migration technique. The interprocessor faces and adjacent cells then become interior faces and interior cells, which may be readily modified through a second adaptation pass. In the second pass only the former interprocessor boundary region needs to be coarsened, refined, or smoothed, as the remainder of the mesh is already consistent with the prescribed point spacing.

Figure 1 illustrates the two-pass approach for solution-based coarsening and refinement of supersonic flow entering a duct. The original mesh partitions, shown in Figure 1(a), are independently coarsened and refined to produce the adapted mesh of Figure 1(b). Note that the interprocessor boundaries are not modified, which leaves a region of the mesh that still requires adaptation. Several layers of cells are migrated from the right processor to the left, as seen in Figure 1(c). The interprocessor boundary is now to the right of its original location. A second coarsening and refinement pass treats the former interprocessor faces and adjacent cells, producing the final adapted grid of Figure 1(d).

3.4 Implications of Cell Migration

A consequence of the cell migration approach is that the shape and extent of the decomposed domains change. The cell migration process may introduce new pairs of adjacent domains that did not initially communicate. Similarly, pairs of processors that once shared common nodes, edges, and faces may become disconnected as a result of cell migration. Updating the interprocessor communication schedule proceeds in two stages. First, the current communication lists are updated for each pair of adjacent domains. If no common nodes are found between two domains, the communication is removed from the cycle. The second stage involves checking for any new communication pairs introduced as a result of migration.

In addition, one can no longer refer to the original decomposed grid to obtain data for solution transfer or for establishing point spacing after the coarsening phase. This issue is remedied by recomposing the global grid arrays at the start of the mesh adaptation process, such that the list of global vertex coordinates, solution vectors, and computed point spacings may be readily available to all processors.

3.5 Load Rebalancing

An adapted grid is inherently unbalanced, with mesh refinement taking place on certain partitions and coarsening taking place on others. Re-establishing load balance is desirable for steady-state problems, and

essential for transient simulations, particularly after several adaptations. Load rebalancing is accomplished using the *ParMETIS* package [19] developed at the University of Minnesota. First the global cell numbers and cell adjacencies are formed for each of the existing partitions. At interprocessor boundaries, the adjacent global cell number on the neighboring processor must also be determined. The adaptive repartitioning routines in *ParMETIS* then redistribute the existing partitions of the adapted grid using a diffusion concept [19]. New processor assignments for each cell in the global grid are obtained. From these new assignments, the balanced partitions may be re-formed.

4.0 APPLICATIONS

4.1 Generic Transonic Store Release

The first application considered is the separation of a finned store from a wing/pylon configuration at Mach 1.2. Inviscid flow is assumed for this tetrahedral grid. A constant ejection force is applied over the first 0.1 seconds of the simulated separation. After the initial ejection stroke, the motion of the store is provided by general 6-degree-of-freedom (6-DOF) equations of motion using the current integrated surface pressure distribution. Gravitational acceleration is also included.

Figure 2 provides an overview of the simulation. A total of ten adaptations were performed at regular intervals. The unstructured grid is comprised of approximately 2.7 million cells, and is decomposed on 16 processors. In this image, the store is colored by the current pressure distribution at each of the four instants shown, and the black lines indicate the changing interprocessor boundaries on the store surface resulting from cell migration and load rebalancing. As it translates, the store yaws nose away from the symmetry plane and pitches nose down. The surface pressure distribution reflects the changing local angle of attack and sideslip angle of the store.

Figure 3 shows the mesh deformation history during the simulation, in which ten adaptations are performed at regular intervals to recover mesh quality. As the distance between the store and pylon surfaces increases, the mesh distortion becomes less severe. With each successive adaptation, the deformation measure reduces to a minimum value greater than the previous minimum. This indicates that mesh movement may likely be applied for a longer period of time before adaptation is warranted. Such strategies and tradeoffs are yet to be investigated.

The evolution of the unstructured mesh as the store falls away is depicted in Figure 4. Interprocessor boundaries are highlighted in red. Through the adaptive coarsening and refinement procedures, overall mesh quality is maintained, and an appropriate cell distribution is provided as the distance between the store and pylon increases. Although Figure 4 illustrates only a slice through the mesh, one can readily see the migration and rebalancing of the interprocessor boundaries.

4.2 Rocket Stage Separation

This demonstration examines lower stage separation from a rocket due to upper stage motor ignition. Accurate tracking of the upper stage exhaust species is of primary concern for estimating plume IR emissions. Motion of the lower stage is coupled to the instantaneous pressure and shear stress distribution. For simplicity, a 5-degree sector is modeled with axisymmetry assumed. The viscous tet/prism grid is decomposed on 16 processors.

One major challenge lies in resolving and containing the high pressure plume. Figure 5 illustrates contours of mixture fraction, the local fraction of mass originating in the upper stage exhaust, along with a mirror image of the current decomposed mesh at 10 msec intervals. At time zero there is no plume and the mesh is relatively coarse away from the rocket. As the plume expands, and produces a large forward separation, additional refinement takes place and the interprocessor boundaries are shifted to reestablish load balance. One aspect of the problem that remains an issue is how frequently to perform mesh adaptation, to keep up with the expanding front rather than "chasing" the evolving flowfield. The extent of the plume and the separation it induces is not known a priori, and hence the mesh must occasionally be regenerated to accommodate this by extending the outer boundaries. In Figure 5 it can be seen that the computational domain is increased upstream, downstream, and radially as required to contain the problem.

A second major challenge is handling the rapid motion of the lower stage. Figure 6 shows a close-up of the interstage connector region during the first 30 msec of the separation event. The large range of motion creates tremendous shear in the mesh, and over 30 adaptations were performed to restore grid quality during this time. The need for corrective adaptation becomes more frequent as the lower stage accelerates.

Once the interstage connector clears the upper stage nozzle exit plane, cell extrusion techniques may be applied to automate the task of handling grid motion [11,20]. Such an approach is suitable for 1-D motions and could simplify the analysis once the initial separation is completed using the adaptive methods shown here.

4.3 Missile Launch from Fighter Aircraft

The final application presented considers missile carriage and launch from a fighter aircraft. Unlike typical store ejection scenarios, aircraft/missile compatibility studies must assess the extent of the missile exhaust plume, and possible impingement and heating of aircraft surfaces, particularly control surfaces.

Figure 7 depicts an example of the missile launch studies currently in progress. A slice of the multi-element unstructured grid is shown, colored by

temperature contours. Significant mass fractions of unburned fuel in the missile exhaust produce afterburning, as evidenced by higher temperatures near the plume perimeter.

Propulsive simulations such as this require the use of multi-element meshes comprised of both tetrahedral and hexahedral cell topologies. Previous work demonstrated that tetrahedral meshes often prove too diffusive for shear layers, and very large meshes must be employed to produce the same results as those obtained with anisotropic, hexahedral cells [21]. While adaptation of such mixed topology grids is possible, as demonstrated in Figure 8, it is presently restricted to single CPU operation. Extension of the parallel implementation to tet/hex grids is planned.

Mesh adaptation is a useful tool for obtaining rapid, grid-converged results for store carriage loads. Table 1 summarizes the effects of mesh adaptation on the force and moment coefficients for the missile prior to launch. While there is an appreciable change between the first two grids, there is less of an incremental change between the first and second adaptation. Subsequent adaptations to further assess incremental changes in the solution may be readily done using the parallel adaptation and rebalancing methods.

The improvement with adaptation is also evident in Figure 9, which illustrates the decomposed mesh, Mach number contours, and turbulent viscosity contours before and after adaptation. A more well resolved pylon and missile wake is clearly seen in both Mach number and the predicted turbulence levels.

5.0 CONCLUDING REMARKS

Parallel mesh adaptation strategies have been developed for deforming unstructured viscous grids. A two-pass approach is shown to effectively permit coarsening and refinement of the interprocessor boundary by shifting its location through cell migration. The parallel mesh movement and adaptation methods are seen to be effective for practical moving body problems such as store separation and rocket staging events. Cell quality is consistently restored as the unstructured mesh undergoes large-scale motion. Parallel mesh adaptation also provides an efficient path for obtaining grid-converged, steady-state solutions.

Future development will focus on extending the parallel adaptation methods to viscous unstructured grids comprised of combinations of tetrahedral and hexahedral topologies. Such mixed element meshes are more suitable for propulsive applications, as tetrahedral grids are often too diffusive to compute turbulent shear layers and reaction zones. Additional computations of realistic weapons bay separation scenarios are also planned, with upcoming flight tests providing telemetry for validation.

6.0 ACKNOWLEDGMENTS

This paper surveys aspects of research conducted under funding from the Air Force Research Laboratory, Wright-Patterson AFB and Edwards AFB, and the Army Aviation and Missile Command. Portions of this work were conducted at the NAVOCEANO Major Shared Resource Center.

The first author thanks Dr. Timothy Baker of Princeton University for many productive discussions throughout our continued collaboration in mesh adaptation research.

7.0 REFERENCES

[1] De Keyser, J., and Roose, D., "Run-Time Load Balancing Techniques for a Parallel Unstructured Multi-Grid Euler Solver with Adaptive Grid Refinement", *Parallel Computing*, Vol. 21, pp. 179-198, 1995.

[2] Flaherty, J.E., Loy, R.M., Shephard, M.S., Szymanski, B.K., Teresco, J.D., and Ziantz, L.H., "Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws", *Journal of Parallel and Distributed Computing*, Vol. 47, pp. 139-152, 1997.

[3] Oliker, L., Biswas, R., and Gabow, H.N., "Parallel Tetrahedral Mesh Adaptation with Dynamic Load Balancing", *Parallel Computing*, Vol. 26, pp. 1583-1608, 2000.

[4] Cavallo, P.A., and Dash, S.M., "Aerodynamics Of Multi-Body Separation Using Adaptive Unstructured Grids", AIAA Paper 2000-4407, 18th Applied Aerodynamics Conference, Denver, CO, August 14-17, 2000.

[5] Baker, T.J., and Cavallo, P.A., "Dynamic Adaptation for Deforming Tetrahedral Meshes", AIAA Paper 99-3253, 14th Computational Fluid Dyanamics Conference, Norfolk, VA, June 28-July 1, 1999.

[6] Cavallo, P.A., Lee, R., Hosangadi, A., Baker, T.J., and Kreeger, R.E., "Simulation of Weapons Bay Store Separation Flowfields Using Unstructured Grids", AIAA Paper 99-3188, 17th AIAA Applied Aerodynamics Conference, Norfolk, VA, June 28-July 1, 1999.

[7] Cavallo, P.A., and Baker, T.J., "Efficient Delaunay-Based Solution Adaptation for Three-Dimensional Unstructured Meshes", AIAA Paper 2000-0809, 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 10-13, 2000.

[8] Hosangadi, A., Lee, R.A., Cavallo, P.A., Sinha, N., and York, B.J., "Hybrid, Viscous, Unstructured Mesh Solver for Propulsive Applications", AIAA Paper 98-3153, 34th Joint Propulsion Conference, Cleveland, OH, July 13-15, 1998.

[9] Hosangadi, A., Lee, R.A., York, B.J., Sinha, N. and Dash, S.M., "Upwind Unstructured Scheme for Three-Dimensional Combusting Flows", *Journal of Propulsion and Power*, Vol. 12, No. 3, pp. 494-503, May-June 1996.

[10] York, B.J., Lee, R.A., Sinha, N., and Dash, S.M., "Progress in the Simulation of Particulate Interactions in Solid Propellant Rocket Exhausts", AIAA Paper 2001-3590, 37th AIAA/ASME/SAE/

ASEE Joint Propulsion Conference, Salt Lake City, UT, July 8-11, 2001.

[11] Cavallo, P.A., Hosangadi, A., Lee, R.A., and Dash, S.M., "Dynamic Unstructured Grid Methodology with Application to Aero/Propulsive Flowfields", AIAA Paper 97-2310, 15th AIAA Applied Aerodynamics Conference, Atlanta, GA, June 23-25, 1997.

[12] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "An Improved Method of Spring Analogy for Dynamic Unstructured Fluid Meshes", AIAA Paper 98-2070, 39th Joint Structures, Structural Dynamics, and Materials Conference and Exhibit, Long Beach, CA, April 20-23, 1998.

[13] Murayama, M., Nakahashi, K., and Matsushima, K., "Unstructured Dynamic Mesh for Large Movement and Deformation", AIAA Paper 2002-0122, 40th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 14-17, 2002.

[14] Baker, T.J., "Mesh Movement and Metamorphosis", *Engineering with Computers*, Vol. 18, No. 3, pp. 188-198, 2002.

[15] Baker, T.J., "Deformation and Quality Measures for Tetrahedral Meshes", Proc. European Congress on Computation Methods in Applied Sciences and Engineering, Barcelona, 2000.

[16] Biswas, R., and Strawn, R.C., "Tetrahedral and Hexahedral Mesh Adaptation for CFD Problems", NAS Technical Report NAS-97-007, 1997.

[17] Baker, T.J., "Mesh Deformation and Modification for Time Dependent Problems", Proc. European Congress on Computation Methods in Applied Sciences and Engineering, Swansea, 2001.

[18] Ilinca, C., Zhang, X.D., Trépanier, J.-Y., and Camarero, R., "A Comparison of Three Error Estimation Techniques for Finite-Volume Solutions of Compressible Flows", *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, pp. 1277-1294, 2000.

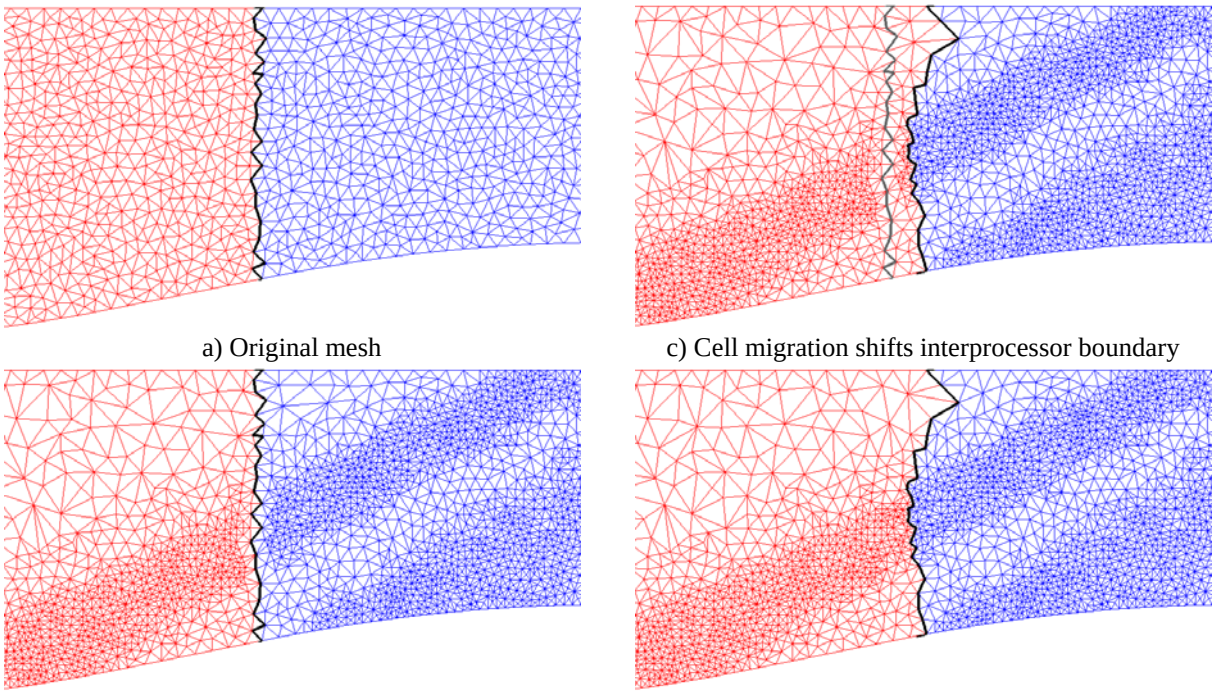
[19] Schloegel, K., Karypis, G., and Kumar, V., "A Unified Algorithm and Load-balancing Adaptive Scientific Simulations", Technical Report, 00-033, University of Minnesota, Department of Computer Science and Engineering, 2000.

[20] Sinha, N., Cavallo, P.A., Lee, R.A., Hosangadi, A., Kenzakowski, D.C., Dash, S.M., Affes, H., and Chu, D., "Novel CFD Techniques for In-Cylinder Flows On Tetrahedral Grids", SAE Paper 980138, SAE International Congress and Exposition, Detroit, MI, February 23-26, 1998.

[21] Shipman, J.D., Cavallo, P.A., and Hosangadi, A., "Efficient Simulation of Aircraft Exhaust Plume Flows using a Multi-Element Unstructured Methodology", AIAA Paper 2001-0598, 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11, 2001.

Table 1. Effect of adaptation on integrated force/moment coefficients.

	Axial Force	Side Force	Normal Force	Roll	Pitch	Yaw
Original Grid	0.0312	-0.0006	0.0329	0.0002	-0.0143	-0.0057
1st Adaptation	0.0267	-0.0027	0.0291	0.0002	-0.0129	-0.0063
% Change	-14.4%	350%	-11.6%	0.0%	-9.8%	10.5%
2nd Adaptation	0.0254	-0.0030	0.0264	0.0002	-0.0119	-0.0064
% Change	-4.9%	11.1%	-9.3%	0.0%	-7.8%	1.6%



a) Original mesh

c) Cell migration shifts interprocessor boundary

b) First adaptation pass

d) Second adaptation pass

Fig. 1. Two-pass approach for parallel coarsening and refinement.

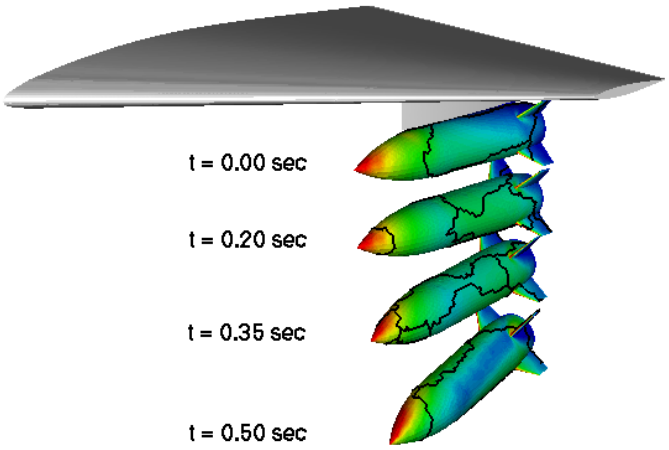


Fig. 2. Store position, orientation, and surface pressures at selected points in trajectory.

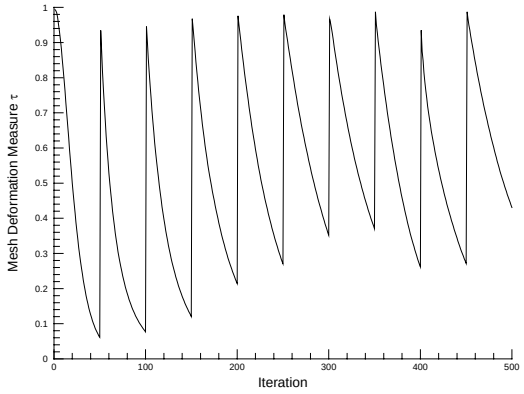
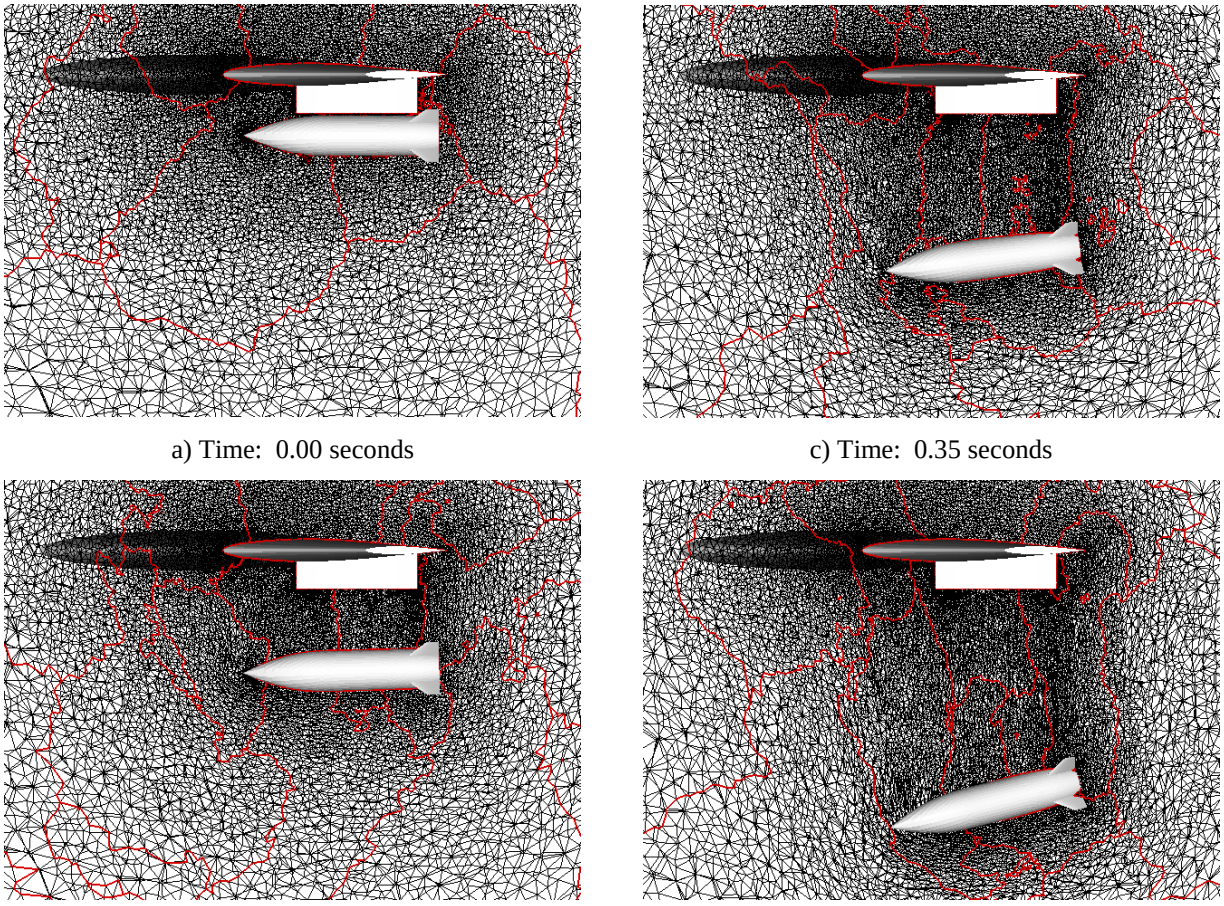


Fig. 3. Mesh deformation measure during store release.



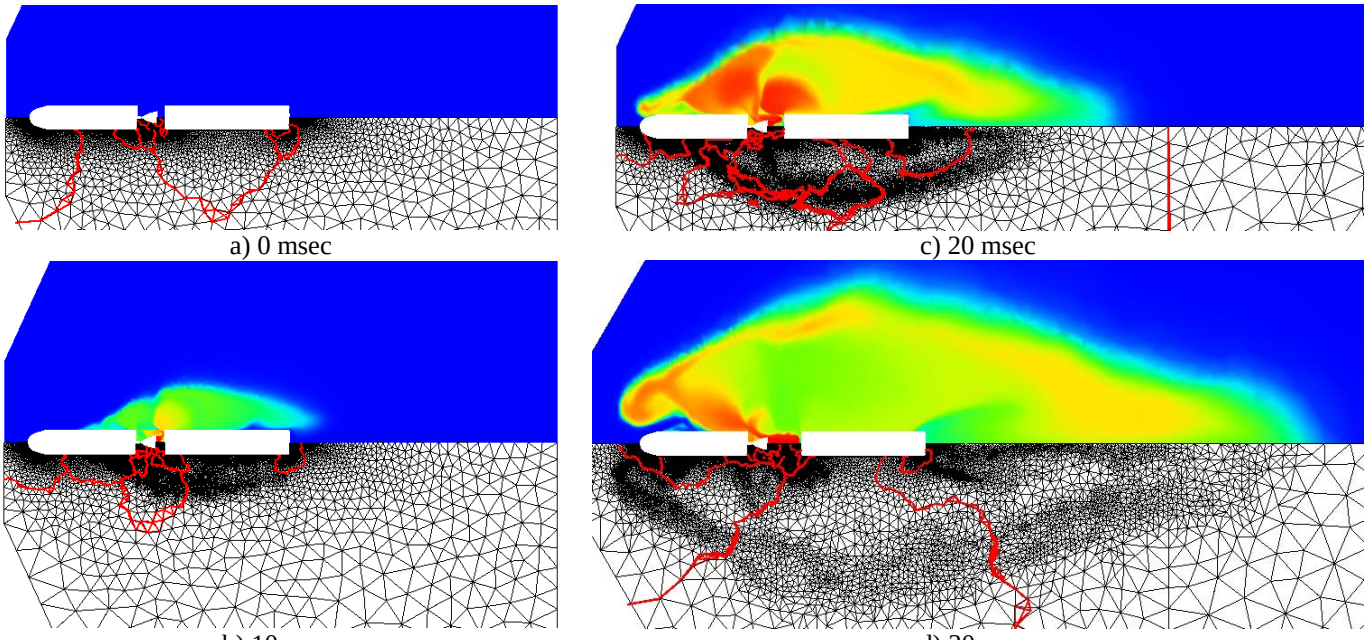
a) Time: 0.00 seconds

c) Time: 0.35 seconds

b) Time: 0.20 seconds

d) Time: 0.50 seconds

Fig. 4. Adapted mesh during store dispense.



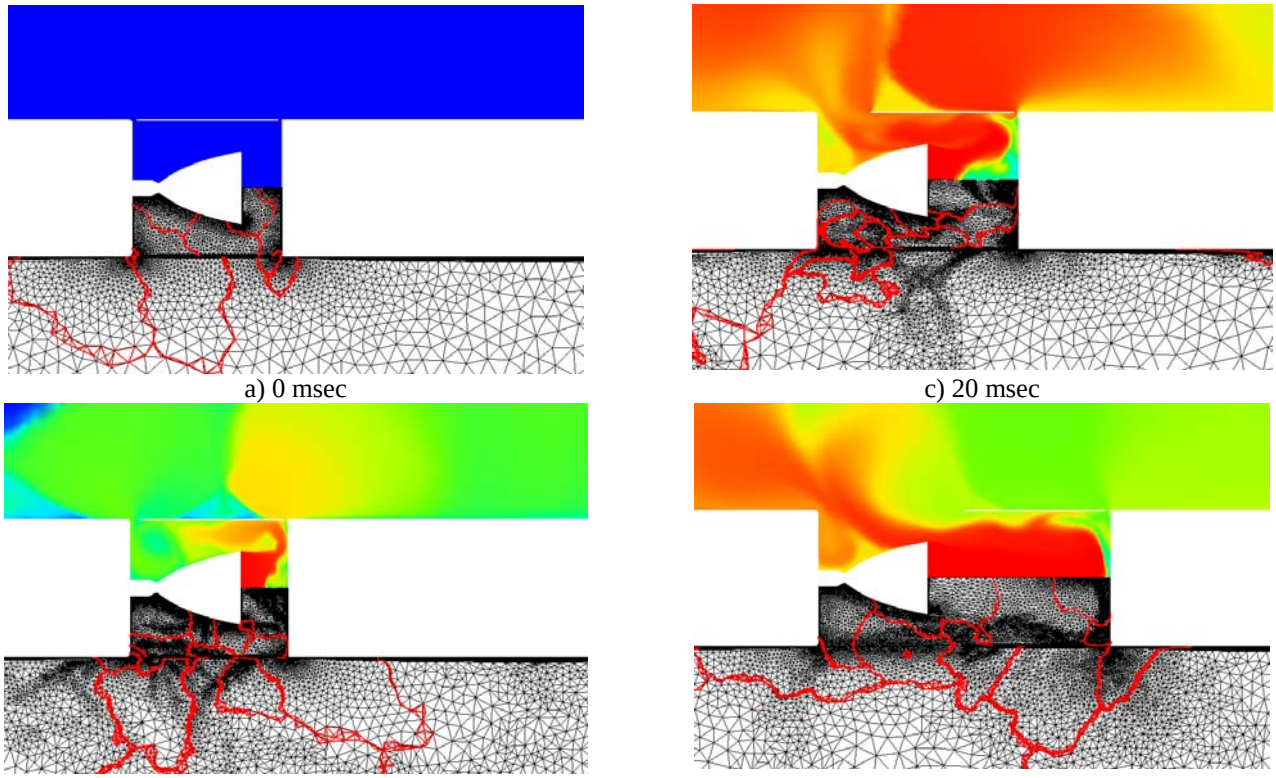
a) 0 msec

c) 20 msec

b) 10 msec

d) 30 msec

Fig. 5. Overview of rocket stage separation.



a) 0 msec

c) 20 msec

b) 10 msec

d) 30 msec

Fig. 6. Close-up of interstage connector region during early separation event.

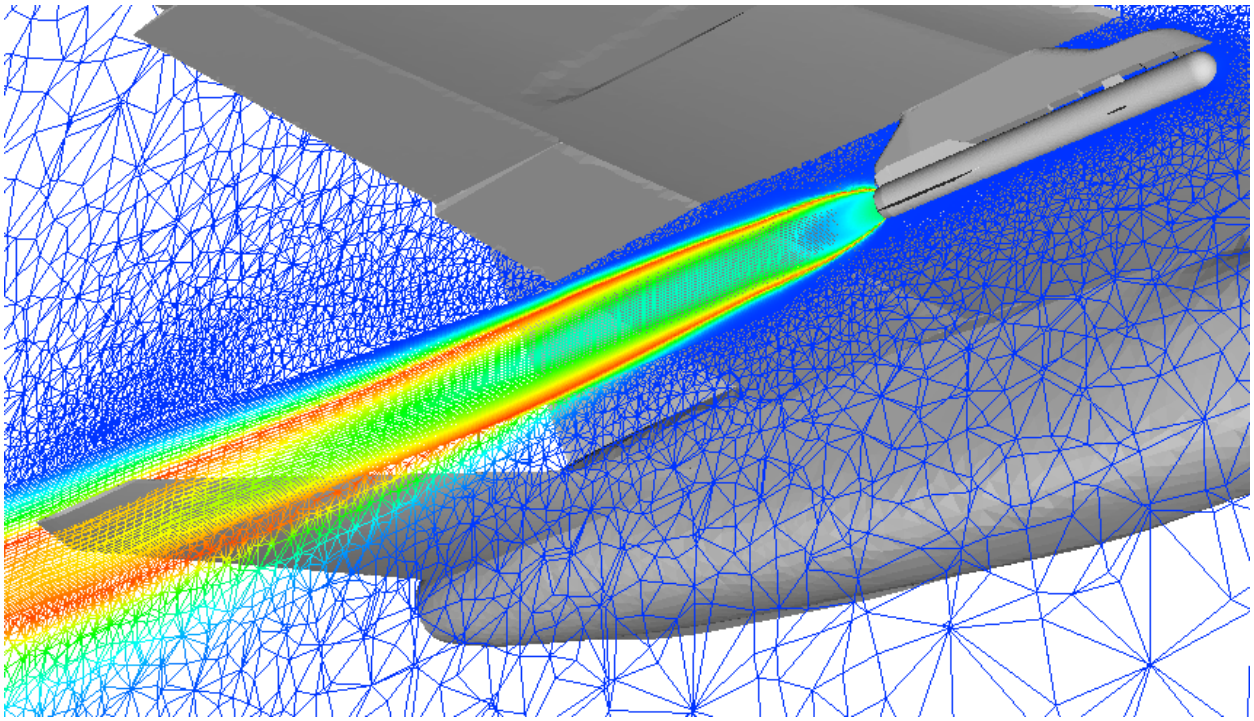


Fig. 7. Missile launch scenario with afterburning, depicting temperature contours and multi-element grid strategy.

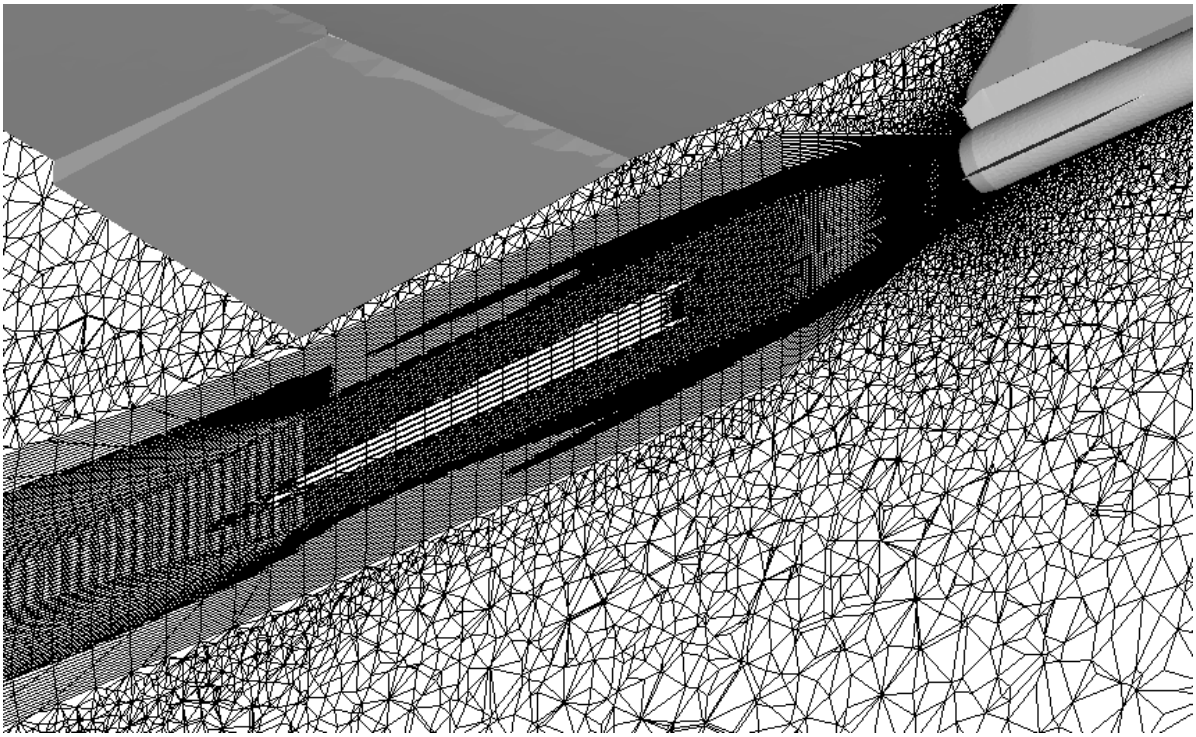
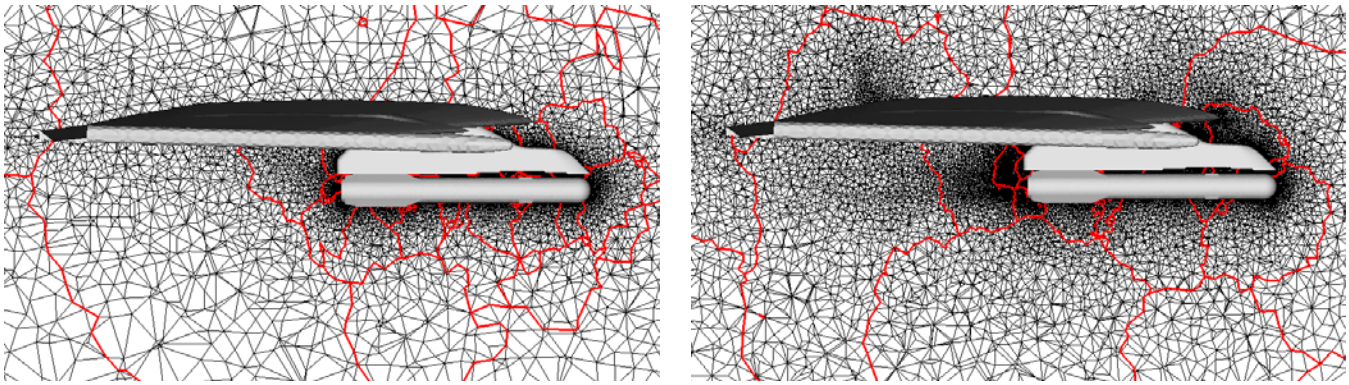
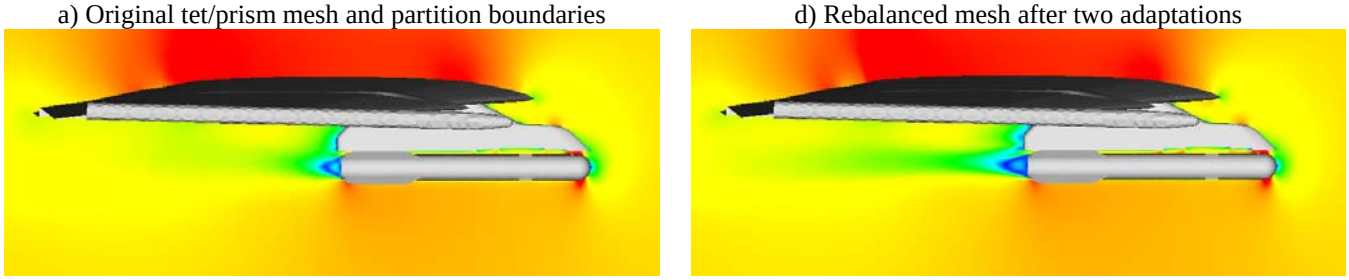


Fig. 8. Adaptation of multi-element (tet/hex) unstructured mesh.



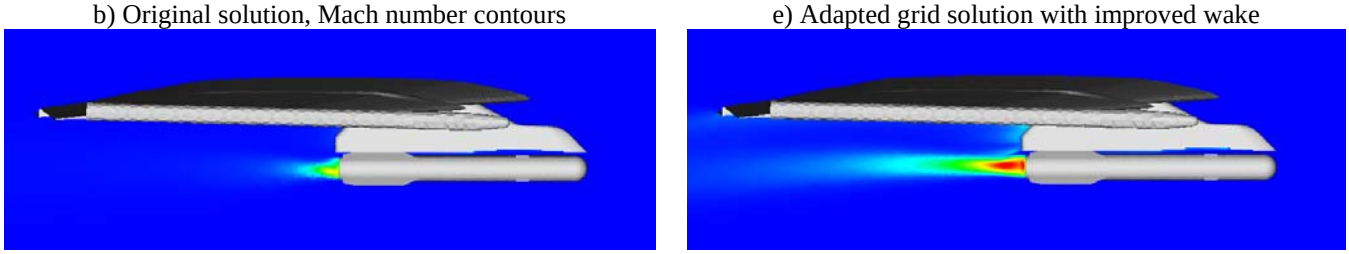
a) Original tet/prism mesh and partition boundaries

d) Rebalanced mesh after two adaptations



b) Original solution, Mach number contours

e) Adapted grid solution with improved wake



c) Original solution, turbulent viscosity contours

f) Adapted viscous solution with increased turbulence levels

Fig. 9. Parallel viscous mesh adaptation for computing carriage loads, missile launch scenario.