24th AIAA Applied Aerodynamics Conference, June 5–8, 2006, San Francisco, CA

Aerodynamic Shape Optimization Using a Cartesian Adjoint Method and CAD Geometry

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

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

We present a new approach for the computation of shape sensitivities using the discrete adjoint and flow-sensitivity methods on Cartesian meshes with general polyhedral cells (cut-cells) at the wall boundaries. By directly linearizing the cut-cell geometric constructors of the mesh generator, an efficient and robust computation of shape sensitivities is achieved. We show that the error convergence rate of the flow solution and its sensitivity, as well as the objective function and its gradient is consistent with the second-order spatial discretization of the three-dimensional Euler equations. The performance of the approach is demonstrated for an airfoil optimization problem in transonic flow, and a CAD-based shape optimization of a reentry capsule in hypersonic flow. The approach is well-suited for conceptual design studies where fast turn-around time is required.

I. Introduction

GRADIENT-based numerical optimization is an effective approach for aerodynamic design problems. When coupled with efficient flow solvers, these methods rapidly determine shape modifications that improve aerodynamic performance and design safety factors. At the heart of the optimization is the computation of the objective function gradient; the sensitivity of some performance measure, such as the lift and drag functionals, to a given set of design variables that control the shape. Well-known techniques for gradient computation are the adjoint and flow-sensitivity (or direct) methods.[1–3] An important issue in the application of adjoints and flow-sensitivities as routine tools of aerodynamic shape optimization is the need to handle geometrically complex engineering designs. Repeated meshing of complex geometry throughout the design process creates an implicit requirement for robust, efficient, and automatic mesh generation. For problems governed by the Euler equations, this requirement is fulfilled by the use of Cartesian meshes with embedded boundaries.[4–9] The purpose of this work is to extend the efficiency and automation capabilities of Cartesian methods to the computation of gradients for shape optimization problems.

A key step in the formulation of the adjoint and flow-sensitivity methods is the linearization of the flow equations, which casts the formulation as continuous or discrete. In the continuous formulation, the linearization is performed before the discretization of the flow equations. This order is reversed in the discrete approach. The most relevant aspect in the present context, however, is the inclusion of shape perturbations in either formulation. Most implementations use body-fitted structured or unstructured meshes in conjunction with mesh-perturbation schemes to smoothly deform the volume mesh in response to changes in the surface geometry.[10–18] A linearization of the mesh-perturbation scheme, or mesh sensitivity, is used to account for the effect of shape perturbations in the flow equations. While this approach provides an accurate computation of the gradient, it also introduces several complications. Efficient computation of mesh sensitivities requires the solution of an additional adjoint equation.[19,20] Furthermore, large shape modifications may cause a breakdown of the mesh-perturbation scheme, and implementations of mesh-perturbation schemes in conjunction with mesh adaptation and embedded-boundary methods are problematic due to discretization changes at each design iteration.

*Research Scientist, Advanced Supercomputing Division, MS T27B; mnemec@mail.arc.nasa.gov. Member AIAA.
†Research Scientist, Advanced Supercomputing Division, MS T27B; maftosmis@mail.arc.nasa.gov. Senior Member AIAA.

The simplest strategy to circumvent some of these difficulties is to limit the extent of the mesh perturbations to near-body cells. Studies have been performed by Anderson and Venkatakrishnan[21] and Lu[22] on unstructured meshes to determine the effect of partial mesh sensitivities on the accuracy of gradient computations. In these body-fitted approaches, the mesh perturbations were restricted to the surface boundary cells, with all interior cells held fixed. The results show that for local shape deformations, away from singularities, reliable approximations of the gradient can be obtained. A similar approach is used by Dadone and Grossman,[23] who presented a discrete-adjoint formulation for the Euler equations on Cartesian meshes. They use an immersed-body methodology and apply finite-difference approximations to compute shape sensitivities only in the boundary intersecting cells. Jameson and Kim[24] introduced a reduced gradient approach for the continuous adjoint formulation on structured grids, where only surface perturbations are considered. Their results indicate that for shape optimization governed by the Euler equations, the reduced approach is accurate and provides significant computational savings. Similar results are also presented by Soto and Löhner[25] on unstructured meshes.

A different strategy is to avoid mesh-perturbation schemes altogether. A good example is the Cartesian TRANAIR code,[4,26] which is based on the full-potential equation with viscous corrections. In this approach, Huffman *et al.*[27] developed and linearized a transpiration boundary condition to approximate the effects of the changing boundary surface. They obtained good gradient accuracy for problems where the design variables involve shape changes normal to the surface, and this approach has been applied to many practical shape-optimization problems. More recently, Duvigneau and Pelletier[28] introduced a continuous formulation that uses higher-order Taylor series approximations near the boundary to obtain accurate shape sensitivities for two-dimensional problems.

In this paper, we propose an automatic and efficient method for the computation of shape sensitivities for problems governed by the three-dimensional Euler equations. We use a discrete formulation based on Cartesian meshes with embedded, cut-cell boundaries. Our approach is similar to a mesh-perturbation scheme where the perturbations are restricted to the boundary cells. The main difference is that we project the shape perturbations onto the Cartesian mesh by directly linearizing the geometry constructors of the mesh generator. In this linearization, we exploit the decoupling of the surface triangulation from the volume mesh to provide a flexible interface for geometry control, which may include parametric computer-aided design (CAD) and in-house geometry modelers.

This work continues the development of the discrete adjoint and flow-sensitivity methods reported in Refs. 29 and 30. In particular, the linearization of the mesh generator and flow solver is extended to include all second-order terms of the finite-volume spatial discretization, with the exception of slope limiters. To assess the accuracy of the gradient, we present detailed verification studies using comparisons with analytic model problems and finite-difference approximations. Thereafter, we investigate the performance of the new method on two representative shape-optimization problems, namely, a lift-constrained drag minimization for an airfoil in transonic flow, and an enhancement of the lift-to-drag ratio for an Apollo-like reentry capsule in hypersonic flow. Factors under consideration include robustness of the adjoint and flow-sensitivity solvers, and efficiency of the gradient computation. In addition, the design examples are also used to demonstrate the capability of the method to use various geometry modeling and manipulation tools, including a direct-CAD interface for the capsule problem and an analytic B-spline approach for the airfoil problem.

II. Optimization Problem

The aerodynamic optimization problem we consider in this work consists of determining values of design variables that minimize a given objective function

$$\min_X \ \mathcal{J}(X, Q) \tag{1}$$

where $\mathcal{J}$ represents a scalar objective function defined by a surface integral, for example lift or drag, X denotes a scalar design variable, for example a shape parameter of the wetted surface, and $Q = [\rho, \rho u, \rho v, \rho w, \rho E]^{\mathrm{T}}$ denotes the continuous flow variables. The flow variables are forced to satisfy the three-dimensional, steady-state Euler equations of a perfect gas within a feasible region of the design space Ω

$$\mathcal{F}(X, Q) = 0 \qquad \forall\ X \in \Omega \tag{2}$$

which implicitly defines $Q = f(X)$. The optimization problem is solved using the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method in conjunction with a backtracking line-search.[31] We use a discrete formulation

for the computation of the gradient, $\mathrm{d}\mathcal{J}/\mathrm{d}X$, where the governing equations, Eqs. 1 and 2, are first discretized and then linearized. In the following section, we present background information on the Cartesian mesh generator and the flow solution method to help anchor the subsequent discussion on linearization.

III. Objective Function Evaluation

A. Mesh Generator

The Euler equations are discretized on a multilevel Cartesian mesh with embedded boundaries.[8] 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 Figs. 1 and 2 , these cells are arbitrary polyhedra. Our focus is on the cut-cells, because an infinitesimal perturbation of the boundary shape affects only these cells in the mesh. We describe the salient features of the cut-cell algorithm that affect the computation of shape sensitivities.

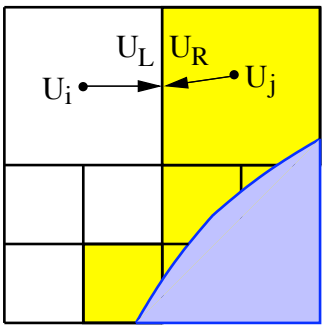


Figure 1. Illustration of a Cartesian mesh in two-dimensions. Cut-cells are shaded and two refinement levels are shown.

The cut-cell algorithm computes the intersection of the surface triangulation with the faces of Cartesian hexahedra. An example is shown in Fig. 2, where a Cartesian hexahedron is split into two cut-cells by the surface triangulation. A key step in the generation of a cut-cell is the construction of triangle-polygons, which are the portions of triangles retained by the cell. Figure 2 shows an example of two triangle-polygons, and we describe the construction of the triangle-polygon (A, B, C, D) for triangle (V_0, V_1, V_2). First, the algorithm computes points C and D formed by the intersection of Cartesian faces with the edge V_0V_1. These points are referred to as on-face pierce points. The geometric constructor for point C (and for on-face pierce points in general) is given by

$$C = V_0 + s(V_1 - V_0) \tag{3}$$

where s denotes the distance fraction of the face location relative to the vertices V_0 and V_1. Next, the intersections of Cartesian edges with the surface triangulation are computed. These are on-edge pierce points, and examples are points A and B in triangle (V_0, V_1, V_2). The computation of these points occurs during the assembly of the triangle-polygons. The geometric constructor for on-edge pierce points involves the vertices of the triangle, as well as the corner points of the hexahedron. Further details are provided in Refs. 32 and 30.

The geometric information required by the finite-volume discretization of the Euler equations is derived from the triangle-polygons. The computation of areas and centroids for the Cartesian faces of the cut-cell is accomplished by subdividing each face into triangles and computing the area and centroid of each triangle. Note that this involves both the on-face and on-edge pierce points. The computation of the surface normal is based on a planar approximation to the variation of the triangulation within each cell. This agglomerated normal vector is computed using the divergence theorem, which requires the geometric closure of each cut-cell

$$\sum_{\mathrm{j}\in\mathrm{V_i}} A_j \cdot \hat{\mathbf{n}}_j = 0 \tag{4}$$

where $\hat{\mathbf{n}}$ denotes the unit normal vector and A represents the face area. The sum is performed over the Cartesian faces j of the cell, which must equal the agglomerated normal vector. The divergence theorem is also applied in the computation of the volume centroid and involves contributions from the triangle-polygons, as well as the Cartesian faces.

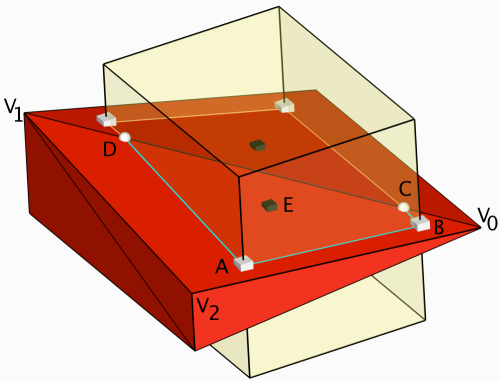


Figure 2. Illustration of a cut-cell: the intersection of a Cartesian hexahedron with triangle (V_0, V_1, V_2) creates on-edge pierce points A and B and on-face pierce points C and D that form a triangle-polygon (A, B, C, D) with a surface centroid E.

B. Flow Solver

The discretization of the Euler equations uses a second-order accurate finite-volume method. A cell-centered approach is used, where the control volumes correspond to the mesh cells and the cell-averaged value of Q, denoted by $\bar{Q}$, is located at the centroid of each cell. The flux residual in each cell i is expressed as

$$R_i = \sum_{\mathrm{j}\in\mathrm{V_i}} \mathbf{F}_j \cdot \hat{\mathbf{n}}_j A_j \tag{5}$$

where j denotes the jth face of volume V_i and $\mathbf{F}$ represents the numerical flux function. The flux function is evaluated at the face centroids using the flux-vector splitting approach of van Leer.[33] Primitive variables, $U = [\rho, u, v, w, p]^{\mathrm{T}}$, are used for the reconstruction of the solution to the cell face. This is illustrated in Fig. 1 for two neighbouring Cartesian cells i, j sharing a common face. The left and right states are given by

$$\begin{aligned} U_{\mathrm{L}} &= \bar{U}_i + d_{\mathrm{L}}\phi_i \nabla U_i \\ U_{\mathrm{R}} &= \bar{U}_j - d_{\mathrm{R}}\phi_j \nabla U_j \end{aligned} \tag{6}$$

where d_{L} and d_{R} are the distances from the cell centroids to the face centroid, ϕ is the slope limiter used to ensure monotonic reconstructions, and ∇U is the solution gradient determined via a linear least-squares procedure. The boundary conditions are enforced weakly using pressure from interior cells in the momentum equations. The resulting discrete system of equations is given by

$$\vec{R}(\vec{Q}, \vec{M}, X) = 0 \tag{7}$$

where $\vec{Q} = [\bar{Q}_1, \bar{Q}_2, \ldots, \bar{Q}_N]^{\mathrm{T}}$ is the discrete solution vector for all N cells of a given mesh $\vec{M}$, and $\vec{R}$ is the flux residual vector.

Steady-state flow solutions are obtained using a five-stage Runge–Kutta scheme with local time stepping, multigrid, and a highly-scalable domain decomposition scheme for parallel computing. For further details on the spatial discretization and flow solution, see Aftosmis *et al.*[8,34,35] and Berger *et al.*[36,37]

IV. Gradient Computation

The design variables that appear directly in Eq. 7 involve parameters that do not change the computational domain, such as the Mach number, angle of attack, and side-slip angle. The influence of shape design variables on the residuals in Eq. 7 is implicit via the computational mesh $\vec{M}$:

$$\vec{M} = f[\vec{T}(X)] \tag{8}$$

where $\vec{T}$ denotes a triangulation of the wetted surface. In the following subsections, we present the linearization of the residual equations and the objective function to obtain the gradient.

A. Discrete Adjoint and Flow-Sensitivity Methods

The gradient of the discrete objective function $\mathcal{J}(X, \vec{M}, \vec{Q})$ with respect to a design variable X is given by

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X} = \frac{\partial \mathcal{J}}{\partial X} + \frac{\partial \mathcal{J}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X} + \frac{\partial \mathcal{J}}{\partial \vec{Q}} \frac{\mathrm{d}\vec{Q}}{\mathrm{d}X} \tag{9}$$

The evaluation of the term $\mathrm{d}\vec{Q}/\mathrm{d}X$, referred to as the flow sensitivities, is obtained by combining Eqs. 7 and 8, and differentiating with respect to the design variables

$$\frac{\partial \vec{R}}{\partial \vec{Q}} \frac{\mathrm{d}\vec{Q}}{\mathrm{d}X} = -\left(\frac{\partial \vec{R}}{\partial X} + \frac{\partial \vec{R}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X} \right) \tag{10}$$

We assume that the implicit function $\vec{Q}(X)$ is sufficiently smooth, and note that $\mathrm{d}\vec{R}/\mathrm{d}X = 0$ because Eq. 2 holds for any design variable. The direct, or *flow-sensitivity*, method results from solving Eq. 10 for the flow sensitivities $\mathrm{d}\vec{Q}/\mathrm{d}X$ and using these values in Eq. 9 to obtain the gradient.

The *adjoint* equation is obtained by combining Eqs. 10 and 9 and defining the following intermediate problem

$$\frac{\partial \vec{R}}{\partial \vec{Q}}^{\mathrm{T}} \vec{\psi} = \frac{\partial \mathcal{J}}{\partial \vec{Q}}^{\mathrm{T}} \tag{11}$$

where the vector $\vec{\psi}$ represents the adjoint variables. The corresponding expression for the gradient is given by

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X} = \frac{\partial \mathcal{J}}{\partial X} + \underbrace{\frac{\partial \mathcal{J}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X}}_{\mathrm{A}} - \vec{\psi}^{\mathrm{T}} \left(\frac{\partial \vec{R}}{\partial X} + \underbrace{\frac{\partial \vec{R}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X}}_{\mathrm{B}} \right) \tag{12}$$

The solution algorithm for the flow-sensitivity and adjoint equations leverages the Runge–Kutta time-marching scheme and the parallel multigrid method of the flow solver. The algorithm is implemented using the duality-preserving approach,[38] such that the asymptotic convergence rate of the flow, flow-sensitivity, and adjoint solvers is the same. The matrix-vector products associated with the flow-Jacobian matrix, left side of Eq. 11, are computed on-the-fly using a two-pass strategy over the faces of the mesh. The flow-Jacobian matrix, as well as the term $\partial \mathcal{J}/\partial \vec{Q}$ in Eq. 11, are derived by hand, where we neglect the linearization of the limiter function used in the solution reconstruction procedure. Overall, the CPU time per iteration and memory usage of the adjoint and flow-sensitivity solvers are roughly equivalent to the flow solver. Further details are provided in Ref. 29.

The computation of partial derivative terms $\partial \mathcal{J}/\partial X$ and $\partial \vec{R}/\partial X$ in Eq. 12 is straightforward, because these terms do not involve derivatives of the surface shape. The remaining partial derivative terms in Eq. 12, labeled as A and B, represent the differentiation of the objective function and residual equations with respect to design variables that alter the surface shape. We discuss the computation of these terms in the next section.

B. Computation of Shape Sensitivities

An important feature of the cut-cell Cartesian approach is that there is no prescribed connectivity between the surface triangulation and the volume mesh. This feature separates the task of geometry modeling and surface triangulation from those of volume mesh generation. Moreover, the computation of shape sensitivities associated with the vertices of the surface triangulation, i.e., the term $\partial \vec{T}/\partial X$ in Eq. 12, is decoupled from the volume mesh sensitivities, $\partial \vec{M}/\partial \vec{T}$. This is an especially important factor when using CAD-based geometry, because the mesh generator needs no knowledge of how the surface triangulation is constructed.

For geometry manipulation and surface triangulation of parametric-CAD models we use the Computational Analysis and PRogramming Interface (CAPRI) developed by Haimes *et al.*[39,40] Our approach is summarized in Ref. 30, where finite-difference approximations are used to compute the term $\partial \vec{T}/\partial X$. Furthermore, we presented a linearization of the on-edge pierce points (see Fig. 2), which allowed us to compute mesh sensitivities consistent with first-order discretization of the Euler equations. This required the linearization of the agglomerated surface normal, Eq. 4, and face areas. We extend the linearization to include on-face pierce points, as well as face and volume centroids, to obtain an exact linearization of the cut-cell geometry. The linearization of the geometric constructor for the on-face pierce point C in Eq. 3 is given by

$$\frac{\partial C}{\partial X} = \frac{\partial V_0}{\partial X} + s\left(\frac{\partial V_1}{\partial X} - \frac{\partial V_0}{\partial X}\right) + (V_1 - V_0)\frac{\partial s}{\partial X} \tag{13}$$

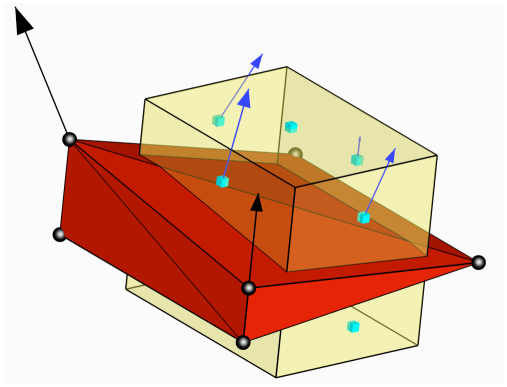


Figure 3. Example sensitivity of face centroid locations (blue vectors) to shape perturbations of two vertices of the surface triangulation (black vectors). Sensitivity vectors are scaled by a factor of three for visualization.

Figure 3 shows an example linearization of the face centroid locations for the Cartesian faces of a cut-cell. The surface triangulation contains two vertices with non-zero shape sensitivities, denoted by black vectors. The resulting sensitivities of the face centroids are

denoted by blue vectors. Note that the"motion" of the face centroids is constrained to the plane of the face. This contrasts with mesh-perturbation approaches for body-fitted meshes, such as the spring-analogy approach, where the cell faces are bound to the vertices of the triangulation and hence are "convected" with the vertices as they move.

Linearization of residual equations follows directly from Eq. 5 and is given by

$$\frac{\partial R_i}{\partial X} = \sum_{\mathrm{j}\in\mathrm{V_i}} \left[\mathbf{F}_j \cdot \frac{\partial \left(\hat{\mathbf{n}}_j A_j \right)}{\partial X} + \frac{\partial \mathbf{F}_j}{\partial X} \cdot \hat{\mathbf{n}}_j A_j \right] \tag{14}$$

where the linearization of the flux function involves the reconstruction of the flow solution to the cell face. For example, the linearization of Eq. 6 for the left state is given by

$$\frac{\partial U_{\mathrm{L}}}{\partial X} = \phi_i \left(d_{\mathrm{L}} \frac{\partial \nabla U_i}{\partial X} + \nabla U_i \frac{\partial d_{\mathrm{L}}}{\partial X} \right) \tag{15}$$

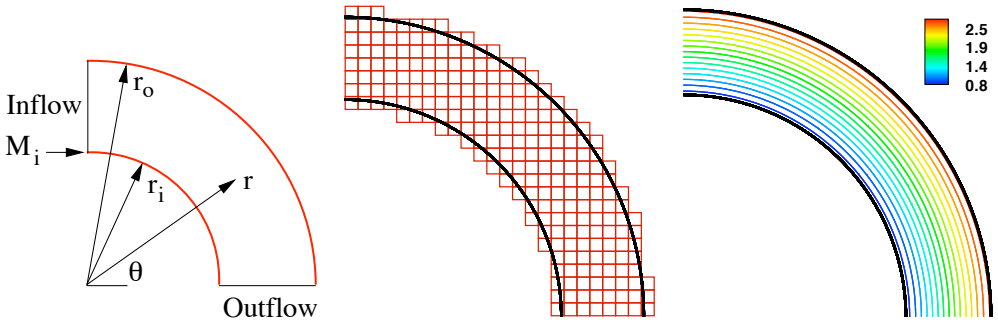
The geometric derivatives, e.g., $\partial d_{\mathrm{L}}/\partial X$ and $\partial \hat{\mathbf{n}}/\partial X$, required for the evaluation of Eqs. 14 and 15 are obtained directly from the linearization of the mesh generator. The flow-solution gradient, ∇U, involves the linearization of the least-squares procedure based on Cholesky factorization. Overall, the linearization of the residual equations is computed on-the-fly for each design variable and reuses the data structures of the mesh generator and flow solver. The CPU time requirements are minimal, because the residual sensitivities in Eq. 14 are non-zero only in the cut-cells and their first and second nearest-neighbors.

V. Verification Studies

We present two verification studies for the adjoint and flow-sensitivity methods. In the first problem, we investigate the convergence of the gradient error for design variables that do not alter the boundary shape. This establishes a benchmark for the second problem, where we examine the accuracy of gradients for shape design variables. In both problems, we use second-order accurate spatial disctretization without limiters, i.e., the slope limiter in Eq. 6 is unity. In these examples, the flow, flow-sensitivity, and adjoint equations are all converged 12 orders of magnitude.

A. Supersonic Vortex Model Problem

To verify the implementation of the adjoint and flow-sensitivity methods, we investigate the error convergence rate of a representative objective function and its gradient. The problem involves isentropic flow between concentric circular arcs at supersonic conditions, for which analytic solutions of the flowfield and flow sensitivities are known. The problem setup is shown in Fig. 4. This problem was used in Ref. 41 to verify the order of accuracy of the flow solver. The exact solution in primitive variables is given by



(a) Problem setup (b) Example mesh (second coarsest mesh of the refinement study) (c) Nondimensional pressure

Figure 4. Supersonic vortex model problem ($M_i = 2.25$, $r_i = 1$, and $r_o = 1.382$).

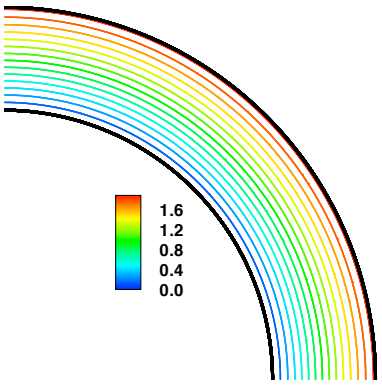
$$
\begin{aligned}
\rho &= \rho_i \left\{ 1 + \frac{\gamma - 1}{2} M_i^2 \left[1 - \left(\frac{r_i}{r} \right)^2 \right] \right\}^{\frac{1}{\gamma - 1}} \\
u &= a_i M_i \left(\frac{r_i}{r} \right) \sin\theta \\
v &= -a_i M_i \left(\frac{r_i}{r} \right) \cos\theta \\
p &= \frac{p_i}{\rho_i^\gamma} \rho^\gamma
\end{aligned}
\tag{16}
$$

where $a_i = \rho_i = 1$, $p_i = 1/\gamma$, $M_i = 2.25$, $r_i = 1$, and $r_o = 1.382$. The integral of pressure along the outer arc provides an objective function that is similar to the lift and drag boundary integrals in aerodynamic design

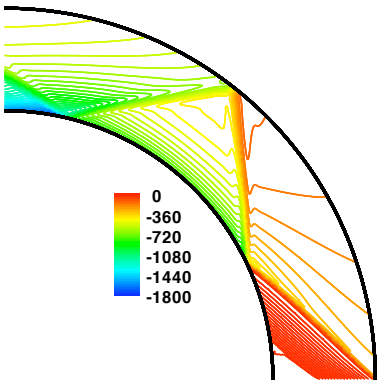
$$
\mathcal{J} = \int_{r_o} p \; \mathrm{d}l = \frac{\pi r_o}{2} p_{r_o} \tag{17}
$$

We compute the objective function gradient and the sensitivities of the flow solution with respect to the inlet Mach number, M_i. Straightforward differentiation of Eqs. 16 and 17 with respect to M_i gives the exact solution for the gradient and flow sensitivities. We use the exact solution to specify Dirichlet inlet and outlet boundary conditions for the solution of the flow, flow-sensitivity, and adjoint equations. The problem is solved on a sequence of nested Cartesian meshes containing 70, 241, 887, 3384, and 13192 cells. We emphasize that although the selected design variable does not alter the shape of the boundary (terms A and B in Eq. 12 are zero), the boundary discretization changes non-smoothly during mesh refinement.

Before presenting error-convergence results, we briefly consider the flow-sensitivity and adjoint solutions shown in Fig. 5 to provide insight into their behavior and physical interpretation. In Fig. 5(a), the gradient of density with respect to the inlet Mach number, $\partial\rho/\partial M_i$, varies directly with radius and is similar to the pressure field shown in Fig. 4(c). The gradient vanishes on the inner arc, because density is independent of Mach number on this boundary. Figure 5(b) shows the adjoint solution for the continuity equation, which can be interpreted as the influence of a point-source of mass on the objective function.[14] The nonlinear variations of the adjoint field correspond to the propagation of point-source perturbations, which include interactions with the inner boundary, for an objective function defined by Eq. 17 along the outer arc. This is clearly seen at the outflow boundary of the duct where the adjoint variable vanishes due to the supersonic nature of the flow. Moreover, the adjoint variable is zero for some distance upstream of the outflow boundary (red contours in Fig. 5(b)), which is defined by the local Mach angle. This shows that perturbations originating past a certain location on the inner arc can not influence an objective function that is defined only on the outer arc. Lastly, an important observation regarding Fig. 5 is the smooth behavior of both the flow-sensitivity and adjoint solutions in the cut-cell boundary, with no visible irregularities due to the arbitrary boundary discretization.



(a) Flow-sensitivity contours, $\partial\rho/\partial M_i$

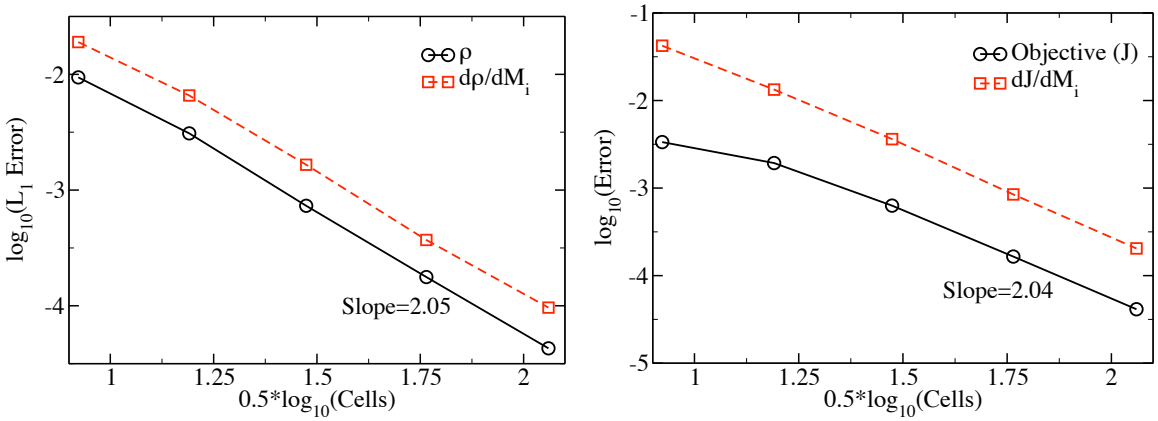


(b) Adjoint contours for the continuity equation. Objective function is given by Eq. 17 and is defined along the outer arc only.

Figure 5. Flow-sensitivity and adjoint solutions for the supersonic vortex model problem.

Figure 6 summarizes the results of the error convergence study. Figure 6(a) shows the error convergence

rates in the L_1 norm of density and its gradient with respect to the inlet Mach number. The error convergence rate of the objective function and its gradient is shown in Fig. 6(b). The adjoint and flow-sensitivity methods compute identical gradients and we display the results obtained via flow sensitivities. The asymptotic convergence rate of errors, which is measured over the three finest meshes, is just slightly over two. These results are consistent with the second-order spatial discretization of the Euler equations and objective function, thereby verifying the accuracy of the linearization and the convergence of these methods to the continuous problem.



(a) L_1 norms of error in density and its gradient

(b) Errors in objective function and its gradient

Figure 6. Error convergence for the supersonic vortex model problem. Slopes of the error lines are indicated for the three finest meshes.

B. Shape Sensitivities for the NACA 0012 Airfoil

The objective of this verification test is to study the convergence of gradients for design variables that alter the shape of the boundary. We consider a subsonic flow over the NACA 0012 airfoil and we choose lift as the objective function. The freestream Mach number is 0.5 and the angle of attack is 1 deg. We investigate the sensitivity of lift to the angle of attack using two equivalent approaches. First, we consider the influence of the angle of attack via a change in the farfield boundary conditions. This approach is similar to the supersonic vortex problem, because freestream perturbations, such as the Mach number and the angle of attack, do not alter the relationship between the mesh and the airfoil. We contrast this with an angle of attack change implemented via a rigid-body rotation of the airfoil about its trailing edge within a fixed mesh. Figure 7 shows the differences in the cut-cells near the leading edge of the airfoil at an angle of attack of 0 and 1 deg. on an intermediate mesh. The mesh-refinement study is performed on a sequence of five nested Cartesian meshes consisting of 3238, 12627, 49770, 192427, and 757531 cells for each airfoil orientation.

The results are summarized in Fig. 8. We observe that lift and its gradient are converging to essentially the same value. This is in agreement with incompressible flow theory for thin airfoils, which predicts a linear variation of lift with the angle of attack. Referring to Fig. 8(b), note that the differences in the gradients between the farfield and rigid-body rotation cases are decreasing as the mesh is refined. Additional regression analysis of this data indicates that the rate of convergence is first-order. This is a consequence of the fact that the mesh perturbations are confined to only the cut-cells. Unlike the supersonic vortex problem, a perturbation of the angle of attack via rigid-body rotation modifies the cut-cell boundary and introduces an error in the objective function proportional to second-order spatial discretization. Nevertheless, the gradient values for the rigid-body rotation case, even on the coarsest mesh of roughly $3,200$ cells, are within 1% of the fine-mesh values. Consequently, such gradients should be sufficient to generate reliable descent directions for an optimization algorithm. In the next section, we investigate this hypothesis for several shape-optimization problems.

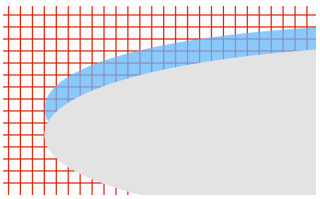
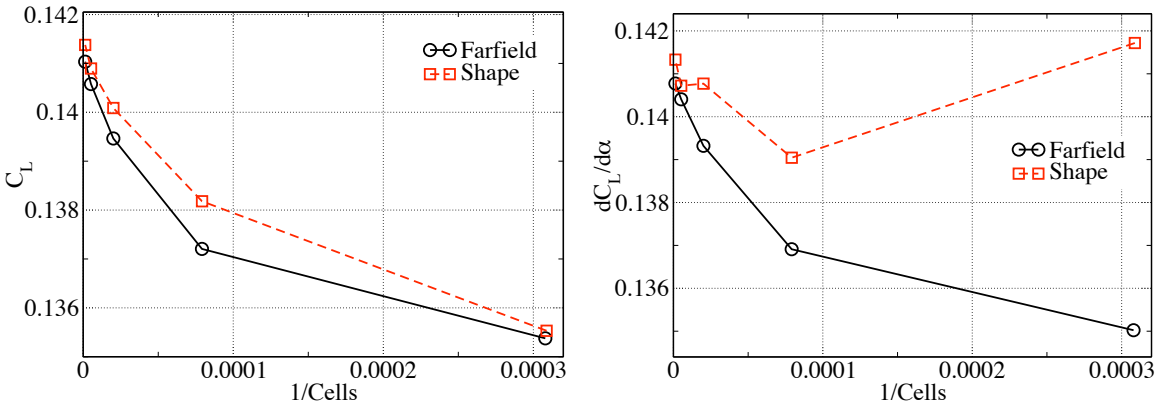


Figure 7. Differences in cut-cells near the leading edge due to a 1° rotation about the trailing edge for the NACA 0012 airfoil.



(a) Lift convergence (b) Gradient of lift with respect to angle of attack

Figure 8. Convergence study of lift and its gradient due to a change in the angle of attack: implemented via the farfield boundary ("Farfield") and rigid-body rotation ("Shape").

VI. Optimization Examples

We present two design examples to demonstrate the effectiveness of the proposed method for aerodynamic shape optimization problems. We use the adjoint method for all gradient computations and we verify the accuracy of the gradient using finite-difference approximations as described in Ref. 42. All parallel computations are performed on Intel® 1.5 GHz IA-64 Itanium 2 processors.

A. Lift-Constrained Drag Minimization

The first design example is a classic airfoil optimization problem in transonic flow. The goal of the optimization is to minimize drag at fixed lift. The problem setup is based on a test case presented in Ref. 42. The freestream Mach number is 0.7 and the initial angle of attack is 3 deg. The computational mesh contains roughly 40,000 cells. Convergence of the flow and adjoint equations is achieved using 64 processors and a 4-level W-cycle multigrid with one pre- and one post-smoothing pass. The second-order accurate spatial discretization uses the van Leer limiter. The objective function is a weighted sum of the lift and drag functionals

$$\mathcal{J} = \omega_{\mathrm{L}} \left(1 - \frac{C_{\mathrm{L}}}{C_{\mathrm{L}}^*}\right)^2 + \omega_{\mathrm{D}} \left(1 - \frac{C_{\mathrm{D}}}{C_{\mathrm{D}}^*}\right)^2 \tag{18}$$

where C_{D}^* and C_{L}^* represent the target drag and lift coefficients, respectively. The target lift coefficient is set to 0.551, which is the lift coefficient for the initial shape and flow conditions, and the target drag coefficient is set to 0.0015, which represents a six-fold reduction in drag from the initial conditions. The weights ω_{L} and ω_{D} are user specified constants set to 1.0 and 0.005, respectively. The initial airfoil is a B-spline approximation of the NACA 0012 airfoil. The B-spline is constructed using 17 control points and we use the vertical position of 2 control points as design variables, as shown in Fig. 9 (points "DV1" and "DV2"). An additional design variable is the angle of attack.

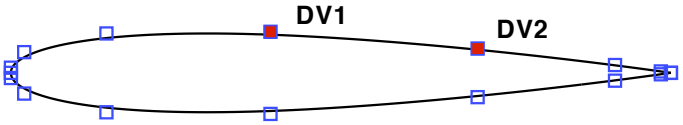


Figure 9. B-spline control points (open symbols) and design variables (shaded symbols) for the initial airfoil shape.

Table 1 compares the adjoint and centered-difference gradients on the first iteration of the optimizer. The agreement is good for the first design variable, as well as the angle of attack. For the second design variable, we find that the centered-difference gradient is highly sensitive to the stepsize, primarily due to incomplete

convergence of the flow solution caused by limiter oscillations. An additional source of error affecting gradient accuracy is the constant limiter assumption in the linearization of the discrete flow equations. Figure 10 shows the convergence of the objective function and gradient. The objective function is converged in 12 design iterations and the L_2 norm of the gradient is reduced by almost 4 orders of magnitude. This indicates that the gradient computation provides sufficiently accurate search directions for the optimization. In terms of wall-clock time, each design iteration requires approximately five minutes. Figure 11 shows the convergence of the lift and drag coefficients. A four-fold reduction in the drag coefficient is achieved within 10 design iterations while maintaining the initial lift coefficient. Figure 12 shows the initial and final pressure distributions and airfoil shapes. The angle of attack is reduced to 2.25 deg. and we obtain a shock-free airfoil very close to the design presented in Ref. 42.

Table 1. Gradient accuracy on first design iteration (lift-constrained drag minimization problem).

Design Variable	Adjoint	Finite-Difference
1	-12.242	-12.389
2	0.5069	0.9055
α	0.3958	0.3848

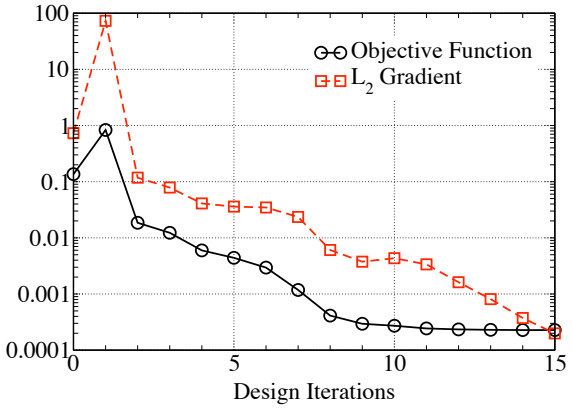


Figure 10. Convergence of objective function and gradient for the lift-constrained drag minimization problem (three design variables).

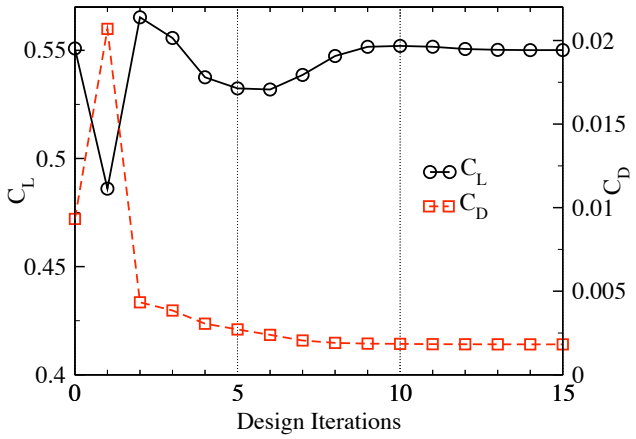


Figure 11. Convergence of lift and drag coefficients for the lift-constrained drag minimization problem.

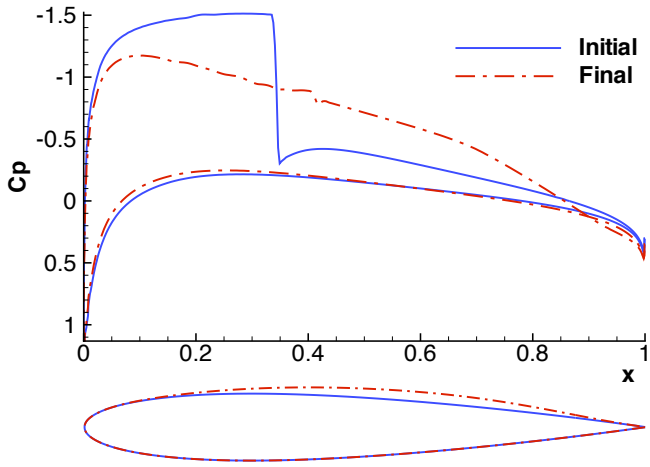


Figure 12. Pressure distribution and airfoil shapes for the lift-constrained drag minimization problem.

B. Reentry Capsule

The second design example targets the optimization of a heat-shield shape for a reentry capsule. The objective of the optimization is to enhance the lift-to-drag ratio, L/D, of the capsule, thereby improving trajectory control for landing-site selection, and reducing the reentry load factor and heat rates. Stability characteristics of the capsule, i.e., trim and negative C_{m_α} slope, are not considered in the present problem, but they could be included by introducing additional penalty terms in the objective function. The purpose of this example is to investigate the robustness of the adjoint method for complex flow problems and demonstrate its effectiveness for CAD-based design.

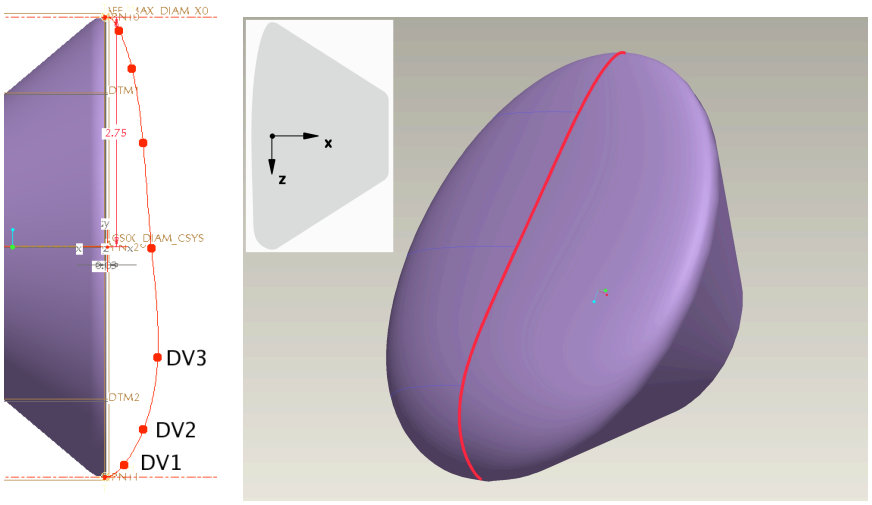
The capsule configuration is shown in Fig. 13 and is derived from the Apollo capsule shape.[43] The Pro/ENGINEER® Wildfire CAD system is used to create the geometry model. The heat-shield is modeled as a two-directional blended surface. Its shape is controlled by the center-line (red line in Fig. 13(b)) and surface tangency conditions at the shoulder. The design variables are associated with the three spline points indicated in Fig. 13(a), each having a single degree-of-freedom in the horizontal direction. Geometry manipulation and surface triangulation of the CAD model is performed via CAPRI and the computation of shape sensitivities for the triangulation, i.e., the term $\partial\vec{T}/\partial X$ in Eq. 12, is accomplished using finite-difference approximations.[30]

We consider a single-point optimization problem at a freestream Mach number of 10 and an angle of attack of 156.5 deg., which represents the trim orientation of the capsule. The angle of attack is measured clockwise from the positive x-axis indicated in Fig. 13. High-temperature effects are approximated by the use of an "effective" ratio of specific heats, γ. We use $\gamma = 1.3$. The initial (symmetric) capsule generates an L/D of 0.37, which is attained using a center-of-gravity offset given by $x/d = 0.124$, $y/d = 0$, and $z/d = 0.03698$, where d represents the capsule diameter set to 5.5m. The target value of L/D is set to 0.4. The objective function is given by

$$\mathcal{J} = \left(\frac{L}{D} - 0.4\right)^2 \tag{19}$$

The volume mesh for the optimization contains roughly 665,000 cells. For the solution of the flow and adjoint equations, we use 64 processors, a 4-level mesh-sequencing startup, and a CFL number of 1.2. The spatial discretization uses a more diffusive *minmod* limiter than the previous transonic example.

The results are presented in Figs. 14 and 15. Pressure contours of the flow solution for the initial capsule shape are shown in Fig. 15(a). The main flow features include a strong bow shock and an unsteady wake behind the capsule. As expected, fluctuations in such an energetic flowfield caused some stability problems for the adjoint solver. However, reducing the spatial discretization to first-order in the cut-cells suppressed this behavior and did not significantly impact the values of the aerodynamic coefficients. With these settings, we obtained a residual reduction of approximately three orders of magnitude for the flow and adjoint equations at each design iteration. Similar issues in high Mach number flow have been confronted in Ref. 19, and this



(a) Spline design variables of heat-shield centerline

(b) Solid model and coordinate system

Figure 13. Parametric-CAD model of capsule shape.

is an area of ongoing work as small scale unsteadiness is common in engineering problems.[44]

Gradient accuracy for the initial iteration of the optimizer is shown in Table 2. The agreement between the central-difference and adjoint gradients is good. Convergence of the optimization problem is shown in Fig. 14, where the target L/D is reached within five design iterations. The L_2 norm of the gradient is reduced by roughly four orders of magnitude. The initial and final heat-shield shapes are shown in Fig. 15, where the shape modifications are relatively minor yet the improvement in L/D is roughly 8%. Larger gains in L/D can be achieved, but additional constraints are required to avoid shapes with concave regions.

The final analysis of the initial and optimized shapes shown in Fig. 15 is performed on a finer mesh containing roughly 1.3 million cells. We use this mesh to provide timing statistics for the computations of mesh sensitivities. The serial algorithm used for mesh generation requires 33.9 seconds to build the mesh. This mesh contains 25,204 cut-cells and intersects a triangulation containing 54,660 triangles. The linearization of the cut-cells adds only *one second* to the execution of the mesh generator for this case. Overall, the wall-clock time per design iteration is approximately 11 minutes on the coarser mesh used in the optimization. This time includes the regeneration and triangulation of the part by CAPRI and the CAD system, as well as the flow solution and adjoint gradient computation. We emphasize that for problems with more design variables, the design-cycle time would remain essentially constant.

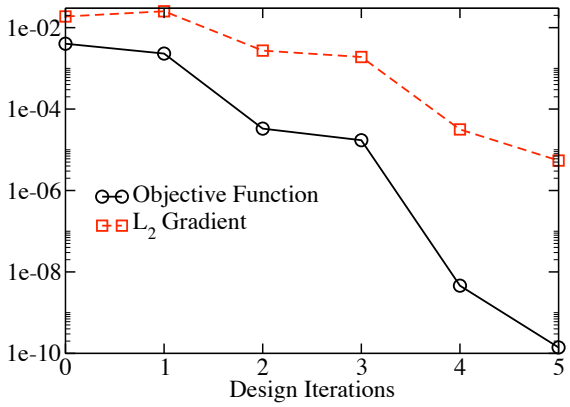
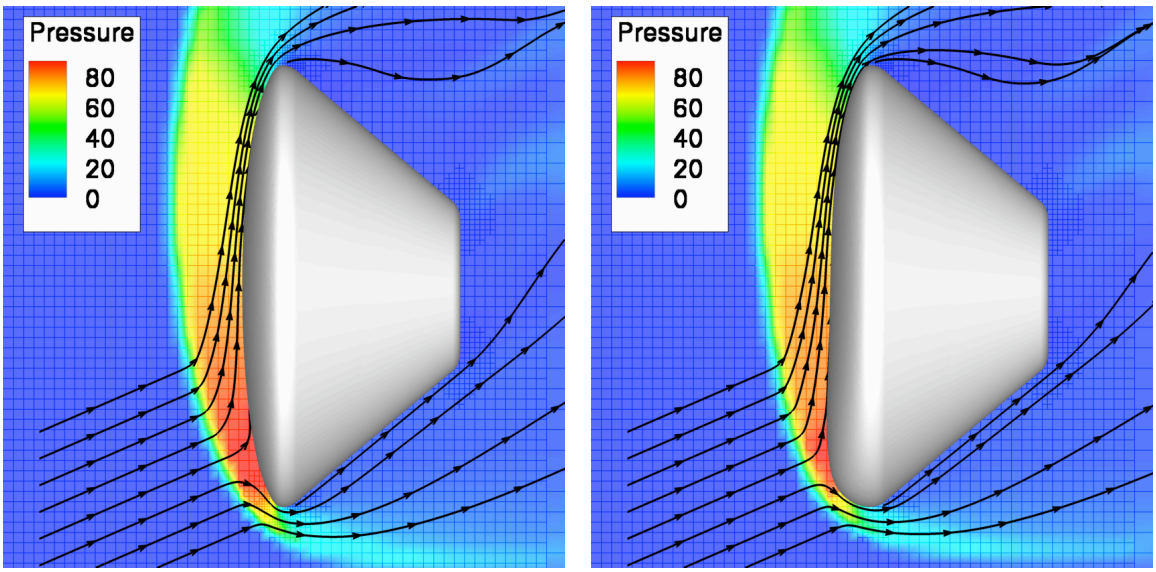


Figure 14. Capsule L/D optimization convergence history.

Table 2. Gradient accuracy on first design iteration (L/D enhancement problem).

Design Variable	Adjoint	Finite-Difference $\epsilon = 0.002$
1	-0.01266	-0.01090
2	-0.01280	-0.01125
3	-0.00538	-0.00566



(a) Initial capsule shape, L/D = 0.37 (b) Optimized capsule shape, L/D = 0.4

Figure 15. Capsule shapes and nondimensional pressure contours on symmetry plane for the optimization of L/D **($M_\infty = 10$, $\alpha = 156.5^\circ$, $\gamma = 1.3$).**

VII. Conclusions and Future Work

A discrete approach for the computation of objective function gradients on embedded-boundary Cartesian meshes has been presented. The effect of shape perturbations in the governing equations was included via a direct linearization of the cut-cell geometric constructors within the mesh generator. The verification studies show that for design variables that do not alter the boundary shape, the error in the gradient value decreases at a rate of $\mathcal{O}(h^2)$. For shape design variables, the convergence rate is reduced to $\mathcal{O}(h)$, which is a consequence of confining the mesh-perturbations to the cut-cells. In both the two- and three-dimensional design examples considered, the new approach provided a reliable approximation of the gradient. The optimizations converged to significantly improved designs, while the L_2 norm of the gradient was reduced by several orders of magnitude. The new approach is well-suited for conceptual design studies and trade-off evaluations where short cycle times are paramount, and complex geometry with many design variables is common.

Future work will focus on applications and testing the performance of the new approach on a broader range of problems. This includes robustness improvements to the adjoint and flow-sensitivity solvers for high Mach number flow, the computation of shape sensitivities near flow singularities, and the inclusion of geometric constraints directly from the CAD system. Furthermore, configurations with multiple intersecting components will be considered.

VIII. Acknowledgments

The authors gratefully acknowledge Robert Haimes (MIT) for his assistance with CAPRI. The authors would also like to thank Scott Murman (ELORET Corp.) for helpful discussions, and Veronica Hawke (ELORET Corp.) for providing the initial CAD model of the capsule. This work was supported by the NASA Ames Research Center contract NNA05BF35C.

References

[1]Pironneau, O., *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, New York, USA, 1984.

[2]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260, Also ICASE report 88–64.

[3]Baysal, O. and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681–688.

[4]Young, D. P., Melvin, R. G., Bieterman, M. B., Johnson, F. T., and Samant, S. S., "A Locally Refined Rectangular Grid Finite Element Method: Application to Computational Fluid Dynamics and Computational Physics," *Journal of Computational Physics*, Vol. 92, No. 1, 1991, pp. 1–66.

[5]Melton, J. E., *Automated Three-Dimensional Cartesian Grid Generation and Euler Flow Solutions for Arbitrary Geometries*, Ph.D. thesis, University of California Davis, 1996.

[6]Karman Jr., S. L., "SPLITFLOW: A 3D Unstructured Cartesian/Prismatic Grid CFD Code for Complex Geometries," AIAA Paper 95–0343, Jan. 1995.

[7]Charlton, E. F. and Powell, K. G., "An Octree Solution to Conservation-laws over Arbitrary Regions (OSCAR)," AIAA Paper 97–0198, Jan. 1997.

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

[9]Lahur, P. R., "Automatic Hexahedra Grid Generation Method for Component-based Surface Geometry," AIAA Paper 2005–5242, Toronto, ON, June 2005.

[10]Mohammadi, B. and Pironneau, O., *Applied Shape Optimization for Fluids*, Oxford University Press, New York, USA, 2001.

[11]Reuther, J. J., *Aerodynamic Shape Optimization Using Control Theory*, Ph.D. thesis, University of California Davis, 1996.

[12]Gunzburger, M. D., "Introduction into Mathematical Aspects of Flow Control and Optimization," *Inverse Design and Optimization Methods*, 1997-05, von Karman Institute For Fluid Dynamics, Brussels, Belgium, April 1997.

[13]Elliott, J. and Peraire, J., "Progress Towards a 3D Aerodynamic Shape Optimization Tool for the Compressible High-Re Navier Stokes Equations Discretized on Unstructured Meshes," AIAA Paper 98–2897, June 1998.

[14]Giles, M. B. and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, No. 3/4, 2000, pp. 393–415.

[15]Kim, C. S., Kim, C., and Rho, O. H., "Sensitivity Analysis for the Navier–Stokes Equations with Two-Equation Turbulence Models," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 838–845.

[16]Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163.

[17]Zingg, D. W., Nemec, M., and Chisholm, T. T., "A Newton–Krylov Algorithm for Aerodynamic Analysis and Design," *Proceedings of the Second International Conference on Computational Fluid Dynamics*, edited by S. Armfield, P. Morgan, and K. Srinivas, ICCFD 2, Springer-Verlag, 2003, pp. 3–18.

[18]Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multi-Point and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.

[19]Nielsen, E. J. and Kleb, B., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables," AIAA Paper 2005–0324, Reno, NV, Jan. 2005.

[20]Mavriplis, D. J., "Formulation and Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes," *AIAA Journal*, Vol. 44, No. 1, 2006, pp. 42–50.

[21]Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Computers & Fluids*, Vol. 28, 1999, pp. 443–480.

[22]Lu, J., *An* a posteriori *Error Control Framework for Adaptive Precision Optimization using Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[23]Dadone, A. and Grossman, B., "Efficient Fluid Dynamic Design Optimization Using Cartesian Grids," AIAA Paper 2003–3959, Orlando, FL, June 2003.

[24]Jameson, A. and Kim, S., "Reduction of the Adjoint Gradient Formula in the Continuous Limit," AIAA Paper 2003–0040, Jan. 2003.

[25]Soto, O. and Löhner, R., "On the Computation of Flow Sensitivities from Boundary Integrals," AIAA Paper 2004–0112, Jan. 2004.

[26]Melvin, R. G., Huffman, W. P., Young, D. P., Johnson, F. T., Hilmes, C. L., and Bieterman, M. B., "Recent Progress in Aerodynamic Design Optimization," *International Journal for Numerical Methods in Fluids*, Vol. 30, 1999, pp. 205–216.

[27]Huffman, W. P., Melvin, R. G., Young, D. P., Johnson, F. T., Bussoletti, J. E., Bieterman, M. B., and Hilmes, C. L., "Practical Design and Optimization in Computational Fluid Dynamics," AIAA Paper 93–3111, 1993.

[28]Duvigneau, R. and Pelletier, D., "Evaluation of Nearby Flows by a Shape Sensitivity Equation Method," AIAA Paper 2005–127, Jan. 2005.

[29]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.

[30]Nemec, M. and Aftosmis, M. J., "Adjoint Algorithm for CAD-Based Shape Optimization Using a Cartesian Method," AIAA Paper 2005–4987, Toronto, ON, June 2005.

[31]Dennis Jr., J. E. and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1983.

[32]Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," Lecture notes, von Karman Institute for Fluid Dynamics, Series: 1997-02, Brussels, Belgium, March 1997.

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

[34]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.

[35]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.

[36]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.

[37]Berger, M. J., Aftosmis, M. J., Marshall, D. D., and Murman, S. M., "Performance of a new CFD flow solver using a hybrid programming paradigm," *J. Parallel Distrib. Comput.*, Vol. 65, 2005, pp. 414–423.

[38]Giles, M. B., "On the Use of Runge–Kutta Time-Marching and Multigrid for the Solution of Steady Adjoint equations," Oxford University Computing Laboratory 00/10, June 2000.

[39]Haimes, R. and Follen, G., "Computational Analysis PRogramming Interface," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by Cross, Eiseman, Hauser, Soni, and Thompson, University of Greenwich, 1998.

[40]Haimes, R. and Crawford, C., "Unified Geometry Access for Analysis and Design," Tech. rep., 12th International Meshing Roundtable, Santa Fe, NM, Sept. 2003.

[41]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.

[42]Nemec, M., Aftosmis, M. J., and Pulliam, T. H., "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method," AIAA Paper 2004–0113, Reno, NV, Jan. 2004.

[43]Moseley, Jr., W. C., Graham, R. E., and Hughes, J. E., "Aerodynamic Stability Characteristics of the Apollo Command Module," NASA TN D–4688, Aug. 1968.

[44]Campobasso, M. S. and Giles, M. B., "Computing Linear Harmonic Unsteady Flows in Turbomachines with Complex Iterative Solvers," AIAA Paper 2005–4705, Toronto, ON, June 2005.