Reentry-Vehicle 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

Abstract submitted to the 24th AIAA Applied Aerodynamics Conference
June 5–8, 2006/San Francisco, CA

I. Introduction

ADJOINT solutions of the governing flow equations are becoming increasingly important for the development of efficient analysis and optimization algorithms. A well-known use of the adjoint method is gradient-based shape optimization.[1–7] Given an objective function that defines some measure of performance, such as the lift and drag functionals, its gradient is computed at a cost that is essentially independent of the number of design variables (e.g., geometric parameters that control the shape). Classic aerodynamic applications of gradient-based optimization include the design of cruise configurations for transonic and supersonic flow, as well as the design of high-lift systems.

Cartesian-mesh methods[8–12] are perhaps the most promising approach for addressing the issues of flow solution automation for aerodynamic design problems. In these methods, the discretization of the wetted surface is decoupled from that of the volume mesh. This not only enables fast and robust mesh generation for geometry of arbitrary complexity,[13,14] but also facilitates access to geometry modeling and manipulation using parametric computer-aided design (CAD). In previous work on Cartesian adjoint solvers, Melvin *et al.*[15] developed an adjoint formulation for the TRANAIR code,[8] which is based on the full-potential equation with viscous corrections. More recently, Dadone and Grossman[16,17] presented an adjoint formulation for the two-dimensional Euler equations using a ghost-cell method to enforce the wall boundary conditions.

In Refs. 18 and 19, we presented an accurate and efficient algorithm for the solution of the adjoint Euler equations discretized on Cartesian meshes with embedded, cut-cell boundaries. Novel aspects of the algorithm were the computation of surface shape sensitivities for triangulations based on parametric-CAD models and the linearization of the coupling between the surface triangulation and the cut-cells. The accuracy of the gradient computation was verified using several three-dimensional test cases, which included design variables such as the freestream parameters and the planform shape of an isolated wing.

The objective of the present work is to extend our adjoint formulation to problems involving general shape changes. Factors under consideration include the computation of mesh sensitivities that provide a reliable approximation of the objective function gradient, as well as the computation of surface shape sensitivities based on a direct-CAD interface. We present detailed gradient verification studies and then focus on a shape optimization problem for an Apollo-like reentry vehicle. The goal of the optimization is to enhance the lift-to-drag ratio of the capsule by modifying the shape of its heat-shield in conjunction with a center-of-gravity (c.g.) offset. This multipoint and multi-objective optimization problem is used to demonstrate the overall effectivness of the Cartesian adjoint method for addressing the issues of complex aerodynamic design.

*Research Scientist, Applications Branch, Advanced Supercomputing Division, MS T27B; mnemec@mail.arc.nasa.gov. Member AIAA.

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

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 steady-state three-dimensional Euler equations within a feasible region of the design space Ω

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

which implicitly defines $Q = f(X)$. Our goal is to compute a reliable approximation to the objective function gradient $\mathrm{d}\mathcal{J}/\mathrm{d}X$. We proceed by using the discrete adjoint method, where the governing equations, Eqs. 1 and 2, are first discretized and then differentiated.

A. Discrete Flow Equations and Solution Method

The discretization of the Euler equations uses a second-order accurate finite-volume method on a multilevel Cartesian mesh with embedded boundaries. The mesh consists of regular hexahedral cells, except for a layer of body-intersecting cells, or cut-cells, that are arbitrary polyhedra adjacent to the boundaries. 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 spatial discretization uses the flux-vector splitting approach of van Leer.[20] The boundary conditions are enforced weakly by appropriate modifications of the reconstructed state and boundary flux. The resulting discrete system of equations is given by

$$\vec{R}(\vec{Q}, \vec{M}, X) = 0 \tag{3}$$

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.*[14,21,22] and Berger *et al.*[23,24]

Note that design variables appear directly in Eq. 3 only when they involve parameters that do not change the computational domain, such as the Mach number, angle of incidence, and side-slip angle. The influence of shape design variables on the residuals is implicit via the computational mesh

$$\vec{M} = f[\vec{T}(X)] \tag{4}$$

where $\vec{T}$ denotes a triangulation for the surface model. The functional dependence of the triangulation on the design variables is determined by the geometry parameterization scheme and is explained further in Sec. III.

B. Discrete Adjoint Method

Combining Eqs. 3 and 4 and differentiating about a steady-state solution $\vec{Q}$ gives the gradient of the discrete objective function $\mathcal{J}(X, \vec{M}, \vec{Q})$

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X} = \frac{\partial \mathcal{J}}{\partial X} + \underbrace{\frac{\partial \mathcal{J}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X}}_{\mathrm{A}} - \vec{\psi}^{\mathrm{T}} \left(\frac{\partial \vec{R}}{\partial X} + \underbrace{\frac{\partial \vec{R}}{\partial \vec{M}} \frac{\partial \vec{M}}{\partial \vec{T}} \frac{\partial \vec{T}}{\partial X}}_{\mathrm{B}} \right) \tag{5}$$

where the vector $\vec{\psi}$ represents adjoint variables given by the adjoint equation

$$\frac{\partial \vec{R}}{\partial \vec{Q}}^{\mathrm{T}} \vec{\psi} = \frac{\partial \mathcal{J}}{\partial \vec{Q}}^{\mathrm{T}} \tag{6}$$

We emphasize that the adjoint equation is independent of the design variables, i.e., the partial derivatives with respect to the flow variables are evaluated at constant X. Hence, for problems with many design variables and one objective function, the use of the adjoint method is attractive since only one solution of the large linear system of Eq. 6 is required.

The solution algorithm for the adjoint equation 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,[25] such that the asymptotic convergence rate of the discrete adjoint is identical to that of the flow solver. The matrix-vector products associated with the flow-Jacobian matrix, left side of Eq. 6, 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. 6, 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 solver are roughly equivalent to the flow solver. For further details on the adjoint formulation and solution procedure see Ref. 18. The computation of partial derivative terms $\partial \mathcal{J}/\partial X$ and $\partial \vec{R}/\partial X$ in Eq. 5 is straightforward, since these terms do not involve derivatives of the surface shape. The remaining partial derivative terms in Eq. 5, 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. An accurate computation of these terms is presented in Ref. 19 and an overview is given in the following section.

III. Objective and Residual Shape Sensitivities

The differentiation of the objective function and residual equations with respect to shape design variables involves the interaction of the surface deformations with the triangulation and the volume mesh. An advantage of our Cartesian approach is that the computation of the surface shape sensitivities, i.e., the term $\partial \vec{T}/\partial X$ in Eq. 5, is cleanly isolated from the residual equations. This is because there is no prescribed connectivity between the volume mesh and the triangulation, which is particularly well-suited for CAD-based geometry control. In subsection A, we outline our approach to geometry manipulation and surface triangulation using a direct CAD interface, and then explain the computation of surface shape sensitivities. In subsection B, we address the binding of the surface to the volume mesh to obtain sensitivities in the cut-cell layer.

A. Shape Sensitivity of Surface Triangulation

Our primary method for surface modeling, control, and triangulation is based on the Computational Analysis and PRogramming Interface (CAPRI) developed by Haimes *et al.*[26,27] CAPRI provides a unified and direct access to most parametric CAD systems. The ability to control CAD solid models is accomplished by exposing the master-model feature tree of parametric parts and assemblies. Design variables can be associated directly with the exposed parameters, or indirectly with splines, such as airfoil sections that are lofted to define a wing.

Upon a successful regeneration of the CAD-model, CAPRI provides an associated water-tight triangulation for each component. The Cartesian position of each vertex in the triangulation, for each face of the CAD model, is given by its parametric values

$$(x, y, z) = f(u, v) \tag{7}$$

Since analytic differentiation of the geometry constructors within the CAD system is presently not possible, we rely on a centered-difference approximation to determine the derivative of each vertex of the triangulation with respect to the design variables. The first step of this procedure involves the regeneration and triangulation of the CAD model to reflect the current values of the design variables, thereby obtaining a baseline model. For each CAD face, we normalize the (u, v) vertex values such that $u, v \in [0, 1]$. The next step involves two additional model regenerations and triangulations for each design variable, which correspond to the plus and minus perturbations. To form the finite-difference approximation, we interrogate the perturbed models using the normalized (u, v) values re-scaled by their new (u, v) range. Hence, we obtain a one-to-one correspondence between vertex coordinates in the physical space for the baseline model and its perturbations. A tacit assumption in this procedure is that the face topology of the baseline and perturbed geometries remains the same. We check this condition after each regeneration of the model and attempt to reduce the stepsize or use one-sided differences if necessary.

The finite-difference procedure is efficient since we query the triangulation in the parameter space, i.e., we only require forward solutions of Eq. 7. An alternate approach is the use of "snap" operations in the physical space to locate nearest vertex coordinates on the baseline and perturbed surfaces. This procedure is less efficient since it involves an inverse solution of Eq. 7. Furthermore, due to internal tolerances of the CAD system, snap operations may introduce excessive noise into the evaluation of the derivative.

B. Sensitivity of Cut-Cells to Shape Deformations

The cut-cell generation procedure involves computing the intersection of the surface triangulation with the faces of Cartesian hexahedra. Details of this procedure are described in Ref. 13. The sensitivity of a cut-cell to a shape deformation is obtained by a linearization of a geometry constructor that closely approximates the mesh generation process. We approximate the cut-cell geometry using the intersection of surface triangles with only the Cartesian edges and neglect contributions from the faces. This is illustrated in Fig. 1, where a Cartesian edge $\overline{\mathbf{de}}$ of a hexahedron intersects triangle $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, creating a pierce point $\mathbf{P}$. We use bold type to represent Cartesian position vectors.

The linearization of the pierce point location with respect to a shape perturbation could be obtained directly from the CAD system using the approach described in the previous section. This would involve the use of "snap" operations to locate the pierce point in the (u, v) space of the appropriate CAD face. Alternatively, the pierce point location can be defined using a set of parametric equations for the triangle $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ and the line segment $\overline{\mathbf{de}}$, which is the approach we pursue in this work. The location of $\mathbf{P}$ along the segment $\overline{\mathbf{de}}$ is given by

$$\mathbf{P} = \mathbf{d} + s^* \cdot \mathbf{D} \tag{8}$$

where $\mathbf{D} = \mathbf{e} - \mathbf{d}$ and $s^* = f(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e})$ represents the parametric value of the intersection point. The expression for s^* is derived in Ref. 13. The linearization of the pierce point is given by

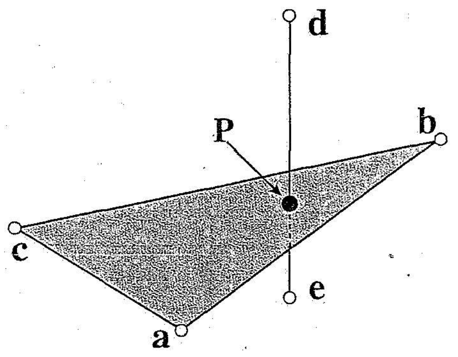


Figure 1. Intersection of a triangle with a Cartesian edge defining a pierce point P

$$\frac{\partial \mathbf{P}}{\partial X} = \frac{\partial s^*}{\partial X}\mathbf{D} \tag{9}$$

Note that only one component of $\mathbf{D}$ is non-zero, since the Cartesian mesh points are coordinate aligned. Furthermore, the points $\mathbf{d}$ and $\mathbf{e}$ are independent of X, i.e., the Cartesian mesh is rigid, and consequently the linearization of s^* is given by

$$\frac{\partial s^*}{\partial X} = \frac{\partial s^*}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial X} + \frac{\partial s^*}{\partial \mathbf{b}}\frac{\partial \mathbf{b}}{\partial X} + \frac{\partial s^*}{\partial \mathbf{c}}\frac{\partial \mathbf{c}}{\partial X} \tag{10}$$

This linearization represents the change in the location of $\mathbf{P}$ along the edge $\overline{\mathbf{de}}$ exclusively as a function of shape sensitivities at the vertices $(\mathbf{a}, \mathbf{b}, \mathbf{c})$. The decoupling of the surface sensitivities from the volume mesh provides a straightforward interface to any geometry parametrization tool.

The linearization of the pierce points is used to determine all the required geometric sensitivities in the linearization of the residual equations. First-order spatial discretization requires the derivative of the face areas and normals, while for second-order discretization, we require additional linearizations for the location of face centroids, volume centroids, and the reconstruction distances used in the evaluation of flow-solution gradients. As a result, a perturbation of the surface shape influences the residuals not only in the cut-cells, but also in the first and second nearest neighbouring cells.

The computation of the surface normal is based on a planar approximation to the variation of the triangulation within each cut-cell. This agglomerated normal vector is computed using the divergence theorem, which requires the geometric closure of each cut-cell

$$\sum_{j \in V_i} \mathbf{n}_j = \sum_{j \in V_i} A_j \cdot \hat{\mathbf{n}}_j = 0 \tag{11}$$

where $\mathbf{n}$ denotes an area scaled normal vector. The sum is performed over the Cartesian faces of the cell, which must equal the agglomerated normal vector. Since geometric closure of a cut-cell should be satisfied for any shape perturbation, we linearize Eq. 11 to obtain

$$\sum_{j\in V_i} \frac{\partial}{\partial X}(A_j \cdot \hat{\mathbf{n}}_j) = 0 \tag{12}$$

The derivative of the Cartesian face areas is computed based on the linearization of the pierce points. The pierce points, in conjunction with the corner points of the cut-cell, form polygons on each Cartesian face. The areas of the polygons are computed by subdividing each polygon into triangles. Central to this procedure is the linearization of an area of a triangle given by the cross-product of two vectors.

The final paper will include a discussion of the computation of volume mesh sensitivities used to obtain accurate gradients.

IV. Results and Discussion

We present several three-dimensional test cases to verify the accuracy of the cut-cell linearization and gradient computations. The Pro/ENGINEER® Wildfire CAD system is used to create all geometry models.

A. NACA 0012 Wing

This test case is used to verify gradient accuracy and involves an isolated wing at transonic flow conditions, $M_\infty = 0.84$ and $\alpha = 3.06°$. The wing geometry is constructed from a generic CAD model with linearly lofted NACA 0012 root and tip sections, a taper ratio of 0.7, aspect ratio of 11, sweep of 15°, and root and tip-twist angles of 3° and −1.5°, respectively. The volume mesh contains 499,716 cells and uses a symmetry plane. Convergence to steady-state is achieved using 64 processors[a], a 4-level W-cycle multigrid with one pre- and one post-smoothing pass, and a CFL number of 1.2. Partial updates of the flow gradients are also used, i.e., the flow gradients are updated only on the first stage of the RK5 scheme. The second-order accurate spatial discretization uses the van Leer limiter. The Mach number contours of the flow solution are shown in Fig. 2.

The objective function represents a lift-constrained drag minimization problem

$$\mathcal{J} = \left(1 - \frac{C_L}{C_L^*}\right)^2 + 0.01\left(1 - \frac{C_D}{C_D^*}\right)^2 \tag{13}$$

where $C_L^* = 0.327$ denotes the initial lift coefficient and $C_D^* = 0.01$. The initial drag coefficient is 0.0487. The design variables are the angle of incidence and the tip-twist angle of the wing.

Convergence of the flow and adjoint equations are shown in Fig. 3. Approximately the first 50 multigrid cycles of the flow solution and the first 75 cycles of the adjoint solution correspond to full-multigrid startup. As shown in Fig. 3(b), the startup time is almost negligible. Furthermore, the convergence characteristics of both solvers are quite similar, with a reduction in residual of over six orders of magnitude in 60 seconds. Note that the flow solver takes advantage of specific compiler directives for code optimization. These directives are not yet implemented in the adjoint code.

Objective function gradient accuracy is presented in Table 1 for both design variables. The agreement between the adjoint and centered-difference gradient values is good. The small differences are attributed to the constant limiter assumption in the linearization of the discrete flow equations, as well as the dependence of the finite-difference gradient on the stepsize due to non-smooth changes in the shock location. The interpretation of the descent direction, which is the negative of the gradient vector, reflects the fact that the objective function is dominated by the drag term. Hence, a reduction in the angle of attack and an increase in the washout of the wing should lower the shock strength and therefore reduce the drag penalty.

The final paper will include additional design variables, as well as an optimization example for this 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 L/D of the capsule, thereby improving its trajectory control and landing-site selection, as well as reducing the reentry load factor and heat rates. The nominal reentry

[a]Intel® 1.5 GHz IA-64 Itanium 2 processor

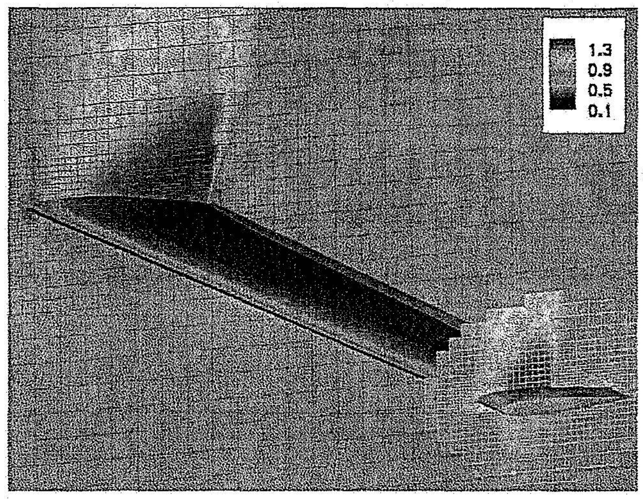
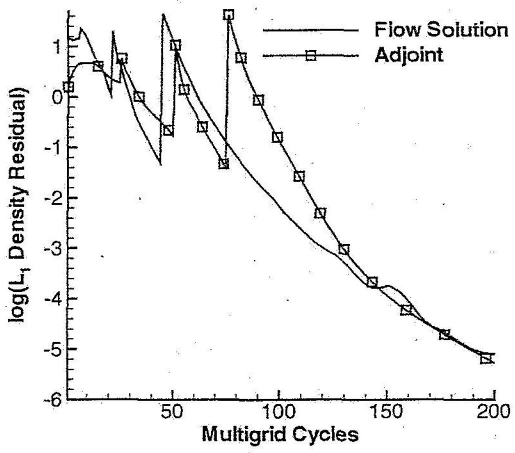
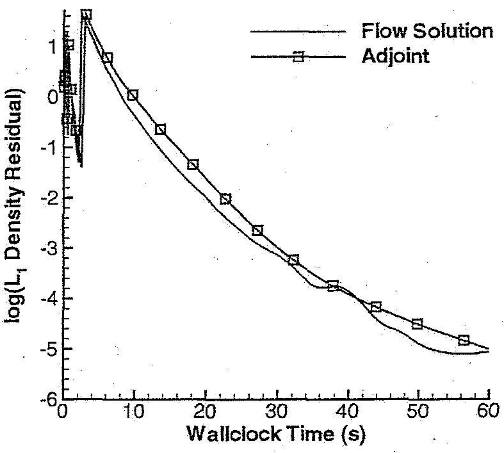


Figure 2. Mach contours for the NACA 0012 Wing ($M_\infty = 0.84$, $\alpha = 3.06°$)



(a) Multigrid Cycles (4-level multigrid with full-multigrid startup)



(b) Wallclock Time (64 Intel® 1.5 GHz Itanium 2 processors, mesh size 499,716 cells)

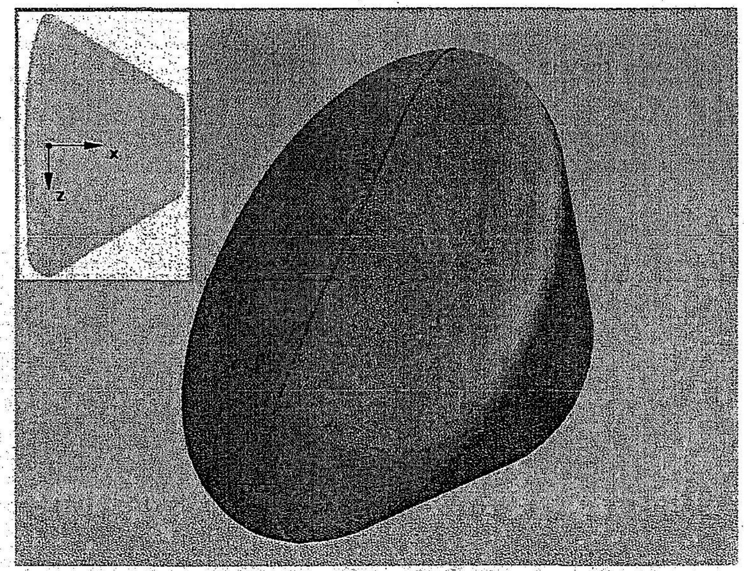
Figure 3. Convergence histories for the NACA 0012 Wing ($M_\infty = 0.84$, $\alpha = 3.06°$)

Table 1. Gradient accuracy for the NACA 0012 Wing

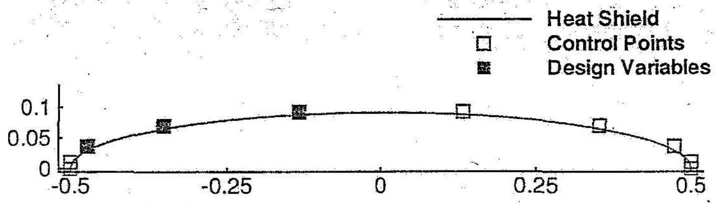
Design Variable	Adjoint	Finite-Difference
α	0.172276	0.177888
Tip Twist	0.062750	0.063810

trajectory assumes a constant L/D, which leads to a multipoint optimization problem. The pitch stability of the capsule, i.e., trim and negative C_{m_α} slope, is ensured by introducing additional penalty terms in the objective function.

The baseline configuration is shown in Fig. 4(a) and is derived from the Apollo capsule shape.[28] The heat-shield is modeled as a two-directional blended surface. Its shape is controlled by the center-line (red line in Fig. 4(a)) and surface tangency conditions at the shoulder. The center-line is parameterized using a cubic B-spline, as shown in Fig. 4(b). The design variables are associated with three control points indicated in Fig. 4(b), each having a single degree-of-freedom in the vertical direction. Additional structural constraints, such as heat-shield length, convex shape, and radius of curvature, can be introduced to account for aerothermal factors.



(a) Solid model and coordinate system



(b) Shape parameterization of heat shield center-line (normalized by capsule diameter $d = 5.5$m)

Figure 4. Parametric-CAD model of capsule shape

We consider a two-point optimization problem where the design Mach numbers are 10 and 25. The corresponding values of γ, the ratio of specific heats, are 1.231 and 1.125, respectively. The target value of L/D at each design point is set to 0.4. This value is based on the aerodynamic characteristics of the Apollo capsule, which attained an L/D of 0.3 using a c.g. offset. We use a similar c.g. offset in the present study, 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 objective function is given by

$$\mathcal{J} = \left[0.1\left(\frac{L}{D} - 0.4\right)^2 + C_m^2\right]_{M=10} + \left[0.1\left(\frac{L}{D} - 0.4\right)^2 + C_m^2\right]_{M=25} \quad (14)$$

The angle of incidence at each design point is used to enforce the pitching-moment constraint, resulting in a total of five design variables. The gradient is computed using finite-differences.

Figures 5 and 6 shown the baseline and optimal designs at the two operating conditions. The shape modifications required to achieve the target performance are relatively minor. The objective function is satisfied within 15 design iterations and the gradient is reduced by roughly 2.5 orders of magnitude, as shown in Fig. 7. Figure 8 shows that value of L/D improved not only for the design Mach numbers, but also at off-design points. The variations in L/D for the final design, as shown in Fig. 8, could be reduced by introducing an additional operating condition in the optimization problem.

In the final paper, the adjoint method will be used to compute the gradient. We will also discuss the pitch slope constraint and investigate c.g. sensitivities. Additional aerodynamic coefficients will be presented and discussed.

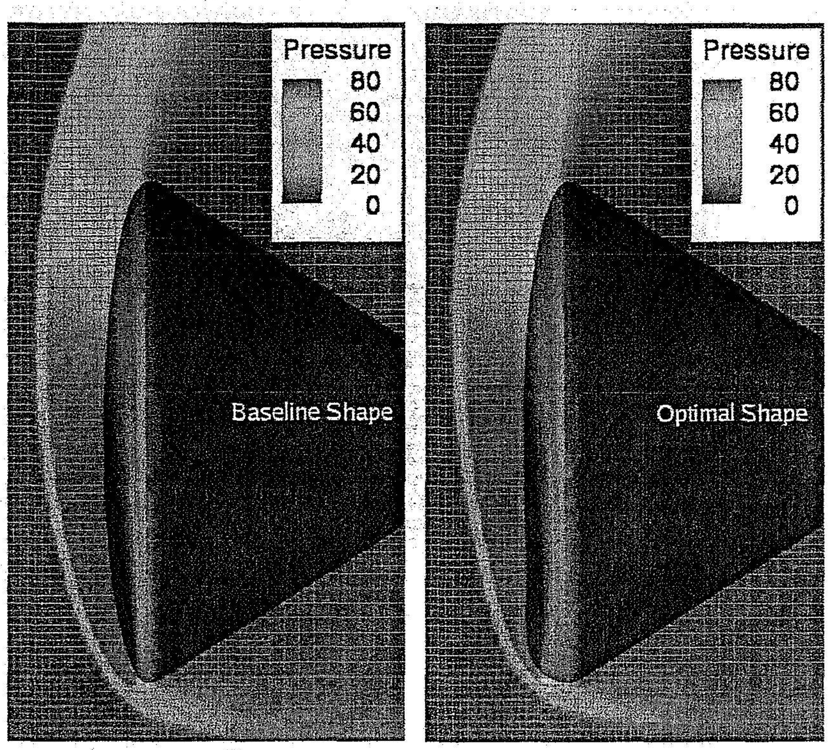


Figure 5. Pressure contours at $M = 10$ (Baseline $\alpha_{\text{trim}} = 156.1$ deg., Optimal Shape $\alpha_{\text{trim}} = 155.7$ deg.)

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

[5]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.

[6]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.

[7]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.

[8]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.

Page intentionally left blank

Page intentionally left blank