https://ntrs.nasa.gov/search.jsp?R=20060022546 2019-08-29T22:03:39+00:00Z

9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, June 5–8, 2006, San Francisco, CA

Parallel, Gradient-Based Anisotropic Mesh Adaptation for Re-entry Vehicle Configurations

Karen L. Bibb,* Peter A. Gnoffo,† Michael A. Park,‡ and William T. Jones§
NASA Langley Research Center, Hampton, VA 23681

Two gradient-based adaptation methodologies have been implemented into the FUN3D–REFINE–GRIDEX infrastructure. A spring-analogy adaptation which provides for nodal movement to cluster mesh nodes in the vicinity of strong shocks has been extended for general use within FUN3D, and is demonstrated for a 70° sphere cone at Mach 2. A more general feature-based adaptation metric has been developed for use with the adaptation mechanics available in FUN3D, and is applicable to any unstructured, tetrahedral, flow solver. The basic functionality of general adaptation is explored through a case of flow over the forebody of a 70° sphere cone at Mach 6. A practical application of Mach 10 flow over an Apollo capsule, computed with the FELISA flow solver, is given to compare the adaptive mesh refinement with uniform mesh refinement. The examples of the paper demonstrate that the gradient-based adaptation capability as implemented can give an improvement in solution quality.

Nomenclature

C	Coarsening factor for adaptation
C_N	Normal force coefficient
F_e^n	Spring force for edge e at node n
h	Isotropic mesh size
$\hat{I}$	Local error adaptation intensity
J_e	Jump condition across edge, for gradient-based adaptation
J_t	Jump tolerance
l	Edge length
l_{targ}	Target length based on a target cell
	Reynolds number
M	Mach number
N	Number of mesh nodes
Re	Reynolds Number
T	Temperature

Symbols

ω	Adaptation relaxation factor
ρ	Density

I. Introduction

NASA spaceflight programs are beginning to rely on Computational Fluid Dynamic (CFD) methodologies to provide both global aerodynamic characteristics and localized flow physics details such as local heating and reaction control system (RCS) jet interactions. As CFD methodologies become more integral to

*Research Engineer, Aerothermodynamics Branch, Senior Member AIAA
†Senior Research Engineer, Aerothermodynamics Branch, AIAA Fellow
‡Research Scientist, Computational AeroSciences Branch, Member AIAA
§Research Scientist, Advanced Engineering Environments Branch, Senior Member AIAA
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

the design process for complex flight vehicles, there is an increased demand for more automatic methods of generating flow solutions and quantifying numerical uncertainties in the flow solution. Such a methodology must provide a *better* answer for the same amount of work, provide an equivalent answer for less effort, or both. Work in this context includes both computer time and human time. *Better* in this context can mean improved flowfield resolution for flowfield diagnostics, a solution with lower numerical uncertainty in desired quantities (C_L, C_D, C_m, for example,) or both.

Some form of mesh adaptation is required for hypersonic problems, where clustering of mesh points in the direction normal to the bow shock has been shown to be of critical importance.[1–3] It is only possible to accomplish this clustering when the shock location is known after computing a flow solution. The manual creation of grids that adequately capture the strong bow shocks of re-entry vehicles requires tedious manipulation of meshing parameters to place points between the bow-shock and the body. Utilizing a manual approach for a large number of calculations, such as the large set of cases required for an entire aerodynamic database, becomes prohibitive because of the human time required to adjust the initial mesh for each case.

Numerous solution adaptive grid methodologies have been developed and applied in an attempt to reduce the dependence on the manual iteration of CFD processes and the required expertise of the practitioner to obtain accurate results.[4,5] The aim is to increase grid resolution where local discretization error estimates are large with the goal of diminishing and equally distributing these estimated errors. A direct method is to target local errors in the solution due to equation residuals, solution derivatives, or entropy generation.[4–17] This local error estimate solution adaptive process tends to focus on strong features in the flow solution (e.g., shocks, boundary layers). Different formulations of the local error estimate target different features in the flowfield. An experienced practitioner using extensive validation (e.g., grid convergence studies, comparison to experiment and other computations) is necessary to consistently produce quality solutions for a variety of simulations.

An alternative method is to estimate the error in the calculation of a specified engineering output functional. Output error indicators utilize the dual or adjoint solution to account for the linearized impact of local error as well as the transport of these local errors throughout the problem domain. This output-based approach has been applied to finite element discretizations[18–25] and extended to other discretizations.[26–30] The output-based approach provides a natural termination criteria that is based on a user-specified functional error tolerance. The 2D adjoint computation and output error specification methods of Venditti and Darmofal,[29,30] have been extended to 3D within the fully unstructured Navier-Stokes three-dimensional[2,3,31–33] (Fun3d) flow solver suite and applied to inviscid, laminar, and turbulent flow problems.[34–36] The methodology shows much promise in enabling the estimation of uncertainties in complex flow simulations and automatically adapting the discretization to reduce these uncertainties. The current method of obtaining the adjoint solution involves a tedious process of differentiating the flow solution process by hand. An alternative method of obtaining the solution derivatives utilizing a complex variable approach has been applied to the generic gas formulation in Fun3d, but some issues remain.[37]

While output-based approach is preferred over gradient-based adaptation, its applicability to high energy flows is currently limited because some numerical difficulties still remain in solving the adjoint equation for these flow regimes. This robustness issue is being investigated. The development and use of gradient-based adaptation techniques within Fun3d is necessary to both substitute for output-based methods where the adjoint methodology is not yet robust and to develop experience with the automated adaptation processes in advance of full implementation of the output-based approach.

The current paper seeks to reduce the manual interaction required to generate meshes with points clus-

tered near strong bow-shocks by using local error estimation adaptation to iteratively create grids for prediction the aerodynamics of re-entry vehicles. The local error estimates in this study are based on solution derivatives. Two types of gradient-based adaptation have been implemented in the FUN3D suite. First, a simple mesh movement algorithm based on a spring analogy, previously available in only the generic gas formulation of the FUN3D solver,[2,3] has been integrated into the standard compressible solver. The spring forces are derived from the temperature gradient, and are especially effective in clustering the mesh points in the direction normal to the shocks. Each mesh movement cycle produces a small perturbation in the mesh, and is performed many times during the flow solution process. This methodology is demonstrated with Mach 2 flow over a 70° sphere-cone.

The second adaptation algorithm provides a gradient-based adaptation metric to the mesh adaptation mechanics developed and implemented in the FUN3D suite.[34–36] An isotropic mesh density metric for the improved mesh is provided by a scalar adaptation parameter based on the 1st derivative of density (estimated by the jump in density across an edge). The anisotropic orientation and scaling is determined by the Mach number Hessian (second derivative tensor). The adaptation mechanics provide for node insertion and deletion to achieve desired mesh density, and edge swapping and node movement to improve element quality. While this methodology has been implemented within the FUN3D suite, it is being structured to allow inputs from other flow solvers.

The full adaptation algorithm is first demonstrated on the 70° sphere cone for Mach 6 flow. Improvement in both the flowfield resolution and accuracy of the final aerodynamic quantities are shown with this example. The methodology is next applied to a series of Mach 10 computations on the Apollo capsule geometry using the inviscid, unstructured FELISA flow solver[38,39] and compared to uniform mesh refinement. The flowfields computed with the adapted meshes clearly resolve the flow, and particularly the bow shock, better than a uniformly refined mesh with a similar number of nodes. Examination of integrated forces and moments as a function of grid size shows that the adaptive methodology generally provides answers with lower numerical uncertainties than achieved through uniform refinement alone.

II. Flow Solvers

The initial series of examples employs the FUN3D suite of codes. The FUN3D standard compressible flow solver employs an unstructured finite-volume tetrahedral method with an implicit point-iterative method or an implicit line relaxation scheme.[40] This formulation is referred to as the standard compressible solver throughout this paper. FUN3D solves Euler and Reynolds-averaged Navier-Stokes (RANS) flow equations. A generic gas formulation, based on the standard compressible solver, is also available, and better accommodates hypersonic flows.[2,3] The present study employs the Euler equations with both the standard compressible and generic gas formulations.

The Mach 10 Apollo capsule examples use the FELISA inviscid flow solver. The FELISA software package consists of a set of computer codes for unstructured grid generation, and the simulation of three-dimensional steady inviscid flows using unstructured tetrahedral meshes.[38] Two flow solvers are available with FELISA– one applicable for transonic flows, and the other for hypersonic flows.[39] The hypersonic flow solver can use numerous gas models, including perfect gas air, equilibrium air, CF_4, and CO_2 in equilibrium.

III. Mesh Adaptation

A. Mesh Movement Gradient-Based Adaptation

A simple spring analogy mesh movement algorithm was introduced into the generic gas formulation of FUN3D by Gnoffo[2,3] and extended to the standard compressible formulation for the present work. The algorithm seeks to move existing mesh points toward strong shocks, thereby reducing cell size to cluster high aspect ratio tetrahedra to captured shocks to produce improved solutions across strong shocks. A tensile force on each node due to edge e is defined as

$$F_e^n = 1 + k f_n \Delta T_e, \tag{1}$$

where $k = 0.3$, $f_n = 1$ for inviscid flows. For viscous flows,

$$f_n = 1 - e^{-\phi/l_{targ}}, \tag{2}$$

where ϕ is the minimum of the radii of the spheres inscribed within each element surrounding node n and l_{targ} is the target length for the edge at node n,

$$l_{targ} = Re_{targ}\mu/\rho a. \tag{3}$$

The displacement at each node is then given by the sum of the tensile forces,

$$\Delta \mathbf{x}^n = \frac{1}{2}\left[\frac{\sum_e F_e^n \mathbf{x}^e}{\sum_e F_e^n} - \mathbf{x}^{\mathbf{n}}\right], \tag{4}$$

where $\mathbf{x}^e$ is the coordinate of the node opposite node n along edge e. The displacement is limited by $\frac{1}{2}\phi$ to avoid negative volumes. Nodal position updates on planar boundary surface nodes are constrained to move in that plane. This particular mesh movement algorithm does not interface with a CAD surface model. Therefore, nodes on curved surfaces are prohibited from moving. This simple methodology produces small movements and only requires the execution time of approximately two flow solver iterations. The adaptation is typically applied every 20-50 flow solver iterations over the span of several thousand flow solver iterations. This methodology has the advantage that is simple to apply within a flow simulation, but will draw points away from other important areas behind the shock if there are not sufficient mesh points.

B. Full Gradient-Based Adaptation

The gradient-based adaptation scheme for FUN3D uses the same mesh modification mechanics as the adjoint-based adaptation.[36] These schemes communicate the desired resolution of the mesh to the mesh modification modules via an anisotropic metric.[6,35] Both the gradient-based and adjoint schemes use a Mach number Hessian[35] to determine the anisotropy. However, for the gradient-based scheme, the isotropic scaling of the desired mesh adaptation is controlled by a gradient-based parameter instead of the adjoint error estimate.

Exact local (or global) error quantification is not possible because the exact solution is not available for general problems. Gradient-based adaptation indicators are computable surrogates. Some of the weaknesses of purely gradient based indicators is addressed by including a local element length scale. Undivided differences have been used by a number of researchers to estimate local error, see references 7 and 9. The jump,

J_e, in ρ over an edge is used for the isotropic scaling of mesh adaptation metric in the initial implementation of the scheme,

$$J_e = |\Delta\rho|. \tag{5}$$

The local error adaptation intensity, $\hat{I}$, is then computed at each node as the largest J_e of an incident edge normalized by a user-specified edge jump tolerance, J_t,

$$\hat{I} = \max_{\text{incident edges}} \left(\frac{J_e}{J_t} \right). \tag{6}$$

A new requested characteristic isotropic mesh size, h_1, is calculated from the current estimated characteristic isotropic mesh size, h^0, and the under-relaxed nodal intensity,

$$h_1 = h^0 \min \left(C, \left(\frac{1}{\hat{I}} \right)^{\omega} \right), \tag{7}$$

where ω is the relaxation factor (typically 0.2) and C is the maximum allowed coarsening factor (typically no larger than 115%) to prevent over-coarsening in smooth regions of the solution. The anisotropic metric is then computed from h_1 as in Reference 35. It is recognized that $\Delta\rho$ is not the only choice available for the adaptive indicator. Gradients of other flow quantities (pressure, Mach, entropy, for example) have been used by numerous other researchers, and the implementation of the methodology is such that this parameter can be easily changed. However, the current study did not investigate alternate formulations of the isotropic scaling parameter.

The basic adaptation mechanics are described by Park and others in several recent papers.[34–36,41] The adaptation modules use local mesh operators to remove edges by collapsing those that are too small as compared to the specified anisotropic metric. Edges that are too long as compared to the specified anisotropic metric are split. Grid element quality is improved by swapping edges to new configurations. Node locations are also modified to improve the quality of the worst incident element. This smoothing is performed in the interior and boundary of the domain. The new nodes that are inserted by an edge split on the boundary are moved to their final position on a high fidelity boundary representation with a global linear elasticity mesh movement scheme. The interface with high fidelity boundary representation is facilitated by the GridEx[42,43] framework, which links various mesh generation and adaptation strategies to geometry through the Computational Analysis Programming Interface[44–46] (`CAPrI`). With this methodology, surface nodes can be moved and inserted on the native CAD geometry. Surface adaptation within the gradient-based adaptation scheme is primarily limited to meshes generated by GridEx, so that boundary fidelity is maintained. The exception is for surface geometries that can be defined analytically.

For inviscid applications, both anisotropic surface and field adaptation can be prescribed. The mechanics to automatically adapt 3-D meshes with highly anisotropic regions near curved boundaries are under development. In high Reynolds number viscous cases, the mesh can only be automatically adapted in the field outside of the high aspect ratio cells near the surface. This is accomplished by freezing the boundary layer points, as demonstrated in Reference 36. The viscous adaptation capability is not demonstrated in the current paper, but is available with the gradient-based adaptation. The adaptation algorithm can additionally be controlled by setting the number of adaptation cycles to be used, setting the relaxation factor to reduce the change to the grid for each iteration, and choosing to smooth the resulting mesh. Mesh size and anisotropy are controlled by the choice of isotropic metric, J_e (`adapt_feature_type`), specified jump toler-

ance, J_t (`adapt_output_error`), and the maximum aspect ratio, edge length, and coarsening, C, allowed. When the adaptation is coupled with a flow solver other than FUN3D, a translation must be made from the mesh and flow solution data structures to FUN3D restart files. A translator for the FELISA software package was completed for this study.

IV. Computational Results

The gradient-based adaptation schemes are initially demonstrated with flow over a 70° sphere cone computed with the FUN3D flow solver. The spring adaptation is applied at Mach 2, and the full gradient-based scheme is applied at Mach 6. The full gradient-based scheme is then compared with uniform mesh refinement for a Mach 10 Apollo capsule case, computed with FELISA.

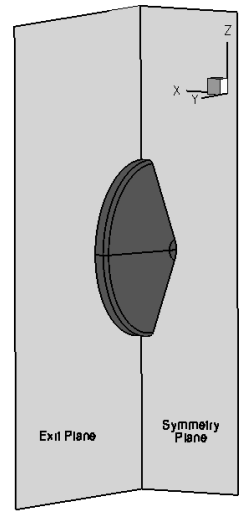


Figure 1. 70° sphere cone geometry with computational domain.

A. 70° Sphere Cone

The 70° sphere cone geometry is typical of planetary entry configurations such as Pathfinder and the recent Mars entry vehicles. Only the forebody is considered here, to focus on resolving the bow shock adequately. The computational domain cuts off the aft section of the vehicle, and is shown in Figure 1.

1. Mesh Movement via Spring Analogy, Mach 2

A spring adaptation cycle was performed on the initial mesh at Mach 2, zero degree angle of attack (α) using the generic gas formulation. The flow solver was run on the initial mesh until the integrated forces were converged ($C_N = -0.00028$). Then the spring adaptation was performed once every 50 flow solver iterations over a period of 5,000 iterations, for a total of 100 adaptations. Finally, the flow solver was allowed to run to iterative convergence without further mesh adaptations.

Figure 2 shows the iterative convergence history of the density residual and the normal force coefficient. The first 1,000 iteration are performed with a first-order spatial operator, and a limited second-order spatial operator is enabled after 1,000 iterations. The spike in residual at 1,000 iterations is due to the change in spatial operator. Once the specified adaptation cycle was completed, the CFL number was able to be increased, the residual dropped five orders of magnitude, and the overall iterative convergence was improved from the solution on the initial mesh. More importantly, the adapted mesh solution gives $C_N = -0.00008$, which is much closer to the expected value of zero than the $C_N = -0.00028$ given by the initial mesh.

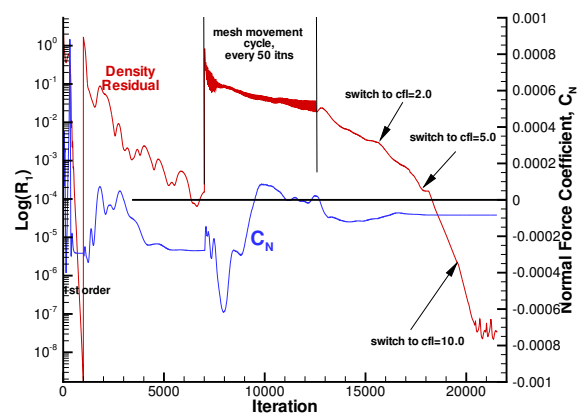
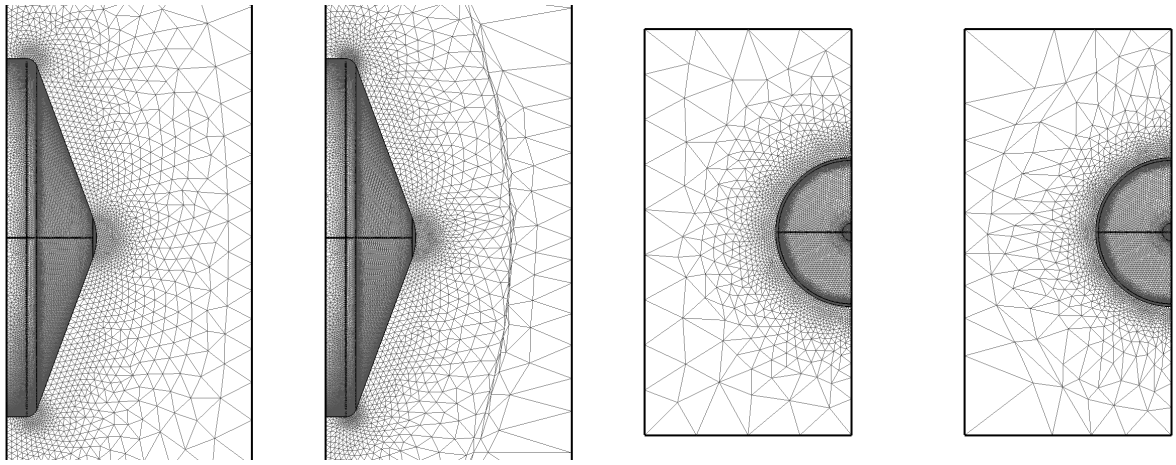


Figure 2. Iterative convergence of 70° sphere-cone case at Mach 20 with spring adaptation.

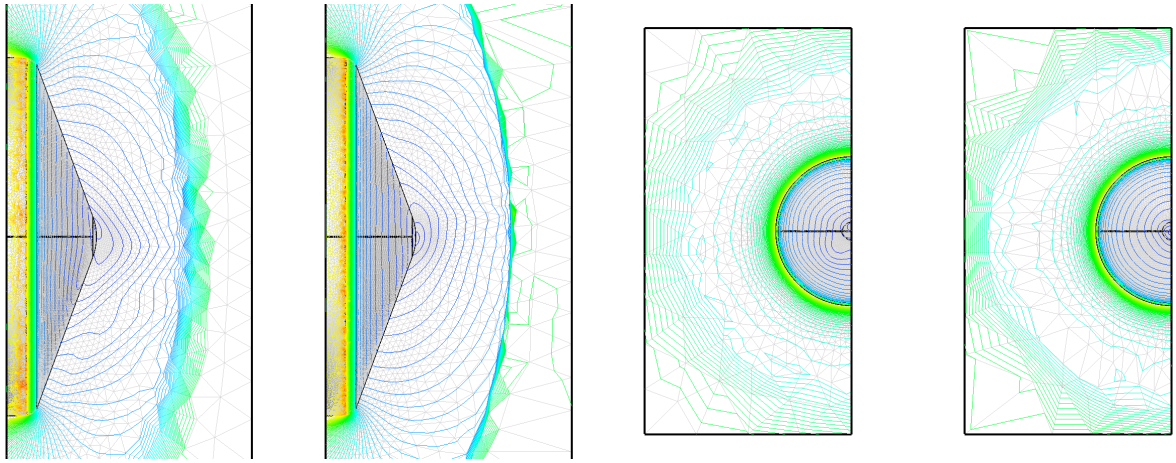
The symmetry and exit plane meshes for the initial and adapted meshes are shown in Figure 3. The clustering of points to the shock in the adapted mesh is very pronounced at the front of the bow shock on the symmetry plane, and the mesh movement has not drawn an excessive number of points out of the region between the shock and the body. There is less clustering where the bow shock as it crosses the exit plane, due both the sparseness of the initial mesh and the weaker shock strength.



(a) Initial. (b) Adapted. (c) Initial. (d) Adapted.

Figure 3. Symmetry and exit plane meshes for Mach 2 flow solution with spring adaptation.

Mach number contours for the initial and adapted meshes are shown in Figure 4. The Mach number contours in the symmetry plane are smoother and more symmetric for the adapted mesh and the resolution normal to the bow shock is improved. The exit plane view of the contours shows the improved symmetry



(a) Initial. (b) Adapted. (c) Initial. (d) Adapted.

Figure 4. Symmetry and exit plane Mach number contours for Mach 2 flow solution with spring adaptation.

of the contours in the stagnation region as well as some sharpening of the shock. Note that the shock is still poorly resolved at the exit plane due to the coarseness of the mesh. Since the spring adaptation does not add nodes to the mesh, the initial mesh must have sufficient mesh density in the vicinity of the shocks. The improvement in the symmetry of the flow solution, the increased bow shock normal resolution, and the reduced error in the normal force is dramatic with the spring adaptation. These improvements demonstrate the importance of properly resolving the bow shock.

This demonstration case was repeated using the standard FUN3D compressible air formulation, and the resulting mesh was similar. The flow solution on the spring-adapted mesh, however, was not as good as the initial flow solution in terms of both convergence and C_N. The poor performance of the standard compressible solver is due to the difference in the flux formulation and limiting scheme between the generic gas and standard compressible solvers. The STVD scheme in the generic gas flow solver is better able to handle the discontinuity across the high aspect ratio cells produced by the spring adaptation. Thus, the usefulness of a mesh movement scheme designed to cluster nodes along strong shocks, effectively providing highly stretched cells with faces aligned to the shock, is limited by the solver algorithm's ability to handle discontinuities across such cells.

2. Full Gradient-Based Adaptation, Mach 6

The full gradient-based adaptation scheme is demonstrated for a Mach 6 case. The initial mesh was similar to the initial mesh in the previous spring adaptation example. The initial solution was computed with the flow solver using the standard compressible path. After each adaptation cycle, the solver was run for 1,500 iterations before the next adaptation. The time for the adaptation is comparable to the time for a complete flow solution, so the full adaptation process is used sparingly and with relatively converged solutions. This contrasts to the simpler mesh movement scheme that is repeatedly employed and gives much smaller perturbations than the full adaptation process. The baseline adaptation parameters used for this series of adaptations are given in Table 1, and the resulting mesh sizes are shown in Table 2.

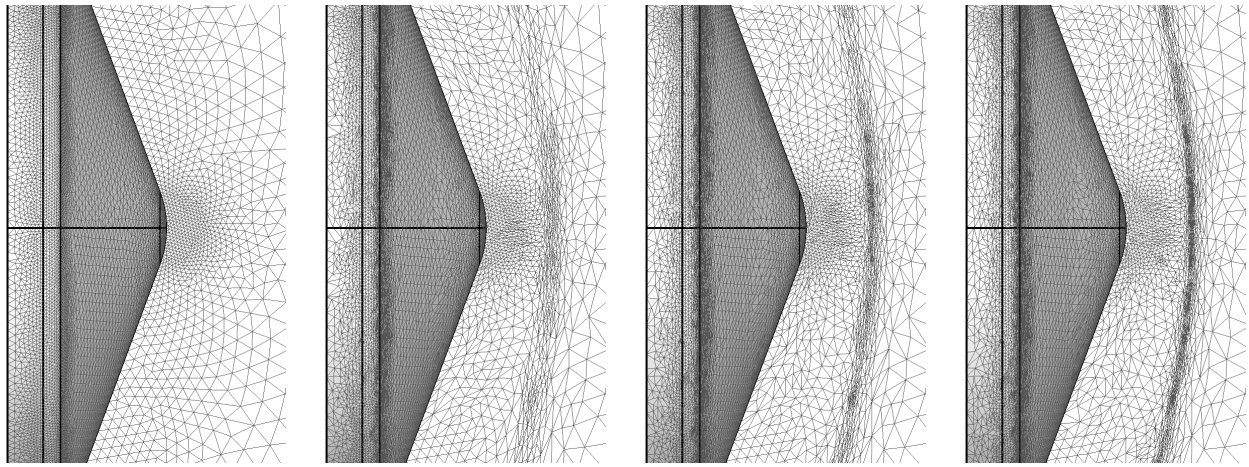
Table 1. Adaptation parameters, 70° sphere cone.

Parameter	Description	Value
`adapt_feature_type`	Definition of scalar adaptation metric, J_e	$\Delta\rho$
`adapt_output_error`	User specified jump tolerance J_t	0.1
`adapt_maxratio`	Maximum aspect ratio allowed for adapted cells	5.0
`adapt_maxedge`	Maximum edge length allowed	15.0
`adapt_coarsen`	Used to compute C, 1.0 allows no mesh coarsening, 2.0 allows 115% coarsening	2.0
`adapt_smooth_surface`	Smoothing of surface mesh after adaptation	*true*
`adapt_cycles`	Number of adaptation steps taken for one level of adaptation	6

Symmetry and exit plane meshes for each adaptation level are shown in Figure 5 and 6. The location of the strongest part of the bow shock is obvious on the symmetry plane, and is less prominent downstream as the shock weakens. There is significant refinement in the shoulder region of the vehicle, as seen in the exit plane plot, Figure 6(d). There is an uneven clustering of nodes evident in several locations, indicating that the adaptation parameter is not smooth. This is a problem typical of local error estimation methods.

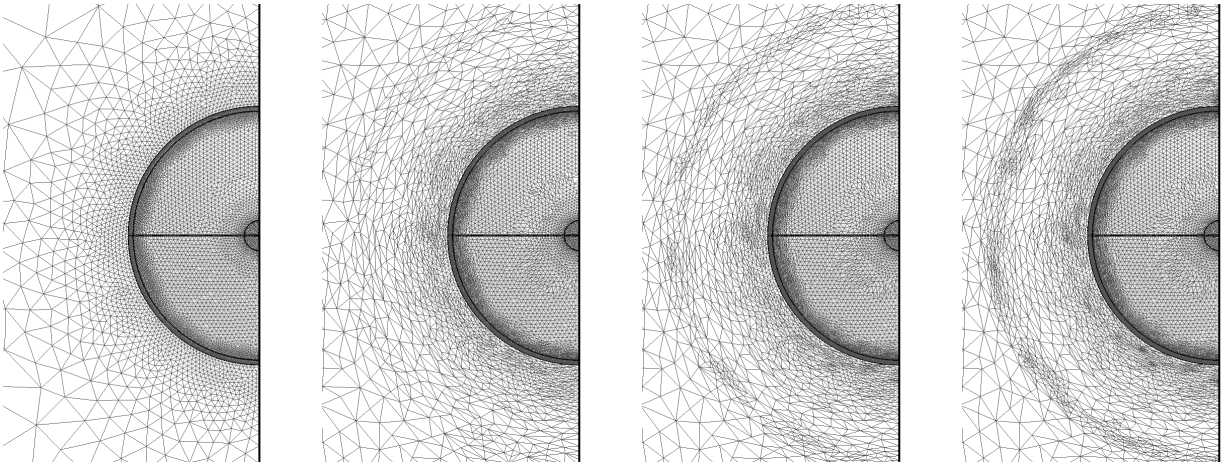
Table 2. Adapted mesh sizes, 70° sphere cone.

	Mesh refinement level					
Parameter	Initial	02	03	04	05	06
Boundary faces	32k	26k	29k	32k	36k	43k
Volume nodes	105k	128k	153k	200k	312k	610k
Volume cells	599k	723k	862k	1.12M	1.75M	3.44M



(a) Initial mesh, Cycle 1 (b) Cycle 3 (c) Cycle 4 (d) Cycle 5

Figure 5. Symmetry plane meshes for multiple adaptation cycles using full adaptation process, Mach 6, α =0°.



(a) Initial mesh, Cycle 1 (b) Cycle 3 (c) Cycle 4 (d) Cycle 5

Figure 6. Exit plane meshes for multiple adaptation cycles using full adaptation process, Mach 6, α =0°.

Convergence of the residual and normal force coefficient is given in Figure 7. The first 1,000 iterations are spatially first-order and the remaining iterations are second-order. The normal force history shows improvement with the mesh refinement. The expected normal force is zero for this symmetric non-lifting case.

Mach number contours for the converged solution on each mesh are given in Figures 8, 9, and 10. The adapted meshes finely resolve the bow shock in its normal direction. While the mesh adaptation has definitely improved the symmetry of the contours, the final mesh still has mild asymmetry in the stagnation region, Figure 10(d).

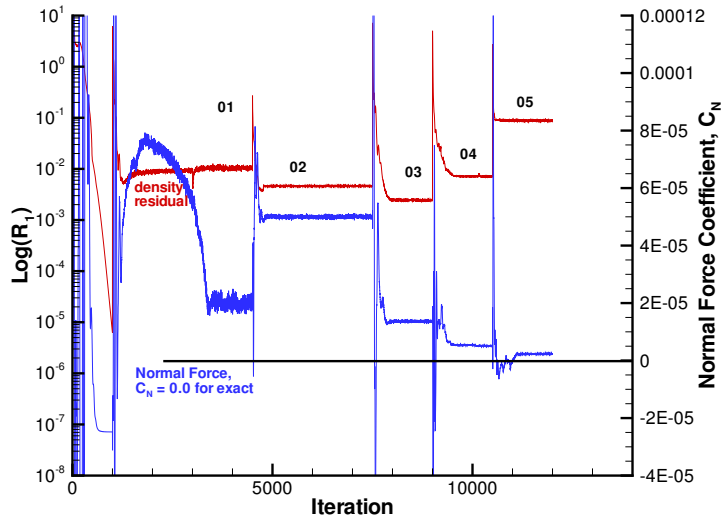
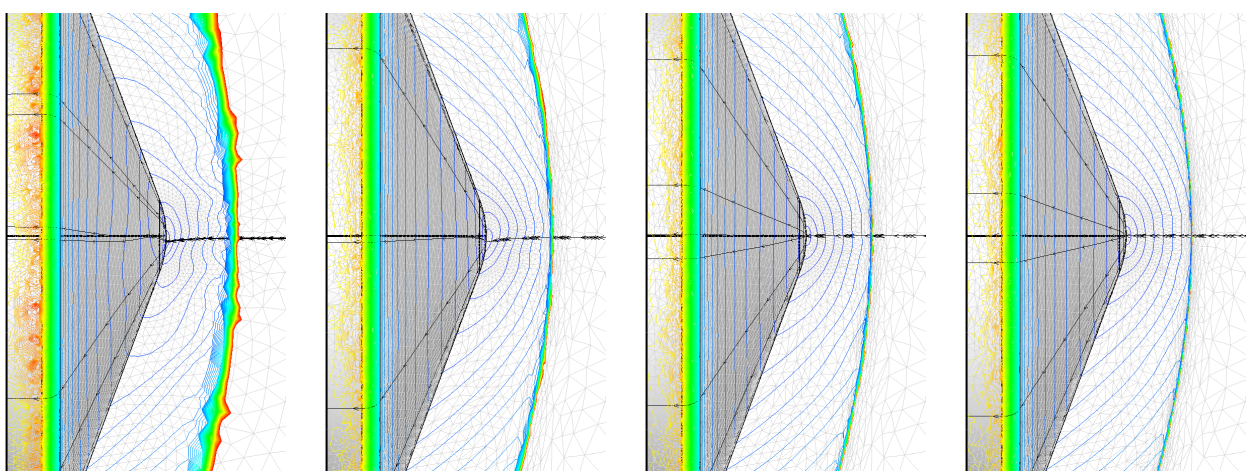
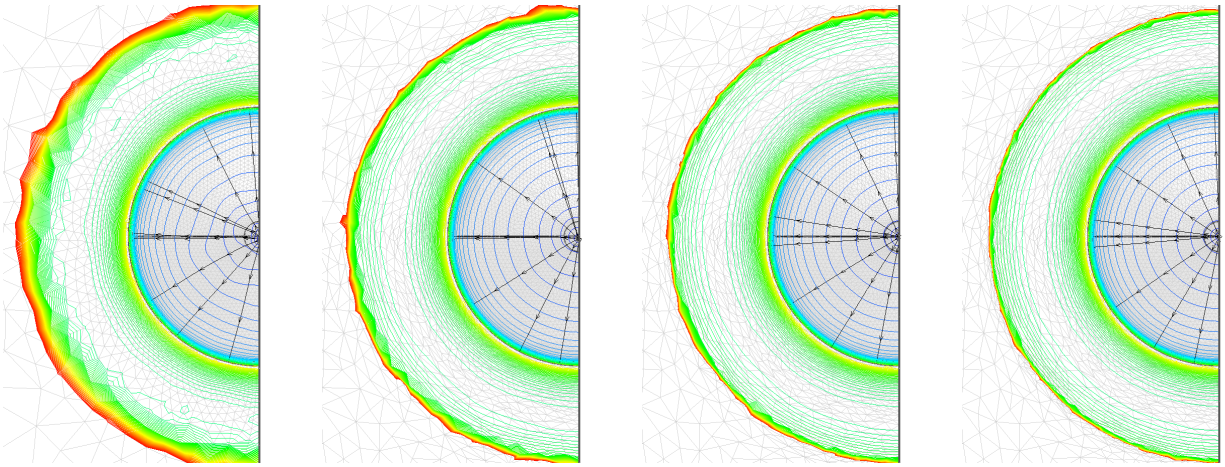


Figure 7. Iterative convergence of Mach 6 flow solution on initial mesh and four adapted meshes.



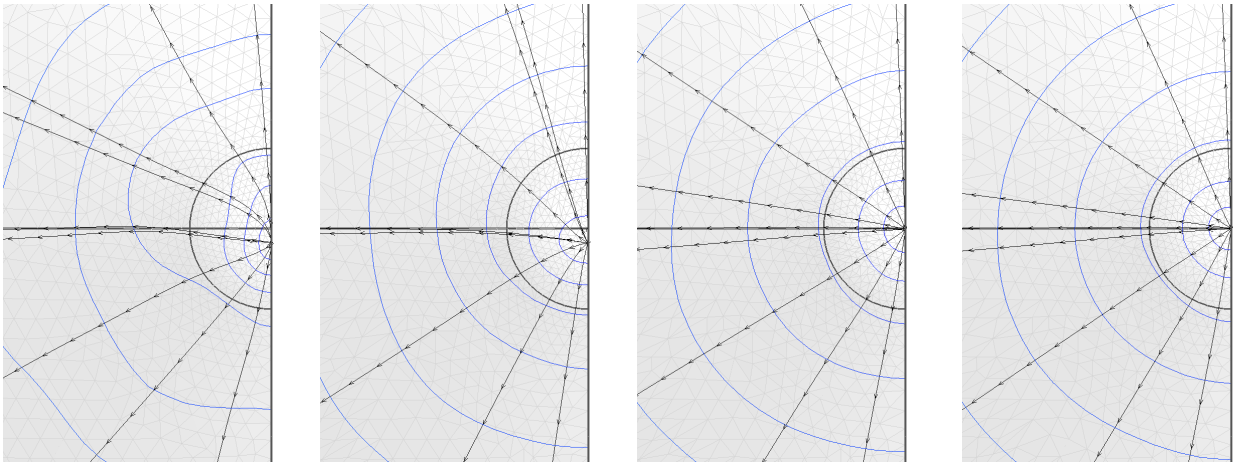
(a) Initial mesh, Cycle 1 (b) Cycle 3 (c) Cycle 4 (d) Cycle 5

Figure 8. Symmetry plane Mach number contours and streamlines for multiple adaptation cycles using full adaptation process, Mach 6, $\alpha = 0°$.



(a) Initial mesh, Cycle 1 (b) Cycle 3 (c) Cycle 4 (d) Cycle 5

Figure 9. Mach number contours and streamlines on the full body and exit plane for multiple adaptation cycles using full adaptation process, Mach 6, α =0°.



(a) Initial mesh, Cycle 1 (b) Cycle 3 (c) Cycle 4 (d) Cycle 5

Figure 10. Mach number contours and streamlines in the stagnation region for multiple adaptation cycles using full adaptation process, Mach 6, α =0°.

B. Apollo Capsule, Mach 10

The aerodynamics for the Apollo capsule at Mach 10 and an angle of attack of 150° have been computed using the FELISA flow solver coupled with the full gradient-based adaptive methodology demonstrated in the previous section. The resulting flow solutions and aerodynamics are compared to a series of uniform mesh refinements in order to establish the improvement in both process time and accuracy for the gradient-based adaptation approach.

1. Problem Setup

The Apollo geometry is shown in Figure 11. The computational geometry was analytically developed based on the dimensions as supplied by the CEV program.

An initial mesh for the baseline geometry was developed using GRIDEX. The spacings for the initial mesh were then scaled to produce a family of meshes to be used in mesh refinement studies. Table 3 gives the mesh sizes for the uniform refinement series meshes. The mesh labeled `ab2na` was the primary mesh, used as a starting point for the adaptations. The coarsest, baseline, and finest meshes in the vicinity of the body are shown in Figure 12.

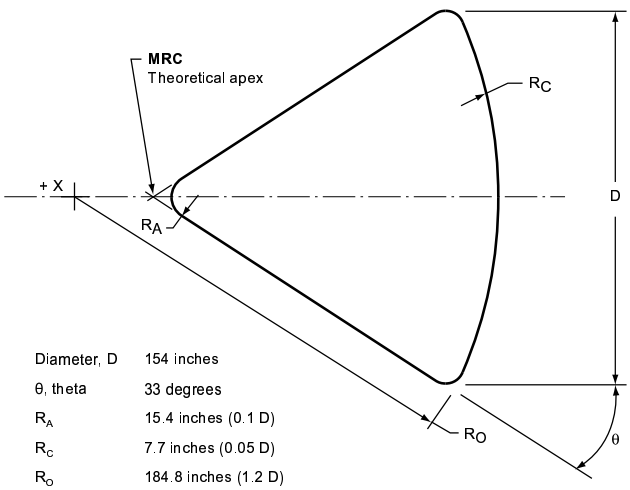


Figure 11. Apollo Capsule Geometry.

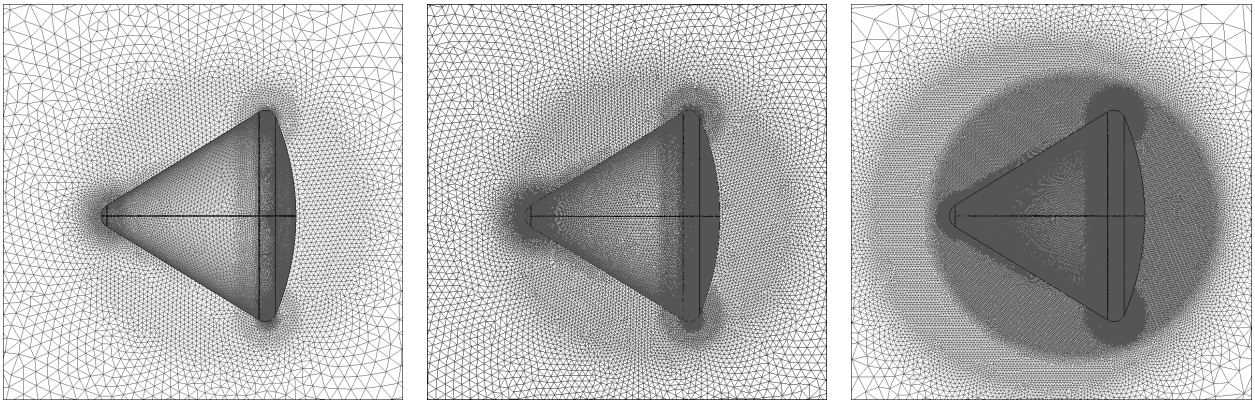
Table 3. Mesh sizes for uniform refinement series meshes, Apollo.

		Volume	
Tag	Boundary faces	Nodes	Cells
`ab2nf`	16k	220k	1.3M
`ab2nb`	21k	330k	1.9M
`ab2na`*	34k	630k	3.7M
`ab2nc`	45k	970k	5.7M
`ab2nd`	73k	2.0M	12M
`ab2ne`	150k	6.0M	36M

* This mesh served as the baseline adaptation mesh.

Flow solutions were computed using FELISA on the Apollo capsule at a Mach 10.14 equilibrium air condition[a] and an angle of attack of 150° on the baseline mesh using FELISA, and then adapted using the full gradient-based methodology. The adaptation parameters used in this series of computations are given in Table 4. Solutions on four adapted meshes and all six uniform-refinement meshes were computed. The sizes of the adapted meshes are given in Table 5.

[a]This corresponds to an AEDC Tunnel C condition used during Apollo testing:[47] $U_\infty = 1614.8 m/s$, $\rho_\infty = 0.00656 kg/m^3$, and $T_\infty = 63.1$.



(a) Coarsest mesh, `ab2nf`.

(b) Baseline mesh for adaptation, `ab2na`.

(c) Finest uniformly-refined mesh, `ab2ne`

Figure 12. Coarsest (`ab2nf`), baseline (`ab2na`), and finest (`ab2ne`) meshes in uniform refinement series.

Table 4. Gradient-based adaptation parameters for FELISA Apollo capsule, Mach 10 cases.

Parameter*	Value
`adapt_output_error`	0.025
`adapt_maxratio`	10.0
`adapt_maxedge`	35.0
`adapt_coarsen`	1.2
`adapt_smooth_surface`	*true*
`adapt_cycles`	3
`adapt_feature_type`	$\Delta\rho$
`current_spacing metric`	$l_{\min}$

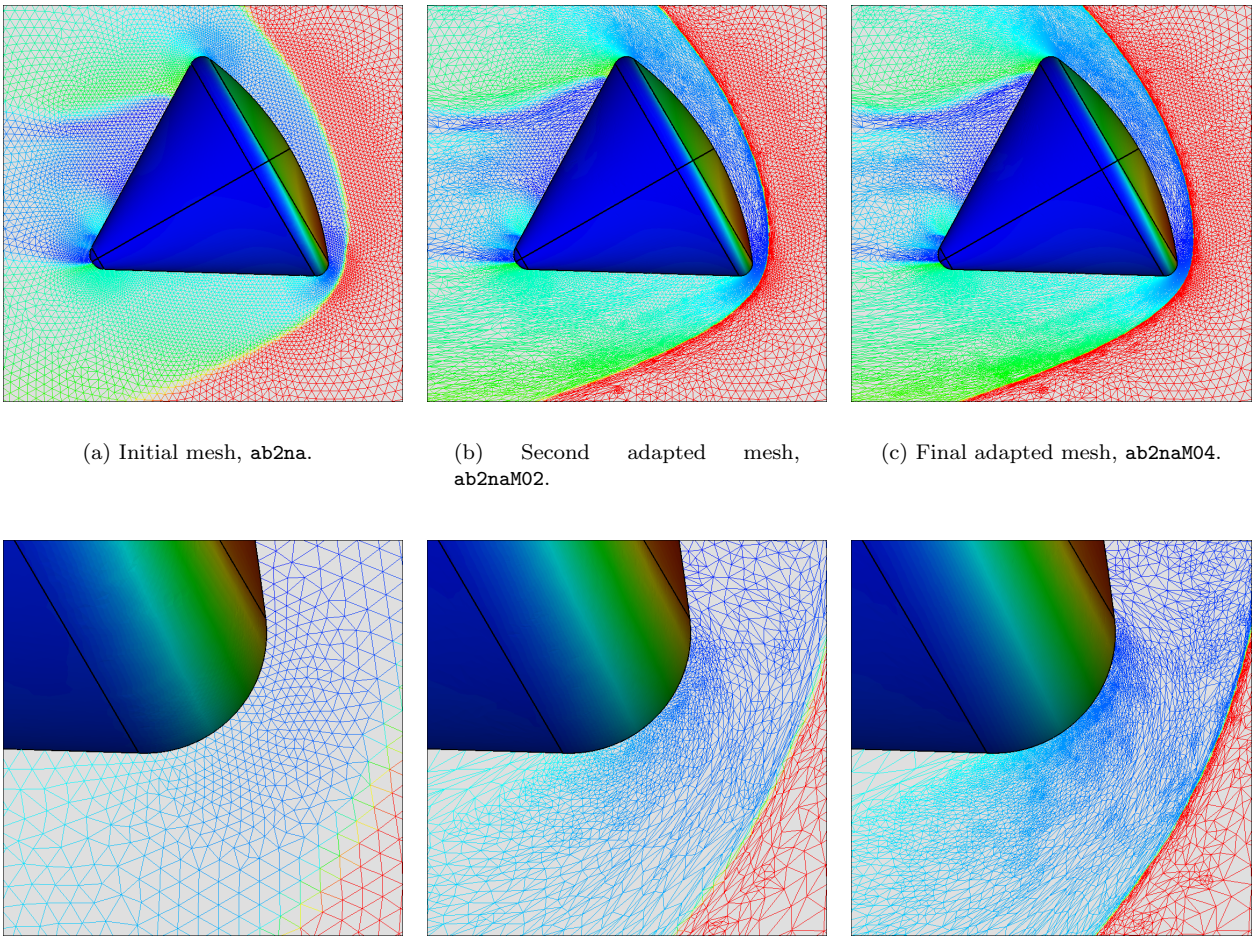
*See table 1 on page 8 for descriptions.

Table 5. Mesh sizes for gradient-based adaptation series, Apollo capsule, Mach 10, $\alpha = 150°$.

Tag	Boundary faces	Volume nodes	Volume cells
`ab2na`	34k	630k	3.7M
`ab2naM01`	40k	900k	5.2M
`ab2naM02`	50k	1.3M	7.6M
`ab2naM03`	67k	2.4M	13M
`ab2naM04`	94k	5.0M	28M

2. *Adapted Meshes and Flowfields*

Figure 13 shows each adapted mesh for the 150° angle of attack case. The symmetry plane mesh is colored by Mach number and the body is colored by C_p. The mesh in the vicinity of the bow shock has been significantly enriched, especially in the direction normal to the shock. There is also significant adaptation downstream of the body, where flow does not appreciably affect the forces and moments on the body. There is some mild coarsening in the wake region that is allowed because of the very low density, and thus small $\Delta\rho$ parameter. The adapted mesh is much finer in the shoulder region, in order to resolve the expansion around the shoulder. The effect of the mild coarsening parameter is seen just aft of the expansion around the shoulder.



(a) Initial mesh, `ab2na`.

(b) Second adapted mesh, `ab2naM02`.

(c) Final adapted mesh, `ab2naM04`.

(d) Initial mesh, `ab2na`, shoulder region.

(e) Second adapted mesh, `ab2naM02`, shoulder region.

(f) Final adapted mesh, `ab2naM04`, shoulder region.

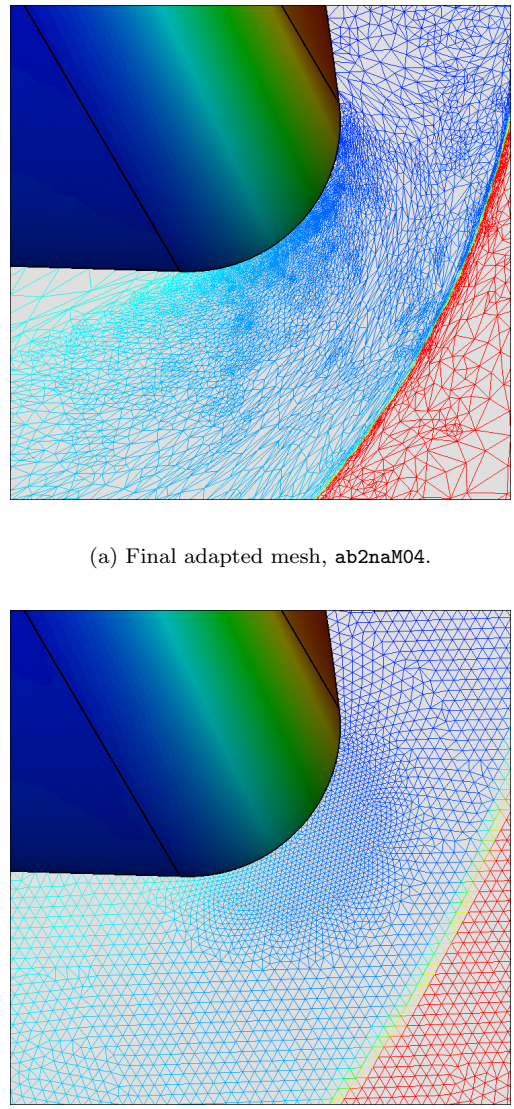
Figure 13. Symmetry plane meshes colored with Mach, body with C_p, Apollo capsule, adaptation series, Mach 10.1, $\alpha = 150°$.

The final adapted mesh in the shoulder region is contrasted with the finest uniformly-refined mesh in Figure 14. The adapted mesh has 1.0M fewer mesh points than the finest uniformly-refined mesh, but clearly has much smaller cells in regions with high gradients.

Figure 15 show flowfield Mach number and C_p contours for the same set of adapted mesh solutions on the symmetry plane and body. The sharpening of the shock due to the adaptation is clearly evident on the symmetry plane. The leeside flow changes significantly with successive adaptation, particularly in the location of the 'inviscid separation' aft of the shoulder. This region, however, is not important from an aerodynamic perspective because C_p is near zero throughout the region. The stagnation point location shifts with the adaptation (Figures 15(d)–15(f)), with the final adaptation showing more circular contours around the stagnation point. There are small but noticeable changes in the contours on the body in the shoulder region, particularly in the beginning of the conical section of the capsule.

The marked improvement in feature resolution for the adaptation series is contrasted with the flowfield countours for the uniform mesh refinement series in Figure 16. The final adapted mesh contours are repeated from Figure 15 for clarity. The final adapted refinement clearly resolves the flow features better than the uniform refinement.

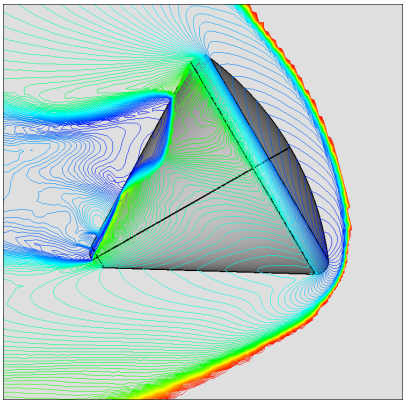
The C_p distribution along the vehicle centerline for several of the uniformly refined meshes is shown in Figure 17(a), with the capsule geometry for reference. Figure 17(b) shows the C_p distribution in the stagnation region for the adapted meshes (symbols) and all of the uniformly-refined meshes (lines). The maximum C_p for the uniformly-refined meshes varies in both maximum value ($1.820 < C_p < 1.846$) and axial location ($10.0 < x_{stag} < 11.8$), with no particular pattern. The C_p distribution is similarly varied. The C_p distributions for the adapted meshes show much less variation in shape and a tighter clustering of the maximum C_p. Figure 17(c) shows the C_p distribution at the end of the expansion around the windward shoulder. Here, the C_p distribution for the uniformly-refined mesh series is varying systematically, with the (over-)expansion and subsequant recompression sharpening with successive refinement. The C_p distribution for the adapted meshes are similar to the finer uniformly-meshes. The final adapted mesh has a sharper minimum than the finest adapted mesh, and has a mesh spacing of approximately one-half, suggesting again that the adapted meshes are providing better flow solutions than similarly sized uniformly refined meshes.



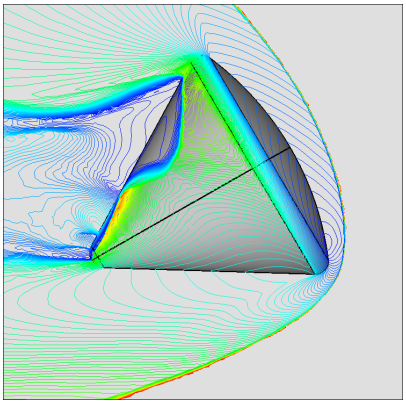
(a) Final adapted mesh, `ab2naM04`.

(b) Finest uniformly-refined mesh.

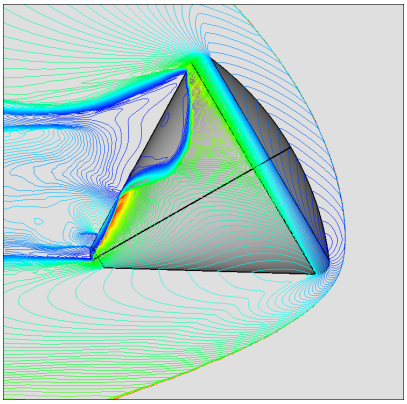
Figure 14. Comparison of adapted and uniformly-refined meshes in the shoulder region, Apollo capsule, adaptation series, Mach 10.1, $\alpha = 150°$.



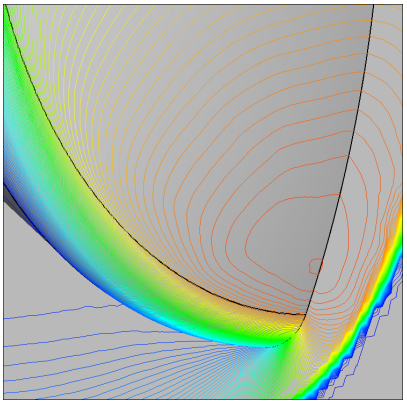
(a) Initial mesh, Mach contours.



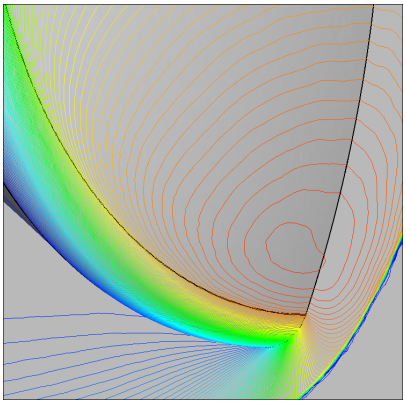
(b) Second adapted mesh, Mach contours.



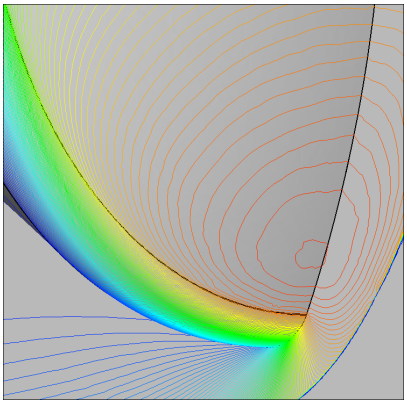
(c) Final adapted mesh, Mach contours.



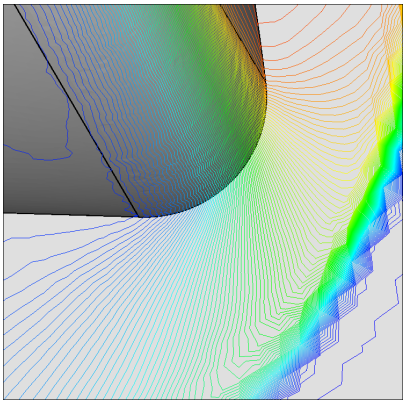
(d) Initial mesh, Stagnation region, C_p contours.



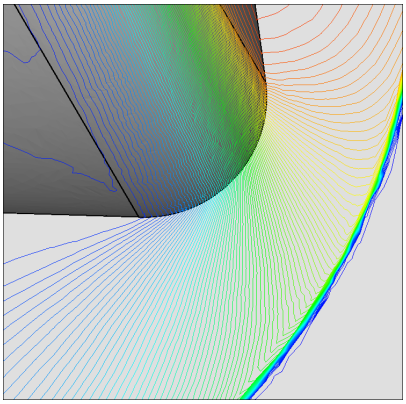
(e) Second adapted mesh, Stagnation region, C_p contours.



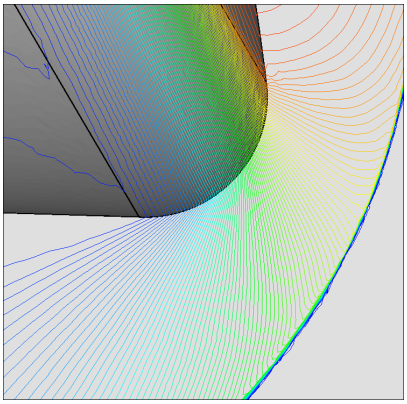
(f) Final adapted mesh, Stagnation region, C_p contours.



(g) Initial mesh, C_p contours.

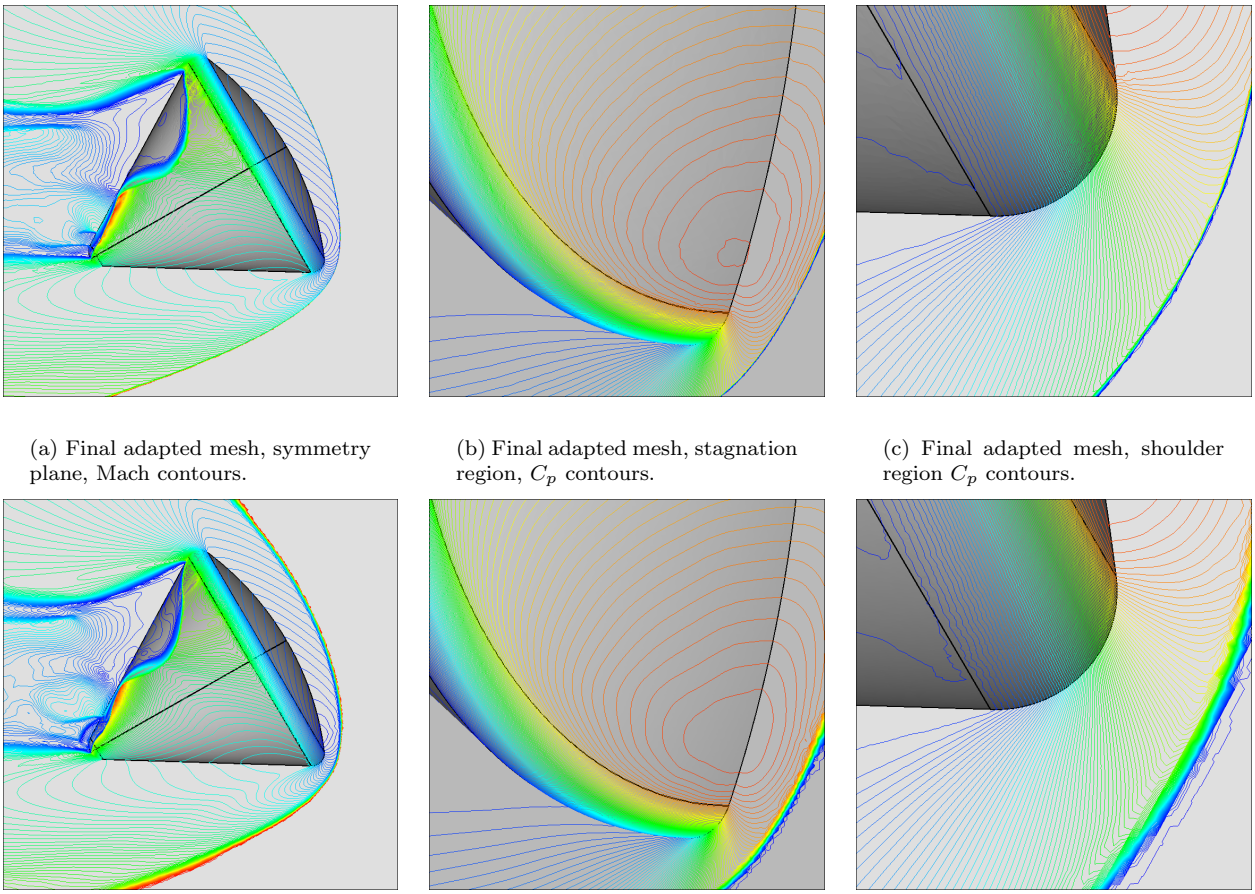


(h) Second adapted mesh, C_p contours.



(i) Final adapted mesh, C_p contours.

Figure 15. Flowfield contours, Apollo capsule, gradient-based adaptation series, Mach 10.1, $\alpha = 150°$.



(a) Final adapted mesh, symmetry plane, Mach contours.

(b) Final adapted mesh, stagnation region, C_p contours.

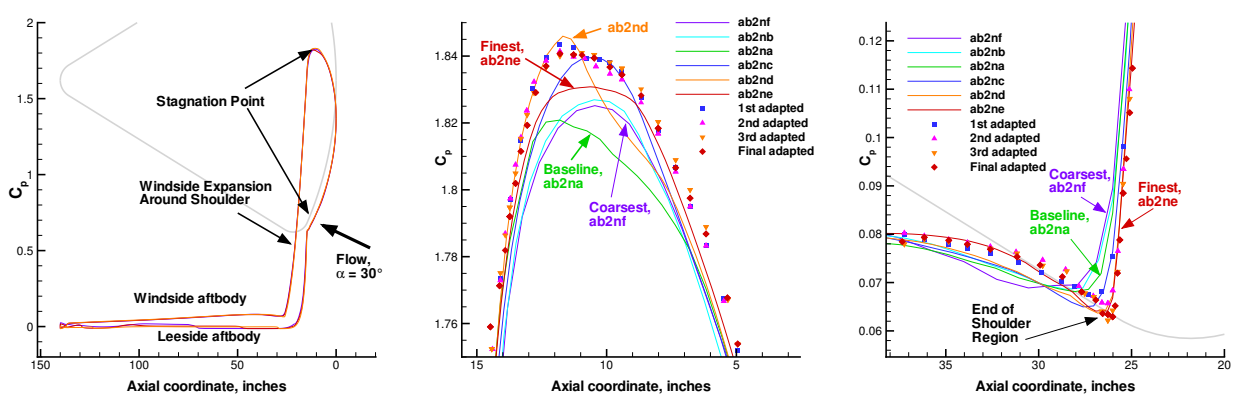
(c) Final adapted mesh, shoulder region C_p contours.

(d) Finest mesh (`ab2ne`), symmetry plane, Mach contours.

(e) Finest mesh (`ab2ne`), stagnation region, C_p contours.

(f) Finest mesh (`ab2ne`), shoulder region, C_p contours

Figure 16. Flowfield contours for final adapted mesh and finest uniform mesh, Apollo, Mach 10.1, $\alpha = 150°$.



(a) Uniform: coarse, mid, and fine.

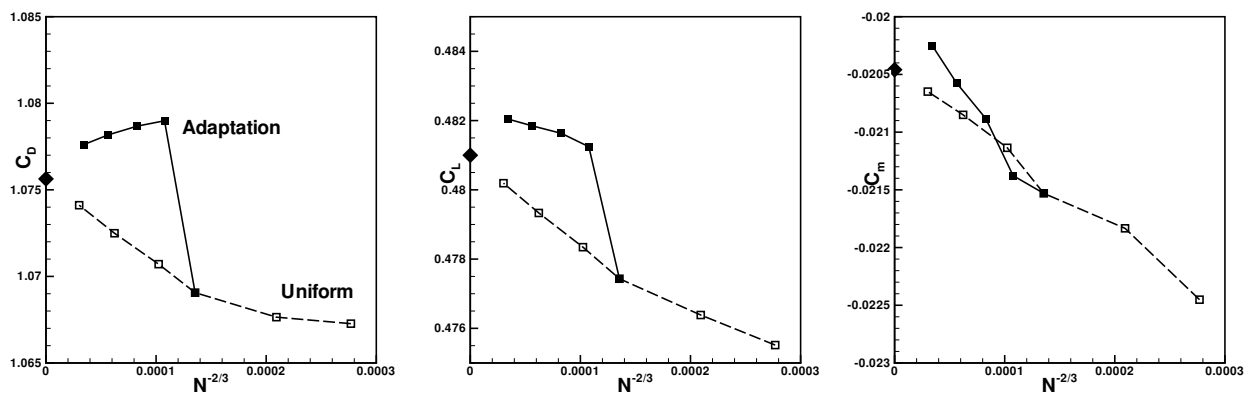
(b) Adapted, stagnation region.

(c) Adapted, shoulder region.

Figure 17. Centerline C_p for uniformly-refined and adapted meshes, Apollo capsule, Mach 10.1, $\alpha = 150°$.

3. Grid Convergence

The computed aerodynamics (C_L, C_D, C_m about apex) are shown in Figure 18, plotted against $N^{-2/3}$, where N is the number of nodes in the mesh. This assumes that the characteristic length of the mesh, h, varies with the inverse of the cube root of the number of nodes, $h \approx N^{-1/3}$. The discretization scheme is also assumed to be second order, so that computed outputs vary linearly with h^2. The axis is such that the finest meshes are on the left side. The dashed lines show the uniform mesh refinement series, and the solid lines show the gradient-based refinement series. The Richardson extrapolation value is shown as a point at $N^{-2/3} = 0$ The jump in the load coefficient between the initial mesh and the first adapted mesh results from the dramatic change in the distribution of points between the initial and first adapted mesh. This jump only occurred for this case in drag an lift, not pitching moment. Also note that the change in the coefficient values between the baseline and finest meshes is not large in absolute terms.



(a) C_D (b) C_L (c) C_m

Figure 18. Grid convergence of lift, drag, and pitching moment, Apollo capsule, Mach 10.1, $\alpha = 150°$.

The last few meshes in the uniform mesh refinement series appear to be in the asymptotic convergence region. However, the adapted meshes are also approaching an asymptotic value, which is different from the uniform mesh extrapolation. Past studies have shown both that the adapted mesh value can be wrong because a feature (such as the shock standoff distance) is refined early in the adaptation process when the feature is far from its true location, and is thus 'forced' into the incorrect location,[7] or that the uniform mesh series is actually not in the asymptotic range and that further refinement could shift the extrapolated value significantly.[36] The erratic C_p distribution in the stagnation region (Figure 17(b)) for the uniformly-refined mesh series indicates that the mesh sequence is not in the asymptotic convergence region in terms of stagnation point resolution, and suggests a rationale for accepting the adapted mesh results over the uniformly refined mesh results.

V. Concluding Remarks and Future Work

Two gradient-based adaptation capabilities have been implemented in the FUN3D suite to give the user an automated ability to improve a computational mesh for a particular application. The spring adaptation capability, integrated into the FUN3D solver, is a simple tool that can provide clustering of points to a shock

for a relatively well designed initial mesh. This study showed a benefit only when used with the STVD flux formulation in the generic gas path of FUN3D. A gradient-based adaptation capability that allows for node insertion and deletion, node movement, and edge swapping has been developed for FUN3D suite through REFINE and GRIDEX. The capability has been extended for use with FELISA. Overall, the 70° sphere cone and Apollo capsule results suggest that gradient-based adaptation gives a benefit in solution quality. The mesh adaptation process clearly gives better resolution of flow features, and will be very useful for problems where the general structure of the flow or a localized flow feature is important. For the examples shown here, the aerodynamics (lift, drag, and pitching moment) improved with the adaptation, both in absolute terms (C_N, 70° sphere cone) and relative to uniform mesh refinement (Apollo capsule).

While the basic implementation of a gradient-based adaptive refinement capability has been developed, there are two primary areas that should be addressed in order to have a fully-functional stand-alone capability. First, to improve process time and reduce disk storage needs for cases that use flow solvers other than FUN3D, the infrastructure to allow input and output of meshes and flow solutions from codes other than FUN3D should be further developed. Secondly, an interface to allow user-configurable specification of the scalar adaptation metric should be implemented to allow a wider range of formulations.

The gradient-based methodologies developed herein, like all gradient-based methodologies, have limitations that must be understood in order to apply the methodology effectively. First, there is no 'stopping' criteria. In the Apollo example, the adaptation was taken further than would have been done for a production case. Most of the improvement in feature resolution and in aerodynamic loads has occurred by the second adaptation, suggesting that the adaptation could have been stopped at this point. Second, the quality of the results is dependent on the formulation of the adaptation parameter. For the Apollo case, there were regions of the wake that were coarsened even though there were strong gradients in Mach number. Finally, there is no firm quantification of the error in the final solution with respect to important integrated quantities such as lift and drag. An adjoint-based adaptation process overcomes these limitations. Until an adjoint solver that is robust for problems with strong shocks is developed, this gradient-based adaptation can be used.

Acknowledgments

The development of the adaptation schemes was funded through the Hypersonics Project of the Agency's Foundations of Aerodynamics Program. The Apollo computations were performed under the CEV Aerosciences Project (CAP). The authors appreciate the supportive and collaborative environment created by the FUN3D development team, and recognize the benefits of working with such a dedicated team.

References

[1]Gnoffo, P. A., Hartung, L. C., and Greendyke, R. B., "Heating Analysis for a Lunar Transfer Vehicle at Near-Equilibrium Flow Conditions," AIAA Paper 93-0270, January 1993.

[2]Gnoffo, P. A., "Computational Fluid Dynamics Technology for Hypersonic Applications," AIAA Paper 2003–3259, 2003.

[3]Gnoffo, P. A. and White, J. A., "Computational Aerothermodynamic Simulation Issues on Unstructured Grids," AIAA Paper 2004–2371, 2004.

[4]Baker, T. J., "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3–4, 1997, pp. 243–273.

[5]Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M., and Vallet, M.-G., "Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles," *International Journal for Numerical Methods in Fluids*, Vol. 32, No. 6, 2000, pp. 725–744.

[6]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.

[7]Warren, G. P., Anderson, W. K., Thomas, J. T., and Krist, S. L., "Grid Convergence for Adaptive Methods," AIAA Paper 91-1592, 1991.

[8]Pirzadeh, S. Z., "A Solution-Adaptive Unstructured Grid Method by Grid Subdivision and Local Remeshing," *AIAA Journal of Aircraft*, Vol. 37, No. 5, September–October 2000, pp. 818–824, See also AIAA Paper 99–3255.

[9]Mavriplis, D., "Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes," *International Journal for Numerical Methods in Fluids*, Vol. 34, No. 2, 2000, pp. 93–111, See also AIAA Paper 97-0857.

[10]Aftosmis, M. J. and Berger, M. J., "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," AIAA Paper 2002-0863, 2002.

[11]Park, Y. M. and Kwon, O. J., "A parallel unstructured dynamic mesh adaptation algorithm for 3-D unsteady flows," *International Journal for Numerical Methods in Fluids*, Vol. 48, No. 6, 2005, pp. 671–690, See also AIAA Paper 2001–0865.

[12]Wood, W. A. and Kleb, W. L., "On multi-dimensional unstructured mesh adaption," AIAA Paper 99-3254, 1999.

[13]Castro-Diáz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., "Anisotropic Unstructured Mesh Adaptation for Flow Simulations," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, 1997, pp. 475–491.

[14]Pelletier, D., Turgeon, E., Lacasse, D., and Borggaard, J., "Adaptivity, Sensitivity, and Uncertainty: Toward Standards of Good Practice in Computational Fluid Dynamics," *AIAA Journal*, Vol. 41, No. 10, October 2003, pp. 1925–1933.

[15]Choi, S., Alonso, J. J., and Van der Weide, E., "Numerical and Mesh Resolution Requirements for Accurate Sonic Boom Prediction of Complete Aircraft Configurations," AIAA Paper 2004–1060, 2004.

[16]Djomehri, M. J. and Erickson, L. L., "An Assessment of the Adaptive Unstructured Tetrahedral Grid, Euler Flow Solver Code FELISA," TP 3526, NASA Ames Research Center, Dec. 1994.

[17]Cavallo, P. A. and Grismer, M. J., "Further Extension and Validation Of A Parallel Unstructured Mesh Adaptation Package," AIAA Paper 2005-0924, 2005.

[18]Monk, P. and Süli, E., "The Adaptive Computation of Far-Field Patterns by A Posteriori Error Estimation of Linear Functionals," *SIAM Journal on Numerical Analysis*, Vol. 36, 1998, pp. 251–274.

[19]Paraschivoiu, M., Peraire, J., and Patera, A., "A Posteriori Finite Element Bounds for Linear-functional Outputs of Elliptic Partial Differential Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 150, No. 1–4, 1997, pp. 289–312.

[20]Braack, M. and Rannacher, R., "Adaptive Finite Element Methods for Low-Mach Flows with Chemical Reactions," Vol. 3 of *30th Computational Fluid Dynamics*, von Karman Institute, 1999, pp. 1–93.

[21]Rannacher, R., "Adaptive Galerkin Finite Element Methods for Partial Differential Equations," *Journal of Computational and Applied Mathematics*, Vol. 128, 2001, pp. 205–233.

[22]Becker, R. and Rannacher, R., "An optimal control approach to error control and mesh adaptation," *Acta Numerica 2001*, 2001.

[23]Giles, M. B. and Süli, E., "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality," *Acta Numerica*, Vol. 11, 2002, pp. 145–236.

[24]Peraire, J. and Patera, A. T., "Asymptotic a posteriori finite element bounds for the outputs of noncoercive problems: the Helmholtz and Burgers equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 171, No. 1–2, 1999, pp. 77–86.

[25]Barth, T. and Deconinck, H., editors, *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Number 25 in Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2002.

[26]Pierce, N. A. and Giles, M. B., "Adjoint Recovery of Superconvergent Functionals from PDE Approximations," *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.

[27]Müller, J. D. and Giles, M. B., "Solution Adaptive Mesh Refinement Using Adjoint Error Analysis," AIAA Paper 2001–2550, 2001.

[28]Venditti, D. A. and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows," *Journal Computational Physics*, Vol. 176, 2002, pp. 40–69, See also AIAA Paper 2000–2244.

[29]Venditti, D. A. and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal Computational Physics*, Vol. 187, 2003, pp. 22–46.

[30]Venditti, D. A., *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[31]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22.

[32]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithm for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Fluids*, Vol. 128, 1996, pp. 391–408.

[33]Nielsen, E. J., *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier-Stokes Equations and a Discrete Adjoint Formulation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 1998.

[34]Park, M. A., "Adjoint–Based, Three–Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, Vol. 42, No. 9, September 2004, pp. 1854–1862, See also AIAA Paper 2002–3286.

[35]Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003–3849, 2003.

[36]Lee-Rausch, E. M., Park, M. A., Jones, W. T., Hammond, D. P., and Nielsen, E. J., "Application of a Parallel Adjoint-Based Error Estimation and Anisotropic Grid Adaptation for Three-Dimensional Aerospace Configurations," AIAA Paper 2005–4842, 2005.

[37]Nielsen, E. J. and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids Using Complex Variables," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 827–836, See also AIAA Paper 2005–324.

[38]Peiro, J., Peraire, J., and Morgan, K., "FELISA Reference Manual and User's Guide, Volume I," University of Wales/Swansea Report CR/821/94, 1994.

[39]Bibb, K. L., Peraire, J., and Riley, C. J., "Hypersonic Flow Computations on Unstructured Meshes," AIAA Paper 97-0625, January 1997.

[40]Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155, See also AIAA Paper 2003–0272.

[41]Jones, W. T., Nielsen, E. J., and Park, M. A., "Validation of 3D Adjoint Based Error Estimation and Mesh Adaptation for Sonic Boom Prediction," AIAA Paper 2006–1150, 2006.

[42]Jones, W. T., "An Open Framework for Unstructured Grid Generation," AIAA Paper 2002–3192, 2002.

[43]Jones, W. T., "GridEx – An Integrated Grid Generation Package for CFD," AIAA Paper 2003–4129, 2003.

[44]Haimes, R., "CAPRI: Computational Analysis PRogramming Interface," Massachusetts Institute of Technology, 2002.

[45]Haimes, R., "Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry," AIAA Paper 99–0776, 1999.

[46]Haimes, R., "CAPRI: Computational Analysis PRogramming Interface," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, July 1998.

[47]Snowden, J. C., "Data Report for Apollo Model FS-3 Wind Tunnel Tests in Tunnels B and C of the AEDC von Karman Gas Dynamics Facility," NASA-CR 117726, December 1962, Also SID-62-1247.