JOURNAL OF COMPUTATIONAL PHYSICS **136,** 366–384 (1997)
ARTICLE NO. CP975743

A PDE Sensitivity Equation Method for Optimal Aerodynamic Design

Jeff Borggaard and John Burns

Interdisciplinary Center for Applied Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0531
E-mail: borggajt@icam.vt.edu and burns@icam.vt.edu

The use of gradient-based optimization algorithms in inverse design is well established as a practical approach to aerodynamic design. A typical procedure uses a simulation scheme to evaluate the objective function (from the approximate states) and its gradient, then passes this information to an optimization algorithm. Once the simulation scheme (CFD flow solver) has been selected and used to provide approximate function evaluations, there are several possible approaches to the problem of computing gradients. One popular method is to differentiate the simulation scheme and compute design sensitivities that are then used to obtain gradients. Although this black-box approach has many advantages in shape optimization problems, one must compute mesh sensitivities in order to compute the design sensitivity. In this paper, we present an alternative approach using the PDE sensitivity equation to develop algorithms for computing gradients. This approach has the advantage that mesh sensitivities need not be computed. Moreover, when it is possible to use the CFD scheme for both the forward problem and the sensitivity equation, then there are computational advantages. An apparent disadvantage of this approach is that it does not always produce consistent derivatives. However, for a proper combination of discretization schemes, one can show *asymptotic consistency* under mesh refinement, which is often sufficient to guarantee convergence of the optimal design algorithm. In particular, we show that when asymptotically consistent schemes are combined with a trust-region optimization algorithm, the resulting optimal design method converges. We denote this approach as the *sensitivity equation method*. The sensitivity equation method is presented, convergence results are given, and the approach is illustrated on two optimal design problems involving shocks. © 1997 Academic Press

1. INTRODUCTION

Optimal design problems consist of selecting design parameters for a system in order to optimize a given design objective, usually constrained to satisfy a partial differential equation. In many of these problems, design parameters describe the shape of an object. Examples of these *shape optimization* problems include drag reduction [24], [25], weight minimization [17], optimal sensor/actuator placement [6], airfoil design [19–22] and the design of wind tunnel elements [18].

Traditionally, approximate solutions of these problems are found by "cut and try" methods, combining a designer's engineering experience with repeated experimental testing. This is often expensive, motivating computational methods which compute the optimal design directly. These methods require defining an objective function and an appropriate PDE model of the states of the system. A comparison of several *optimal design* methods may be found in [13].

While there are examples of shape optimization problems solved using derivative free optimization algorithms (see, e.g., [15]), many popular approaches couple a gradient-based optimization algorithm with function evaluations provided by a proven simulation scheme. One of the disadvantages of these approaches is the expense of computing the gradient. Using finite differences is often too costly, even if appropriate step sizes can be found and the simulation scheme can take advantage of "nearby" solutions (as is the case with iterative solvers for nonlinear equations).

Two strategies for alleviating the computational expense of gradient evaluations are adjoint variables [20] and design sensitivities [17]. Adjoint methods are advantageous when either the problem is self-adjoint or there are a large number of design parameters. However, when there are relatively few design parameters, using design sensitivities, quantities which describe the influence of the design parameters on the states of the system, is an attractive alternative. In addition to efficient gradient computations, they can be used in some problems to construct an effective update of the approximate Hessian for quasi-Newton optimization algorithms, e.g., [10].

A standard approach often used to compute the design sensitivities is based on (implicitly) differentiating the simulation scheme (for the states) with respect to the design variables. Using the chain rule to carry out this calculation results in an efficient numerical scheme for the sensitivities. The efficiency arises from reusing many of the quantities computed in the simulation scheme. In fact, the "inversion" of the system matrix (i.e., the matrix factorization) can often be reused.

A disadvantage of this approach is that for shape optimization problems, the discretization is parameter depen-

0021-9991/97 $25.00

dent. Thus, derivatives of the discretization (mesh sensitivities) are required for each shape parameter. Depending on the simulation scheme used for the states, determining the discretization can require the solution of a partial differential equation (as is the case for finite difference solutions of viscous flow problems [29]). This requires a strategy for computing the mesh sensitivities [23] or for computing an approximation to them [27, 28].

Another approach to finding design sensitivities relies on approximating the partial differential equation, known as the *sensitivity equation*. This equation is obtained by implicitly differentiating the (infinite-dimensional) state equation with respect to each design parameter. As shown in [2], using the same numerical scheme to approximate the sensitivity equation which is used to approximate the states leads to an efficient scheme with similar computational advantages as the design sensitivity approach described above. Furthermore, since the discretization is applied directly to the sensitivity equations, no sensitivity of the mesh is required. The sensitivity equation is always *linear* in the design sensitivity, even if the state equation is nonlinear. Since there is no requirement to use the same numerical scheme, it is possible to gain additional computational savings by using a scheme which takes advantage of the linearity in the sensitivity equations.

An apparent disadvantage of this approach is that it does not compute *consistent derivatives*. In other words, the sensitivity equation approach does not capture the sensitivity of the truncation errors in the scheme. Thus, there is a concern that providing an optimization algorithm with an approximation of the gradient of the infinite-dimensional objective function instead of the gradient of the approximate objective function would cause the algorithm to fail. One might expect, however, that if the gradients are ‘‘close enough’’ to the true gradients, then the optimization algorithm should still converge. We show that this convergence can be established if one combines compatible simulation and optimization schemes.

Trust-region optimization algorithms are constructed to be globally convergent by minimizing a model of the objective function in a region where the model is ‘‘trusted.’’ This leads to robust algorithms capable of handling inaccuracies in the model. In fact, convergence results have been given for these algorithms when the model is based on inaccurate gradient information [7, 8]. The results hold provided the gradients satisfy a given error condition. Therefore, it is natural to consider an optimal design method which couples a trust-region optimization algorithm with gradients computed using the sensitivity equation approach. We denote this combined sensitivity/trust-region algorithm by the *sensitivity equation method* (SEM).

In this work, we present and analyze the sensitivity equation method. The method can be applied to a wide class of optimal design problems, including those mentioned above, however, we focus on the particular example of shape optimization of Euler flows in order to illustrate the method. In Section 2, we describe two design problems. In Section 3, we present the sensitivity equation method including the trust-region algorithm and the use of the sensitivity equation to find the design sensitivities. Furthermore, we compare various numerical approximations of the sensitivity equation with approaches based on the discretized equations. Section 4 discusses a number of convergence issues and includes a convergence theorem for the sensitivity equation method. In Section 5, we use a one-dimensional duct design problem to describe the implementation of the sensitivity equation approach. Finally, we describe the implementation and perform shape optimization for a two-dimensional forebody simulator design problem where the steady-state Euler equations are used to model the state variables.

2. ILLUSTRATIVE EXAMPLES

We present two optimal design problems below which are used to illustrate the sensitivity equation method. These problems consist of determining shape parameters which produce a solution to the Euler equations that matches a desired flow ‘‘as closely as possible.’’ The first problem is motivated by the design of a wind tunnel element in order to produce a desired flow in the test section. We study a two-dimensional analogue of this problem. The second problem consists of prescribing the cross-sectional area of a one-dimensional duct to produce a duct flow which matches a desired flow profile. This problem was used by Frank and Shubin [13] in their study of optimal design.

2.1. *Forebody Simulator Design Problem*

The Arnold Engineering Development Center (AEDC) operates a free-jet test facility which is used for full-scale testing of commercial and military aircraft engines. Engines are evaluated for performance and safety under various free flight conditions. While this facility is large enough to house engines, it is not large enough to house an entire aircraft forebody. Thus, the effect of the aircraft forebody on the engine inlet flow profile must be simulated. One way of doing this is to replace the actual forebody by a smaller object, called a *forebody simulator* (FBS). The use of the FBS is illustrated in Fig. 1. The FBS design problem is to specify the shape of this FBS so that it produces an engine inlet flow profile which is as close to some desired profile as possible [18]. The desired profile can be determined by performing either a wind tunnel simulation or a computational simulation of a model configuration resembling a test condition of the aircraft engine.

In order to demonstrate the applicability of the SEM, we consider a two-dimensional analogue of this problem.

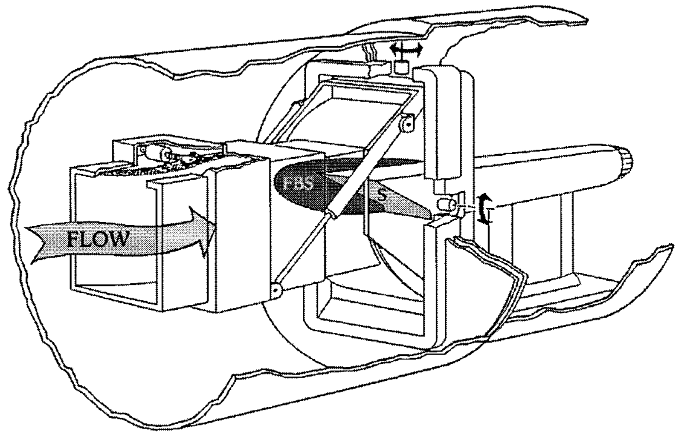


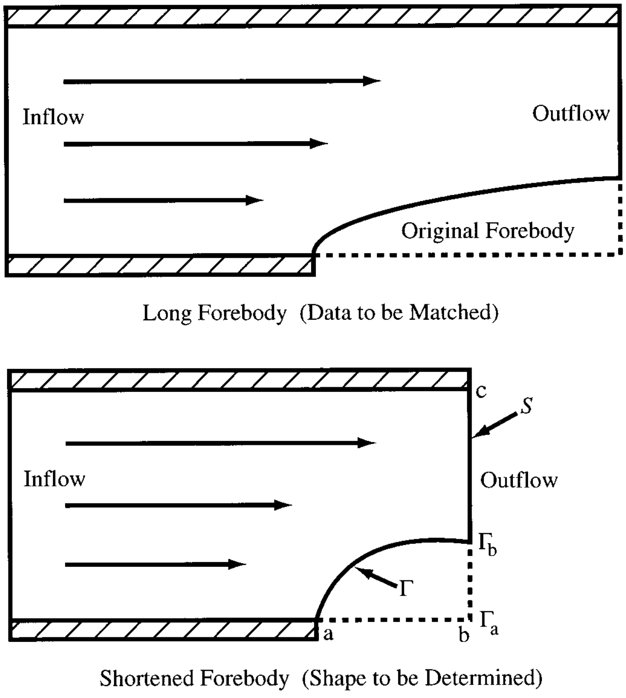
FIG. 1. Forebody simulator design problem.

This problem, depicted in Fig. 2, is to find the shape of the curve Γ, which produces an outflow that matches the outflow generated by the original (longer) forebody as closely as possible. The flow, Q (consisting of the density ρ, the momentum $\rho u \hat{\imath} + \rho v \hat{\jmath}$ and the sum of the internal and kinetic energy E) is modeled using the steady state Euler equations,

$$\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{1}$$

where

$$F = uQ + \begin{bmatrix} 0 \\ P \\ 0 \\ Pu \end{bmatrix}, \qquad G = vQ + \begin{bmatrix} 0 \\ 0 \\ P \\ Pv \end{bmatrix}, \quad \text{and} \quad Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}. \tag{2}$$



Long Forebody (Data to be Matched)

Shortened Forebody (Shape to be Determined)

FIG. 2. 2D forebody simulator design problem.

The pressure P is related to the elements of Q by

$$P = (\gamma - 1)\left[E - \frac{1}{2}\rho(u^2 + v^2)\right], \tag{3}$$

where γ is the ratio of specific heats ($\gamma = 1.4$ for air). Given a forebody simulator shape Γ, the flow $Q(\Gamma)$ is determined by solving the Euler equations (1) in the test cell domain $\Omega(\Gamma)$ subject to the boundary conditions (for supersonic flow)

$$Q = Q_{\text{in}} \quad \text{at the test cell inflow,} \tag{4}$$

$$(u, v) \cdot \hat{n} = 0 \quad \text{and} \tag{5}$$

$$\frac{\partial}{\partial n}((u, v) \cdot \hat{t}) = 0 \quad \text{at the walls (no flow penetration),} \tag{6}$$

where $\hat{n}$ and $\hat{t}$ are the normal and tangential vectors at the boundary, respectively. The set of admissible forebody simulator shapes is

$$\begin{aligned} \mathscr{A} = \{\Gamma \in C^1(a, b) \mid \Gamma(a) = \Gamma_a, \\ \Gamma(b) = \Gamma_b \text{ and } \Gamma(x) \geq \Gamma_a, \forall x \in (a, b)\}. \end{aligned} \tag{7}$$

A statement of the design problem is given below.

Problem 2.1 (Forebody Simulator Design). Let $\hat{Q}$ be a desired flow at the outflow (engine inlet),

$$S = \{(x, y) \mid x = b, \Gamma_b \leq y \leq c\}. \tag{8}$$

Define the objective function

$$\mathscr{J}(\Gamma) = \int_S \|Q(\Gamma) - \hat{Q}\|_2^2 \, dS, \tag{9}$$

where $Q(\Gamma)$ represents the solution of (1) with boundary conditions (4)–(6) in the test cell $\Omega(\Gamma)$. The forebody simulator design problem is to find $\Gamma_* \in \mathscr{A}$ such that

$$\mathscr{J}(\Gamma_*) \leq \mathscr{J}(\Gamma) \quad \text{for all } \Gamma \in \mathscr{A}. \tag{10}$$

Closed-form solutions to (1) with (4)–(6) are available only for special domains. Therefore, we consider approximate solutions of (1) and hence the approximation of Problem 2.1.

The discretization is performed by selecting mesh points in the flow domain $\Omega(\Gamma)$ where to flow variables will be approximated. It is desirable to select this mesh in such a way that the points are more dense in regions where flow gradients are expected to be "large" (in order to have more accurate differencing) and more coarse in regions where the flow is nearly constant (in order to save computer time). Other issues, such as selecting points with no sharp changes in density and with sufficient resolution to treat the boundary conditions, make the mesh generation a science in and of itself (see, e.g., [29]).

Another constraint on the discretization, to simplify the implementation of a finite difference scheme, is to use a regular mesh, i.e., a mesh where there exists a bijective map taking the mesh points to a lattice of points in the computational space. For example, suppose that $\mathscr{M}$ is a C^1 mapping,

$$\mathscr{M}: (x, y) \to (\xi, \eta); \tag{11}$$

then derivatives in the physical space are easily approximated on the lattice using the chain rule. Denoting the Jacobian of the mapping by $J_{\mathscr{M}}$, the transformed Euler equations become

$$\frac{\partial \overline{F}}{\partial \xi} + \frac{\partial \overline{G}}{\partial \eta} = 0, \tag{12}$$

where

$$\overline{F} = U\overline{Q} + PJ_{\mathscr{M}}^{-1} \begin{bmatrix} 0 \\ \dfrac{\partial \xi}{\partial x} \\ \dfrac{\partial \xi}{\partial y} \\ U \end{bmatrix}, \quad \overline{G} = V\overline{Q} + PJ_{\mathscr{M}}^{-1} \begin{bmatrix} 0 \\ \dfrac{\partial \eta}{\partial x} \\ \dfrac{\partial \eta}{\partial y} \\ V \end{bmatrix}, \tag{13}$$

$$\overline{Q} = J_{\mathscr{M}}^{-1} Q, \quad U = \frac{\partial \xi}{\partial x} u + \frac{\partial \xi}{\partial y} v, \quad \text{and} \quad V = \frac{\partial \eta}{\partial x} u + \frac{\partial \eta}{\partial y} v. \tag{14}$$

A standard finite difference scheme developed by Beam and Warming [1] is used to approximate the transformed equations. The scheme introduces a time variable, t, as a means of iterating an initial guess for the solution, to a solution of the steady state equations. Second- and fourth-order artificial dissipation terms are added for stability, represented by $\Psi^{(2)}$ and $\Psi^{(4)}$, respectively. This scheme is implemented in the PARC2D code [9]. Several implementation issues are discussed briefly below which are referred to in later sections. Readers interested in more code details or the actual expressions used for $\Psi^{(2)}$ and $\Psi^{(4)}$ should consult [9].

The difference scheme produces a system of equations for the update of the flow variables, $\Delta\overline{Q}^n$. Thus, the solution at the nth iteration, $\overline{Q}^n$, is determined from

$$\overline{Q}^n = \overline{Q}^{n-1} + \Delta\overline{Q}^{n-1}. \tag{15}$$

The system matrix produced by the approximation above is quite large due to differencing in each direction. However, this problem is circumvented using an *approximate factorization* into a product of two matrices, each corresponding to differencing in one of the lattice directions. The final system has the form

$$\begin{aligned} &[I + \Delta t \delta_\xi \overline{A}^n - \nabla_\xi(\Psi_\xi^{(2)} + \Psi_\xi^{(4)})\, \Delta_\xi J_{\mathscr{M}}] \\ &\quad \times [I + \Delta t \delta_\eta \overline{B}^n - \nabla_\eta(\Psi_\eta^{(2)} + \Psi_\eta^{(4)})\, \Delta_\eta J_{\mathscr{M}}]\, \Delta\overline{Q}^n \\ &= -\Delta t \delta_\xi \overline{F}^n - \Delta t \delta_\eta \overline{G}^n \\ &\quad + \Delta t\, \nabla_\xi\, (\Psi_\xi^{(2)} - \Psi_\xi^{(4)}\, \Delta_\xi\, \nabla_\xi)\, \Delta_\xi\, (J_{\mathscr{M}} \overline{Q}^n) \\ &\quad + \Delta t\, \nabla_\eta\, (\Psi_\eta^{(2)} - \Psi_\eta^{(4)}\, \Delta_\eta\, \nabla_\eta)\, \Delta_\eta\, (J_{\mathscr{M}} \overline{Q}^n), \end{aligned} \tag{16}$$

where

$$\overline{A}^n = \frac{\partial \overline{F}}{\partial \overline{Q}}(\overline{Q}^n) \quad \text{and} \quad \overline{B}^n = \frac{\partial \overline{G}}{\partial \overline{Q}}(\overline{Q}^n). \tag{17}$$

The subscripted terms δ, ∇, and Δ represent the central, backward, and forward difference operators, respectively, in the lattice direction indicated by the subscript. The converged solution is denoted by $Q^N(x, y) \equiv \overline{Q}^{n*}(\mathscr{M}(x, y))$.

We introduce Bezier curves to parameterize the forebody simulator. Bezier polynomials possess several nice properties when used in approximations. The most important for the examples presented here are the *convex hull* and *endpoint interpolation* properties (see, e.g., Farin [12]). For this problem, we consider a set of two-parameter, $q = (q^1, q^2)$, Bezier curves

$$\begin{aligned} \mathscr{B} = \{\Gamma \in C^1[0, 1] \mid \Gamma(s) = (\Gamma_x(s), \Gamma_y(s; q)), \\ \Gamma_y(s; q) \geq \Gamma_a, s \in [0, 1], q \in \mathbb{R}^2\}, \end{aligned} \tag{18}$$

where

$$\Gamma_x(s) = aB_{0,3}(s) + 0.6B_{1,3}(s) + 0.8B_{2,3}(s) + bB_{3,3}(s), \tag{19}$$

$$\Gamma_y(s;q) = \Gamma_a B_{0,3}(\mathrm{s}) + q^1 B_{1,3}(s) + q^2 B_{2,3}(s) + \Gamma_b B_{3,3}(s), \tag{20}$$

and

$$B_{i,r}(x) = \binom{r}{i} x^i (1-x)^{r-i}. \tag{21}$$

We also assume $a = 0.5$ and $b = 1.0$. We can now introduce the approximate forebody simulator design problem.

Problem 2.2 (Approximate Forebody Simulator Design). Let $\{\hat{Q}_i\}_{i=1}^g$ be desired flow measurements at S. We assume that the data measurements are given at the quadrature points; otherwise interpolation must be used. Define the objective function

$$\mathscr{J}_g^N(\Gamma) = \sum_{i=1}^g c_i \|Q^N(x_i;\Gamma) - \hat{Q}_i\|_2^2, \tag{22}$$

where $Q^N(x_i;\Gamma)$ represents the approximate solution to (1) in the domain $\Omega(\Gamma)$ at the quadrature point x_i. The approximate forebody simulator design problem is to find $\Gamma_* \in \mathscr{B}$ such that

$$\mathscr{J}_g^N(\Gamma_*) \leq \mathscr{J}_g^N(\Gamma) \quad \text{for all } \Gamma \in \mathscr{B}. \tag{23}$$

Let

$$\mathscr{Q} = \{(q^1, q^2) \in \mathbb{R}^2 \mid \Gamma(\cdot; q^1, q^2) \in \mathscr{B}\}; \tag{24}$$

then the problem can be equivalently stated as finding $(q_*^1, q_*^2) \in \mathscr{Q}$ such that

$$\mathscr{J}_g^N(q_*^1, q_*^2) \leq \mathscr{J}_g^N(q^1, q^2) \quad \text{for all } (q^1, q^2) \in \mathscr{Q}. \tag{25}$$

2.2. *Duct Design Problem*

This problem consists of designing the cross-sectional area of a one-dimensional duct such that, under specified inlet and outlet conditions, it produces a flow which is as close to a desired transonic flow as possible. The governing conservation laws (steady-state continuity, momentum, and energy equations) can be reduced to a single two-point boundary value problem (BVP) for the velocity. It was shown in [13] that the velocity, u, is the solution of

$$\frac{\partial}{\partial x} f(u) + g(u, A) = 0, \qquad u(0) = u_{\text{in}} \quad \text{and} \quad u(1) = u_{\text{out}}, \tag{26}$$

where u_{in} and u_{out} are the velocities at the inlet and outlet of the duct, and A is the cross-sectional area of the duct,

$$f(u) = u + \frac{\overline{H}}{u}, \quad g(u,A) = \frac{1}{A}\left(\frac{\partial}{\partial x}A\right)\left(\overline{\gamma}u - \frac{\overline{H}}{u}\right), \quad \text{and} \quad \overline{\gamma} = \frac{\gamma - 1}{\gamma + 1}, \tag{27}$$

where $\overline{H}$ and γ are flow constants taken to be 1.14 and 1.4, respectively. The Rankine–Hugoniot condition yields the speed of sound as $u_s = \sqrt{\overline{H}}$. Unique solutions of this BVP are guaranteed for monotone area functions, therefore, cross-sectional areas, A, are restricted to

$$\mathscr{A} = \left\{ A \in C^1(0,1) \mid A(0) = A_{\text{in}}, A(1) = A_{\text{out}} \text{ and } \frac{\partial}{\partial x}A(x) > 0, \forall x \in (0,1) \right\} \tag{28}$$

for fixed inlet and outlet areas of A_{in} and A_{out}. We now describe the optimal design problem.

Problem 2.3 (Duct Design). Let $\hat{u}(\cdot) \in L^2(0,1)$ be a desired transonic flow profile for the duct and define the objective function by

$$\mathscr{J}(A) = \int_0^1 [u(x;A) - \hat{u}(x)]^2 \, dx, \tag{29}$$

where $u(\cdot; A)$ is the solution to (26) corresponding to A. The optimal design problem is to find an $A_* \in \mathscr{A}$ such that

$$\mathscr{J}(A_*) \leq \mathscr{J}(A) \quad \text{for all} \quad A \in \mathscr{A}. \tag{30}$$

While the BVP has a closed form solution [13], we consider approximations of (26) and consequently of Problem 2.3 in order to study the more general case. We begin by discretizing the duct length into N cells (of length $h = 1/N$) with centers, $x_j = (j - \frac{1}{2})h$, $j = 1, \ldots, N$ and define u_j^N to be the average velocity in the jth cell, i.e.,

$$u_j^N(A) = \frac{1}{h}\int_{x_j - h/2}^{x_j + h/2} u(x;A)\, dx. \tag{31}$$

A system of nonlinear equations for $u^N(A) = \{u_j^N(A)\}_{j=1}^N$ can be found by integrating (26) over each cell,

$$\frac{f(u(x_j + h/2;A)) - f(u(x_j - h/2;A))}{h} + g(u_j^N(A), A(x_j)) = 0, \quad j = 1, \ldots, N, \tag{32}$$

where it was assumed that $(1/A)\,(\partial/\partial x)\,A$ was nearly constant over each cell. An approximation to u^N is found by replacing the fluxes at the cell edges, $f(u(x_j + h/2))$, using the cell center values $f_j = f(u_j^N)$ and $f_{j+1} = f(u_{j+1}^N)$. Two standard first-order "Godunov-type" methods are the Enquist–Osher scheme

$$f\left(u\left(x_j+\frac{h}{2}\right)\right)\approx F^{\mathrm{EO}}_{j+1/2}$$
$$=\begin{cases} f_{j+1} & u_j^N, u_{j+1}^N \le u_s; \\ f_j & u_j^N, u_{j+1}^N \ge u_s; \\ f(u_s) & u_j^N < u_s < u_{j+1}^N; \\ f_j+f_{j+1}-f(u_s) & u_{j+1}^N < u_s < u_j^N; \end{cases} \tag{33}$$

and the artificial viscosity scheme

$$f\left(u\left(x_j+\frac{h}{2}\right)\right)\approx F^{\mathrm{AV}}_{j+1/2}=\frac{1}{2}(f_{j+1}+f_j-\alpha(u_{j+1}-u_j)), \tag{34}$$

where α has been selected as 1 for this study. These approximations were used in [13], but are included above for completeness.

We turn now to the approximation of the cross-sectional area A. The space $\mathscr{A}$ is replaced by a subset of Bezier quadratic polynomials. The properties of Bezier polynomials allow us to easily impose both the monotonicity requirement and the matching of inflow and outflow cross-sectional areas. Consider

$$\begin{aligned}\mathscr{B}=\{A\in C^1(0,1)|A(x)&=A_{\mathrm{in}}B_{0,2}(x)+qB_{1,2}(x)\\ &+A_{\mathrm{out}}B_{2,2}(x); x\in(0,1), q\in[A_{\mathrm{in}}, A_{\mathrm{out}}]\},\end{aligned} \tag{35}$$

where $B_{i,r}$ is defined in (21). Thus, $\mathscr{B}$ is a one parameter set of curves in $\mathscr{A}$. We restrict our optimization problem to this set $\mathscr{B}$.

Our final step in the approximation of Problem 2.3 is replacing the integral by a quadrature rule, with the set of quadrature weights and points $\{(c_i, x_i)\}_{i=1}^g$. We now state the approximate design problem.

Problem 2.4 (Approximate Duct Design). Let $\{\hat{u}_i\}_{i=1}^g$ represent data for a desired transonic flow profile in the duct. We assume that the data and the approximate solution are given at the quadrature points, otherwise interpolation must be used. Define the objective function

$$\mathscr{J}_g^N(A)=\sum_{i=1}^g c_i[u_i^N(A)-\hat{u}_i]^2, \tag{36}$$

where $u^N(A)$ is an approximate solution to (26) with the cross-sectional area A. The approximate design problem is to find an $A_* \in \mathscr{B}$ such that

$$\mathscr{J}_g^N(A_*)\le\mathscr{J}_g^N(A) \quad \text{for all} \quad A\in\mathscr{B}. \tag{37}$$

Note that we can identify any $A \in \mathscr{B}$ with the parameter $q \in \mathscr{Q} \equiv [A_{\mathrm{in}}, A_{\mathrm{out}}]$ which uniquely represents it. Thus we can equivalently state the problem as to find $q_* \in \mathscr{Q}$ such that

$$\mathscr{J}_g^N(q_*)\le\mathscr{J}_g^N(q) \quad \text{for all} \quad q\in\mathscr{Q}. \tag{38}$$

3. SENSITIVITY EQUATION METHOD

3.1. *Trust-Region Algorithms*

We shall use a trust-region algorithm for the optimization loop. The reason for selecting this type of scheme is its robustness property [7], which will be clear when we discuss the convergence properties in Section 4. We give a brief description of a trust-region algorithm below in order to prepare for the formulation of the sensitivity equation method.

The quasi-Newton optimization algorithm produces a sequence of iterates which are obtained by minimizing a local quadratic model of the objective function. This model is constructed using the evaluation of the objective function, $\mathscr{J}_g^N(q_k)$; its gradient, $\nabla\mathscr{J}_g^N(q_k)$; and a secant approximation to its Hessian, H_k, at the current iterate q_k. The minimization of this model produces the next iterate q_{k+1}, i.e.,

$$\begin{aligned}m_k(q_{k+1})&=\min_{s_k} m_k(q_k+s_k)\\ &=\min_{s_k}\left(\mathscr{J}_g^N(q_k)+\nabla\mathscr{J}_g^N(q_k)^Ts_k+\frac{1}{2}s_k^TH_ks_k\right).\end{aligned} \tag{39}$$

Thus the next step is

$$q_{k+1}=q_k-H_k^{-1}\nabla\mathscr{J}_g^N(q_k).$$

For sufficiently close initial guesses of q_0 and H_0 (and assumptions on the objective function), the iterates converge q-superlinearly to the minimum, q_* (see, e.g., [11, p. 206]).

However, the initial guess may not be in this superlinear region. Thus globalization strategies are employed to bring the iterates into the superlinear region. It is desirable to choose strategies which reduce to the quasi-Newton algorithm close to the minimum. One such strategy is a *trust-region* algorithm. In this algorithm, a quantity δ, known as the trust-region radius, is used to measure the region in which the local quadratic model, m_k, is "trusted" as an approximation of the actual objective function,$\mathscr{J}_g^N$. Thus, the next iterate, q_{k+1}, is now found by minimizing the model in this region, i.e.,

$$m_k(q_{k+1})=\min_{\|s_k\|\le\delta_k} m_k(q_k+s_k), \tag{40}$$

where δ_k is the trust-region radius at the kth iteration, and $\|\cdot\|$ indicates the Euclidean norm.

A heuristic for changing the trust-region radius needs to be developed which increases δ_k when the model prediction is good and decreases δ_k when the model prediction is poor. One such strategy uses the ratio

$$\rho_k = \frac{\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1})}{m_k(q_k) - m_k(q_{k+1})}, \tag{41}$$

which is the ratio of the computed reduction to the reduction predicted by the model. If this ratio is small (or negative), then the model did a poor job of predicting $\mathcal{J}_g^N$ and the trust-region is decreased. On the other hand, if the ratio is near 1, then the model did very well at predicting $\mathcal{J}_g^N$ and the trust-region radius is increased.

We present the resulting trust-region algorithm below.

ALGORITHM 3.1 (Trust-Region). Select an initial guess $q_0 \in \mathcal{Q}$, an initial trust-region radius δ_0, and constants $0 < \eta_1 < \eta_2 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$. Compute $\mathcal{J}_g^N(q_0)$, $\nabla\mathcal{J}_g^N(q_0)$ and select or initialize H_0.

Do $k = 0, 1, \ldots,$ until "convergence"

1. Determine the approximate solution s_k to Eq. (40). We chose the optimally constrained hook-step method [11, 14] to do this.
2. If $\rho_k < \eta_1$, then set $\delta_{k+1} \in (0, \gamma_1\delta_k)$ and $q_{k+1} = q_k$, $\mathcal{J}_g^N(q_{k+1}) = \mathcal{J}_g^N(q_k)$, $\nabla\mathcal{J}_g^N(q_{k+1}) = \nabla\mathcal{J}_g^N(q_k)$, and $H_{k+1} = H_k$.
3. If $\eta_1 < \rho_k < \eta_2$, then set $\delta_{k+1} \in (0, \delta_k]$ and $q_{k+1} = q_k + s_k$. Compute $\mathcal{J}_g^N(q_{k+1})$, $\nabla\mathcal{J}_g^N(q_{k+1})$ and the update H_{k+1}.
4. If $\eta_2 < \rho_k$, then set $\delta_{k+1} \in [\delta_k, \gamma_2\delta_k]$ and $q_{k+1} = q_k + s_k$. Compute $\mathcal{J}_g^N(q_{k+1})$, $\nabla\mathcal{J}_g^N(q_{k+1})$ and the update H_{k+1}.

Continue

3.2. *Design Sensitivities*

In order to apply a gradient-based optimization algorithm, such as the trust-region algorithm described above, we need to consider methods for computing the gradient of $\mathcal{J}_g^N$. In this discussion, we consider finding the gradient of $\mathcal{J}_g^N$ (or a suitable approximation) with respect to the single design parameter q. This discussion can be easily extended to find the gradient of $\mathcal{J}_g^N$ with respect to multiple design parameters. A straightforward approach is to use a finite difference approximation, e.g.,

$$\frac{\partial}{\partial q}\mathcal{J}_g^N(q) \approx \frac{\mathcal{J}_g^N(q + \Delta q) - \mathcal{J}_g^N(q)}{\Delta q}. \tag{42}$$

Unfortunately, this approach may not be practical for problems where the approximation of the PDE is computationally expensive and is overly complex in shape optimization problems due to the necessity of computing a new mesh. One way of alleviating the computational burden is by using design sensitivities, quantities which describe the influence of the design variables on the flow variables. For example, we can directly compute the gradient by differentiating (36) as

$$\frac{\partial}{\partial q}\mathcal{J}_g^N(q) = 2\sum_{i=1}^{q} c_i[u_i^N(q) - \hat{u}_i]\frac{\partial}{\partial q}u_i^N(q). \tag{43}$$

The quantity $(\partial/\partial q)u^N = \{(\partial/\partial q)\, u_i^N\}_{i=1}^N$ is the design sensitivity for the discretized flow u^N.

There are several ways to compute this sensitivity. As above, one might use finite differences, yielding the approximation

$$\frac{\partial}{\partial q}u^N(x_i; q) \approx \frac{u^N(x_i; q + \Delta q) - u^N(x_i; q)}{\Delta q}. \tag{44}$$

When the discretization is parameter dependent, it is easier to compute this approximation using

$$\frac{\partial}{\partial q}u^N(x_i; q) \approx \frac{u^N(x_i + (\partial/\partial q)\mathcal{M}(x_i)\Delta q; q + \Delta q) - u^N(x_i; q)}{\Delta q} - \frac{\partial}{\partial x}u^N(x_i; q)\frac{\partial}{\partial q}\mathcal{M}(x_i) \tag{45}$$

in order to avoid interpolating back to the unperturbed mesh. This approach has the advantage that it may be possible to select a step size Δq using error estimates for u^N. However, it is as computationally expensive as computing finite differences on $\mathcal{J}_g^N$.

A more efficient approach can be obtained by differentiating the simulation scheme used to approximate the flow (the discrete sensitivity approach). For example, in the FBS design problem, the simulation scheme (16) could be differentiated with respect to q, leading to a numerical scheme for terms like $(\partial/\partial q)u^N$. Since the chain rule must be used to carry this out, the resulting scheme for the sensitivities contains terms similar to those found in the simulation scheme. Thus, the sensitivities can be computed efficiently along with the flow. A disadvantage of this approach is that when the discretization is parameter dependent, as in shape optimization problems, then derivatives of the discretization (terms like $(\partial/\partial q)\mathcal{M}$) need to be considered, see, e.g., [23].

An alternative approach is based on differentiating the original flow equation with respect to the design parameter and then approximating the resulting *sensitivity equation*. The result is $((\partial/\partial q)u)^{N,M}$, where the superscript N refers to the approximation of the flow equation and the superscript M refers to the approximation of the sensitivity equation. Since this approach interchanges the order of differentiation and approximation, no mesh sensitivities are required. Furthermore, it has been shown [2] that applying the same approximation scheme to the sensitivity equation leads to computational advantages similar to those provided by the discrete approach described above. Moreover, additional computational savings could be obtained by applying a scheme which takes advantage of the linearity of the sensitivity equation. A potential disadvantage of this approach, however, is that in general $(\partial/\partial q)u^N \neq ((\partial/\partial q)u)^{N,M}$, even if the same approximation scheme is used for both the flow and sensitivity equations.

However, if we consider the gradient of the infinite-dimensional objective function,

$$\frac{\partial}{\partial q}\mathcal{J}(q) = 2\int_0^1 [u(x;A) - \hat{u}(x)]\,\frac{\partial}{\partial q}u(x;A), \tag{46}$$

then using the sensitivity equation approach provides an approximation of this gradient, i.e.,

$$\begin{aligned}\frac{\partial}{\partial q}\mathcal{J}(q) &\approx \left(\frac{\partial}{\partial q}\mathcal{J}\right)_g^{N,M}(q)\\ &= 2\sum_{i=1}^{g} c_i(u_i^N(q) - \hat{u}_i)\left(\frac{\partial}{\partial q}u\right)_i^{N,M}(q).\end{aligned} \tag{47}$$

Thus, we have reason to expect that this approach could produce feasible gradients for the optimization scheme. These two sensitivity approaches are described in detail in later sections using concrete examples.

3.3. *Sensitivity Equation Method*

The sensitivity equation method couples a trust-region optimization algorithm with gradient evaluations provided by approximating the sensitivity equation. Thus we consider applying Algorithm 3.1 with the quadratic model

$$\begin{aligned}\psi_k(q_{k+1}) &= \min_{\|s_k\|\le\delta_k} \psi_k(q_k + s_k)\\ &= \min_{\|s_k\|\le\delta_k}\left(\mathcal{J}_g^N(q_k) + g_k^T s_k + \frac{1}{2}s_k^T H_k s_k\right).\end{aligned} \tag{48}$$

Note that we replace the quadratic model m_k by ψ_k to emphasize the fact that $\nabla\mathcal{J}_g^N$ is approximated by g_k, computed as $((\partial/\partial q)\mathcal{J})_g^{N,M}(q_k)$.

The intent is to use the robustness of the trust-region optimization algorithm to compensate for the nonconsistent gradients. The result is an optimal design method which is often more efficient and considerably easier to implement than current methods. In the sections below, we discuss convergence issues and describe the implementation of this method.

4. CONVERGENCE ISSUES

Algorithm 3.1 is based on a quadratic model of the objective function $\mathcal{J}_g^N$ and expects the gradient $(\partial/\partial q)\mathcal{J}_g^N$ in order to formulate the next step. Whether or not that gradient is produced depends on the numerical schemes used for both the states and the sensitivities. This motivates our definition below.

Definition 4.1. A numerical scheme is said to produce *consistent derivatives* with respect to approximations N (for the states) and M (for the sensitivities) if

$$\frac{\partial}{\partial q}\mathcal{J}_g^N(\cdot) = \left(\frac{\partial}{\partial q}\mathcal{J}\right)_g^{N,M}(\cdot). \tag{49}$$

This is exactly the case for the discrete sensitivity approach, since one actually defines (computes) $((\partial/\partial q)\mathcal{J})_g^{N,M}(\cdot)$ to be $(\partial/\partial q)\mathcal{J}_g^N(\cdot)$.

In some cases, it may be more advantageous to compute gradients which are not consistent. However, it is possible that these schemes produce gradients which are "close enough" to ensure a convergent algorithm. We introduce the definition of asymptotically consistent derivatives below to facilitate our investigation of convergence later in this section.

Definition 4.2. A numerical scheme is said to produce *asymptotically consistent derivatives* with respect to approximations N (for the states) and M (for the sensitivities) if

$$\left|\frac{\partial}{\partial q}\mathcal{J}_g^N(q) - \left(\frac{\partial}{\partial q}\mathcal{J}\right)_g^{N,M}(q)\right| \to 0, \quad \forall q \in \mathcal{Q}_0 \tag{50}$$

is satisfied as the approximations N and M are refined.

We now consider the convergence of the sensitivity equation method. To begin with, we assume that the following hypotheses hold.

(H1) For a given q_0 in the design space $\mathcal{Q}$, let $\mathcal{Q}_0$ be an open convex subset containing the level set of $\mathcal{J}_g^N$ at q_0, i.e.,

$$\mathcal{L}_0 = \{q \in \mathcal{Q} | \mathcal{J}_g^N(q) \le \mathcal{J}_g^N(q_0)\} \subset \mathcal{Q}_0 \subseteq \mathcal{Q}. \tag{51}$$

(H2) $\mathcal{J}_g^N$ is bounded below.

(H3) $\mathcal{J}_g^N$ is Frechet differentiable on $\mathcal{Q}_0$.

(H4) The Frechet derivative of $\mathcal{J}_g^N$, denoted by $\nabla\mathcal{J}_g^N$, is Lipschitz continuous on $\mathcal{Q}_0$ with Lipschitz constant L, i.e.,

$$\|\nabla \mathscr{J}_g^N(q^1) - \nabla \mathscr{J}_g^N(q^2)\| \le L\|q^1 - q^2\|, \quad \forall q^1, q^2 \in \mathscr{Q}_0. \qquad (52)$$

(H5) The approximate gradient, g_k is asymptotically consistent to $\nabla \mathscr{J}_g^N(q_k)$.

(H6) There exists a constant $c_1 \in (0, 1]$ such that

$$c_1\|g_k\|\,\|s_k\| \le \langle -g_k, s_k\rangle \le \|g_k\|\,\|s_k\|, \quad \forall k = 1, 2, ... \qquad (53)$$

(H7) There exist constants $c_2, c_3 \in (0, \infty)$ such that

$$-c_2\,\langle d, d\rangle \le \langle H_k d, d\rangle \le c_3\,\langle d, d\rangle, \quad \forall k = 1, 2, ... \qquad (54)$$

The following discussion parallels the proof given in [7] which treats the use of trust-region algorithms with inexact gradient and function values. This discussion makes use of the fact that we seek the minimum of $\mathscr{J}_g^N$ and have asymptotically consistent derivatives.

Lemma 4.1. *Under assumptions* (*H*6) *and* (*H*7), *Algorithm* 3.1 *produces iterates which satisfy*

$$\psi_k(q_k) - \psi_k(q_{k+1}) \ge \frac{1}{2} c_1\|g_k\| \min\left\{\delta_k, \frac{c_1\|g_k\|}{c_3}\right\}. \qquad (55)$$

Proof. Note that since $\psi_k(q_k) = \mathscr{J}_g^N(q_k)$,

$$\psi_k(q_k) - \psi_k(q_{k+1}) = -\langle g_k, s_k\rangle - \frac{1}{2}\langle H_k s_k, s_k\rangle. \qquad (56)$$

Now, let

$$s_k = \|s_k\|\frac{s_k}{\|s_k\|} \equiv a_* d_k;$$

then a_* solves

$$\min_{0 \le a \le \delta_k} a\langle g_k, d_k\rangle + \frac{1}{2}a^2\,\langle H_k d_k, d_k\rangle. \qquad (57)$$

We can break this up into two cases, when $\langle H_k d_k, d_k\rangle \ge 0$ and when $\langle H_k d_k, d_k\rangle < 0$.

Case 1. Assume $\langle H_k d_k, d_k\rangle \ge 0$; then either

$$a_* = -\frac{\langle g_k, d_k\rangle}{\langle H_k d_k, d_k\rangle},$$

in which case

$$\begin{aligned}\psi_k(q_k) - \psi_k(q_{k+1}) &= \frac{\langle g_k, d_k\rangle}{\langle H_k d_k, d_k\rangle}\langle g_k, d_k\rangle \\ &\quad - \frac{1}{2}\frac{\langle g_k, d_k\rangle^2}{\langle H_k d_k, d_k\rangle^2}\langle H_k d_k, d_k\rangle \\ &= \frac{1}{2}\frac{\langle g_k, d_k\rangle^2}{\langle H_k d_k, d_k\rangle} \ge \frac{1}{2}c_1^2\frac{\|g_k\|^2}{c_3}\end{aligned}$$

using hypotheses (H6) and (H7), or

$$a_* = \delta_k,$$

in which case

$$\delta_k < -\frac{\langle g_k, d_k\rangle}{\langle H_k d_k, d_k\rangle}$$

implies

$$\begin{aligned}\psi_k(q_k) - \psi_k(q_{k+1}) &= -\delta_k\langle g_k, d_k\rangle - \frac{1}{2}\delta_k^2\langle H_k d_k, d_k\rangle \\ &\ge -\delta_k\,\langle g_k, d_k\rangle + \frac{1}{2}\delta_k\,\langle g_k, d_k\rangle \ge \frac{1}{2}c_1\delta_k\|g_k\|\end{aligned}$$

by hypothesis (H6).

Case 2. Assume $\langle H_k d_k, d_k\rangle < 0$; then $a_* = \delta_k$. Therefore

$$\begin{aligned}\psi_k(q_k) - \psi_k(q_{k+1}) &= -\delta_k\langle g_k, d_k\rangle - \frac{1}{2}\delta_k^2\langle H_k d_k, d_k\rangle \\ &\ge -\delta_k\,\langle g_k, d_k\rangle \ge c_1\delta_k\|g_k\| \ge \frac{1}{2}c_1\delta_k\|g_k\|. \quad \blacksquare\end{aligned}$$

Lemma 4.2. *Assume* (*H*7) *holds; then*

$$\liminf_{k\to\infty}\|g_k\| > 0 \quad \textit{and} \quad \lim_{k\to\infty}\delta_k = 0 \qquad (58)$$

imply

$$\lim_{k\to\infty} -\frac{\langle s_k, g_k\rangle}{\|s_k\|\,\|g_k\|} = 1. \qquad (59)$$

Proof. It was shown [11] that, if $\|s_k\| = \delta_k$, then the solution to (48) is given by $s(\mu_k)$, where

$$s(\mu) = -\,(H_k + \mu I)^{-1}\,g_k$$

and μ_k is the unique real number that satisfies $\|s(\mu_k)\| =$

δ_k. Therefore, if $\delta_k \to 0$, then $\mu_k \to \infty$ (since H_k is bounded, by (H7)). Thus $s_k \to -\mu_k^{-1} g_k$. ▌

Lemma 4.3. *Let $\mathcal{J}_g^N$ satisfy (H3), (H4), and (H7); then the iterates satisfy*

$$\begin{aligned}[\psi_k(q_k) - \psi_k(q_{k+1})] &- [\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1})] \\ &\leq \frac{1}{2}(c_2 + L)\,\|s_k\|^2 - \langle g_k - \nabla\mathcal{J}_g^N(q_k), s_k\rangle.\end{aligned} \tag{60}$$

Proof. Using the Cauchy–Schwartz inequality and (H3), we obtain

$$\begin{aligned}\mathcal{J}_g^N(q_{k+1}) - \mathcal{J}_g^N(q_k) &= \int_0^1 \langle \nabla\mathcal{J}_g^N(q_k + \lambda s_k), s_k\rangle\, d\lambda \\ &= \langle \nabla\mathcal{J}_g^N(q_k), s_k\rangle + \int_0^1 \langle \nabla\mathcal{J}_g^N(q_k + \lambda s_k) \\ &\quad - \nabla\mathcal{J}_g^N(q_k), s_k\rangle\, d\lambda \\ &\leq \langle \nabla\mathcal{J}_g^N(q_k), s_k\rangle + \int_0^1 \|\nabla\mathcal{J}_g^N(q_k + \lambda s_k) \\ &\quad - \nabla\mathcal{J}_g^N(q_k)\|\,\|s_k\|\, d\lambda.\end{aligned}$$

By the Lipschitz hypothesis (H4),

$$\begin{aligned}\mathcal{J}_g^N(q_{k+1}) - \mathcal{J}_g^N(q_k) &\leq \langle \nabla\mathcal{J}_g^N(q_k), s_k\rangle \\ &\quad + \int_0^1 L\|\lambda s_k\|\,\|s_k\|\, d\lambda \\ &= \langle \nabla\mathcal{J}_g^N(q_k), s_k\rangle + \frac{1}{2}L\|s_k\|^2.\end{aligned}$$

Thus, using (H7),

$$\begin{aligned}[\psi_k(q_k) - \psi_k(q_{k+1})] &- [\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1})] \\ &\leq -\langle g_k, s_k\rangle - \frac{1}{2}\langle H_k s_k, s_k\rangle + \langle \nabla\mathcal{J}_g^N(q_k), s_k\rangle + \frac{1}{2}L\|s_k\|^2 \\ &\leq -\langle g_k - \nabla\mathcal{J}_g^N(q_k), s_k\rangle + \frac{1}{2}(c_2 + L)\|s_k\|^2,\end{aligned}$$

which completes the proof. ▌

Lemma 4.4. *Assume $\mathcal{J}_g^N$ satisfies (H2), (H3), and (H4), and assume (H7) holds; then $\nabla\mathcal{J}_g^N$ is bounded on $\mathcal{L}_0$.*

Proof. Let c be a constant such that $\mathcal{J}_g^N(q) \geq c$, $\forall q \in \mathcal{Q}_0$ (as guaranteed by (H2)). Assume to the contrary that there exists a point $\overline{q} \in \mathcal{L}_0$ such that

$$\|\nabla\mathcal{J}_g^N(\overline{q})\|^2 > 8L(\mathcal{J}_g^N(q_0) - c).$$

Define $\overline{s} = -(\alpha/2L)\ \nabla\mathcal{J}_g^N(\overline{q})$, where we chose α small enough so that $\overline{q} + \overline{s} \in \mathcal{Q}_0$. Then

$$\begin{aligned}\mathcal{J}_g^N(\overline{q}) - \mathcal{J}_g^N(\overline{q} + \overline{s}) &= -\int_0^1 \langle \nabla\mathcal{J}_g^N(\overline{q}), \overline{s}\rangle\, d\lambda - \int_0^1 \langle \nabla\mathcal{J}_g^N(\overline{q} + \lambda\overline{s}) \\ &\quad - \nabla\mathcal{J}_g^N(\overline{q}), \overline{s}\rangle\, d\lambda \\ &\geq \frac{\alpha}{2L}\|\nabla\mathcal{J}_g^N(\overline{q})\|^2 - \frac{1}{2}L\|\overline{s}\|^2 \\ &\geq \frac{\alpha}{4L}\|\nabla\mathcal{J}_g^N(\overline{q})\|^2\left(1 - \frac{\alpha}{2}\right) \\ &> \frac{\alpha}{4L}\left(1 - \frac{\alpha}{2}\right)[8L(\mathcal{J}_g^N(q_0) - c)].\end{aligned}$$

This is positive for $\alpha \in (0, 2)$, and thus $\mathcal{J}_g^N(\overline{q}) > \mathcal{J}_g^N(\overline{q} + \overline{s})$, which implies $\overline{q} + \overline{s} \in \mathcal{L}_0$. In addition

$$\mathcal{J}_g^N(\overline{q}) - \mathcal{J}_g^N(\overline{q} + \overline{s}) > \mathcal{J}_g^N(q_0) - c$$

holds, but this is a contradiction since $\overline{q}$ and $\overline{q} + \overline{s}$ are in $\mathcal{L}_0$. ▌

Theorem 4.1. *Assume $\mathcal{J}_g^N$ satisfies (H2), (H3), and (H4). Furthermore, assume the approximate gradient satisfies conditions (H5) and (H6) and that the update is constructed so that (H7) holds. Then, for a sufficiently fine discretization, the sensitivity equation method produces a sequence of iterates such that*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{61}$$

Proof. Assume to the contrary that $\liminf_{k\to\infty} \|g_k\| > 0$ and define θ_k such that

$$\cos(\theta_k) = \frac{\langle -g_k, s_k\rangle}{\|g_k\|\,\|s_k\|}$$

and $w_k \in \mathcal{Q}$ such that

$$w_k = \begin{cases} 0 & \sin(\theta_k) = 0 \\ \dfrac{1}{\sin(\theta_k)}\left(\dfrac{s_k}{\|s_k\|} + \cos(\theta_k)\dfrac{g_k}{\|g_k\|}\right) & \sin(\theta_k) \neq 0 \end{cases}.$$

Then $\langle g_k, w_k\rangle = 0$ by construction, and

$$\langle g_k - \nabla\mathcal{J}_g^N(q_k), w_k\rangle = -\langle \nabla\mathcal{J}_g^N(q_k), w_k\rangle.$$

If $\sin(\theta_k) \neq 0$, then $\|w_k\| = 1$ and

$$s_k = \|s_k\|\left(-\cos(\theta_k)\frac{g_k}{\|g_k\|} + \sin(\theta_k) w_k\right). \tag{62}$$

Let $\mathcal{K}$ denote the set of successful iterations, then

$$\rho_k = \frac{\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1})}{\psi_k(q_k) - \psi_k(q_{k+1})} > \eta_1$$

for each $k \in \mathcal{K}$. Lemma 4.1 implies

$$\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1}) \geq \frac{\eta_1 c_1}{2} \|g_k\| \min\left\{\delta_k, \frac{c_1\|g_k\|}{c_3}\right\}.$$

Since $\mathcal{J}_g^N$ is bounded below, by (H2), the above condition implies $\lim_{k\to\infty, k\in\mathcal{K}}\delta_k = 0$. Therefore, as δ_k is decreased in unsuccessful iterations, $\lim_{k\to\infty}\delta_k = 0$. We now have the conditions for Lemma 4.2, and

$$\lim_{k\to\infty} \frac{\langle -g_k, s_k\rangle}{\|g_k\|\,\|s_k\|} = 1.$$

Thus $\lim_{k\to\infty} \cos(\theta_k) = 1$ and $\lim_{k\to\infty} \sin(\theta_k) = 0$.

Consider the expression

$$1 - \rho_k = \frac{\psi_k(q_k) - \psi_k(q_{k+1}) - (\mathcal{J}_g^N(q_k) - \mathcal{J}_g^N(q_{k+1}))}{\psi_k(q_k) - \psi_k(q_{k+1})};$$

by Lemma 4.3 and the definition of ψ_k, we get

$$1 - \rho_k \leq \frac{\frac{1}{2}(c_2 + L)\|s_k\|^2 - \langle g_k - \nabla\mathcal{J}_g^N(q_k), s_k\rangle}{\langle -g_k, s_k\rangle - \frac{1}{2}\langle H_k s_k, s_k\rangle}.$$

Using hypothesis (H7),

$$1 - \rho_k < \frac{\frac{1}{2}(c_2 + L)\|s_k\|^2 - \langle g_k - \nabla\mathcal{J}_g^N(q_k), s_k\rangle}{\langle -g_k, s_k\rangle}.$$

Substituting expression (62) and using $\|s_k\| < \delta_k$, we get

$$1 - \rho_k \leq \frac{\frac{1}{2}(c_2 + L)\delta_k - (\cos(\theta_k)/\|g_k\|)\,\langle g_k - \nabla\mathcal{J}_g^N(q_k), g_k\rangle - \langle g_k - \nabla\mathcal{J}_g^N(q_k), w_k\rangle \sin(\theta_k)}{\|g_k\| \cos(\theta_k)}.$$

By Lemma 4.4 and the Cauchy–Schwartz inequality, $\langle\nabla\mathcal{J}_g^N(q_k), w_k\rangle$ is bounded and we consider the limit as $k \to \infty$,

$$\lim_{k\to\infty} 1 - \rho_k = \lim_{k\to\infty} \frac{\langle g_k - \mathcal{J}_g^N(q_k), g_k\rangle}{\|g_k\|^2} \leq \frac{\|g_k - \mathcal{J}_g^N(q_k)\|}{\|g_k\|}.$$

Since $\liminf_{k\to\infty} \|g_k\| > 0$ and g_k is asymptotically consistent, we can select a sufficiently fine discretization such that

$$\lim_{k\to\infty} 1 - \rho_k < 1 - \eta_2.$$

Hence, $\rho_k > \eta_2$ which implies $\delta_{k+1} > \delta_k$, a contradiction. ∎

The convergence of the sensitivity equation method is possible due to the robustness of the trust-region algorithm (Algorithm 3.1). Secant methods have also been shown to be robust [16] and it may be possible to use this fact to study the convergence rate of this method. This will be investigated in future work.

5. DUCT DESIGN PROBLEM

In this section, we use the duct design problem to illustrate the implementation of the sensitivity equation method. To begin with, we will introduce the discrete approach for finding design sensitivities in order to compare it with the sensitivity equation approach.

5.1. *Discrete Sensitivities*

To obtain an algorithm for the sensitivities $(\partial/\partial q)u^N(q) = \{(\partial/\partial q)u_j^N(q)\}_{j=1}^N$, the system of nonlinear equations (32) is differentiated, yielding

$$\frac{\overline{F}_{j+1/2} - \overline{F}_{j-1/2}}{h} + \overline{g}\left(u_j^N, \frac{\partial}{\partial q}u_j^N, A(x_j), \frac{\partial}{\partial q}A(x_j)\right) = 0, \quad (63)$$

where $\overline{F}_{j+1/2}$ is determined by the scheme used to compute the flow. If the Enquist–Osher scheme was used,

$$\overline{F}_{j+1/2}^{\,\mathrm{EO}} = \begin{cases} \overline{f}_{j+1} & u_j^N, u_{j+1}^N \leq u_{\mathrm{s}}; \\ \overline{f}_j & u_j^N, u_{j+1}^N \geq u_{\mathrm{s}}; \\ 0 & u_j^N < u_s < u_{j+1}^N; \\ \overline{f}_j + \overline{f}_{j+1} & u_{j+1}^N < u_s < u_j^N; \end{cases} \quad (64)$$

or if the artificial viscosity scheme was used,

$$\overline{F}_{j+1/2}^{\,\mathrm{AV}} = \frac{1}{2}\left(\overline{f}_{j+1} + \overline{f}_j - \alpha\left(\frac{\partial}{\partial q}u_{j+1}^N - \frac{\partial}{\partial q}u_j^N\right)\right), \quad (65)$$

where $\overline{f}_j = \overline{f}(u_j^N, (\partial/\partial q)u_j^N)$,

$$\overline{f}\left(u, \frac{\partial}{\partial q}u\right) = \left(1 - \frac{\overline{H}}{u^2}\right)\frac{\partial}{\partial q}u \quad (66)$$

and

$$\bar{g}\left(u, \frac{\partial}{\partial q}u, A, \frac{\partial}{\partial q}A\right) = \frac{\partial}{\partial q}\left(\frac{1}{A}\frac{\partial}{\partial x}A\right)\left(\bar{\gamma}u - \frac{\overline{H}}{u}\right) + \left(\frac{1}{A}\frac{\partial}{\partial x}A\right)\left(\bar{\gamma} + \frac{\overline{H}}{u^2}\right)\frac{\partial}{\partial q}u. \tag{67}$$

This differentiated scheme can now be used to compute $(\partial/\partial q)u^N$.

5.2. *Sensitivity Equation*

We now present the implementation for the sensitivity equation approach. We begin by differentiating the flow equation (26) with respect to the parameter q. Thus

$$\frac{\partial}{\partial x}\bar{f}\left(u, \frac{\partial}{\partial q}u\right) + \bar{g}\left(u, \frac{\partial}{\partial q}u, A, \frac{\partial}{\partial q}A\right) = 0 \tag{68}$$

$$\frac{\partial}{\partial q}u(0) = 0 \quad \text{and} \quad \frac{\partial}{\partial q}u(1) = 0 \tag{69}$$

is the sensitivity equation for this problem. Note that the sensitivity equation is a linear equation with variable coefficients (determined by u). There has been little analysis of numerical schemes to approximate equations of this type. However, for this two-point boundary value problem, the same numerical schemes (Enquist–Osher and artificial viscosity) provide convergent algorithms. As in the approximation of (26), we consider $((\partial/\partial q)u)_j^N$ to be the average sensitivity in the jth cell. A system of nonlinear equations for $((\partial/\partial q)u)^N(q) = \{((\partial/\partial q)u)_j^N(q)\}_{j=1}^N$ can be found by integrating (68) over each cell,

$$\frac{\bar{f}(u(x_j + h/2), (\partial/\partial q)u(x_j + h/2)) - \bar{f}(u(x_j - h/2), (\partial/\partial q)u(x_j - h/2))}{h} + \bar{g}\left(u_j^N, \left(\frac{\partial}{\partial q}u\right)_j^N, A(x_j), \frac{\partial}{\partial q}A(x_j)\right) = 0, \tag{70}$$

$j = 1, ..., N$, where we assume A and $(\partial/\partial q)A$ are nearly constant over each cell. As before, the terms $\bar{f}(u(x_j + h/2), (\partial/\partial q)u(x_j + h/2))$ are replaced by the cell center values $\bar{f}_j$ and $\bar{f}_{j+1}$. Using the Enquist–Osher scheme, we obtain

$$\overline{F}_{j+1/2}^{\mathrm{EO}} = \begin{cases} \bar{f}_{j+1} & u_j^N, u_{j+1}^N \le u_s; \\ \bar{f}_j & u_j^N, u_{j+1}^N \ge u_s; \\ \bar{f}\left(u_s, \left(\dfrac{\partial}{\partial q}u\right)_s\right) & u_j^N < u_s < u_{j+1}^N; \\ \bar{f}_j + \bar{f}_{j+1} + \bar{f}\left(u_s, \left(\dfrac{\partial}{\partial q}u\right)_s\right) & u_{j+1}^N < u_s < u_j^N; \end{cases} \tag{71}$$

and obtain

$$\overline{F}_{j+1/2}^{\mathrm{AV}} = \frac{1}{2}\left(\bar{f}_{j+1} + \bar{f}_j - \alpha\left(\left(\frac{\partial}{\partial q}u\right)_{j+1}^N - \left(\frac{\partial}{\partial q}u\right)_j^N\right)\right) \tag{72}$$

for the artificial viscosity scheme. It is obvious that the approximation of the sensitivity equations depends on the approximation of the flow equations. As described earlier, we use the notation $((\partial/\partial q)u)^{N,M}$ to represent using scheme N to approximate the flow equation and scheme M to approximate the sensitivity equation.

5.3. *Convergence Results*

The convergence result provided in Theorem 4.1 can be proved for the case when the artificial viscosity scheme is used to appproximate the flow and the Enquist–Osher scheme is used to approximate the sensitivities in Algorithm 3.1. For this problem, we assume (the (H1) in Theorem 4.1) that

$$\mathscr{Q} = [A_{\mathrm{in}}, A_{\mathrm{out}}], \quad \mathscr{Q}_0 = (A_{\mathrm{in}}, A_{\mathrm{out}}),$$

and

$$\mathscr{L}_0 = \{q \in \mathscr{Q} \,|\, \mathscr{J}_g^{N_{\mathrm{AV}}}(q) \le \mathscr{J}_g^{N_{\mathrm{AV}}}(q_0)\}.$$

The objective function $\mathscr{J}_g^{N_{\mathrm{AV}}}$ given above is obviously bounded below (by zero if all of the quadrature weights are nonnegative) satisfying (H2). The hypothesis (H3), the differentiability of

$$\mathscr{J}_g^{N_{\mathrm{AV}}}(q) = \sum_{i=1}^{g} c_i\,(u^{N_{\mathrm{AV}}}(x_i; q) - \hat{u}(x_i))^2$$

on $\mathscr{Q}_0$ and hypothesis (H4), the Lipschitz continuity of the derivative, follow from the following.

LEMMA 5.1. *The approximate solution $u^{N_{\mathrm{AV}}}$ is differentiable and the derivative is Lipschitz continuous on $\mathscr{Q}_0$.*

Proof. The approximate solution, $u^{N_{\mathrm{AV}}}$ is the root of the nonlinear equations

$$W(u^{N_{\mathrm{AV}}}, q) = [F_{j+1/2}^{\mathrm{AV}}(u^{N_{\mathrm{AV}}}, q) - F_{j-1/2}^{\mathrm{AV}}(u^{N_{\mathrm{AV}}}, q)] + g_j\,(u^{N_{\mathrm{AV}}}, q) = 0, \tag{73}$$

where $F_{j+1/2}^{\mathrm{AV}}$ and g are C^∞ functions of their arguments (for $u^{N_{\mathrm{AV}}} > 0$). Then by the implicit function theorem, the map

$$q \to u^{N_{\mathrm{AV}}}(q)$$

is Lipschitz continuously differentiable. ▌

We point out that the differentiability of the approximate objective functional is strongly dependent on the discretization scheme used in the approximation. For example, the objective functional associated with a Godunov approximation of the flow is not differentiable, a result of matching a parameter dependent discontinuity on a discrete set of points [4]. Finding feasible optimization strategies for this problem has been the focus of recent work, see, e.g., [4, 22, 26]. However, for the purpose of this discussion, the artificial viscosity scheme provides a smooth enough approximate objective function.

The hypothesis (H5) is guaranteed (for some discretization level) by the asymptotic consistency shown below.

THEOREM 5.1. *For the one-dimensional Euler equations, the derivative* $((\partial/\partial q)\mathscr{J})_g^{N_{\mathrm{AV}},M_{\mathrm{EO}}}$*, where the flow is approximated using the artificial viscosity approximation and the sensitivities are approximated using the Enquist–Osher scheme, is asymptotically consistent to* $(\partial/\partial q)\mathscr{J}_g^{N_{\mathrm{AV}}}$.

Proof. Consider the norm used in the definition of asymptotic consistency:

$$\begin{aligned}
&\left|\frac{\partial}{\partial q}\mathscr{J}_g^{N_{\mathrm{AV}}} - \left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{EO}}}\right| \\
&\quad\le \left|\frac{\partial}{\partial q}\mathscr{J}_g^{N_{\mathrm{AV}}} - \left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{AV}}}\right| \\
&\quad\quad + \left|\left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{AV}}} - \left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{Ex}}}\right| \\
&\quad\quad + \left|\left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{Ex}}} - \left(\frac{\partial}{\partial q}\mathscr{J}\right)_g^{N_{\mathrm{AV}},M_{\mathrm{EO}}}\right|.
\end{aligned}$$

The first term on the right-hand side vanishes since using the artificial viscosity scheme for approximating both the flow and sensitivity equations leads to consistent derivatives. The last two terms go to zero as the approximations N_{AV}, M_{AV}, and M_{EO} are refined, since the artificial viscosity and Enquist–Osher schemes converge when used to approximate the sensitivity equation, $((\partial/\partial q)u)^{N_{\mathrm{AV}},M_{\mathrm{Ex}}}$ is the exact solution to the sensitivity equation given $u^{N_{\mathrm{AV}}}$. ▌

The hypothesis (H6) can be enforced by the optimization algorithm by rejecting steps which violate this condition and shrinking the trust-region radius. This procedure eventually creates a step which satisfies (H6), since the limit of this procedure would produce a step in the steepest descent direction. As noted above, the existence of the constant c_1 is a theoretical result. This is a relatively weak condition stating that the angle between the step s_k and the steepest descent direction $-g_k$ are uniformly bounded away from 90°. In practice, we never actually calculate a c_1.

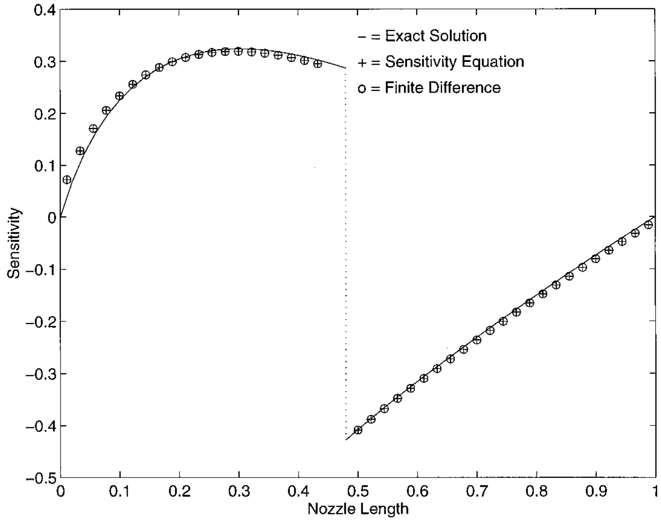


FIG. 3. Design sensitivity approximations using Enquist–Osher scheme.

Finally, (H7) can be enforced by the secant update strategy. Therefore, we have shown that these approximation schemes satisfy the conditions of Theorem 4.1. Numerical computations using these sensitivity schemes are provided below.

5.4. *Numerical Results*

The sensitivity of the velocity with respect to the Bezier parameter, q, is presented using the numerical schemes described above. For this computation, the cross-sectional area corresponds to an element of $\mathscr{B}$ (see (35)) with $q = 1.37125$. The interval [0, 1] is divided into 45 cells. In Fig. 3, the sensitivity solution using the Enquist–Osher scheme to compute both the flow $u^{N_{\mathrm{EO}}}$ and the sensitivity $((\partial/\partial q)u)^{N_{\mathrm{EO}},M_{\mathrm{EO}}}$ is compared with the closed form sensitivity solution. In addition, the sensitivities computed via finite differences of Enquist–Osher solutions using a finite difference step size of $\Delta q = (1 \times 10^{-6})\, q$ are also provided. Excellent agreement is seen for both of these methods. The only discrepancy is in the cell to the left of the shock, where numerical dissipation appears in the flow solution.

The corresponding design sensitivities which are computed using only the artificial viscosity schemes are shown in Fig. 4. As above, the agreement is excellent except where dissipation errors appear in the flow approximations. In this case, these errors appear over more cells near the shock.

Note that the computations of these sensitivities were performed efficiently, relative to the cost of a flow approxi-

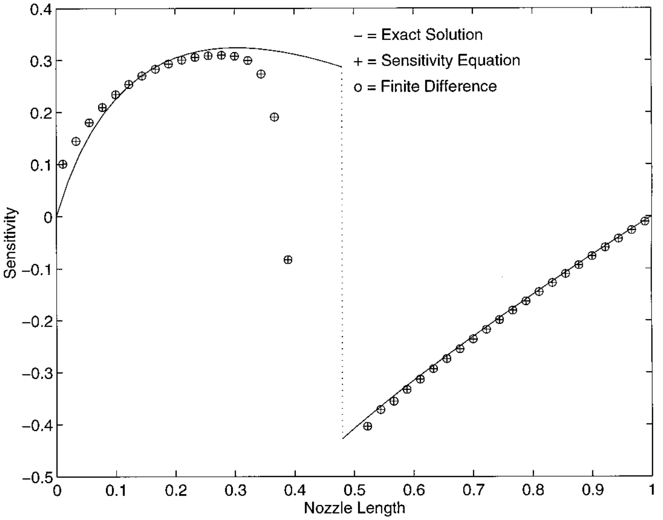


FIG. 4. Design sensitivity approximations using artificial viscosity scheme.

mation. The flow approximation requires solving a system of nonlinear equations. The sensitivity approximation, on the other hand, only requires solving a linear system since the sensitivity appears only linearly in the definition of $\bar{f}$ and $\bar{g}$. Moreover, if the Newton method is used to solve the nonlinear system, then the linear system is already available in factored form. Therefore, the sensitivities can be computed using less computational time than required for one Newton step. Computational efficiencies such as this can be missed if the flow algorithm is simply differentiated.

Note that as long as $((\partial/\partial q)u)_s$ is bounded,

$$\bar{f}\left(u_s, \left(\frac{\partial}{\partial q}u\right)_s\right) = \left(1 - \frac{\overline{H}}{u_s^2}\right)\left(\frac{\partial}{\partial q}u\right)_s = 0,$$

since $\overline{H} = u_s^2$. Thus, one observes that the numerical algorithms to compute either $(\partial/\partial q)u^{N_{\mathrm{EO}}}$ or $((\partial/\partial q)u)^{N_{\mathrm{EO}},M_{\mathrm{EO}}}$ are equivalent. This leads to the fact that using the Enquist–Osher scheme to approximate both the flow and the sensitivity equations produces consistent gradients. In addition, it is easily seen that using the artificial viscosity scheme to approximate both equations also produces consistent gradients. However, if the artificial viscosity scheme is used to approximate the flow and the Enquist–Osher scheme is used to approximate the sensitivity equations, the gradients are not consistent but asymptotically consistent.

Numerical results for this asymptotically consistent case are provided in Table I.

TABLE I

A Comparison of Gradients at the Optimum for Various Mesh Sizes

N	q_*	$\mathcal{J}_g^{N_{AV}}$	$(\nabla_q\mathcal{J})_q^{N_{AV},M_{EO}}$
15	1.3398	0.011707	−0.056521
45	1.3437	0.004800	−0.001566
135	1.3525	0.002485	−0.000012
225	1.3543	0.002476	0.007602
315	1.3553	0.002645	0.026731
405	1.3554	0.002816	0.001584

6. FOREBODY SIMULATOR DESIGN PROBLEM

We now describe the implementation of the sensitivity equation method for the forebody simulator design problem described in Section 2. As in the duct design problem, we begin by presenting the equations which comprise the discrete sensitivity scheme in order to compare and contrast the two methods. Unlike the duct problem, we have no theoretical convergence results for the FBS design problem. However, the numerical experiments below show that the SEM still converges.

6.1. *Discrete Sensitivities*

Differentiating the numerical scheme (16) with respect to a design parameter, represented by q, leads to the scheme

$$\begin{aligned}
&[I + \Delta t\delta_\xi\overline{A}^n - \nabla_\xi(\Psi_\xi^{(2)} + \Psi_\xi^{(4)})\Delta_\xi J_{\mathscr{M}}] \\
&\qquad \times [I + \Delta t\delta_\eta\overline{B}^n - \nabla_\eta(\Psi_\eta^{(2)} + \Psi_\eta^{(4)})\Delta_\eta J_{\mathscr{M}}]\, \Delta\frac{\partial}{\partial q}\overline{Q}^n \\
&\quad = -\left[\Delta t\delta_\xi\frac{\partial}{\partial q}\overline{A}^n - \nabla_\xi\left(\frac{\partial}{\partial q}\Psi_\xi^{(2)} + \frac{\partial}{\partial q}\Psi_\xi^{(4)}\right)\Delta_\xi J_{\mathscr{M}}\right. \\
&\qquad \left. - \nabla_\xi(\Psi_\xi^{(2)} + \Psi_\xi^{(4)})\,\Delta_\xi\frac{\partial}{\partial q}J_{\mathscr{M}}\right] \\
&\qquad \times [I + \Delta t\delta_\eta\overline{B}^n - \nabla_\eta(\Psi_\eta^{(2)} + \Psi_\eta^{(4)})\,\Delta_\eta J_{\mathscr{M}}]\,\Delta\overline{Q}^n \\
&\qquad - [I + \Delta t\delta_\xi\overline{A}^n - \nabla_\xi(\Psi_\xi^{(2)} + \Psi_\xi^{(4)})\,\Delta_\xi J_{\mathscr{M}}] \\
&\qquad \times \left[\Delta t\delta_\eta\frac{\partial}{\partial q}\overline{B}^n - \nabla_\eta\left(\frac{\partial}{\partial q}\Psi_\eta^{(2)} + \frac{\partial}{\partial q}\Psi_\eta^{(4)}\right)\Delta_\eta J_{\mathscr{M}}\right. \qquad (74) \\
&\qquad \left. - \nabla_\eta(\Psi_\eta^{(2)} + \Psi_\eta^{(4)})\,\Delta_\eta\frac{\partial}{\partial q}J_{\mathscr{M}}\right]\Delta\overline{Q}^n \\
&\qquad - \Delta t\delta_\xi\frac{\partial}{\partial q}\overline{F}^n - \Delta t\delta_\eta\frac{\partial}{\partial q}\overline{G}^n \\
&\qquad + \Delta t\nabla_\xi\left(\frac{\partial}{\partial q}\Psi_\xi^{(2)} - \frac{\partial}{\partial q}\Psi_\xi^{(4)}\,\Delta_\xi\nabla_\xi\right)\Delta_\xi(J_{\mathscr{M}}\overline{Q}^n)
\end{aligned}$$

$$+ \Delta t \nabla_\xi (\Psi_\xi^{(2)} - \Psi_\xi^{(4)} \Delta_\xi \nabla_\xi) \Delta_\xi \frac{\partial}{\partial q} (J_{\mathscr{M}} \overline{Q}^n)$$

$$+ \Delta t \nabla_\eta \left(\frac{\partial}{\partial q} \Psi_\eta^{(2)} - \frac{\partial}{\partial q} \Psi_\eta^{(4)} \Delta_\eta \nabla_\eta \right) \Delta_\eta (J_{\mathscr{M}} \overline{Q}^n)$$

$$+ \Delta t \nabla_\eta (\Psi_\eta^{(2)} - \Psi_\eta^{(4)} \Delta_\eta \nabla_\eta) \Delta_\eta \frac{\partial}{\partial q} (J_{\mathscr{M}} \overline{Q}^n).$$

The equations representing the boundary conditions are also differentiated. Note that the above sensitivity scheme requires derivatives of the mapping, $(\partial/\partial q)\,\mathscr{M}$ (denoted as mesh sensitivities) and the dissipation terms, $(\partial/\partial q)\Psi^{(2)}$ and $(\partial/\partial q)\Psi^{(4)}$. Evaluation of $(\partial/\partial q)\mathscr{M}$ is given by differentiating the scheme which determines $\mathscr{M}$; see, e.g., [23]. Other methods for approximating $(\partial/\partial q)\,\mathscr{M}$ have also been investigated; see, e.g., [28]. We see from (74) that terms containing these expressions represent a significant portion of the computational effort, aside from the fact that $(\partial/\partial q)\mathscr{M}$, $(\partial/\partial q)\Psi^{(2)}$, and $(\partial/\partial q)\Psi^{(4)}$ themselves need to be determined.

6.2. *Sensitivity Equation*

The sensitivity equation approach to computing design sensitivities is presented below. To begin with, we differentiate the Euler equations and associated boundary conditions with respect to the design parameter q, which leads to

$$\frac{\partial F_q}{\partial x} + \frac{\partial G_q}{\partial y} = 0, \qquad (75)$$

where

$$F_q = \frac{\partial}{\partial q} uQ + uQ_q + \begin{bmatrix} 0 \\ \frac{\partial}{\partial q} P \\ 0 \\ \frac{\partial}{\partial q} Pu + P \frac{\partial}{\partial q} u \end{bmatrix},$$

$$G_q = \frac{\partial}{\partial q} vQ + vQ_q + \begin{bmatrix} 0 \\ 0 \\ \frac{\partial}{\partial q} P \\ \frac{\partial}{\partial q} Pv + P \frac{\partial}{\partial q} v \end{bmatrix},$$

$$Q_q = \begin{bmatrix} \frac{\partial}{\partial q} \rho \\ \frac{\partial}{\partial q} (\rho u) \\ \frac{\partial}{\partial q} (\rho v) \\ \frac{\partial}{\partial q} E \end{bmatrix},$$

and where

$$\frac{\partial}{\partial q} u = \left[\frac{\partial}{\partial q} (\rho u) - \frac{\partial}{\partial q} \rho \frac{\rho u}{\rho} \right] \Big/ \rho \quad \text{and}$$

$$\frac{\partial}{\partial q} v = \left[\frac{\partial}{\partial q} (\rho v) - \frac{\partial}{\partial q} \rho \frac{\rho v}{\rho} \right] \Big/ \rho,$$

since $\rho \neq 0$.

We are now free to apply any appropriate scheme to solve (75). In particular, it is possible to use a method which takes advantage of the linearity of the sensitivity equation. However, in this work, the same scheme used to solve the flow equations is used to approximate the sensitivity equations, which leads to an efficient computational scheme as in the discrete approach [2]. This scheme is described below.

This equation may now be transformed to generalized coordinates, so that the finite differencing can be done more easily. It makes sense to use the same transformation (which is equivalent to using the same mesh) that was used in the solution of the Euler equations. Thus the resulting system is

$$\frac{\partial \overline{F}_q}{\partial \xi} + \frac{\partial \overline{G}_q}{\partial \eta} = 0, \qquad (76)$$

where

$$\overline{F}_q = U\overline{Q}_q + U_q\overline{Q} + \frac{\partial}{\partial q} PJ_{\mathscr{M}}^{-1} \begin{bmatrix} 0 \\ \frac{\partial \xi}{\partial x} \\ \frac{\partial \xi}{\partial y} \\ U \end{bmatrix} + PJ_{\mathscr{M}}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ U_q \end{bmatrix},$$

$$\overline{G}_q = V\overline{Q}_q + V_q\overline{Q} + \frac{\partial}{\partial q} PJ_{\mathcal{M}}^{-1} \begin{bmatrix} 0 \\ \dfrac{\partial \eta}{\partial x} \\ \dfrac{\partial \eta}{\partial y} \\ V \end{bmatrix} + PJ_{\mathcal{M}}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ V_q \end{bmatrix},$$

where

$$U = \nabla\xi \cdot (u, v) \quad \text{and} \quad U_q = \nabla\xi \cdot \left(\frac{\partial}{\partial q} u, \frac{\partial}{\partial q} v\right),$$

and

$$V = \nabla\eta \cdot (u, v) \quad \text{and} \quad V_q = \nabla\eta \cdot \left(\frac{\partial}{\partial q} u, \frac{\partial}{\partial q} v\right).$$

It can be shown that

$$\overline{A} = \frac{\partial \overline{F}}{\partial \overline{Q}} = \frac{\partial \overline{F}_q}{\partial \overline{Q}_q} \quad \text{and} \quad \overline{B} = \frac{\partial \overline{G}}{\partial \overline{Q}} = \frac{\partial \overline{G}_q}{\partial \overline{Q}_q},$$

so that the discretization has the same factored form as the Euler equations; thus

$$\begin{aligned} &[I + \Delta t \delta_\xi \overline{A}^n - \nabla_\xi(\Psi_\xi^{(2)} + \Psi_\xi^{(4)})\Delta_\xi J_{\mathcal{M}}] \\ &\qquad \times [I + \Delta t \delta_\eta \overline{B}^n - \nabla_\eta(\Psi_\eta^{(2)} + \Psi_\eta^{(4)})\Delta_\eta J_{\mathcal{M}}]\, \Delta\left(\frac{\partial}{\partial q}\overline{Q}\right)^n \\ &\quad = -\Delta t \delta_\xi(\overline{F}_q)^n - \Delta t \delta_\eta(\overline{G}_q)^n \\ &\qquad + \Delta t \nabla_\xi(\Psi_\xi^{(2)} - \Psi_\xi^{(4)}\Delta_\xi\nabla_\xi)\Delta_\xi(J_{\mathcal{M}}\overline{Q})^n \\ &\qquad + \Delta t \nabla_\eta(\Psi_\eta^{(2)} - \Psi_\eta^{(4)}\Delta_\eta\nabla_\eta)\Delta_\eta(J_{\mathcal{M}}\overline{Q})^n. \end{aligned} \tag{77}$$

Since the left-hand-side matrices are the same, a right-hand-side vector needs to be formed for each design sensitivity. In addition, the boundary condition type is the same for both the Euler and sensitivity equations. The boundary conditions are determined using implicit differentiation.

Note that this scheme is similar to the discrete sensitivity approach. However, since the approximation is applied after the differentiation, there are no mesh sensitivity or dissipation sensitivity terms. The other obvious difference is that the boundary condition on the parameter-dependent boundary is different.

6.3. *Boundary Conditions*

The boundary conditions for the sensitivity equation (75) are provided below for the case where the forebody simulator is described by a two-parameter Bezier curve (18)–(20). Extensions to other forebody descriptions will be obvious. The appropriate conditions are obtained by differentiating the corresponding boundary conditions for the Euler equations. For example, at the inlet, the flow Q_{in} is prescribed and will not vary as the forebody parameters $q = (q^1, q^2)$ are changed; thus

$$Q_q = 0$$

at the test cell inflow. The walls are treated in a similar fashion. However, the boundary condition at the forebody simulator surface requires more attention. This is because the points where the condition is evaluated are parameter dependent.

We study the treatment of condition (5) in detail. The normal vector to the forebody surface is

$$\hat{n} = \left(-\frac{\partial}{\partial s}\Gamma_y(s; q), \frac{\partial}{\partial s}\Gamma_x(s)\right) \Bigg/ \sqrt{\left(\frac{\partial}{\partial s}\Gamma_x(s)\right)^2 + \left(\frac{\partial}{\partial s}\Gamma_y(s; q)\right)^2}. \tag{78}$$

Thus, the boundary condition (5) can be written as

$$\begin{aligned} &-u\,(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial s}\Gamma_y(s; q) \\ &\qquad + v\,(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial s}\Gamma_x(s) = 0. \end{aligned} \tag{79}$$

The corresponding sensitivity equation boundary condition for the first parameter, q^1, can be obtained via differentiation, i.e.,

$$\begin{aligned} &-\frac{\partial}{\partial q^1} u(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial s}\Gamma_y(s; q) \\ &\quad + \frac{\partial}{\partial q^1} v\,(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial s}\Gamma_x(s) \\ &\quad = \frac{\partial}{\partial y} u(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial q^1}\Gamma_y(s; q)\frac{\partial}{\partial s}\Gamma_y(s; q) \\ &\quad + u(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial^2}{\partial s \partial q^1}\Gamma_y(s; q) \\ &\quad - \frac{\partial}{\partial y} v(\Gamma_x(s), \Gamma_y(s; q); q)\frac{\partial}{\partial q^1}\Gamma_y(s; q)\frac{\partial}{\partial s}\Gamma_x(s). \end{aligned}$$

This is simply a nonhomogeneous version of condition (5), namely,

$$\left(\frac{\partial}{\partial q^1}u, \frac{\partial}{\partial q^1}v\right)\cdot\hat{n} = \frac{\partial}{\partial y}u\frac{\partial}{\partial q^1}\Gamma_y\frac{\partial}{\partial s}\Gamma_y + u\frac{\partial^2}{\partial s\partial q^1}\Gamma_y - \frac{\partial}{\partial y}v\frac{\partial}{\partial q^1}\Gamma_y\frac{\partial}{\partial s}\Gamma_x. \tag{80}$$

Using the same techniques, the boundary conditions corresponding to (6) are

$$\begin{aligned}\frac{\partial}{\partial n}\left(\frac{\partial}{\partial q^1}u_t\right) &= \frac{\partial^2}{\partial x\partial y}u_t\frac{\partial}{\partial q'}\Gamma_y\frac{\partial}{\partial s}\Gamma_y \\ &\quad + \frac{\partial}{\partial x}u_t\frac{\partial^2}{\partial s\partial q^1}\Gamma_y - \frac{\partial^2}{\partial y^2}u_t\frac{\partial}{\partial q^1}\Gamma_y\frac{\partial}{\partial s}\Gamma_x, \\ \frac{\partial}{\partial n}\left(\frac{\partial}{\partial q^1}v_t\right) &= \frac{\partial^2}{\partial x\partial y}v_t\frac{\partial}{\partial q'}\Gamma_y\frac{\partial}{\partial s}\Gamma_y \\ &\quad + \frac{\partial}{\partial x}v_t\frac{\partial^2}{\partial s\partial q^1}\Gamma_y - \frac{\partial^2}{\partial y^2}v_t\frac{\partial}{\partial q^1}\Gamma_y\frac{\partial}{\partial s}\Gamma_x.\end{aligned}$$

The analogous boundary conditions for q^2 are obvious.

6.4. *Numerical Results*

The sensitivity equation approach, which computes design sensitivities for the two-dimensional Euler equation, is illustrated below. In this implementation, a right-hand-side vector for each design sensitivity is formed along with the corresponding vector for the flow approximations. The updates for the flow and the sensitivity variables are obtained simultaneously, exploiting the fact that the left-hand-side matrices are the same.

The design sensitivities with respect to the first Bezier parameter q^1 were computed for a forebody described by the curve

$$\hat{\Gamma} = (\hat{x}(s), \hat{y}(s)), \quad s \in [0, 1],$$

where

$$\begin{aligned}\hat{x}(s) &= 0.0B_{0,3}(s) + 0.1B_{1,3}(s) + 0.55B_{2,3}(s) + 1.0B_{3,3}(s), \\ \hat{y}(s) &= \Gamma_a B_{0,3}(s) + q^1 B_{1,3}(s) + q^2 B_{2,3}(s) + \Gamma_b B_{3,3}(s),\end{aligned}$$

$q^1 = 0.1$, $q^2 = 0.15$, $\Gamma_a = 0$, and $\Gamma_b = 0.2$. This curve is twice as long in the x-direction as the admissible forebody simulators given in $\mathscr{B}$ (see (18)). Under a uniform inlet flow profile described by the inlet Mach number, $M_a = 2.0$, the approximate flow variables and sensitivities are computed on a 43 × 49 mesh. The sensitivity of the x-component of momentum with respect to the Bezier parameter q^1, computed using the sensitivity equation approach and the finite difference approach (for four different step sizes) are plotted along the outflow plane in Fig. 5. The corresponding plots for the energy sensitivity are provided in Fig. 6. Observe that the step size of 0.00001 produces noisy sensitivity values close to the forebody (presumably due to round-off errors). A larger step size of 0.01 gives the best results (when compared to the sensitivity equation approach) near the shock location. The best qualitative behavior appears when the step size is 0.001. Figures 5 and 6 demonstrate the difficulty of obtaining a satisfactory step size at all resolution levels in the flow domain.

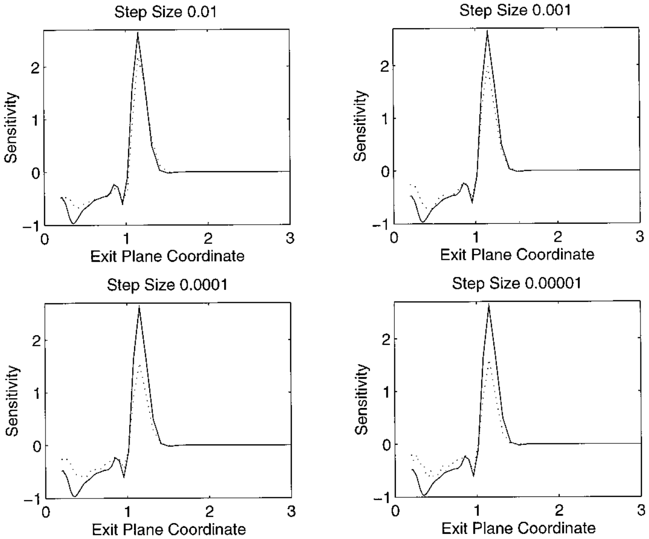


FIG. 5. Comparison of sensitivities at outflow: x-component of momentum.

A model forebody simulator design problem is discussed below. To begin with, we seek the optimum value of the inlet Mach number and two Bezier parameters ((q^1, q^2),

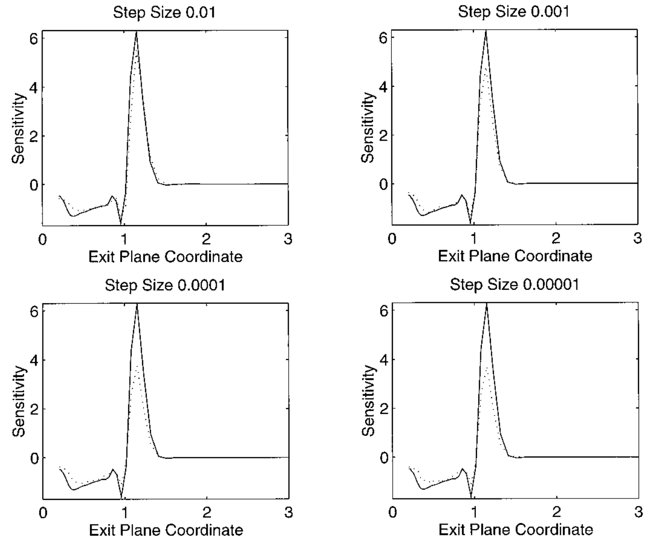


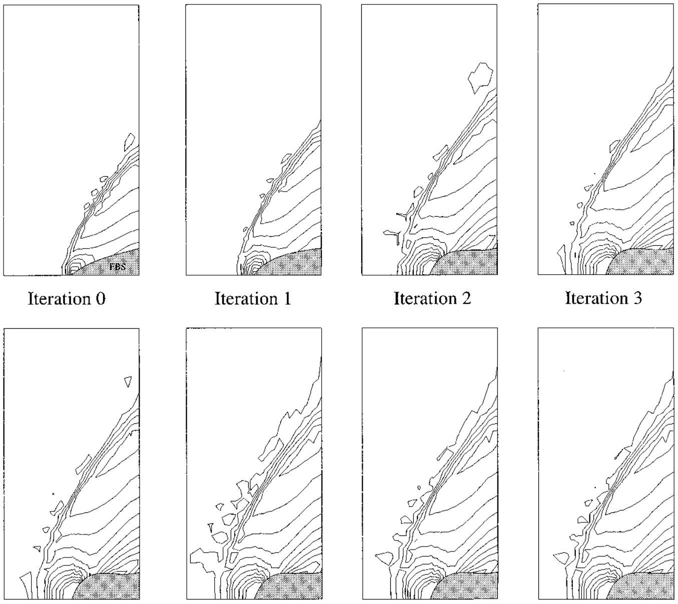
FIG. 6. Comparison of sensitivities at outflow: energy.

TABLE II

Shortened Forebody Optimization

Iteration	M_a	q_1	q_2	Cost functional	Gradient
0	2.00000	0.10000	0.15000	3.2339	27.1283
1	2.00108	0.14608	0.17177	1.6000	11.6285
2	2.01054	0.26846	0.14152	0.3332	3.7955
3	2.00897	0.30765	0.13671	0.2334	0.4621
4	2.01027	0.30139	0.14007	0.2306	0.5963
5	2.01307	0.29367	0.14737	0.2289	0.6861
6	2.01670	0.28891	0.15564	0.2271	0.5009
7	2.01900	0.29011	0.15921	0.2249	0.1513
8	2.01940	0.29278	0.15821	0.2237	0.0576
9	2.01936	0.29420	0.15669	0.2232	0.0571
10	2.01952	0.29439	0.15604	0.2230	0.0275
11	2.01994	0.29417	0.15603	0.2229	0.0173
12	2.02006	0.29415	0.15609	0.2229	0.0153

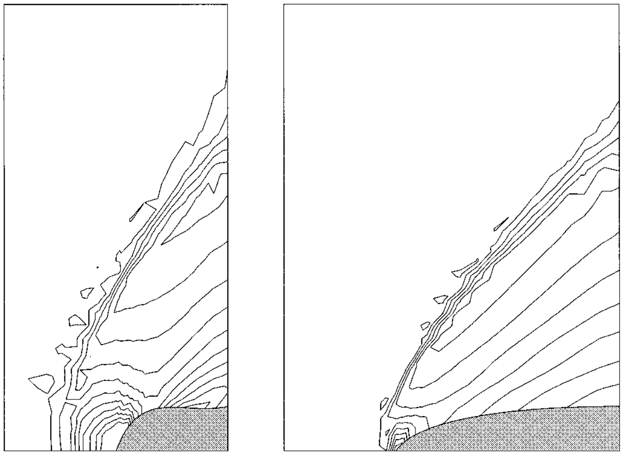
describing a shortened forebody simulator in the admissible set $\mathscr{B}$) which minimize the approximate cost functional $\mathscr{J}_g^N$ (given in Eq. (25)). The flow data $\hat{Q}$ to be matched is given by the flow Q^N corresponding to the forebody shape $\hat{\Gamma}$ described above. We point out that the artificial dissipation in the flow solver produces a "smearing" effect on the flow variables. Therefore, based on the results for the duct design problem, we expect a sufficiently smooth approximate cost functional. Furthermore, the comparison of the sensitivities in Figs. 5 and 6 leads us to believe that the sensitivity equation approach may produce asymptotically consistent derivatives.



Iteration 0 Iteration 1 Iteration 2 Iteration 3

Iteration 4 Iteration 6 Iteration 8 Iteration 12

FIG. 7. Sensitivity equation method iterations.



Final Converged Solution Original Long Forebody (data to be matched)

FIG. 8. Comparison of optimal solution with data.

The sensitivity equation method was applied to the FBS design problem with initial values of the parameters: $M_a = 2.0$, $q^1 = 0.10$, and $q^2 = 0.15$. These parameters correspond to those used to generate $\hat{Q}$ (even though that forebody is longer). We present the iteration history in Table II. Observe that there is a drastic reduction in the approximate cost functional in the first three iterations. The iteration history for the x-component of momentum is given in Fig. 7. Note that the front end of the forebody simulator becomes more blunt during the first two iterations in which a stagnation region is set up in front of the FBS. This has the effect of moving the shock forward, which comes close to the shock location created by the long forebody. The remaining iterations are used to "fine tune" the solution near the FBS. The comparison of the optimal forebody simulator to the flow generated by the long forebody is displayed in Fig. 8. Notice that the shock location is the same in both flows.

6.5. *Comments*

While no rigorous proof of asymptotically consistent gradients has been shown for Euler equations, numerical evidence in [3] suggests that the gradients may indeed be asymptotically consistent. Similar numerical evidence exists for finite element approximations of the Navier–Stokes equations [5].

ACKNOWLEDGMENTS

This research was supported in part by the Air Force Office of Scientific Research under Grants F49620-92-J-0078, F49620-93-1-0280, and F49620-96-1-0329, and by the National Aeronautics and Space Administration under Contract NASI-19480 while John Burns was a visiting scientist at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

REFERENCES

1. R. Beam and R. F. Warming, *J. Comput. Phys.* **22,** 87 (1976).
2. J. Borggaard, J. Burns, E. Cliff, and M. Gunzburger, Sensitivity calculations for a 2D, inviscid, supersonic forebody problem, in *Identification and Control of Systems Governed by Partial Differential Equations*, edited by H. T. Banks, R. Fabiano, and K. Ito, (SIAM, Philadelphia, PA, 1993), p. 14.
3. J. T. Borggaard, *The Sensitivity Equation Method for Optimal Design*, Ph.D. thesis, Virginia Tech, Blacksburg, VA, 1994.
4. J. T. Borggaard, On the presence of shocks in domain optimization of Euler Flows, in *Flow Control*, edited by M. Gunzburger, Proceedings of the IMA (Springer-Verlag, Berlin/New York, 1995). Vol. 68.
5. J. V. Burkardt, *Sensitivity Analyses and Computational Shape Optimization for Incompressible Flows*, Ph.D. thesis, Virginia Tech, Blacksburg, VA, 1995.
6. J. A. Burns, B. B. King, and Y.-R. Ou, A computational approach to sensor/actuator location for feedback control of fluid flow systems, in *Sensing, Actuation, and Control in Aeropropulsion*, edited by J. D. Paduano, Proceedings of the International Society for Optical Engineering (1995), Vol. 2494, p. 60.
7. R. G. Carter, *Numerical Optimization in Hilbert Space Using Inexact Function and Gradient Evaluations*, Technical Report 89-45, ICASE, 1989.
8. R. G. Carter, *SIAM J. Numer. Anal.* **28,** 251 (1991).
9. G. K. Cooper and J. R. Sirbaugh, *PARC Code: Theory and Usage*, Technical Report AEDC-TR-89-15, Arnold Engineering Development Center, Arnold AFB, TN, 1989.
10. J. E. Dennis Jr., D. M. Gay, and R. E. Welsch, *Trans. Math. Software* **7,** 348 (1981).
11. J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).
12. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide* (Academic Press, San Diego, CA, 1988).
13. P. D. Frank and G. R. Shubin, *J. Comput. Phys.* **98,** 74 (1992).
14. D. M. Gay, *SIAM J. Sci. Statist. Comput.* **2,** 186 (1981).
15. R. Glowinski, T.-W. Pan, A. J. Kearsley, and J. Periaux, *Int. J. Numer. Methods Fluids* **20,** 695 (1995).
16. A. Griewank, *SIAM J. Numer. Anal.* **24,** 684 (1987).
17. E. J. Haug, K. K. Choi, and V. Komkov, *Design Sensitivity Analysis of Structural Systems*, Mathematics in Science and Engineering (Academic Press, New York, 1986), Vol. 177.
18. D. Huddleston, *Development of a Free-Jet Forebody Simulator Design Optimization Method*, Technical Report AEDC-TR-90-22, Arnold Engineering Development Center, Arnold AFB, TN, 1990.
19. A. Jameson, *J. Sci. Comput.* **3,** 233 (1988).
20. A. Jameson and J. Reuther, Control theory based airfoil design using the Euler equations, in *Proceedings from the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994*, p. 206.
21. C.-Y. Joh, B. Grossman, and R. T. Haftka, *Eng. Opt.* **21,** 1 (1993).
22. R. Narducci, B. Grossman, and R. Haftka, *Inverse Prob. Eng.* **2,** 49 (1995).
23. N. Pagaldipti and A. Chattopadhyay, A discrete semi-analytical procedure for aerodynamic sensitivity analysis including grid sensitivity, in *Proceedings from the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994*, p. 161.
24. O. Pironneau, *J. Fluid Mech.* **59,** 117 (1973).
25. O. Pironneau, *J. Fluid Mech.* **60,** 97 (1974).
26. A. Shenoy and E. M. Cliff, An optimal control formulation for a flow matching problem, in *Proceedings from the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994*, p. 520.
27. A. C. Taylor III, G. W. Hou, and V. M. Korivi, A methodology for determining aerodynamic sensitivity derivatives with respect to variation of geometric shape, in *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, 1991*, AIAA paper 91-1101.
28. A. C. Taylor III, G. W. Hou, and V. M. Korivi, Sensitivity analysis, approximate analysis and design optimization for internal and external viscous flows, in *Proceedings of AIAA Aircraft Design Systems and Operations Meeting, 1991*, AIAA paper 91-3083.
29. J. Thompson, Z. Warsi, and C. Mastin, *Numerical Grid Generation: Foundations and Applications* (North-Holland, New York, 1985).