46th AIAA Aerospace Sciences Meeting, Jan. 7–10, 2008, Reno, NV

Adjoint-Based Adaptive Mesh Refinement for Complex Geometries

Marian Nemec*
ELORET Corp., Moffett Field, CA 94035

Michael J. Aftosmis
NASA Ames Research Center, Moffett Field, CA 94035

Mathias Wintzer
Stanford University, Stanford, CA 94305

This paper examines the robustness and efficiency of an adjoint-based mesh adaptation method for problems with complicated geometries. The method is used to drive cell refinement in an embedded-boundary Cartesian mesh approach for the solution of the three-dimensional Euler equations. Detailed studies of error distributions and the evolution of cell-wise error histograms with mesh refinement are used to formulate an adaptation strategy that minimizes the run-time of the flow simulation. The effectiveness of this methodology for controlling discretization errors in engineering functionals of non-smooth problems is demonstrated using several test cases in two and three dimensions. The test cases include a model problem for sonic-boom applications and parametric studies of launch-vehicle configurations over a wide range of flight conditions. The results show that the method is well-suited for the generation of aerodynamic databases of prescribed quality without user intervention.

I. Introduction

Estimation of discretization errors and the influence of these errors on the output quantities of flow simulations is an essential step in applying computational fluid dynamics to practical problems. Users endeavor to construct meshes that appropriately balance accuracy with computational cost to reliably predict forces, moments and other output functionals of interest. Such hand-crafted meshes encapsulate the knowledge and experience of experts for the simulation at hand, capturing the relevant flow features in sufficient detail and even avoiding weaknesses of the solver or mesh generator learned from past simulations. As the complexity of simulations increases, perhaps due to flow features that span many length scales, such as shocks, or the need to analyze problems with complicated geometries, the relative importance of features becomes difficult to judge. With uniform mesh refinement being too expensive, many adaptive approaches are found in the literature that seek to automate this task and alleviate the burden of expert-driven meshing.[1 5] Unfortunately, while most of these approaches work well for isolated analysis cases and simple geometries, it seems that solution-adaptive mesh refinement is rarely used in practice for tasks that involve parametric and optimization studies of complex geometries. With a few exceptions,[6] reliability and efficiency issues make the use of hand-crafted meshes the dominant practice across the industry.

A solution-adaptive approach typically consists of three components: 1) the formulation of cell-refinement parameters from localized error estimates, 2) a meshing algorithm that uses the refinement parameters to construct an improved mesh, and 3) a mesh-adaptation strategy that minimizes the cost of the simulation. In this paper, we carefully consider all three components to evaluate the usefulness of an approach we proposed in Ref. 7, for parametric studies of complex configurations. We explore if this approach is sufficiently robust

*Research Scientist, Advanced Supercomputing Division, MS T27B; Marian.Nemec@nasa.gov. Member AIAA.
Aerospace Engineer, Advanced Supercomputing Division, MS T27B; Michael.Aftosmis@nasa.gov. Senior Member AIAA.
Graduate Student, Department of Aeronautics and Astronautics; mwintzer@stanford.edu. Student Member AIAA.
Copyright © 2008 by the American Institute of Aeronautics and Astronautics, Inc. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

and efficient to be applied in practice, such that the accuracy of user selected outputs is ensured in every flow simulation without supervision.

The proposed approach uses adjoint-weighted residual error estimates to localize mesh refinement to those features of the computational domain that influence the accuracy of the selected outputs. Such error estimates have been initially derived in the context of finite element methods,[8] and later extended to finite volume methods by Giles and Pierce,[9] Barth,[10] and Venditti and Darmofal.[11] The accuracy of these error estimates has been successfully demonstrated in three-dimensional problems on structured and unstructured meshes, see for example Refs. 12 17. These studies verify and validate the adjoint-based approach in conjunction with various flow regimes and output functionals, including field pressure functionals for sonic-boom applications; however, they consider relatively simple geometries, for example clean wing, wing-body, and cone-cylinder configurations.

The routine use of adjoint-based adaptive mesh refinement in practice, where the analysis of several thousand flight conditions and vehicle configurations may be required, introduces several challenges. Among the key issues are the requirements for robust and fast re-meshing of complex geometry, and the stability of the adjoint solver. If a typical flow simulation requires roughly five to ten adaptation cycles to attain sufficient output accuracy, then the construction of databases for parametric studies with prescribed error bounds may easily invoke the mesh generator ten-thousand times. Moreover, while convergence failures of the flow and adjoint solvers can usually be detected and circumvented automatically (for example by choosing a stronger preconditioner), failures in mesh generation usually terminate the simulation. Unlike previous studies, we use an embedded-boundary Cartesian-mesh method to address this issue. Given a watertight surface discretization, these methods are known for their ability to efficiently handle surfaces of arbitrary shape.[18,19] Our mesh-refinement scheme relies on the work of Aftosmis and Berger,[5] who presented an incremental adaptation strategy for refinement of nested Cartesian meshes. We extend this strategy to include adjoint error estimates and propose several improvements to minimize the run-time of the simulation.

We demonstrate the effectiveness of the proposed approach on three test cases. The first test case considers a two-dimensional model problem for sonic-boom applications. We discuss several ideas that improve wave propagation through the computational mesh to the near-field, thereby reducing the number of cells in the mesh. The remaining examples consider launch vehicle configurations and we perform Mach number and angle of attack studies to evaluate the robustness and efficiency of the approach for realistic geometries. In these cases, the presence of fine geometric details and complicated flow features generally prevents deep convergence of the flow solver. We investigate the robustness of the adjoint solver and the utility of the adjoint error estimates when confronted with flow solutions that may be less than ideally converged. Some comparisons with experimental data are also provided.

We begin with a brief derivation of adjoint-weighted residual error estimates in the next section and review the salient features of our implementation from Ref. 7. The main contributions of this work are presented in the subsequent sections, where we discuss our adaptation strategy and results.

II. Background

The goal of the following derivation is to compute a reliable approximation of a functional $\mathcal{J}(Q)$, such as lift, that is a function of a flow solution $Q = [\rho \;\; \rho u \;\; \rho v \;\; \rho w \;\; \rho E]^{\mathrm{T}}$ satisfying the steady-state three-dimensional Euler equations of a perfect gas

$$\mathcal{F}(Q) = 0 \tag{1}$$

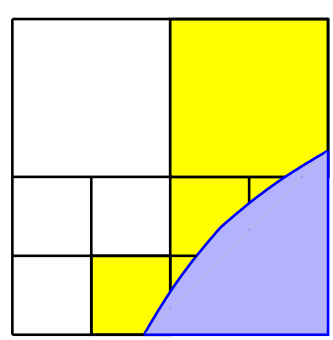


Figure 1. Multilevel Cartesian mesh in two-dimensions with a cut-cell boundary

Let $J(\mathbf{Q}_H)$ denote an approximation of the functional computed on an affordable Cartesian mesh with a characteristic cell size H, where $\mathbf{Q} = [\bar{Q}_1 \;\; \bar{Q}_2 \;\; \ldots \;\; \bar{Q}_N]^{\mathrm{T}}$ is the discrete solution vector of the cell-averaged values for all N cells of the mesh and J is the discrete operator used to evaluate the functional. The governing equations are discretized on a multilevel Cartesian mesh with embedded boundaries. The mesh consists of regular Cartesian hexahedra everywhere, except for a layer of body-intersecting cells, or *cut-cells*, adjacent to the boundaries as illustrated in Fig. 1. The spatial discretization of Eq. 1 uses a cell-centered, second-order accurate finite volume method with a weak imposition of boundary conditions,

resulting in a system of equations

$$\mathbf{R}(\mathbf{Q}_H) = 0 \tag{2}$$

The flux-vector splitting approach of van Leer[20] is used. The mesh is viewed as an unstructured collection of control volumes, which makes this approach well-suited for solution-adaptive mesh refinement. Steady-state flow solutions are obtained using a five-stage Runge Kutta scheme with local time stepping, multigrid, and a domain decomposition scheme for parallel computing; for more details see Aftosmis *et al.*[21,22] and Berger *et al.*[23]

To approximate the functional error $\mathcal{J}(Q) - J(\mathbf{Q}_H)$, we consider isotropic refinement of an initial Cartesian mesh to obtain a finer mesh with average cell size h containing approximately $8N$ cells (in three dimensions), and seek to compute the discrete error $J(\mathbf{Q}_h) - J(\mathbf{Q}_H)$ without solving the problem on the embedded, fine mesh. Our approach follows the work of Venditti and Darmofal,[11] where Taylor series expansions of the functional and residual equations are used on the embedded mesh about the coarse mesh solution. The result is an estimate of the functional on the embedded mesh, given by

$$J(\mathbf{Q}_h) \approx J(\mathbf{Q}_h^H) - \underbrace{(\ {}_h^H)^{\mathrm{T}}\,\mathbf{R}(\mathbf{Q}_h^H)}_{\text{Adjoint Correction}} - \underbrace{(\ {}_h - \ {}_h^H)^{\mathrm{T}}\,\mathbf{R}(\mathbf{Q}_h^H)}_{\text{Remaining Error}} \tag{3}$$

where $\mathbf{Q}_h^H$ and ${}_h^H$ denote a reconstruction of the flow and adjoint variables from the coarse mesh to the embedded mesh, and the adjoint variables satisfy the following linear system of equations

$$\left[\frac{\mathbf{R}(\mathbf{Q}_H)}{\mathbf{Q}_H}\right]^{\mathrm{T}} \ {}_H = \frac{J(\mathbf{Q}_H)}{\mathbf{Q}_H}^{\mathrm{T}} \tag{4}$$

Ref. 24 gives details on the implementation of the adjoint solver.

Referring to Eq. 3, the adjoint variables provide a correction term that improves the accuracy of the functional on the coarse mesh, and a remaining error term that is used to form an error-bound estimate and a localized refinement parameter. A difficulty with the remaining error term is that it depends on the solution of the adjoint equation on the embedded mesh, ${}_h$. We approximate ${}_h$ with an interpolated adjoint solution from the coarse mesh. A piecewise linear and constant reconstruction is used, so that a computable approximation to Eq. 3 becomes

$$J(\mathbf{Q}_h) \approx J(\mathbf{Q}_{\mathrm{L}}) - (\ {}_{\mathrm{L}})^{\mathrm{T}}\,\mathbf{R}(\mathbf{Q}_{\mathrm{L}}) - (\ {}_{\mathrm{L}} - \ {}_{\mathrm{C}})^{\mathrm{T}}\,\mathbf{R}(\mathbf{Q}_{\mathrm{L}}) \tag{5}$$

where the subscript $(\cdot)_{\mathrm{L}}$ denotes piecewise linear reconstruction and subscript $(\cdot)_{\mathrm{C}}$ denotes a piecewise constant solution value. Details of this approximation and its performance are presented in Ref. 7. The advantages of this approach are robustness, especially in regions where the flow and adjoint solutions exhibit steep gradients and discontinuities, and simplicity of implementation. The disadvantage is a reduction in accuracy of the adjoint correction and remaining error terms. The remaining error term in Eq. 5 is used to estimate a bound on the local error in each cell k of the coarse mesh

$$e_k = \left|(\ {}_{\mathrm{L}} - \ {}_{\mathrm{C}})^{\mathrm{T}}\,\mathbf{R}(Q_{\mathrm{L}})\right|_k \tag{6}$$

where the sum is performed over the children of each coarse cell. An estimate of the net functional error E is the sum of the cell-wise error contributions to the functional

$$E = \sum_{k=0}^{N} e_k \tag{7}$$

Given a user-specified global tolerance `TOL` for the functional of interest, the termination criterion for the simulation is satisfied when

$$E \leq \texttt{TOL} \tag{8}$$

To define a local refinement parameter on the coarse mesh, we specify a maximum allowable error level, t, for each cell by equidistributing the user-specified tolerance over the cells of the coarse mesh

$$t = \frac{\texttt{TOL}}{N} \tag{9}$$

The refinement parameter, r_k, is given by

$$r_k = \frac{e_k}{t} \tag{10}$$

which is the ratio of the cell-wise error, given by Eq. 6, to the maximum allowable error. A cell is flagged for refinement if

$$r_k > \lambda \tag{11}$$

where $\lambda \geq 1$ is a global *refinement threshold*. The refinement parameter drives an incremental adaptation strategy. Starting from a coarse initial mesh, cells are flagged for refinement as indicated by Eq. 11. The solution is computed on the refined mesh and the adaptation cycles continue until the termination criterion is satisfied. The refinement region is typically enlarged by one coarse cell to ensure that the interface boundary of the refined region of the mesh is in a region of low error.

We use two heuristics to fine-tune the adaptation procedure. The first allows control over the maximum refinement level in the new mesh. Usually one level of refinement is allowed at each adaptation cycle, but in some cases, especially for field functionals, we find that it is beneficial to perform one or two adaptation cycles with the maximum refinement level held fixed. This allows the propagation of mesh-interface boundaries to regions of low error while maintaining relatively low cell counts. The second heuristic allows adjustment of the threshold λ in Eq. 11 on early adaptation cycles. We use the expected convergence rate of the functional and statistical analysis of error distribution on the coarse mesh to determine a sequence of thresholds that reduce the run-time of the simulation and improve accuracy. The next section describes details of this threshold adjustment strategy.

III. Threshold Selection and Adaptation Strategy

A. Error histograms

In our earlier work,[5] we introduced the use of $\log_2$ histograms for insight into distributions of cell-wise error with mesh refinement. Figure 2 reviews the main ideas. The left frame of the sketch shows an example error distribution for some hypothetical discrete solution. In the sketch, the vertical axis is the number of cells in each bin of the histogram. The x-axis is typically a refinement parameter that represents the cell-wise error in the solution. In this work, it is given by Eq. 10, which is the scaled cell-wise contribution of error in the output functional. High error cells lie to the right and low-error cells are on the left. The goal of adaptation is *equidistribution*. In the present context, this implies that we wish to adjust the mesh such that all cells contribute equally to the functional error. Achieving this goal would put all the cells into the same bin of the error histogram, thus transforming it into a delta function with height equal to the number of cells in the mesh.

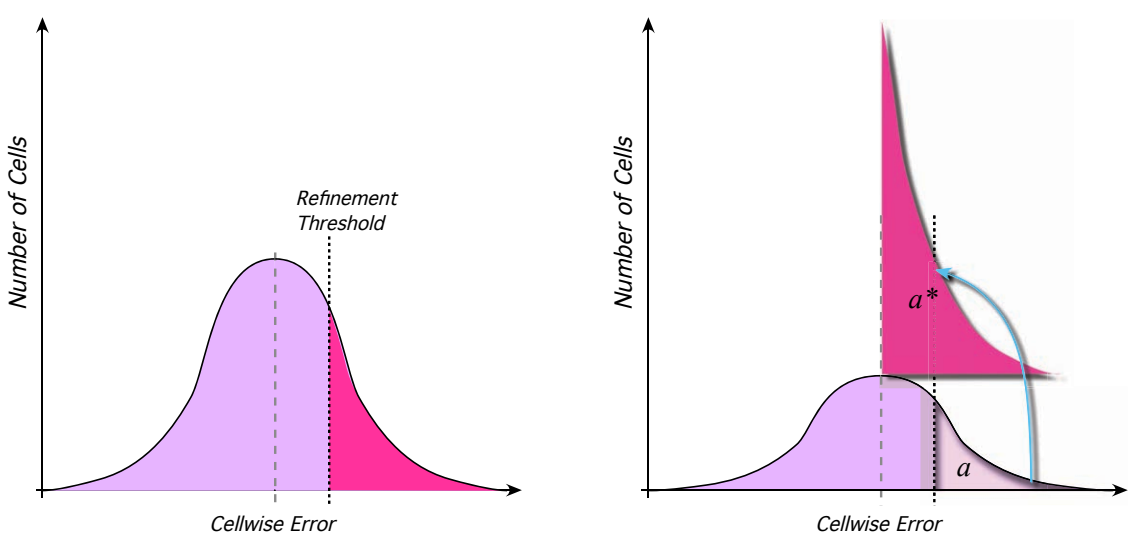


Figure 2. Histogram of cell-wise error in computational domain (Eq. 10) and error migration with adaptive refinement. h-refinement moves the cells in region a to the left according to the accuracy of the scheme. If a contains N_a cells, then a^* contains mN_a cells, where m is the parent-to-child ratio of h-refinement

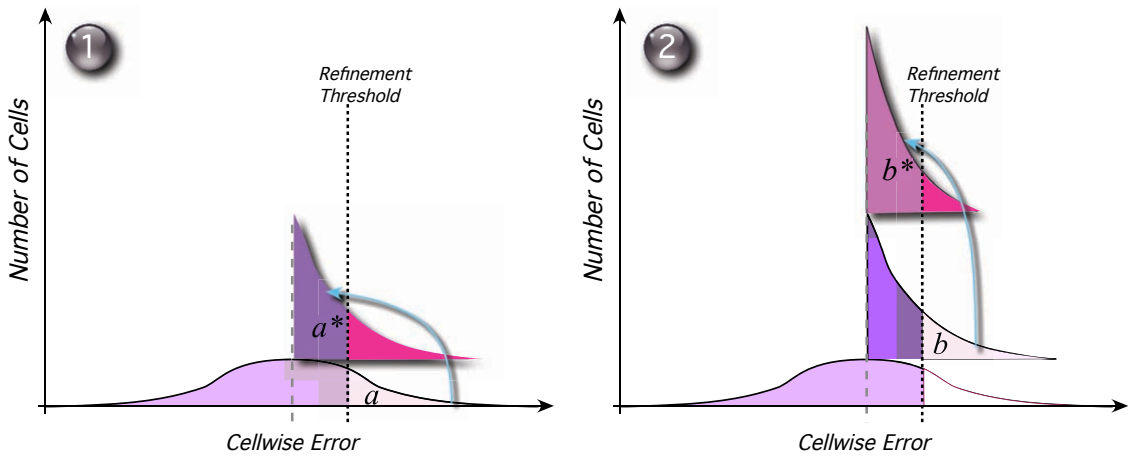


Figure 3. Sketch of $\log_2$-error histograms for 2-stage adaptation sequence using a constant threshold strategy

The sketch at the right of Fig. 2 illustrates the basic mechanism of adaptive refinement. The refinement seeks to replace high-error cells in the domain with more cells that contain less error. Near grid convergence, each 2:1 cell-refinement of a pth-order scheme reduces the local truncation error by a factor of 2^p. Accordingly, the children of an h-refined Cartesian cell are translated an absolute distance of p units to the left on a base-2 histogram of cell-wise error.

B. Adaptation strategies

1. Constant Threshold

The simplest refinement strategy is to refine any cell that exceeds the cell-wise error tolerance as given by Eq. 9. The refinement threshold, λ in Eq. 11, is set at unity and held constant for all adaptation cycles. This threshold is located at 0 on our $\log_2$ histogram, and we seek to build a delta function at zero. Figure 3 sketches the error histograms following this refinement strategy through two cycles. In the first refinement, cells to the right of the refinement threshold (set "a") migrate to a^* after subdivision according to the rules outlined in Fig. 2. After this first refinement, however, we note that a^* still has a tail which exceeds the threshold (the pink portion of a^* in Fig. 3). The second refinement pass addresses these cells. With b denoting the set of offending cells in a^*, the second refinement pass attacks this tail, amplifying the cell count in b and moving the cells to the left on the histogram, $b \rightarrow b^*$. As before, with a second-order scheme, 2:1 cell refinement moves the cells 2 units to the left on a $\log_2$ histogram, and the cell count increases following the parent-to-child ratio of the h-refinement scheme. After the second subdivision, b^* still has a tail to the right of the refinement threshold and the refinement process continues until Eq. 8 is satisfied.

With this perspective on the error migration over two adaptation cycles, we make some general observations about the Constant Threshold strategy. To begin with, note that since the threshold is fixed, all refinements pile cells at the same abscissa on the evolving histogram. As a result, a large peak gets built early in the refinement process. This is good from the standpoint of driving toward a delta function, but has the drawback that it results in large cell counts relatively early on. Put differently, the Constant Threshold strategy is slow to address the high-error "tail" of the histogram. The worst offending cells in the histogram (those on the extreme right) are the *last* to get addressed, and it may take many adaptation cycles before the most extreme errors in the domain are pulled sufficiently to the left. Consider a limiting case with only a handfull of high-error cells to the right of the refinement threshold. We end up recomputing the flow and adjoint problems on essentially the same mesh, perhaps several times, before finally cleaning up the error in the worst few cells.

While this presentation has focused on incremental h-refinement, our discussion has implications for global re-meshing as well. Generating many cells early in the adaptive process means carrying those cells through until the the worst offending regions are addressed in the later adaptation cycles. Moreover, if mesh generation is expensive, then this cost is compounded if many fine meshes are needed with cell counts nearly that of the finest mesh. With a Constant Threshold strategy, there is a good chance that mesh growth factors will decrease substantially in later refinement passes.

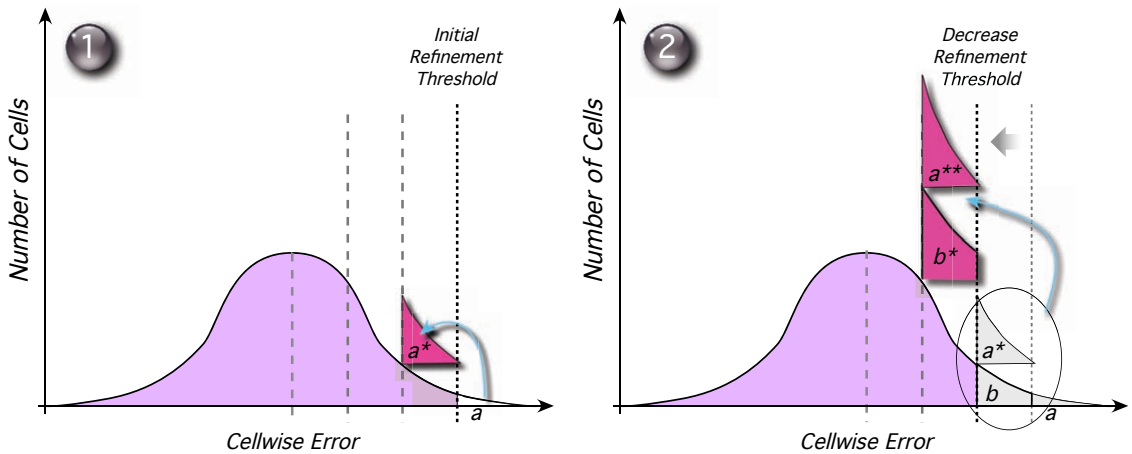


Figure 4. Sketch of $\log_2$ error histograms for 2-stage adaptation sequence using a decreasing threshold strategy

2. Decreasing Threshold

The observation that a constant threshold strategy may require multiple solves on essentially the most expensive mesh suggests an alternative approach. Why not attack the worst offending cells as early as possible in the process? By starting with a high initial refinement threshold, i.e. large λ in Eq. 11, it may be possible to first address the worst errors, and then "roll-up" the histogram's error-tail by decreasing the threshold as the adaptation cycles continue. This approach is also attractive because it prevents excessive refinement on coarse meshes when the accuracy of the flow and adjoint solutions is poor. Similar ideas have been proposed by Jones *et al.*[13] and Fidkowski[16] in the context of global re-meshing, who define parameters that adjust the target error level as the adaptation progresses.

Figure 4 presents a 2-stage adaptive sequence outlining the Decreasing Threshold strategy. As in Fig. 3, the first frame shows the initial refinement with cells in region a again migrating to a^*. In this case, however, we start with an initial threshold several units higher along the histogram's abscissa. Because the initial mesh is coarse with only a rough error-map, cells containing only the highest errors are selected for refinement. In contrast to our earlier example, when cells in region a now migrate to a^* very little tail remains to the right of the refinement threshold. Moreover, we reiterate that region a contains only a handful of the worst cells in the domain, and therefore the mesh growth starts out slowly.

As illustrated in the second frame of Fig. 4, subsequent adaptation cycles are accompanied by a reduction in the refinement threshold. This refines cells originally in a a second time ($a \rightarrow a^* \rightarrow a^{**}$), while also refining cells from b to b^*. The histogram's peak gets built to the right of the final target threshold and shifts to the left with each threshold decrease. Ideally, the final adaptation cycle should flag the most cells of all the cycles, and shift and amplify the peak to satisfy Eq. 8 and terminate the simulation. With this strategy, mesh growth factors starts out low and continually increase, requiring only one solution of the flow and adjoint problems on an expensive mesh.

3. An Illustrative Example

We examine the two refinement strategies using the familiar ONERA M6 wing transonic test case[25] at $M_\infty = 0.84$ and $\alpha = 3.06°$. Figure 5 presents an overview of this case showing the final solution and adapted mesh. Adaptation was driven by specifying a 0.001 tolerance on the drag coefficient C_D. The Constant Threshold strategy maintained a scaled error threshold of unity, $\lambda = 1$ in Eq. 11, throughout the process, while the Decreasing Threshold strategy started at $\lambda = 32$ and decreased by a factor of two with each subsequent adaptation.

Figure 5 shows convergence histories of the functional and error estimate on the right side of the figure, respectively. At the upper right, we observe that both approaches obtain the same final value of C_D. Convergence is plotted against cell growth on a logarithmic scale. Since it is adapting more cells earlier in the process, the Constant Threshold curve is continually to the right of the Decreasing Threshold data. The initial cell growth factor for the Constant Threshold strategy is 5.9:1 vs. only 2.6:1 for the Decreasing Threshold strategy. In both cases, refinement systematically reduces the total error in the solution, but by focusing early effort on only the most severe errors, the Decreasing Threshold strategy is more aggressive

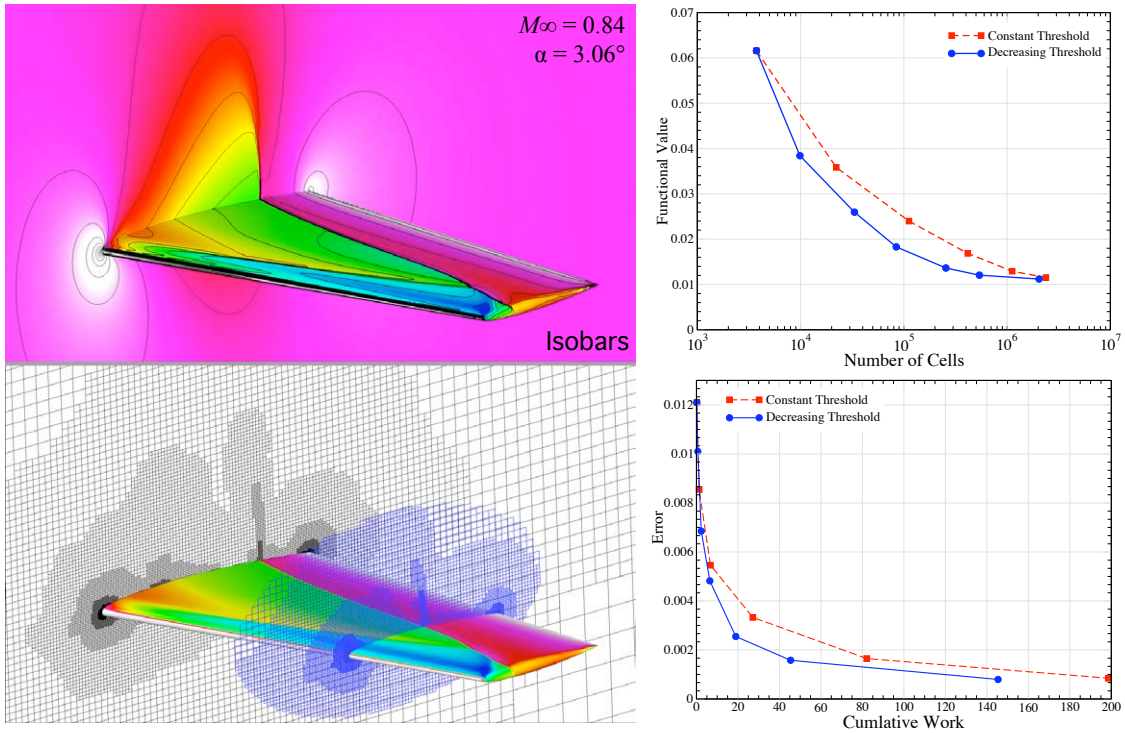


Figure 5. Transonic flow over ONERA M6 wing at $M_\infty = 0.84$ and $\alpha = 3.84°$ contrasting Constant Threshold and Decreasing Threshold refinement strategies. *Upper Left*: Isobars in discrete solution. *Lower Left*: Final adapted mesh (Decreasing Threshold approach) with 2.04 M cells. *Upper Right*: C_D convergence *Lower Right*: Computational work normalized by one fine-grid multigrid W-cycle

earlier in the process. This results in lower error levels on smaller meshes, making the process both quicker and more efficient. With the Decreasing Threshold, the last mesh in the sequence nearly quadrupled the total cell count (from 540k to 2.04M cells) and about 79% of the cells in the final mesh were children created at the last adaptation cycle. These statistics contrast nicely with statistics from the Constant Threshold strategy. In that case, the final adaptation cycle attacked the tail of the histogram, marking only 8.21% of the control volumes for refinement so that only 60% of the cells in the final mesh were newly created at the final adaptation.

The final meshes and cell counts resulting from both strategies are similar (not shown), however, the Decreasing Threshold mesh is somewhat more efficient in that it achieves comparable error levels with about 10-15% fewer cells. Only slight differences are observed in the placement and shape of refinement boundaries. The *most* significant difference between the strategies is the cost to achieve the results. The lower right frame of Fig. 5 focuses on this cost difference by examining the accumulation of work with mesh refinement. The x-axis labeled "Cumulative Work" is accumulated CPU time normalized by the time required to perform one Multigrid W-cycle (flow and adjoint solution) on the final mesh from the "Decreasing Threshold" case (2.04M cells). Accordingly, the Decreasing Threshold strategy required the equivalent of 146 fine-mesh multigrid cycles to achieve the error tolerance. The Constant Threshold strategy required about 36% more work with a slightly higher final error (although still within the prescribed tolerance).

Figure 6 contains a lay-up of $\log_2$-scaled histograms of the refinement parameter, Eq. 10, extracted from four meshes in the adaptive sequence using the Decreasing Threshold approach. The left two histograms illustrate the first adaptation cycle (Mesh 0 $\rightarrow$ Mesh 1), while the right two histograms illustrate the last (Mesh 5 $\rightarrow$ Mesh 6). In the first adaptation cycle, cells to the right of $\lambda = 32$ are tagged for refinement. In the Mesh 0 histogram of Fig. 6, these cells are located to the right of $x = \log_2(32) = 5$. Its interesting to note that after converging on Mesh 1, new cells with very high error have appeared to the right of the histogram. This is typical early in the adaptive process as the geometry is poorly resolved on these extremely coarse meshes, and new geometry is "revealed" with each refinement pass.

The right two histograms in Fig. 6 illustrate the last refinement pass. Note in comparing these that the vertical axis scale changes by more than a factor of four between the two histograms. The cell count in this final adaptation cycle has increased by a factor of 3.78, while the standard deviation has dropped from just

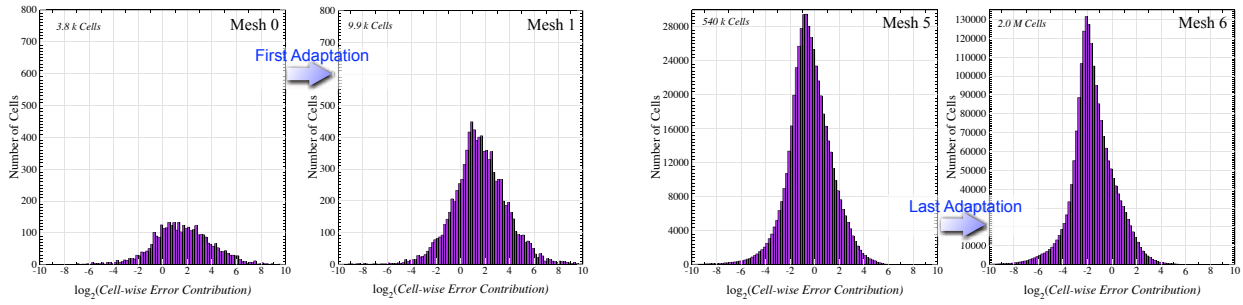


Figure 6. Histograms of normalized cell-wise error, Eq. 10, showing the first (left two plots) and last (right two plots) adaptation cycles for the transonic wing example

over 3 to about 2.7. Thus the peak is increasing not only due to the increased cell count, but also due to a sharpening of its profile. This last adaptation was carried out with a refinement threshold $\lambda = 1$ and we note that the peak shifts to the left following our earlier discussion. The peak of the distribution is located at -2, which is due to the discrete subdivision of Cartesian cells, and the sum of the cell-wise errors is less than the user-specified tolerance. Lastly, note the steepness of the distribution to the left of the peak. This indicates that the potential benefits of cell coarsening are minor for this example.

IV. Results

A. Near-Field Pressure Functional for Diamond Airfoil

The first test case considers a two-dimensional model problem to investigate the usefulness of the proposed approach for sonic-boom applications involving field functionals. The goal is to minimize the number of cells required to produce sharply resolved pressure signals several body lengths away from the object, which can then be propagated to the ground using a sonic-boom prediction code.

The case setup is similar to one presented by Fidkowski and Darmofal.[26] A unit-chord-length diamond airfoil with wedge half-angle of 4.995° is placed at 0° angle of attack in Mach 1.5 freestream flow. The far-field boundaries are located 200 units away from the airfoil. The near-field pressure signal is sampled along a line segment (the "sensor") that is parallel to the freestream flow, 9 units in length, and placed 20 units above and 18 units behind the airfoil. The mesh refinement seeks to reduce the error in the functional

$$J = \int_0^L \frac{\Delta p}{p_\infty} ds \tag{12}$$

evaluated along a sensor of length L, with $\Delta p = p - p_\infty$ and $p_\infty = 1/\gamma$.

The starting mesh, consisting of 1,543 cells, is shown in Fig. 7. The rotation of the mesh aligns the cells with the freestream Mach angle, in this case 41.81°, which significantly improves the propagation of the leading-edge shock wave from the airfoil to the sensor. The solid line in the upper right corner indicates the location of the sensor where the pressure signal is sampled. Note that the sensor is contained in just three cells. The signal error is evaluated by comparison with a solution computed on a uniformly refined and rotated mesh containing 32 million cells. This solution is labeled as "Reference" in subsequent plots.

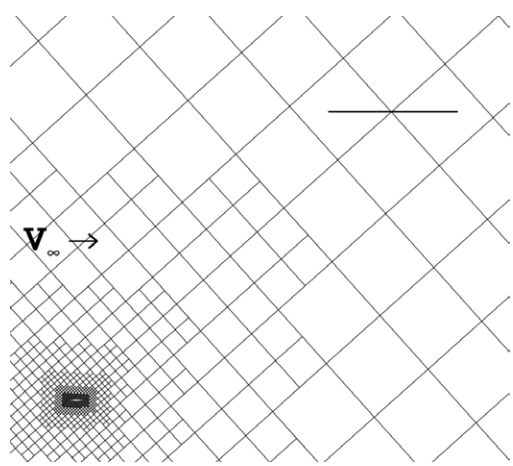
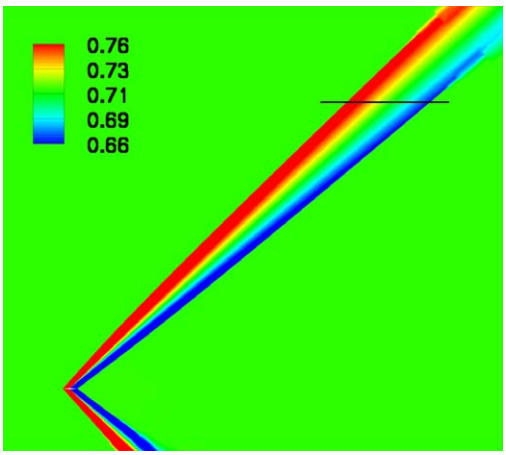
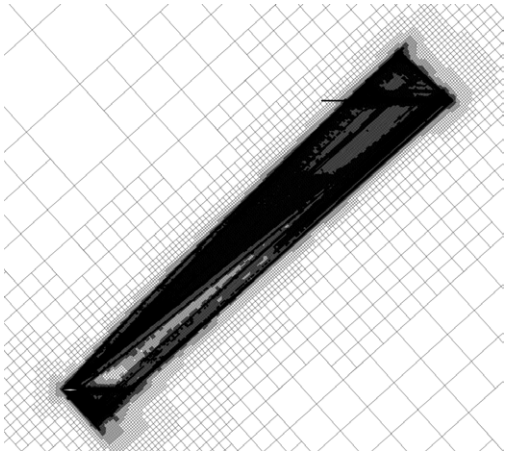


Figure 7. Starting grid for the diamond airfoil; 1,543 cells

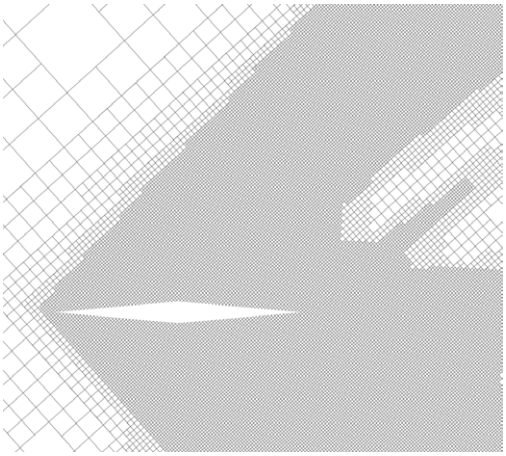
We specify an error tolerance of 0.0001 for the functional in Eq. 12. Nine adaptation cycles are performed, with the maximum cell refinement level being increased only every third adaptation cycle. The final adapted mesh, presented in Fig. 8, contains 628,291 cells in the domain and 1,085 sample



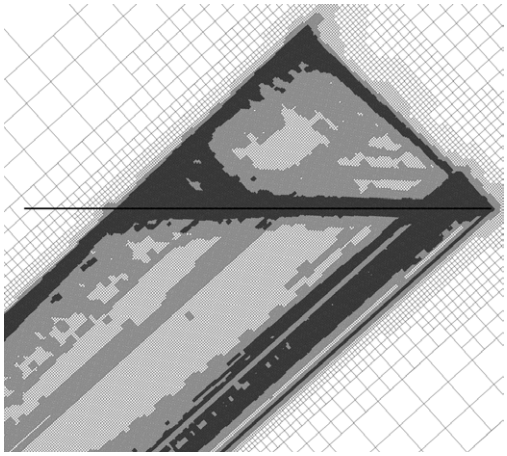
(a) Pressure contours



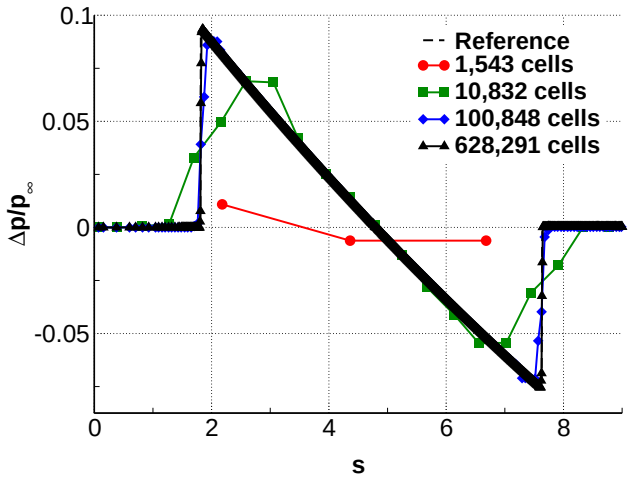
(b) Adapted region of final mesh; 628,291 cells



(c) Detail of airfoil region



(d) Sensor region; 1,085 points along sensor



(e) Pressure signals measured after each refinement

Figure 8. Final solution based on functional in Eq. 12 (9 adaptation cycles)

points along the pressure sensor. Contours of pressure in the adaptation region are shown in Fig. 8(a). The associated mesh is shown in Fig. 8(b), while Fig. 8(c) and Fig. 8(d) show detail views of the airfoil and sensor regions, respectively. The evolution of the pressure signature at the sensor line towards an N-wave shape is shown in Fig. 8(e).

Viewed from a distance, the mesh refinement structure is as expected; the most attention is paid to shock and expansion regions and along the sensor itself. It is interesting to note that regions where discretization errors have a strong influence on the functional also include the lower surface of the airfoil and the wake region. While some refinement of these regions is required to capture the sharp pressure rise of the N-wave, it is clear from Figs. 8(d) and (e) that significant refinement also occurs in the constant flow region, behind the N-wave ($s = 7.7$ to 9 in Fig. 8(e)). The total number of cells appears to be a function of the user-specified sensor length, which is undesirable.

Examination of Eq. 12 suggests that while the value of $\Delta p/p_\infty$ is essentially zero past $s = 7.7$, the derivative of the functional in each cell

$$\frac{J}{Q} = \frac{1}{p_\infty}\frac{p}{Q} \tag{13}$$

is non-zero in general. This term forms the right hand side of Eq. 4 and ultimately leads to some refinement along the sensor due to contributions of adjoint interpolation values in Eq. 6 (residual errors are relatively small past the N-wave). To remedy this situation, we propose a change in the integrand of the target functional to a quadratic form

$$J = \int_0^L \left(\frac{\Delta p}{p_\infty}\right)^2 ds \tag{14}$$

This introduces a $\Delta p/p_\infty$ term in the cell-wise derivative

$$\frac{J}{Q} = \frac{2}{p_\infty}\left(\frac{\Delta p}{p_\infty}\right)\frac{p}{Q} \tag{15}$$

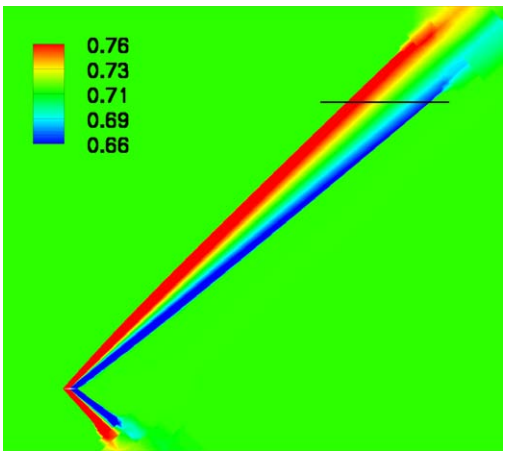
the effect of which is to emphasize regions of sharp pressure rise in the N-wave independent of line length.

The final mesh adapted using this squared functional is shown in Fig. 9. Presented in a similar format to the linear functional result, this mesh has 426 sample points along the sensor and only 137,430 cells in the domain a factor of 5 reduction relative to the linear case. The heavy cell refinement aft of the trailing edge shock jump has been suppressed, and the detail views (Figs. 9(c) and (d)) reveal far more selective refinement of the mesh. Compression, expansion, and cross-coupled waves are clearly visible near the airfoil, and the number of sample points in the expansion region of the pressure signal is reduced. The complete lack of refinement of the wake region for this case, despite identical starting conditions, suggests its influence on the N-wave is secondary. Figure 10 compares the error convergence rate of the two approaches with respect to the reference solution. The convergence rates are dominated by the shocks and are therefore only slightly better than first-order. Nevertheless, the plot shows that the reduction in cell counts for the squared functional pays off over the entire history of the adaptation.

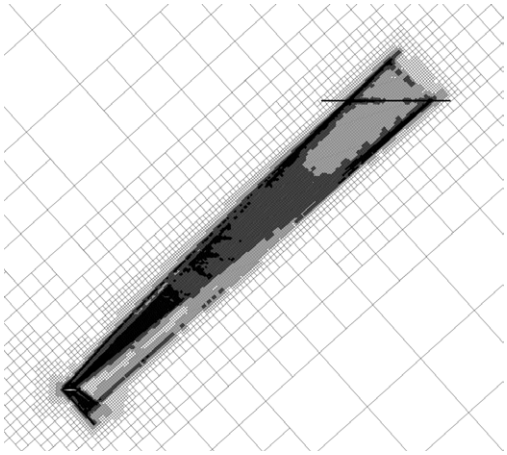
In these propagation problems, it is possible to further capitalize upon the Mach-angle alignment by stretching the mesh cells along the dominant wave direction. Using the squared functional, three adaptation cycles were performed on a mesh with cell aspect ratio (AR) of 4. This time, the adapted mesh, shown in Fig. 11(a), contains only 11,929 cells. With unit AR cells and again applying the squared functional, six adaptation cycles were required to produce a mesh of comparable signal error. As noted in Fig. 11(b) where the N-waves are compared, the unit AR mesh contains 33,565 cells a factor of 3 increase over the stretched case. To achieve an error level comparable to these two cases using the linear functional on a unit AR mesh would require 100,848 cells. The 11,929 cell result obtained with the stretched mesh thus represents an order of magnitude reduction in cell count, indicating this to be a promising approach for problems in three dimensions.

B. Aerodynamic Analysis of a Launch Abort Vehicle

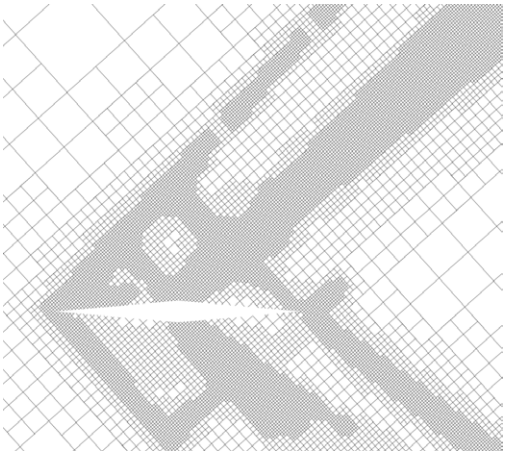
The second example tests the proposed approach on a complex geometry problem. We change our focus from airfoils to a simulation of a Launch Abort Vehicle (LAV) wind-tunnel model. The vehicle geometry is shown in Fig. 12, where the inset frame details the geometric features near the abort motor nozzles. The model also includes plugs for steering jets near the nose and a wind-tunnel sting that extends several body lengths behind the vehicle. The model is not symmetric, there is a small raceway running the length of the



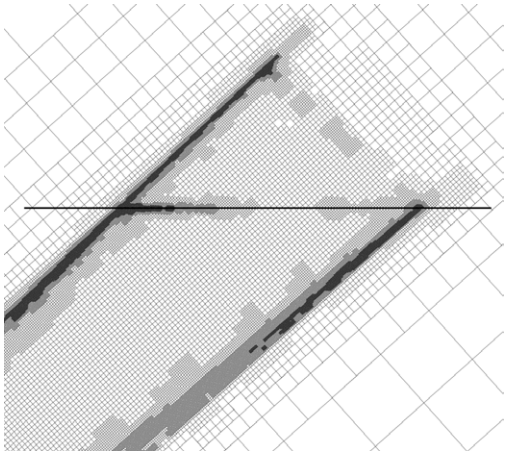
(a) Pressure contours



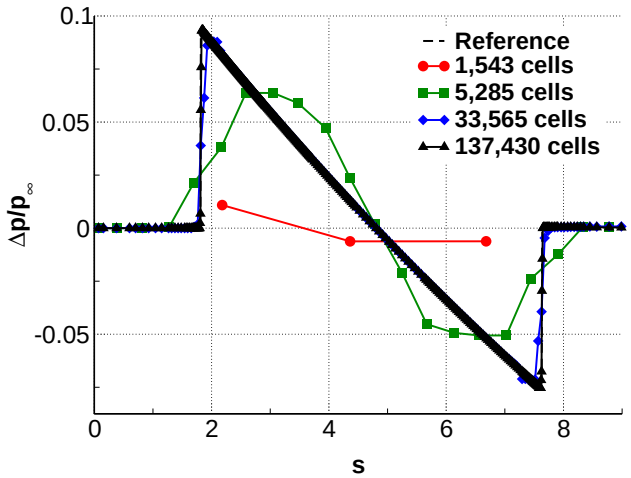
(b) Adapted region of final mesh; 137,430 cells



(c) Detail of airfoil region



(d) Sensor region; 426 points along sensor



(e) Pressure signals measured after each refinement

Figure 9. Final solution based on the squared functional, Eq. 14 (9 adaptation cycles)

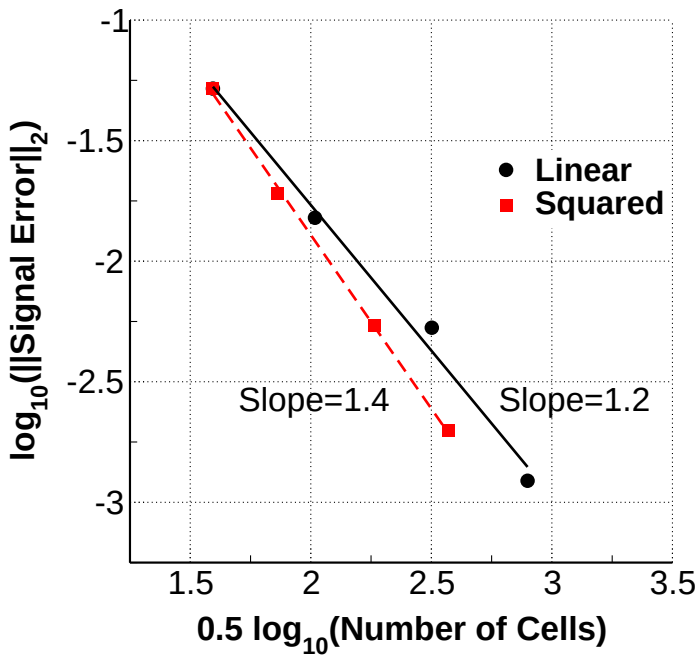
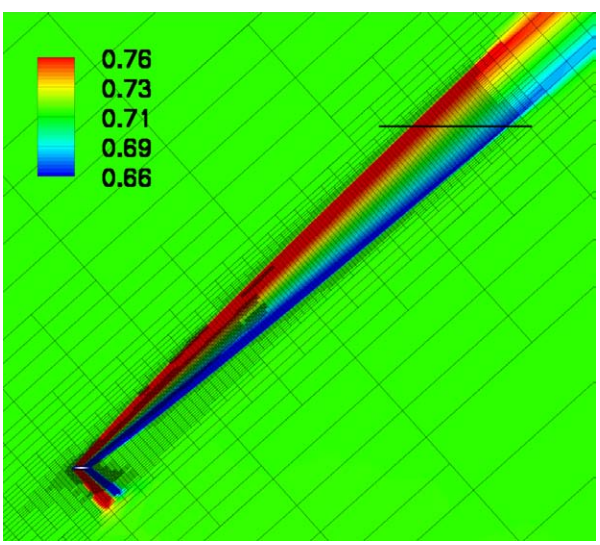
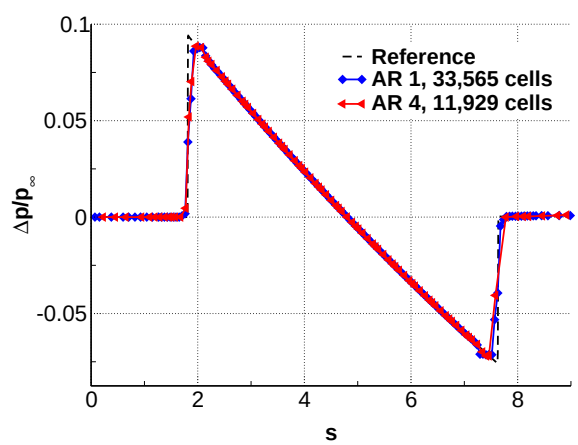


Figure 10. Signal error convergence for adaptive mesh refinement using linear and squared functionals



(a) $AR = 4$, 11,929 cells, 96 points along sensor



(b) Comparison of pressure signals

Figure 11. Comparison of signals on isotropic ($AR = 1$) and stretched ($AR = 4$) meshes with similar signal error

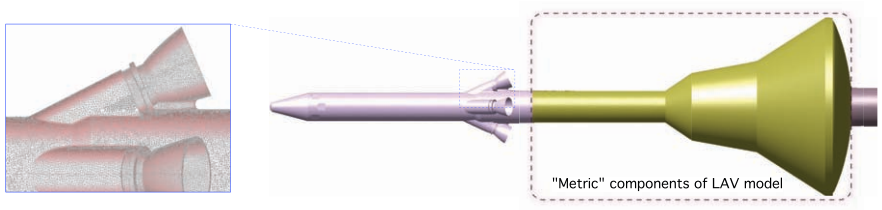


Figure 12. Launch Abort Vehicle (LAV) geometry with wind-tunnel sting

tower (not visible in Fig. 12). Aerodynamic forces and moments are evaluated on the "Metric" portion of the model, as shown in Fig. 12, which is similar to earlier wind-tunnel tests for this model. We evaluate the proposed approach for three Mach numbers (0.5, 1.1, 1.3), and a range of angles of attack from $-25°$ to $6°$.

The flow around this vehicle is characterized by bluff-body aerodynamics. An example flow-field is shown in Fig. 13 for freestream conditions $M_\infty = 1.1$ and $\alpha = -25°$. There is a sharp lip near the shoulder of the heat-shield that initiates a significant wake behind the vehicle. In addition, small regions of separated flow are observed downstream of the abort motor nozzles. While such geometric complexity and "unsteady" flow features are a departure from the smooth and steady flow assumptions made in the present adjoint error analysis, it is important to evaluate the usefulness of the proposed approach on these cases. The size of aerodynamic databases needed for the development of such a vehicle creates an engineering requirement to do the best job possible with inexpensive solvers, such as this steady and inviscid analysis. Efficient use of inexpensive simulations reduces the number of cases that require unsteady and/or viscous analysis. The run-time of a typical adjoint-based mesh refinement case presented in this section is roughly 30 minutes on 16 CPUs, while unsteady and viscous simulations may require hours of wall-clock time. Furthermore, the adjoint error estimates obtained in these inexpensive simulations may provide guidance for mesh refinement in the higher-fidelity solvers.

We choose a functional for the adaptation that is a linear combination of normal and axial forces evaluated on the Metric portion of the model

$$J = C_N + 0.2C_A \tag{16}$$

We experimented with several functional formulations and found that Eq. 16 results in good mesh convergence of both forces and moments. The error tolerance for this functional is set to 0.05 in all cases considered. We use the decreasing threshold adaptation strategy outlined in Sec. III, where we start with $\lambda = 32$ in Eq. 11 and reduce it by a factor of two on each subsequent cycle. The surface triangulation is created directly from a CAD model of the vehicle and contains roughly 380,000 triangles. This triangulation is fixed in all mesh refinement studies.

Before presenting results for the Mach number and angle of attack sweeps, we first examine the cell-wise error contributions to the output functional on a sample case. We choose the case shown in Fig. 13 at $M_\infty = 1.1$ and $\alpha = -25°$, and construct a functional error-map using a uniform mesh with six million cells. The left-side of Fig. 14 shows the adjoint solution for the density variable, which represents the influence of point sources of mass on the output functional. Note the large values near the heat-shield shoulder, the continuity of the adjoint solution across shocks, and the upwind bias due to the supersonic nature of the flow. The influence of the wake is relatively small for this case. We use the adjoint solution in conjunction with the flow residual errors to form the refinement parameter, Eq. 10, which is shown on the right side of Fig. 14. Regions of the field colored in grey-scale mark cells that do not exceed the refinement threshold. The features in the grey-scale region highlight the zone of dependence for the output functional, which is stretched in the cross-flow direction due to the mild supersonic conditions. The black contour line marks the boundary of the refinement region, with the colored region denoting cells that exceed the refinement threshold, and therefore require refinement. Notice that the error scale is logarithmic,

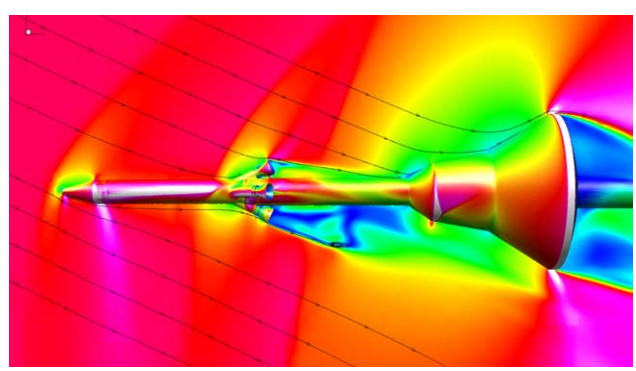


Figure 13. Mach number contours around LAV ($M_\infty = 1.1$, $\alpha = -25°$)

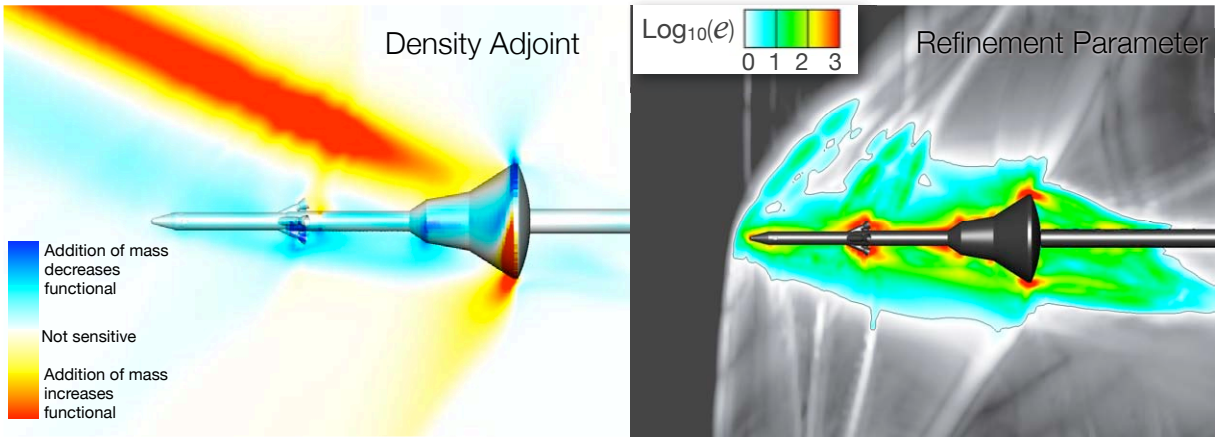


Figure 14. Adjoint solution contours for the continuity equation (left) and refinement parameter given by Eq. 10 (right), $M = 1.1$, $\alpha = -25°$

and red regions contribute almost three orders of magnitude more error than cells near the refinement boundary. Consequently, cells in the red regions of the error map require roughly five additional levels of refinement for the present second-order scheme. Regions of largest errors are at the shoulder of the heat-shield and near the abort motors. Only a moderate level of refinement is required near shocks and in the wake.

Next, we present an example adaptation case for the same flow conditions considered above. We start with a coarse initial mesh that contains roughly 3,700 cells, which is shown in the left inset frame on Fig. 15, and perform eight adaptation cycles. The final mesh contains almost two million cells. The refinement regions are in close agreement with the refinement parameter computed on the uniform mesh in Fig. 14, with the finest cells near the shoulder of the heat-shield and the abort motors. Figure 16 shows convergence of the functional and its error-bound estimate for this case. This figure illustrates a typical example of a mesh refinement study. Over the first five adaptation cycles the error estimate rapidly decreases accompanied by a steadily improving value of the functional. Note that the corrected functional value accurately predicts

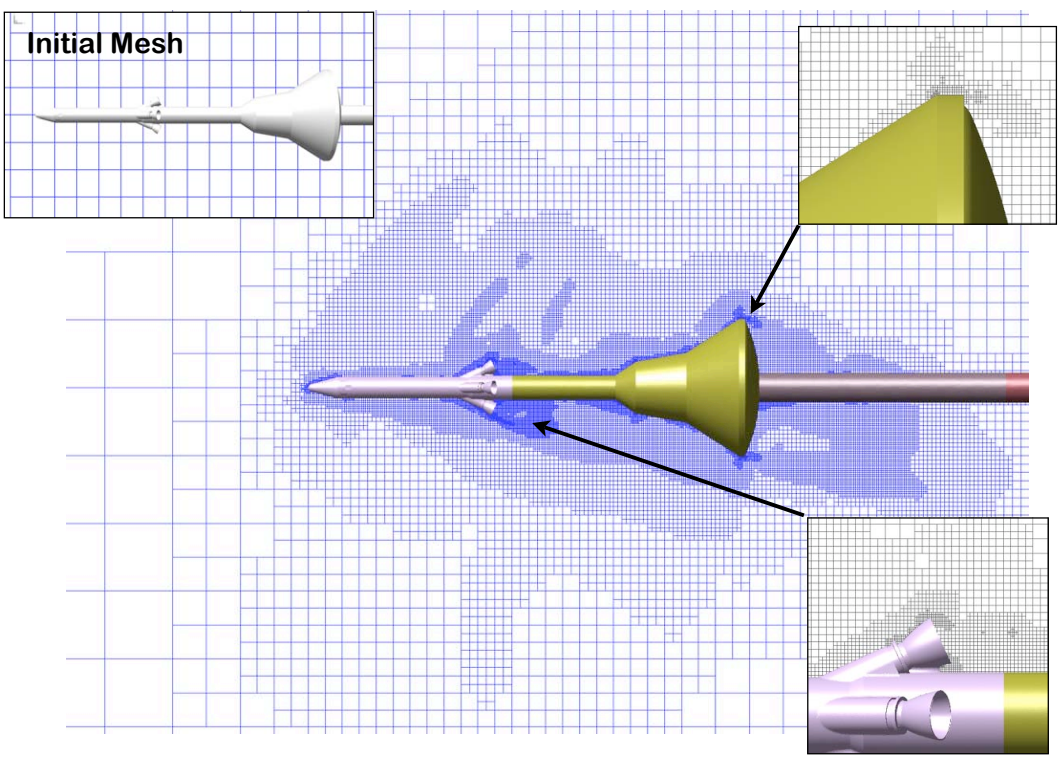


Figure 15. Near-body view of initial and final meshes on symmetry plane, $M_\infty = 1.1$, $\alpha = -25°$

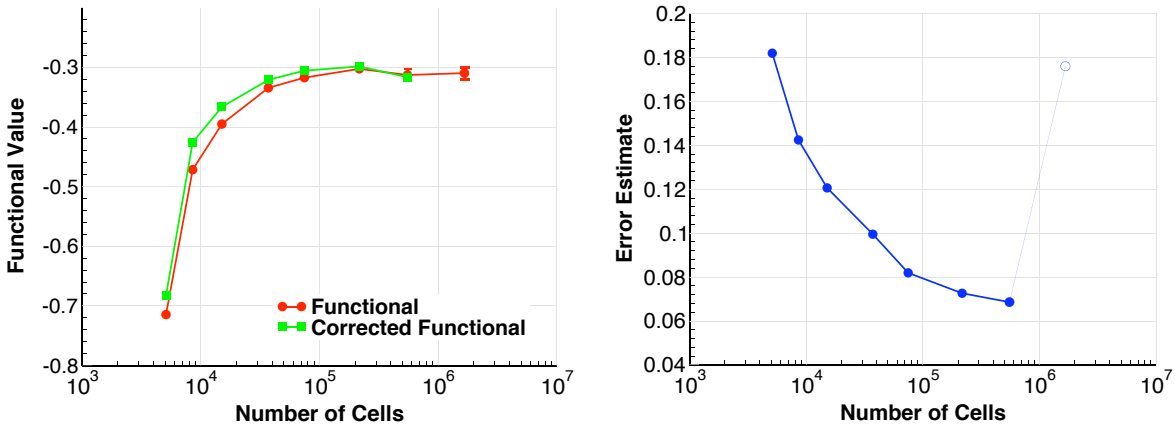


Figure 16. Convergence of functional (left) and error bound estimate (right), $M_\infty = 1.1$, $\alpha = -25°$

the functional on the next mesh of the adaptation. The functional appears to be approaching an asymptotic value by the fifth adaptation cycle, despite the complex geometry and flow features. On the final two meshes, small "unsteadiness", i.e. noise due to incomplete flow convergence, appears in the functional. We crudely indicate the peak-to-peak variation in the functional with error bars in the left plot of Fig. 16, and we also observe a reduction in the convergence rate of the error-bound estimate, see right plot of Fig. 16. On the final adapted mesh, shown in Fig. 15, the unsteadiness increases, and although we obtain a partially converged adjoint solution, the error-bound estimate increases significantly. The adaptation stops just short of reaching the target error tolerance of 0.05.

Similar results are presented in Fig. 17, where we show convergence of the force and moment coefficients with mesh refinement, but now plotted with respect to the number of multigrid cycles of the flow solver. The flow solution on each new mesh is warm-started from the solution on the previous mesh. The figure shows that the iterative convergence of forces and moments is excellent, with some fluctuations on the final mesh. We performed an unsteady simulation on the final mesh and the time-averaged coefficients are in good agreement with the steady-state results, see right side of Fig. 17. This behavior was verified for many cases of the parametric study. Our current interpretation of these results is that a meaningful estimate of the steady-state functional value can be reliably obtained on meshes with accurate adjoint corrections and decreasing error bound. In Figs. 16 and 17, the fluctuations in the flow solution were small, and did not have a large projection into the functional. We note that this is not always the observed behavior. In some of the subsonic cases the unsteadiness occurred much earlier in the mesh refinement, and this may provide guidance for which cases there is no alternative to unsteady simulations.

Snapshots of the final meshes and the corresponding flow solutions are shown in Figs. 18 and 19, respectively. The matrix of runs consisted of 30 flow conditions, which are indicated along the left and top borders of the figures. All computations were started from the same initial mesh, shown in the left inset of Fig. 15, and the final meshes contain between 1.2 to 2.2 million cells. Most simulations ran for eight adaptation cycles. We observe reasonable symmetry in the flow solutions. Convergence was worst for the subsonic cases at small angles of attack, and best for the $M = 1.3$ cases. Comparison with experiment on this model is shown in Fig. 20, where we plot normal and axial forces, as well as the pitching moment. We obtain reasonable agreement with the experimental values, in particular for normal force and pitching moment, despite the limitations of the present steady and inviscid analysis.

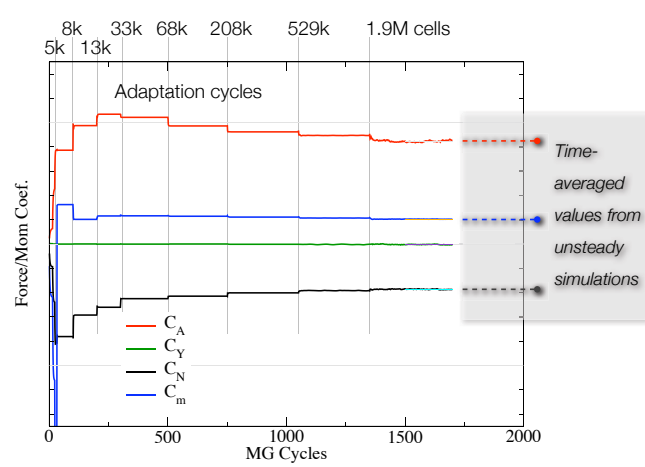


Figure 17. Convergence of force and moment coefficients, $M = 1.3$, $\alpha = -25°$

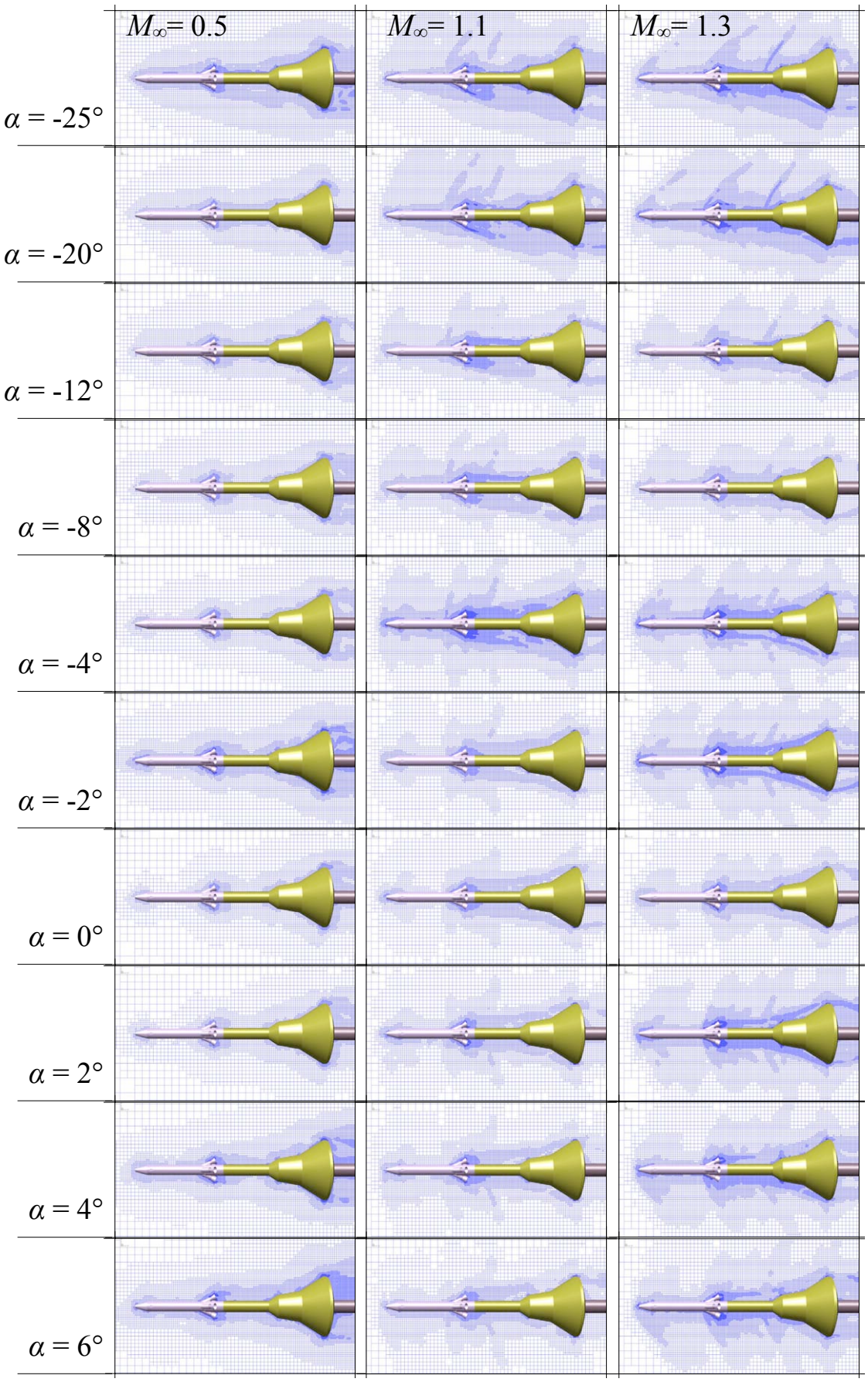


Figure 18. Final adapted meshes for parametric study of the LAV geometry

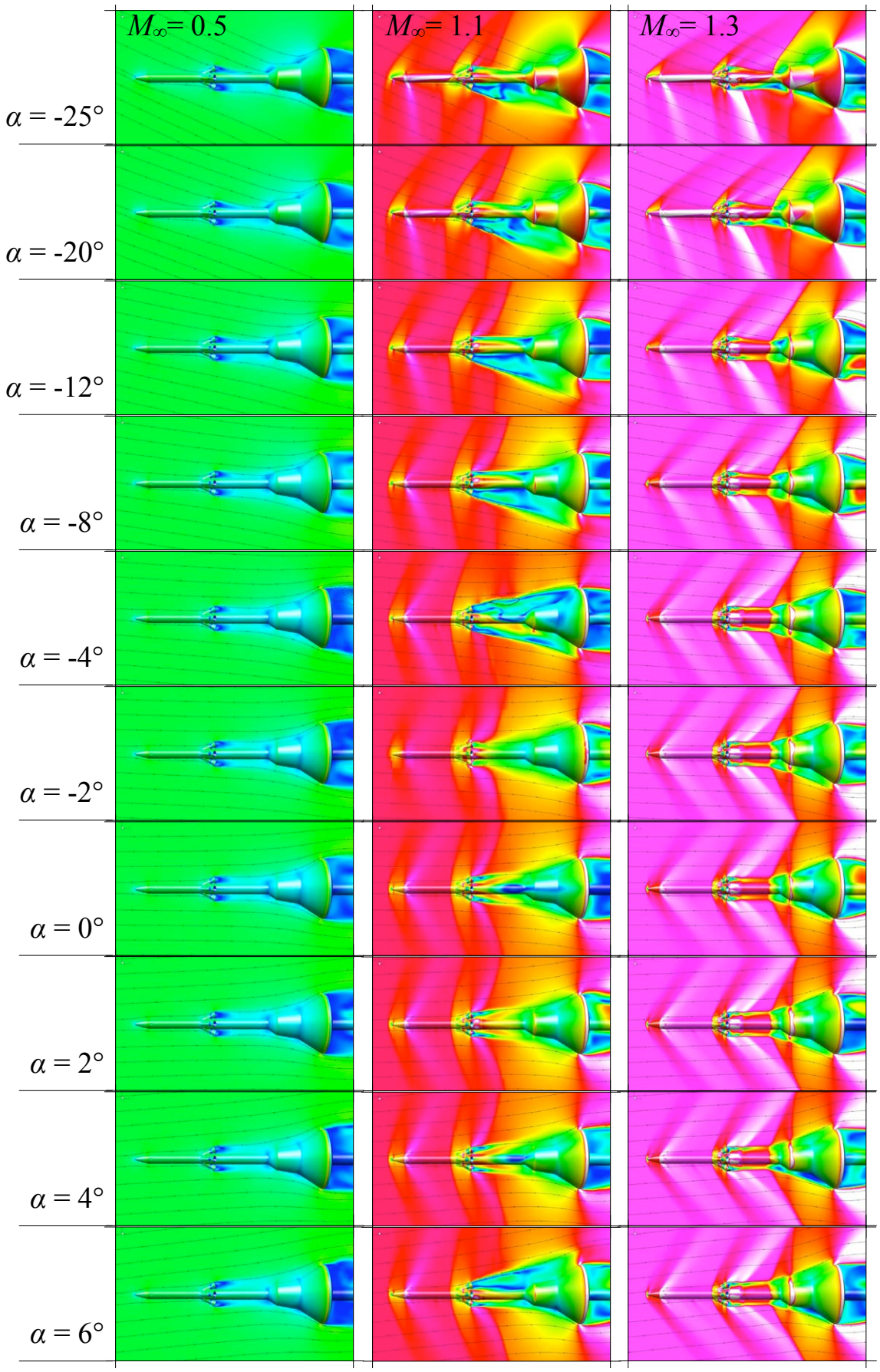


Figure 19. Mach number contours for all flow conditions of the LAV geometry

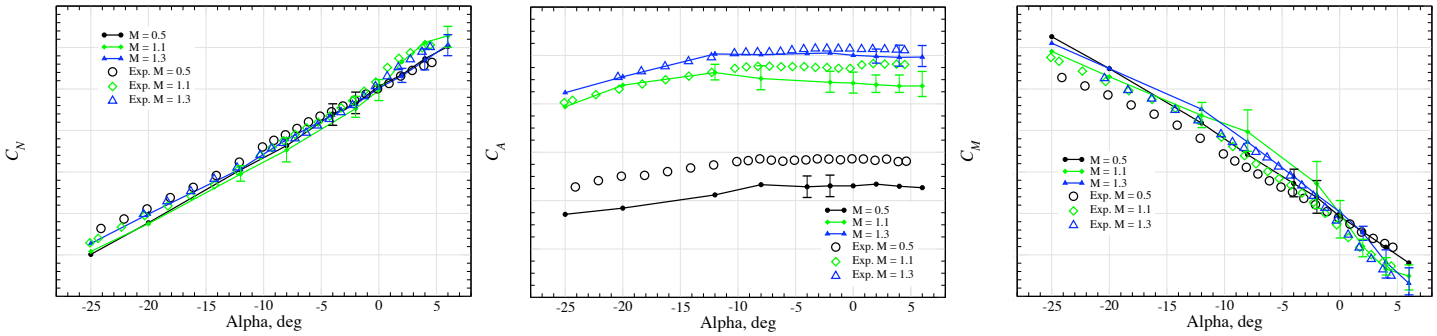


Figure 20. Comparison with experiment for the LAV geometry

In Fig. 20, computations that include error bars crudely indicate the degree of "unsteadiness" of the flow solution on the final mesh. We plot the average value over the last 50 multigrid cycles and indicate peak-to-peak variation with the error bars. Cases with large error bars may require additional unsteady analysis, but overall, the agreement with experimental values is reasonable, despite the complex geometry and bluff body flow with many length scales.

C. Parametric Study of Launch-Vehicle Stage Separation

The final test case considers a parametric study of stage-separation for a launch vehicle. The geometry is a wind-tunnel model of the full-scale vehicle. As shown in Fig. 21, the vehicle consists of two stages, with a launch abort system of the second stage similar to the configuration examined in the previous example. There is a large wind-tunnel strut attached to the second stage and we do not model the sting of the first stage. An interesting feature of this geometry is the inter-stage, which is shown in the right inset of Fig. 21. As the two stages separate, the inter-stage becomes exposed to the flow in the wake of the second stage, and causes interactions that influence forces and moments of both stages. At small separation distances, we use steady-state analysis to examine the dominant flow features, evaluate the forces and moments acting on the two stages, and also examine the interference effect of the strut on the forces and moments.

The run matrix for the steady-state portion of the parametric study consists of roughly 25 configurations with and without the wind-tunnel strut and various offset distances between the stages in the steamwise and

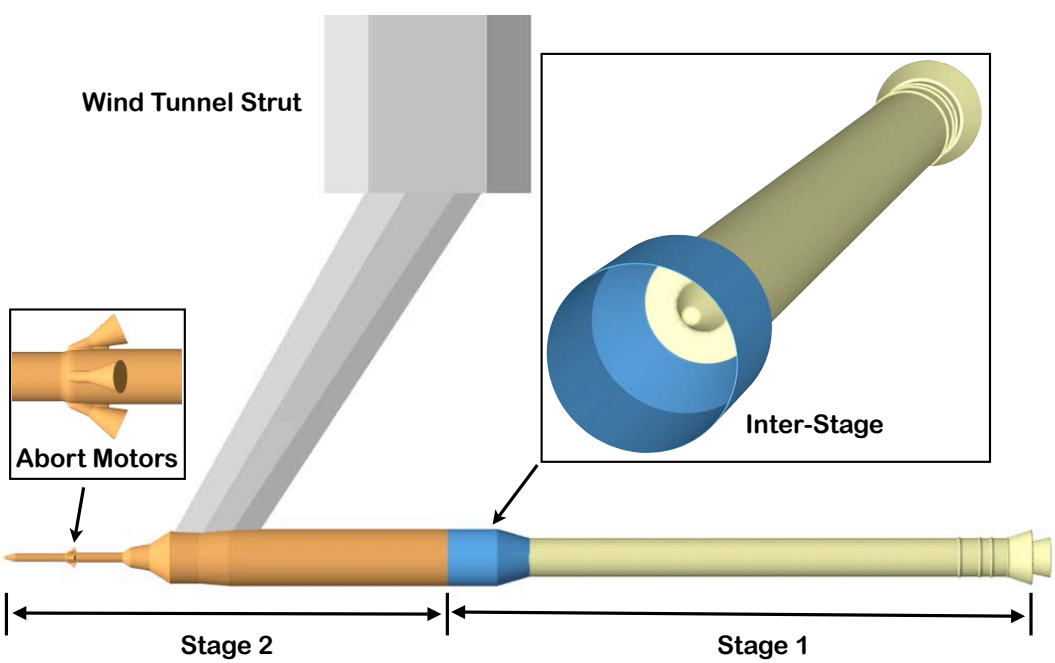
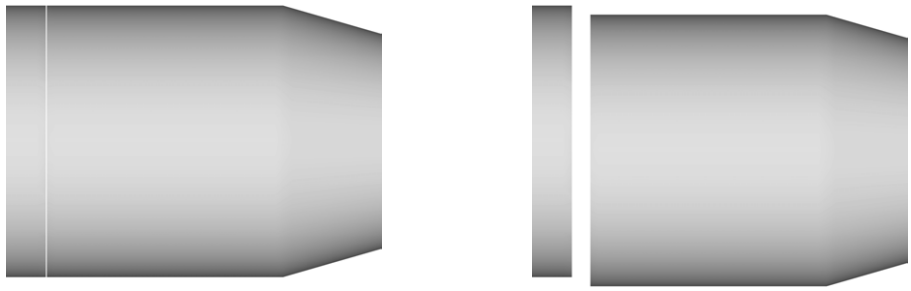


Figure 21. Launch vehicle geometry with wind-tunnel strut



(a) Case A (b) Case B

Figure 22. Example configurations for the launch vehicle stage-separation study. Shown are close-up views of the inter-stage region, with relative positions of Stages 1 and 2

cross-flow directions, as well as two angles of attack. The freestream Mach number is fixed at 4.5. All cases were evaluated with the proposed adjoint-based approach, and from these, we choose two configurations to discuss the reliability and efficiency of our approach. Figure 22 shows the two configurations: Case A considers the smallest relative offset in the streamwise direction of the parametric study, which is 0.05% of the vehicle diameter, and Case B that involves larger separation distances in both the streamwise and cross-flow directions. The angle of attack for these simulations is set to zero. The surface triangulation is obtained from a CAD model of the vehicle and contains roughly 381,000 triangles. As in the previous test case, the triangulation is obtained in a preprocessing step to the flow simulation. We select a linear combination of Stage 1 and Stage 2 drag as the output functional. Note that the nozzle of Stage 1 is not included in the evaluation of the functional, as indicated in Fig. 21, which is similar to previous wind-tunnel measurements. The error tolerance is set at 0.1. We use the decreasing threshold strategy, starting from $\lambda = 8$ in Eq. 11 and reducing λ by a factor of two on every other adaptation cycle to a floor of unity. The initial mesh for all cases in the parametric study is shown in Fig. 23. The mesh contains roughly 13,000 cells with only 1,400 cut-cells. Because of the tiny offset distance relative to the length of the vehicle in Case A, we use an extra level of refinement in the inter-stage region.

Figure 24 shows the final mesh for Case A after eight adaptation cycles. This mesh is also a good example of a typical mesh obtained in the full parametric study for the zero angle-of-attack cases, because the main differences occur only in the inter-stage region. In Fig. 24, the top images show overall views of the vehicle. Comparison of the top-left frame with Fig. 23 reveals that the vertical portion of the wind-tunnel strut has no influence on the selected functional, as there was no refinement of the cells in this region of the mesh. Furthermore, the "Side View" of the vehicle in Fig. 24 shows that the refinement is constrained to a diamond-shaped region around the geometry due to the supersonic flow conditions. This region is actually biconic and bounded by the intersection of the bow shock and the local Mach angle that strikes the base of the nozzle, which is the end location of the selected functional. A close-up of the nozzle region is shown in the lower-right inset of Fig. 24, where we observe rapid coarsening of the mesh at the end of the nozzle skirt. The lower-left inset shows the nose region of the vehicle with remarkably precise refinement of the shock systems.

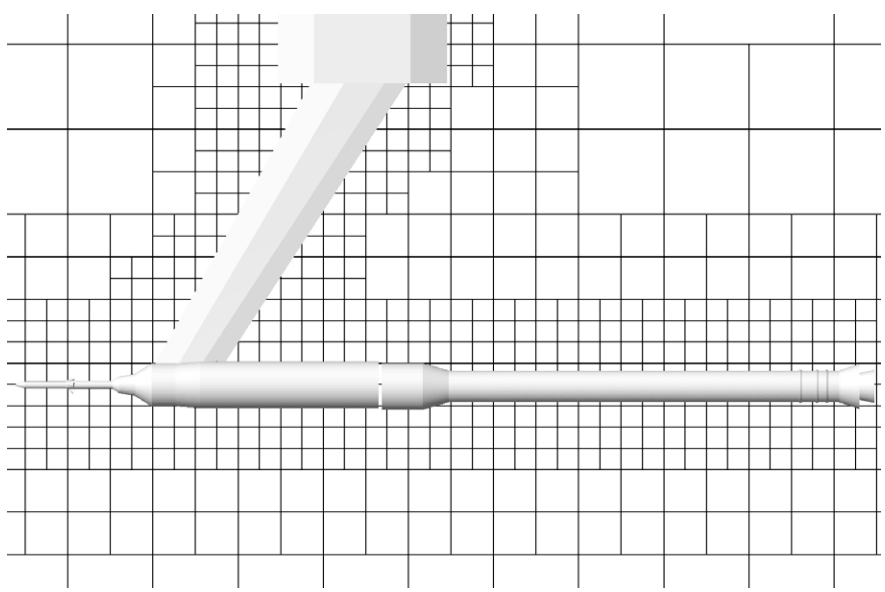


Figure 23. Near-body view of the initial mesh for the launch vehicle configuration ($\approx$ 13,000 cells)

Convergence of the functional and its error estimate is

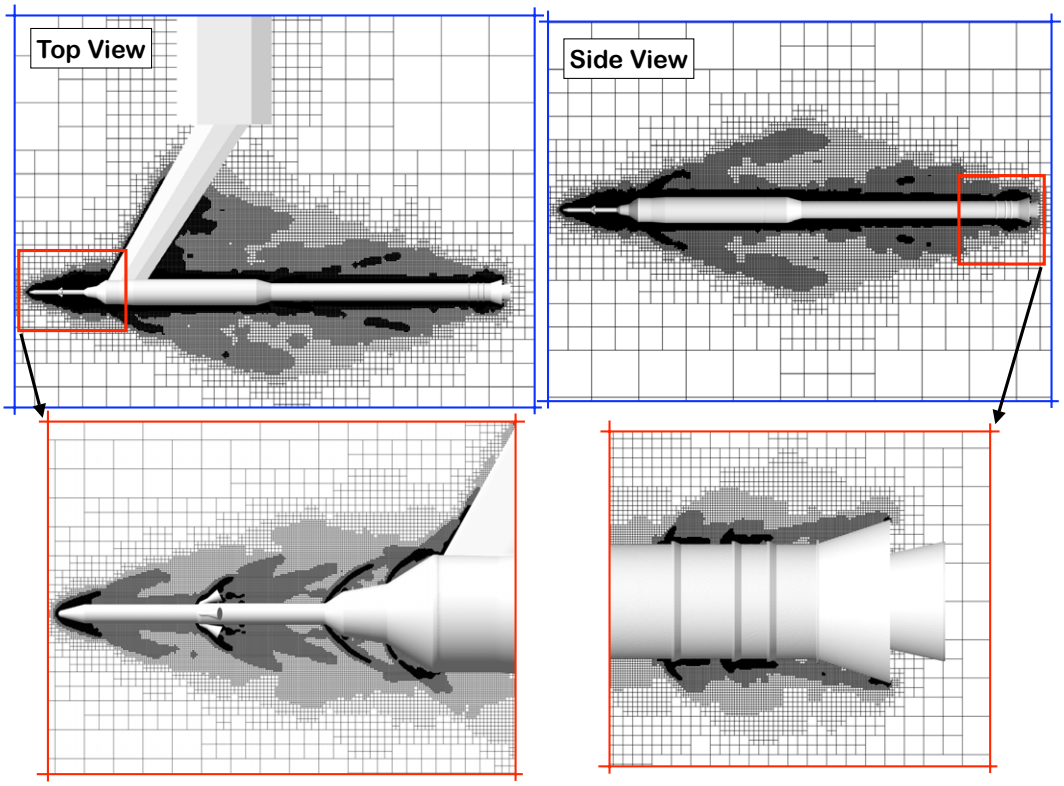
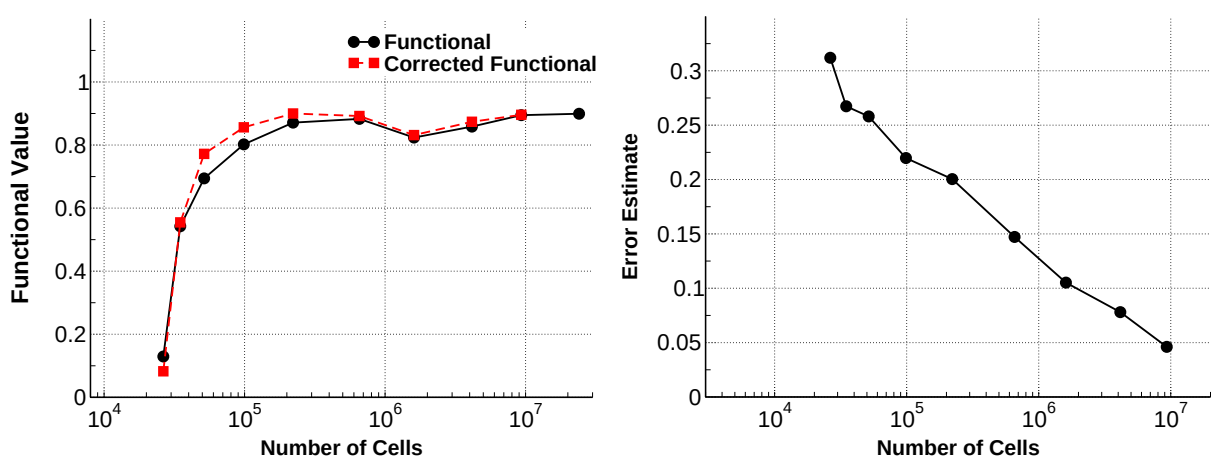


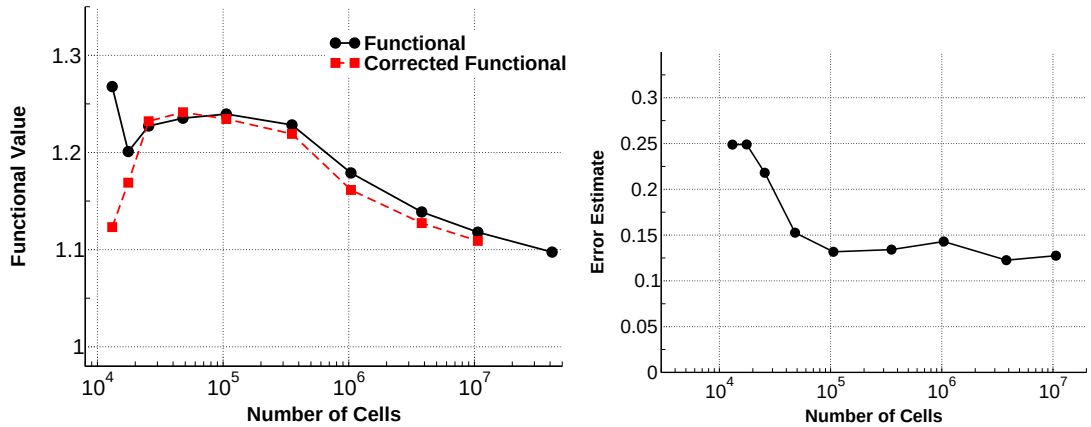
Figure 24. Near-body views of the final mesh for the launch vehicle configuration with roughly 20 million cells, $M_\infty = 4.5$, $\alpha = 0°$

shown in Figs. 25 and 26 for Cases A and B, respectively. Case A converges well over the first five adaptation cycles - the adjoint error correction is accurate in predicting the functional on the next mesh. The dip in the convergence on the sixth adaptation cycle corresponds to achieving a sufficient refinement in the gap between the two stages. At this point, the cells have become small enough in this region to define all the geometric features of the gap accurately. Over the final two adaptation cycles, the functional appears to be approaching an asymptotic value. The error estimate is consistently reduced at each adaptation iteration and reaches the desired tolerance on the eighth adaptation cycle on a mesh with roughly five million cells. Convergence of the functional is not as convincing for Case B, as shown in Fig. 26. Although the adjoint correction term predicts the functional value on the next mesh reasonably well, the convergence of the error estimate stalls after the first four adaptation cycles. The stall is most likely due to flow fluctuations in the inter-stage region, as shown in Fig. 27 for both cases. Case A requires minimal refinement inside the inter-



(a) Functional (b) Error-bound estimate

Figure 25. Convergence of functional and error estimate for Case A



(a) Functional (b) Error-bound estimate

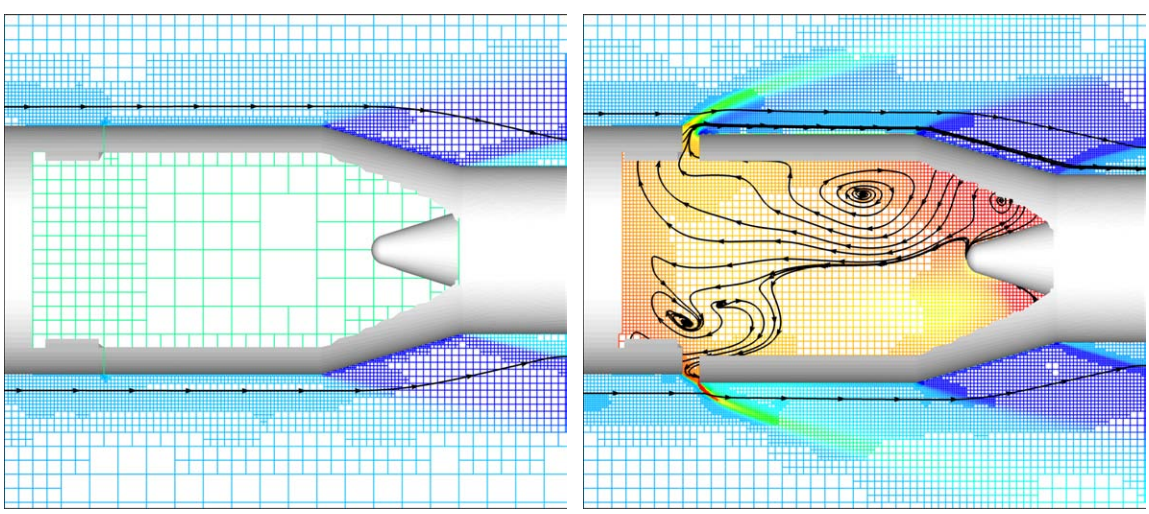
Figure 26. Convergence of functional and error estimate for Case B

stage, although the gap between the stages is highly refined. For Case B, the refinement of the inter-stage is much more detailed due to a complicated flow pattern. Convergence of the flow solution became noisy on the final mesh preventing further adjoint analysis.

Overall, the proposed approach provided reliable estimates of forces and moments for both stages of the vehicle without user intervention throughout the study. Moreover, it improved our understanding of the important features of the flow and allowed the user to focus on validation and analysis of the results instead of mesh generation.

V. Conclusions

This work evaluated the ability of an adjoint-based mesh adaptation method to guide refinement and control discretization errors in inviscid simulations around complex geometries. The adjoint error estimates were used to drive h-refinement of cells in an embedded-boundary Cartesian mesh approach for selected output functionals. Robustness and speed of the Cartesian approach were key ingredients as these simulations involved several hundred meshing/simulation cycles on complex three-dimensional geometries. Detailed investigations studied cell-wise discretization error estimates and traced the evolution of this error through the use of $\log_2$ error histograms. These investigations led to the proposal of an adaptation strategy that reduces the run-time of the flow simulation by using a large refinement threshold on coarse meshes and



(a) Case A (b) Case B

Figure 27. Cutaway view of the inter-stage region. Mesh is colored by contours of pressure with red > 1.5 and blue < 0.4, $M_\infty = 4.5$, $\alpha = 0°$

systematically decreasing the threshold value as the adaptation progresses. This Decreasing Threshold adaptation strategy substantially outperformed our baseline strategy in all cases considered.

The robustness and effectiveness of the adjoint-driven adaptation was examined using several test cases in two and three dimensions using output functionals defined both on the body surface and using off-body sensors. A sonic-boom model problem was used to demonstrate the exceptional economy of the approach. This case showed that results on adaptively refined meshes were nearly indistinguishable from reference solutions computed on fine meshes with $\mathcal{O}(10^3)$ more cells.

Parametric studies of launch-vehicle configurations were used to evaluate the method's reliability when applied to complex flows with many length scales. These examinations focused on the automatic generation of aerodynamic databases with a prescribed error tolerance for two realistically complex three-dimensional geometries over a wide range of flight conditions. These examples evaluated the ability of the approach to drive appropriate mesh adaptation and offer meaningful error estimates when confronted with non-smooth problems that include multi-scale and potentially unsteady flow physics. By focusing on these difficult applications, we hope to advance the use of adjoint-guided meshing and simulation toward practical use in engineering settings.

VI. Acknowledgments

The authors gratefully acknowledge support from the CEV Aerosciences Project, the Space Shuttle Vehicle Integration Office, and the NASA Ames Research Center contract NNA06BC19C. In addition, we wish to thank Shishir Pandya and Goetz Klopfer (NASA Ames) for providing the launch vehicle geometry and many interesting discussions.

References

[1]Berger, M. J. and Jameson, A., Automatic Adaptive Grid Refinement for the Euler Equations, *AIAA Journal*, Vol. 23, No. 4, 1985, pp. 561 568.

[2]Habashi, W. G., Dompierre, J., Bourgault, Y., Fortin, M., and Vallet, M., Certifiable Computational Fluid Dynamics Through Mesh Optimization, *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 703 711.

[3]Dannenhoffer III, J. F., A Comparison of Adaptive-Grid Redistribution and Embedding for Steady Transonic Flows, *Inter. J. for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 653 663.

[4]Bibb, K. L., Gnoffo, P. J., Park, M. A., and Jones, W. T., Parallel, Gradient-Based Anisotropic Mesh Adaptation for Re-entry Vehicle Configurations, AIAA Paper 2006 3579, San Francisco, CA, June 2006.

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

[6]Bussoletti, J. E., Johnson, F. T., Bieterman, M. B., Hilmes, C. L., Melvin, R. G., Young, D. P., and Drela, M., TRANAIR: Solution adaptive CFD modeling for complex 3D configurations, *Recent developments and applications in aeronautical CFD*, Proceedings of the 1993 European Forum, London, United Kingdom, 1993, pp. 10.1 10.14.

[7]Nemec, M. and Aftosmis, M. J., Adjoint Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes, AIAA Paper 2007 4187, Miami, FL, June 2007.

[8]Becker, R. and Rannacher, R., An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numerica 2000*, 2001, pp. 1 102.

[9]Giles, M. B. and Pierce, N. A., Adjoint error correction for integral outputs, *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, edited by T. Barth and H. Deconinck, Vol. 25 of *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, 2002.

[10]Barth, T., Numerical Methods and Error Estimation for Conservation Laws on Structured and Unstructured Meshes, Lecture notes, von Karman Institute for Fluid Dynamics, Series: 2003-04, Brussels, Belgium, March 2003.

[11]Venditti, D. A. and Darmofal, D. L., Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flow, *Journal of Computational Physics*, Vol. 176, 2002, pp. 40 69.

[12]Park, M. A., Adjoint-Based, Three Dimensional Error Prediction and Grid Adaptation, *AIAA Journal*, Vol. 42, No. 9, 2004, pp. 1854 1862.

[13]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, Jan 2006.

[14]Barthet, A., Airiau, C., Braza, M., and Tourrette, L., Adjoint-based error correction applied to far-field drag breakdown on structured grid, AIAA Paper 2006 3315, 2006.

[15]Balasubramanian, R. and Newman III, J. C., Comparison of Adjoint-based and Feature-based Grid Adaptation for Functional Outputs, AIAA Paper 2006 3314, 2006.

[16]Fidkowski, K. J., *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier–Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[17]Dwight, R. P., Goal-Oriented Mesh Adaptation using a Dissipation-Based Error Indicator, AIAA Paper 2007 4093, Miami, FL, June 2007.

[18]Hunt, J. D., *An Adaptive 3D Cartesian Approach for the Parallel Computation of Inviscid Flow about Static and Dynamic Configurations*, Ph.D. thesis, University of Michigan, 2004.

[19]Aftosmis, M. J., Berger, M. J., and Melton, J. E., Robust and Efficient Cartesian Mesh Generation for Componenet-Based Geometry, *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952 960.

[20]van Leer, B., Flux-Vector Splitting for the Euler Equations, ICASE Report 82-30, Sept. 1982.

[21]Aftosmis, M. J., Berger, M. J., and Adomavicius, G., A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries, AIAA Paper 2000 0808, Reno, NV, Jan. 2000.

[22]Aftosmis, M. J., Berger, M. J., and Murman, S. M., Applications of Space-Filling-Curves to Cartesian Methods for CFD, AIAA Paper 2004 1232, Reno, NV, Jan. 2004.

[23]Berger, M. J., Aftosmis, M. J., and Murman, S. M., Analysis of Slope Limiters on Irregular Grids, AIAA Paper 2005 0490, Reno, NV, Jan. 2005.

[24]Nemec, M., Aftosmis, M. J., Murman, S. M., and Pulliam, T. H., Adjoint Formulation for an Embedded-Boundary Cartesian Method, AIAA Paper 2005 0877, Reno, NV, Jan. 2005.

[25]Schmitt, V. and Charpin, F., Pressure Distributions on the ONERA-M6-Wing at transonic Mach numbers, *Experimental Data Base for Computer Program Assessment*, AGARD-R-138, May 1979.

[26]Fidkowski, K. J. and Darmofal, D. L., Output-based Adaptive Meshing Using Triangular Cut Cells, MIT Aerospace Computational Design Laboratory Report TR-06-2, 2006.