

NASA Contractor Report 191598

ICASE Report No. 94-8

1N-64

311

p-35

ICASE

A SENSITIVITY EQUATION APPROACH TO SHAPE OPTIMIZATION IN FLUID FLOWS

Jeff Borggaard
John Burns

(NASA-CR-191598) A SENSITIVITY
EQUATION APPROACH TO SHAPE
OPTIMIZATION IN FLUID FLOWS Final
Report (ICASE) 35 p

N94-27415

Unclas

Q3
1/64 0000311

NASA Contract Nos. NAS1-19480, NAS1-18605
January 1994

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

A SENSITIVITY EQUATION APPROACH TO SHAPE OPTIMIZATION IN FLUID FLOWS

*Jeff Borggaard*¹

*John Burns*²

Interdisciplinary Center for Applied Mathematics
Department of Mathematics
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

ABSTRACT

In this paper we apply a sensitivity equation method to shape optimization problems. An algorithm is developed and tested on a problem of designing optimal forebody simulators for a 2D, inviscid supersonic flow. The algorithm uses a BFGS/Trust Region optimization scheme with sensitivities computed by numerically approximating the linear partial differential equations that determine the flow sensitivities. Numerical examples are presented to illustrate the method.

¹Supported in part by the Air Force Office of Scientific Research under grant F49620-92-J-0078.

²Supported in part by the Air Force Office of Scientific Research under grants F49620-92-J-0078 and F49620-93-1-0280, the National Science Foundation under grant INT-89-22490 and by the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-19480 and NAS1-18605 while the author was a visiting scientist at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

The development of practical computational methods for optimization based design and control often relies on cascading simulation software into optimization algorithms. Black-box methods are examples of this approach. Although the precise form of the overall “optimal design” (OD) algorithm may change, there is an often unstated assumption that properly combining the “best” simulation algorithm with the “best” optimization scheme will produce a good OD algorithm. There are many examples to show that in general this assumption is not valid. However, in many cases it is a valid assumption and often this approach is the only practical way of attacking complex optimal design problems. If one uses this cascading approach, then it is still important to carefully pass information between the simulation and the optimizer. Typically, one uses a simulation code to produce a finite dimensional model and this discrete model is then used to supply approximate function evaluations to the optimization algorithm. Moreover, the approximate functions are then differentiated to supply gradients needed by the optimizer. Although there are numerous variations on this theme, they all may be formulated as “approximate-then-optimize” approaches. There are other approaches that first formulate the problem as an infinite dimensional optimization problem and then use numerical schemes to approximate the optimal design. All-at-once, one-shot and adjoint methods are examples of this “optimize-then-approximate” approach. Regardless of which approach one chooses, some type of approximation must be introduced at some point in the design process.

The sensitivity equation (SE) method is an approach that views the simulation scheme as a device to produce approximations of both the function and the sensitivities. The basic idea is to produce approximations of the infinite dimensional sensitivities and to pass these “approximate derivatives” to the optimizer along with the approximate function evaluations. There are several theoretical and practical issues that need to be considered when this approach is used. For example, there is no assurance that the SE method produces “consistent derivatives.” This will depend on the particular numerical scheme used to discretize the problem. However, the SE method allows one the option of using separate numerical schemes for flow solves and sensitivities, so that consistent derivatives can be forced. We shall not address these issues in this short paper. The goal here is to illustrate that a SE based method can be used with standard optimization schemes to produce a practical fast algorithm for optimal design. We concentrate on a particular application (the optimal forebody design problem) and use a specific iterative solver for the flow equations (PARC). Many flow solvers are iterative and for these types of codes, the SE method has perhaps the maximum potential for improving speed and accuracy.

In the next section we describe the forebody design problem and formulate the optimal design problem. In Sections 3 and 4 we review the derivation of the sensitivity equations and in Section 5 we discuss modifications to an existing simulation code that are needed in order to use that code for computing sensitivities. In Section 6, we present numerical results for the optimal design problem and Section 7 contains conclusions and suggestions for future work.

2 Optimal Design of a Forebody Simulator

This problem is a 2D version of the problem described in [1,4,8] . The Arnold Engineering Development Center (AEDC) is developing a free-jet test facility for full-scale testing of engines in various free flight conditions. Although the test cells are large enough to house the jet engines, they are too small to contain the full airplane forebody and engine. Thus, the effect of the forward fuselage on the engine inlet flow conditions must be “simulated.” One approach to solving this problem is to replace the actual forebody by a smaller object, called a “forebody simulator” (FBS), and determine the shape of the FBS that produces the best flow match at the engine inlet. The 2D version of this problem is illustrated in Figure 2.1 (see [1],[4], [8] and [9]).

The underlying mathematical model is based on conservation laws for mass, momentum and energy. For inviscid flow, we have that

$$\frac{\partial}{\partial t}Q + \frac{\partial}{\partial x}F_1 + \frac{\partial}{\partial y}F_2 = 0 \quad (1)$$

where

$$Q = \begin{pmatrix} \rho \\ m \\ n \\ E \end{pmatrix}, \quad F_1 = \begin{pmatrix} m \\ mu + P \\ mv \\ (E + P)u \end{pmatrix} \quad \text{and} \quad F_2 = \begin{pmatrix} n \\ nu \\ nv + P \\ (E + P)v \end{pmatrix}. \quad (2)$$

The velocity components u and v , the pressure P , the temperature T , and the Mach number M are related to the conservation variables, i.e., the components of the vector Q , by

$$\begin{aligned} u &= \frac{m}{\rho}, & v &= \frac{n}{\rho}, & P &= (\gamma - 1) \left(E - \frac{1}{2}\rho(u^2 + v^2) \right), \\ T &= \gamma(\gamma - 1) \left(\frac{E}{\rho} - \frac{1}{2}(u^2 + v^2) \right) & \text{and} & & M^2 &= \frac{u^2 + v^2}{T}. \end{aligned} \quad (3)$$

At the inflow boundary, we want to simulate a free-jet, so that we specify the total pressure P_0 , the total temperature T_0 and the Mach number M_0 . We also set $v = 0$ at the

inflow boundary. If u_I , P_I and T_I denote the inflow values of the x-component of the velocity, the pressure and the temperature, these may be recovered from P_0 , T_0 and M_0 by

$$T_I = \frac{T_0}{(1 + \frac{\gamma-1}{2} M_0^2)}, \quad P_I = \frac{P_0}{(1 + \frac{\gamma-1}{2} M_0^2)^{\frac{\gamma}{\gamma-1}}} \quad \text{and} \quad u_I^2 = M_0^2 T_I = \frac{M_0^2 T_0}{(1 + \frac{\gamma-1}{2} M_0^2)}. \quad (4)$$

The components of Q at the inflow may then be determined from (4) through the relations

$$\rho_I = \frac{\gamma P_I}{T_I}, \quad m_I = \rho_I u_I, \quad n_I = 0 \quad \text{and} \quad E_I = \frac{P_I}{\gamma-1} + \rho_I \frac{u_I^2}{2}. \quad (5)$$

The forebody is a solid surface, so that the normal component of the velocity vanishes, i.e.,

$$u\eta_1 + v\eta_2 = 0 \quad \text{on the forebody}, \quad (6)$$

where η_1 and η_2 are the components of the unit normal vector to the boundary. Note that we impose (6) on the velocity components u and v , and not on the momentum components m and n . Insofar as the state is concerned, it is clear that it does not make any difference whether (6) is imposed on m and n or on u and v , since $m = \rho u$ and $n = \rho v$ and $\rho \neq 0$. It can be shown that it does not make any difference to the sensitivities as well.

Assume that at $x = \beta$ the desired steady state flow $\hat{Q} = \hat{Q}(y)$ is given as data on the line (called the Inlet Reference Plane)

$$IRP = \{(x, y) | x = \beta, \sigma \leq y \leq \delta\}.$$

Also, we assume here that the inflow (total) Mach number M_0 can be used as a design (control) variable along with the shape of the forebody. Let the forebody be determined by the curve $\Gamma = \Gamma(x)$, $\alpha \leq x \leq \beta$ and let $p = (M_0, \Gamma(\cdot))$. The problem can be stated as the following optimization problem:

Problem FBS. Given data $\hat{Q} = \hat{Q}(y)$ on the *IRP*, find the parameters $p^* = (M_0^*, \Gamma^*(\cdot))$ such that the functional

$$\mathcal{J}(p) = \frac{1}{2} \int_{\sigma}^{\delta} \| Q_{\infty}(\beta, y) - \hat{Q}(y) \|^2 dy \quad (7)$$

is minimized, where $Q_{\infty}(x, y) = Q_{\infty}(x, y, p)$ is the solution to the steady state Euler equations

$$G(Q, p) = \frac{\partial}{\partial x} F_1 + \frac{\partial}{\partial y} F_2 = 0. \quad (8)$$

In the FBS design problem, the data \hat{Q} is generated both experimentally and numerically. In particular, the full airplane forebody (which is longer and larger than the desired FBS) is used to generate the data. Since the FBS is “constrained” to be shorter and smaller, we shall

consider the optimization problem illustrated in Figure 2.2 below. The data \hat{Q} is generated by solving (1)-(6) for the long forebody in Figure 2.2-(a) and the problem is to find p^* to minimize \mathcal{J} where the shortened FBS is constrained to be one half the length of the “real forebody.” This problem provides a realistic test of the optimal design algorithm in that the data can not be fitted exactly. Also, we note that we have a problem with shocks in the flow field. As shown in [2], optimization of flows with shocks can be difficult and requires some understanding of the impact that shocks have on the smoothness of the cost functional.

Clearly the statement of the problem is not complete. For example, one should carefully specify the set of admissible curves $\Gamma(\cdot)$ and questions remain about existence, uniqueness and integrability of “the” solution Q_∞ . We will not address these issues in this short paper.

Most optimization based design methods require the computation of the derivatives $\frac{\partial}{\partial p}Q_\infty(x, y, p)$. These derivatives are called sensitivities and various schemes have been developed to approximate the sensitivities numerically (see [7], [8], [10] and [11]). A common approach is to use finite differences. In particular, the steady state equation (8) is solved for \tilde{p} and again for $\tilde{p} + \Delta p$ and then $\frac{\partial}{\partial p}Q_\infty(x, y, \tilde{p})$ is approximated by $\frac{Q_\infty(x, y, \tilde{p} + \Delta p) - Q_\infty(x, y, \tilde{p})}{\Delta p}$. This method is often costly and can introduce large errors. Another approach is to first derive an equation (the sensitivity equation) for $\frac{\partial}{\partial p}Q_\infty(x, y, p)$ and then numerically solve this equation. We shall illustrate this approach for the forebody design problem. In the next two sections we derive the sensitivity equations. Although these derivations may be found in [3] we repeat them here for completeness.

3 Sensitivities with Respect to the Inflow Mach Number

First, we consider the design parameter M_0^2 . We will derive equations for the sensitivity

$$Q' \equiv \frac{\partial Q}{\partial M_0^2} \equiv \begin{pmatrix} \rho' \\ m' \\ n' \\ E' \end{pmatrix}, \quad (9)$$

where

$$\rho' \equiv \frac{\partial \rho}{\partial M_0^2}, \quad m' \equiv \frac{\partial m}{\partial M_0^2}, \quad n' \equiv \frac{\partial n}{\partial M_0^2} \quad \text{and} \quad E' \equiv \frac{\partial E}{\partial M_0^2}. \quad (10)$$

The differential equation system (1) has no explicit dependence on the design parameter M_0^2 , so that equations for the components of Q' are easily determined by formally differentiating (1) with respect to M_0^2 . The result is the system

$$\frac{\partial Q'}{\partial t} + \frac{\partial F'_1}{\partial x} + \frac{\partial F'_2}{\partial y} = 0, \quad (11)$$

where

$$F'_1 = \begin{pmatrix} m' \\ mu' + m'u + P' \\ mv' + m'v \\ (E + P)u' + (E' + P')u \end{pmatrix} \quad \text{and} \quad F'_2 = \begin{pmatrix} n' \\ nu' + n'u \\ nv' + n'v + P' \\ (E + P)v' + (E' + P')v \end{pmatrix}, \quad (12)$$

and where,

$$u' = \frac{\partial u}{\partial M_0^2}, \quad v' = \frac{\partial v}{\partial M_0^2}, \quad P' = \frac{\partial P}{\partial M_0^2} \quad \text{and} \quad T' = \frac{\partial T}{\partial M_0^2}, \quad (13)$$

and where, through (3), the sensitivities (10) and (13) are related by

$$\begin{aligned} u' &= \frac{1}{\rho}m' - \frac{m}{\rho^2}\rho', & P' &= (\gamma - 1) \left(E' - \frac{1}{2}\rho'(u^2 + v^2) - \rho(uu' + vv') \right), \\ v' &= \frac{1}{\rho}n' - \frac{n}{\rho^2}\rho' & \text{and} \quad T' &= \gamma(\gamma - 1) \left(\frac{1}{\rho}E' - \frac{E}{\rho^2}\rho' - (uu' + vv') \right). \end{aligned} \quad (14)$$

Note that (11) is of the same form as (1), with a different flux vector. In particular, (11) is in conservation form. As a result of the fact that (11) is *linear* in the primed variables, and that by (14) u' , v' and P' are linear in the components of Q' , (11) is a linear system in the sensitivity (9), i.e., in the components of Q' .

Now, we need to discuss the boundary conditions for Q' . Except for the inflow conditions, all boundary conditions are independent of the design parameter M_0^2 . Thus, the latter may be differentiated with respect to M_0^2 to obtain boundary conditions for the sensitivities. For example, at the forebody where (6) holds, we simply would have that

$$u'\eta_1 + v'\eta_2 = 0 \quad \text{on the forebody.} \quad (15)$$

Similar operations yield boundary conditions for the sensitivities along symmetry lines, other solid surfaces and at the outflow boundary. Note that if instead of (6), one interprets the no penetration condition as one on the momentum, i.e., $m\eta_1 + n\eta_2 = 0$ on the forebody, then instead of (15) we would have that

$$m'\eta_1 + n'\eta_2 = 0 \quad \text{on the forebody} \quad (16)$$

which is seemingly different from (15). However, (6) and (14) can be used to show that

$$m'\eta_1 + n'\eta_2 = \rho(u'\eta_1 + v'\eta_2) + \rho'(u\eta_1 + v\eta_2) = \rho(u'\eta_1 + v'\eta_2) \quad (17)$$

so that, since $\rho \neq 0$, (15) and (16) are identical.

The inflow boundary conditions for the sensitivities may be determined by differentiating (4) and (5) with respect to the design parameter M_0^2 . Note that this parameter appears

explicitly in the right-hand-sides of the equations in (4) and (5). Without difficulty, one finds from (5) that

$$\begin{aligned} \rho'_I &= \frac{\gamma}{T_I} P'_I - \frac{\gamma P_I}{T_I^2} T'_I, & m'_I &= \rho_I u'_I + u_I \rho'_I, \\ n'_I &= 0 & \text{and} & \quad E'_I = \frac{1}{\gamma - 1} P'_I + \frac{1}{2} u_I^2 \rho'_I + \rho_I u_I u'_I, \end{aligned} \quad (18)$$

where, from (4),

$$\begin{aligned} T'_I &= - \left(\frac{\gamma - 1}{2} \right) \frac{T_0}{\left(1 + \frac{\gamma - 1}{2} M_0^2 \right)^2}, & P'_I &= - \left(\frac{\gamma}{2} \right) \frac{P_0}{\left(1 + \frac{\gamma - 1}{2} M_0^2 \right)^{\frac{2\gamma - 1}{\gamma - 1}}} \\ \text{and} \quad u'_I &= \frac{\sqrt{T_I}}{2M_0} + \frac{M_0}{2\sqrt{T_I}} T'_I = \frac{\sqrt{T_0}}{2M_0 \left(1 + \frac{\gamma - 1}{2} M_0^2 \right)^{3/2}} \left(1 + (\gamma - 1) M_0^2 \right). \end{aligned} \quad (19)$$

4 Sensitivities with Respect to the Forebody Design Parameters

We assume that the forebody is described in terms of a finite number of design parameters which we denote by P_k , $k = 1, \dots, K$, and that the forebody may be described by the relation

$$y = \Phi(x; P_1, P_2, \dots, P_K), \quad \alpha \leq x \leq \beta \quad (20)$$

We express the dependence of the state variable Q on the coordinates and the design parameters by $Q = Q(t, x, y; M_0^2, P_1, P_2, \dots, P_K)$. We have already seen what equations can be used to determine the sensitivity of the state with respect to M_0^2 , i.e., for Q' . We now discuss what equations can be used to determine the sensitivities with respect to the forebody design parameters P_k , $k = 1, \dots, K$, i.e., for

$$Q_k \equiv \frac{\partial Q}{\partial P_k} \equiv \begin{pmatrix} \rho_k \\ m_k \\ n_k \\ E_k \end{pmatrix}, \quad (21)$$

where

$$\rho_k \equiv \frac{\partial \rho}{\partial P_k}, \quad m_k \equiv \frac{\partial m}{\partial P_k}, \quad n_k \equiv \frac{\partial n}{\partial P_k} \quad \text{and} \quad E_k \equiv \frac{\partial E}{\partial P_k}, \quad k = 1, \dots, K. \quad (22)$$

System (1) has no explicit dependence on the design parameters P_k , so that equations for the components of Q_k are easily determined by differentiating (1) with respect to P_k , $k = 1, \dots, K$. This produces the systems, $k = 1, \dots, K$, given by

$$\frac{\partial Q_k}{\partial t} + \frac{\partial F_{k1}}{\partial x} + \frac{\partial F_{k2}}{\partial y} = 0, \quad (23)$$

where

$$F_{k1} = \begin{pmatrix} m_k \\ mu_k + m_k u + P_k \\ mv_k + m_k v \\ (E + P)u_k + (E_k + P_k)u \end{pmatrix} \quad \text{and} \quad F_{k2} = \begin{pmatrix} n_k \\ nu_k + n_k u \\ nv_k + n_k v + P_k \\ (E + P)v_k + (E_k + P_k)v \end{pmatrix}, \quad (24)$$

and where,

$$u_k = \frac{\partial u}{\partial P_k}, \quad v_k = \frac{\partial v}{\partial P_k}, \quad P_k = \frac{\partial P}{\partial P_k} \quad \text{and} \quad T_k = \frac{\partial T}{\partial P_k}. \quad (25)$$

Moreover, by (3), the sensitivities (22) and (25) are related by

$$\begin{aligned} u_k &= \frac{1}{\rho} m_k - \frac{m}{\rho^2} \rho_k, & P_k &= (\gamma - 1) \left(E_k - \frac{1}{2} \rho_k (u^2 + v^2) - \rho (uu_k + vv_k) \right), \\ v_k &= \frac{1}{\rho} n_k - \frac{n}{\rho^2} \rho_k & \text{and} \quad T_k &= \gamma(\gamma - 1) \left(\frac{1}{\rho} E_k - \frac{E}{\rho^2} \rho_k - (uu_k + vv_k) \right), \end{aligned} \quad (26)$$

for $k=1, \dots, K$.

All boundary conditions except the one on the forebody also do not depend on the forebody design parameters P_k , $k = 1, \dots, K$. For example, consider the inflow boundary conditions (4)-(5). Differentiating these with respect to P_k , $k = 1, \dots, K$ yields that

$$\rho_{kI} = m_{kI} = n_{kI} = E_{kI} = T_{kI} = P_{kI} = u_{kI} = v_{kI} = 0 \quad (27)$$

at the inflow boundary. Now consider the boundary condition (6) on the forebody. We have that on the forebody

$$\frac{\eta_1}{\eta_2} = -\frac{\partial \Phi}{\partial x}. \quad (28)$$

Combining (6) and (28) we have that

$$u \frac{\partial \Phi}{\partial x} - v = 0 \quad (29)$$

along the forebody or, displaying the full functional dependence on the coordinates and design parameters, we have at a point (x, y) on the forebody, and at any time t ,

$$\begin{aligned} u \left(t, x, y = \Phi(x; P_1, P_2, \dots, P_K); M_0^2, P_1, P_2, \dots, P_K \right) \frac{\partial \Phi}{\partial x}(x; P_1, P_2, \dots, P_K) \\ - v \left(t, x, y = \Phi(x; P_1, P_2, \dots, P_K); M_0^2, P_1, P_2, \dots, P_K \right) = 0. \end{aligned} \quad (30)$$

We can proceed to differentiate (30) with respect to any of the forebody design parameters P_k , $k = 1, \dots, K$. The result is that, along the forebody for $k = 1, \dots, K$,

$$u_k \frac{\partial \Phi}{\partial x} - v_k = - \left(\frac{\partial u}{\partial y} \right) \left(\frac{\partial \Phi}{\partial P_k} \right) \left(\frac{\partial \Phi}{\partial x} \right) - u \frac{\partial}{\partial x} \left(\frac{\partial \Phi}{\partial P_k} \right) + \left(\frac{\partial v}{\partial y} \right) \left(\frac{\partial \Phi}{\partial P_k} \right), \quad (31)$$

where u , v and their derivatives are evaluated at the forebody ($x, y = \Phi(x)$).

If an iterative scheme is used to find a steady state solution of this system ((23), (27), (31)), then we assume that present guesses for the state variables u and v and their derivatives $\partial u/\partial y$ and $\partial v/\partial y$ and for the design parameters M_0^2 and P_k , $k = 1, \dots, K$, are known. It follows that the right-hand-side of (31) is known as well and equation (31), the boundary conditions along the forebody for the sensitivities with respect to the forebody design parameters, is merely an inhomogeneous version of (29), the boundary condition along the forebody for the state.

Let us now specialize to the type of forebodies considered by Huddleston, [8,9], i.e.,

$$\Phi(x; P_1, P_2, \dots, P_K) = \sum_{k=1}^K P_k \phi_k(x), \quad (32)$$

where $\phi_k(x)$, $k = 1, \dots, K$, are prescribed functions, e.g., Bezier curves (see [6]). In this case,

$$\frac{\partial \Phi}{\partial P_k} = \phi_k(x) \quad \text{and} \quad \frac{\partial}{\partial x} \left(\frac{\partial \Phi}{\partial P_k} \right) = \frac{d\phi_k}{dx}(x), \quad (33)$$

and

$$\frac{\partial \Phi}{\partial x} = \sum_{k=1}^K P_k \frac{d\phi_k}{dx}(x). \quad (34)$$

Combining (31)-(34), one obtains that, at any point $(x, \Phi(x))$ on the forebody and for each $k = 1, \dots, K$,

$$\left(\sum_{j=1}^K P_j \frac{d\phi_j}{dx} \right) u_k - v_k = - \left(\frac{\partial u}{\partial y} \right) \left(\sum_{j=1}^K P_j \frac{d\phi_j}{dx} \right) \phi_k - u \frac{d\phi_k}{dx} + \left(\frac{\partial v}{\partial y} \right) \phi_k. \quad (35)$$

For forebodies of the type (32), (35) gives the boundary conditions along the forebody for the sensitivities with respect to the forebody design parameters P_k , $k = 1, \dots, K$. It is now clear that, given guesses for the state variables u and v and their derivatives $\partial u/\partial y$ and $\partial v/\partial y$ and for the design parameters M_0^2 and P_k , $k = 1, \dots, K$, then the right-hand-side of (35) is known.

Consider now the problem of minimizing $\mathcal{J}(p)$ as defined above. Most optimization algorithms use gradient information. In particular, if P_k denotes one of the shape parameters, then the derivative

$$\frac{\partial}{\partial P_k} \mathcal{J}(\tilde{p}) = \int_{\sigma}^{\delta} < \left[\frac{\partial}{\partial P_k} Q_{\infty}(\beta, y, \tilde{p}) \right], Q_{\infty}(\beta, y, \tilde{p}) - \hat{Q}(y) > dy \quad (36)$$

may be required in the optimization loop. The sensitivity $\frac{\partial}{\partial P_k} Q_{\infty}(x, y, \tilde{p})$ satisfies the steady-state version of the sensitivity equations (23). In practice one must construct approximations to $\frac{\partial}{\partial P_k} Q_{\infty}(x, y, \tilde{p})$ and feed this information into the optimizer.

Assume that one has a particular simulation scheme (finite differences, finite elements, etc.) to approximate the flow $Q_\infty(x, y, \tilde{p})$ on a given grid, i.e.

$$Q_h(x, y, \tilde{p}) \rightarrow Q_\infty(x, y, \tilde{p}). \quad (37)$$

as the “step size” $h \rightarrow 0$. Given the design parameter \tilde{p} , one constructs a grid (depending on \tilde{p}) and then computes $Q_h(x, y, \tilde{p}) \approx Q_\infty(x, y, \tilde{p})$. This process may require some type of iterative scheme. We will address this issue below. In theory, one could use the same grid and computational scheme to approximate $\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p})$ so that one generates “approximate sensitivities”

$$\left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h \rightarrow \frac{\partial}{\partial P_k} Q_h(x, y, \tilde{p}) \quad (38)$$

as $h \rightarrow 0$. It is important to note that in general

$$\left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h \neq \frac{\partial}{\partial P_k} [Q_h(x, y, \tilde{p})], \quad (39)$$

i.e. this approach may not provide “consistent sensitivities”. However, some schemes do provide consistent derivatives and even if (39) holds, the error

$$ED_h = \left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h - \frac{\partial}{\partial P_k} [Q_h(x, y, \tilde{p})] \quad (40)$$

may be sufficiently small so that the optimization algorithm converges. Trust region methods are particularly well suited for problems of this type, where derivative information may contain (small) errors. As we shall see below, there are certain cases where $\left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h$ can be computed fast and accurately. Hence, the SE method provides estimates for sensitivities that may prove “good enough” for optimization and yet relatively cheap to compute. A comparison of $\left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h$ and various finite difference approximations of $\frac{\partial}{\partial P_k} [Q_h(x, y, \tilde{p})]$ may be found in [3].

It is important to note that the details of the computations needed to approximate a sensitivity are not the central issue here. For example, the sensitivity equations (11) and (23) are viewed as independent partial differential equations that must be solved by “some” numerical scheme. This scheme does not necessarily have to be the same scheme used to solve the flow equation (1), although as we shall see below, there are cases where using the same scheme is a useful approach.

Also, note that the sensitivity equations are derived for the problem formulated on the “physical” domain. If one uses a computational method that maps the problem to a computational domain (as does PARC), then the SE method does not require derivatives of this mapping. One simply maps the sensitivity equation (including the necessary boundary conditions), grids the computational domain, solves the resulting transformed equations

and then maps back to the physical domain. If, on the other hand, one mapped the flow equation (1) and derived a sensitivity equation in the computational domain, then to obtain the correct sensitivities one would have to compute the mapping sensitivity. Therefore, it is more efficient to derive the sensitivity equations in the physical domain.

Finally, we note that the SE method described here has one additional benefit. To compute a sensitivity, say $\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p})$, then one first selects the parameter value \tilde{p} , constructs a computational grid and solves for $\left[\frac{\partial}{\partial P_k} Q_\infty(x, y, \tilde{p}) \right]_h$. There is no need to compute grid sensitivities.

5 Computing Sensitivities using an Existing Code for the State

Suppose one has available a code to compute the state variables, i.e., to find approximate solutions of (1) along with boundary and initial conditions. In principle, it is an easy matter to amend such a code so that it can also compute sensitivities.

First, let us compare (1) with (11). If one wishes to amend the existing code that can handle (1) so that it can treat (11) as well, one has to change the definitions of the flux functions from those given in (2) to those given in (12). Note that the solution for the state is needed in order to evaluate the flux functions of (12).

Next, note that (11) and (23) are identical differential equations. Thus, the changes made to the code in order to treat (11) can also be used to treat (23). In fact, as long as the differential equation and any other part of the problem specification do not explicitly depend on the design parameters, the analogous relations will be the same for all the sensitivities.

The only changes that vary from one sensitivity calculation to another are those that arise from conditions in which the design parameters appear explicitly. In our example, for the sensitivity with respect to M_0^2 , one must change the portion of the code that treats the inflow conditions (4)-(5) so that it can instead treat (18)-(19). In the problem considered here, the nature (i.e. what variables are specified) of the boundary conditions at the inflow, and everywhere else, is not affected. Note that for the sensitivity with respect to M_0^2 , the boundary condition (15) on the forebody is the same as that for the state, given by (6).

For the sensitivities with respect to the forebody design parameters, the inflow boundary conditions simplify to (27), i.e., they become homogeneous. The boundary condition at the forebody is now given by (31) or (35). Once again, the nature of the boundary conditions is unchanged from that for the state and only the specified data is different. For the inflow boundary conditions, we may still specify the same conditions for the sensitivities, but now they would be homogeneous. The boundary conditions along the forebody change in that

they become inhomogeneous, (compare (29) and (35)).

In summary, to change a code for the state so that it also handles the sensitivities, one must redefine the flux functions in the differential equations, and the data in the boundary conditions. The changes necessary in the code to account for any particular relation that does not explicitly involve the design parameters are independent of which sensitivity one is presently considering.

The previous remarks are concerned only with the changes one must effect in a state code in order to handle the fact that one is discretizing a different problem when one considers the sensitivities. We have seen that these changes are not major in nature. However, there are additional changes that may be needed when one attempts to solve the discrete equations. In the numerical results presented below we use the finite difference code “PARC” (see [4] and [8]) to solve the state and sensitivity equations. However, the following comments apply equally well to other CFD codes of this type.

Since we are interested in steady design problems, the time derivative in (1) is considered only to provide a means for marching to a steady state. Now, suppose that at any stage of a Gauss-Newton, or other iteration, we have used PARC to find an approximate steady state solution of (1) plus boundary conditions. In order to do this, one has to solve a sequence of linear algebraic systems of the type

$$\left(I + \Delta t A(Q_h^{(n)})\right) Q_h^{(n+1)} = \left(Q_h^{(n)} + \Delta t B(Q_h^{(n)})\right), \quad n = 0, 1, 2, \dots, \quad (41)$$

where the sequence is terminated when one is satisfied that a steady state has been reached and where $Q_h^{(n)}$ denotes the discrete approximation to the state Q at the time $t = n\Delta t$. We denote this steady state solution for the approximation to the state by Q_h . One problem of the type (41) is solved for every time step. In (41), the matrix A and vector B arise from the spatial discretization of the fluxes and the boundary conditions. Both of these depend on the state at the previous time level.

Having computed a steady state solution by (41), the task at hand is now to compute the sensitivities. We will focus on Q' , the sensitivity with respect to the inflow Mach number. Analogous results hold for the sensitivities with respect to the forebody design parameters. Recall that given a state, the sensitivity equations are linear in the sensitivities. Therefore, if one is interested in the steady state sensitivities, instead of (11) one may directly treat its stationary version

$$\frac{\partial F'_1}{\partial x} + \frac{\partial F'_2}{\partial y} = 0. \quad (42)$$

Since (42) is linear in the components of Q' , one does not need to consider marching algorithms in order to compute a steady sensitivity. One merely discretizes (42) and solves the

resultant linear system, which has the form

$$\mathcal{A}(Q_h)Q'_h = \mathcal{B}(Q_h), \quad (43)$$

where Q'_h denotes the discrete approximation to the steady sensitivity. The matrix \mathcal{A} and vector \mathcal{B} differ from the A and B of (41) because we have discretized different differential equations and boundary conditions. Note that \mathcal{A} and \mathcal{B} in (43) depend only on the steady state Q_h and thus (43) is a *linear system of algebraic equations* for the discrete sensitivity Q'_h .

The cost of finding a solution of (43) is similar to that for finding the solution of (41) for a single value of n , i.e. for a single time step. The differences in the assembly of the coefficient matrices and right-hand-sides of (41) and (43) are minor. Thus, in theory at least, *one can obtain a steady sensitivity in the same computer time it takes to perform one time step in a state calculation*. If one wants to obtain all the sensitivities, e.g., $K + 1$ in our example, one can do so at a cost similar to , e.g., $K+1$ time steps of the state calculation. This is very cheap compared to the multiple state calculations necessary in order to compute sensitivities through the use of difference quotients.

Although (43) is in theory no more complex than one time step in (41), we can solve (42) by using the same iterative (or another) scheme. The simplest approach (but certainly not the optimal approach) is to use the PARC code to solve (42) by time marching. In particular, assume that $Q_h^{(n)}$ is a solution to (41), then the system

$$\left[I + \Delta t A'(Q_h^{(n)}) \right] (Q')_h^{(n+1)} = \left[(Q')_h^{(n)} + \Delta t B'(Q_h^{(n)}) \right] \quad (44)$$

can be used to find $(Q')_h^{(n+1)}$ given $(Q')_h^{(n)}$. Thus, one makes an initial guess for $Q_h^{(0)}$ and $(Q')_h^{(0)}$ and then iterates (41) and (44) simultaneously. Also, the same scheme can be used to compute any $Q_k = \frac{\partial Q}{\partial P_k}$, i.e.,

$$\left[I + \Delta t A'(Q_h^{(n)}) \right] (Q_k)_h^{(n+1)} = \left[(Q_k)_h^{(n)} + \Delta t B'(Q_h^{(n)}) \right]. \quad (45)$$

In practice, these “optimal” estimates of speed up are rarely achieved. Moreover, as noted above, it is important to note that finite difference (FD) and sensitivity equation (SE) methods do not necessarily produce the same results. Since the ultimate goal is to find useful and cheap gradients for optimization, the most important issue is whether or not the SE method combined with an optimization algorithm produces a convergent optimal design as fast as possible. We have tested this scheme on the forebody design problem and the next section contains a summary of these results.

6 An Optimal Design Example

In order to illustrate the SE method and to test its use in an optimization problem, we used the PARC code as described above to compute sensitivities and the used these sensitivities in a BFGS/Trust Region scheme to find an optimal shortened forebody simulator. As shown in Figure 2.2, data was generated by solving the Euler equations over the long forebody at a Mach number of 2.0. The objective is to find a forebody simulator with length one half of the long forebody and such that the resulting flow matches the data as well as possible, i.e. minimizes \mathcal{J} along the outflow boundary.

The shortened forebody was parameterized by a Bezier curve using two parameters. Thus, there are three design parameters $p = (M_0^2, P_1, P_2)$. The algorithm used in this numerical experiment was based on using the PARC code to simultaneously march to the steady state solutions of the flow and sensitivity equations. We made no attempt to optimize the algorithm since the main goal was to test for convergence.

The design algorithm proceeds as follows. First, an initial guess for the optimal design is made, i.e., we select a $p^0 = ((M_0^2)^0, P_1^0, P_2^0)$. A good selection of initial parameters can be made knowing the operating conditions of the aircraft and some rough guess of the shape from the aircraft forebody. In our example, we chose M_0^2 as the inlet Mach number from the computation which generated our data. The initial guess for the parameters were those used to generate the long forebody (although corresponding to different x-locations). These parameters, p^0 , are used to generate a grid, the inflow and forebody boundary conditions for both the flow (1) and sensitivity equations ((11) and (23)) and an initial guess for both $Q_h^{(0)}$ and $(\frac{\partial Q}{\partial p} Q)_h^{(0)}$. In our example, a rough guess for the flow field $Q_h^{(0)}$ uses the constant inflow boundary condition throughout the flow domain. Likewise, the initial guess for $(Q')_h^{(0)}$ is taken as the inflow boundary conditions (given in equation (18)) throughout the flow domain. The initial guess for $(Q_k)_h^{(0)}$ is initially taken as zero (except on the forebody). The systems (41), (44) and (45) are then solved simultaneously (in our case the left hand side matrix is the same for (41) as for the sensitivity equations (44) and (45), i.e. $A = A'$) for the updated $Q_h^{(1)}$, $(Q')_h^{(1)}$, $(\frac{\partial Q}{\partial P_1})_h^{(1)}$ and $(\frac{\partial Q}{\partial P_2})_h^{(1)}$. The updated $Q_h^{(n)}$ is then used to formulate (41), (44) and (45) and solve for $(Q_h)^{(n+1)}$ and $(\frac{\partial Q}{\partial p} Q)_h^{(n+1)}$. Then one iterates until the desired convergence is achieved. In our example, the residuals, $\Delta Q_h = [Q_h^{(n+1)} - Q_h^{(n)}]$ were converged to approximately 10^{-15} (in 800 time steps). The outflow data Q_h and $(\frac{\partial Q}{\partial p} Q)_h$ are then used to compute $\mathcal{J}(p^0)$ and $\nabla \mathcal{J}(p^0)$.

The optimization algorithm consisted of a BFGS secant method coupled with a “hook” step model trust region method [5]. The initial Hessian was obtained by finite differences on $\nabla \mathcal{J}(\hat{p})$. The function and gradient information needed by the optimization algorithm is

obtained by calling the modified PARC code with $p = \tilde{p}$.

This algorithm was tested for the case where the forebody simulator was allowed to have the full length of the body generating the data. In this case the optimization algorithm produced exact data fits, i.e. $\mathcal{J}(p^*) = 0$ and it recovered the parameters used to generate the data. However, the more realistic test (constraining the length of the forebody simulator) also produced a convergent design and reduced the cost functional significantly.

Figure 6.1 shows the flow field over the long forebody. Observe, that there is a shock in the flow. As noted in [2], shocks can cause difficulties if one is not careful in the selection of an appropriate numerical scheme. High order schemes can produce (numerically generated) local minimum that can cause the optimization loop to fail. This problem is avoided here because the numerical viscosity in PARC (required for stability) is sufficient to “smooth” the cost functional (see [2] for details).

Figure 6.2 shows the shape and flow field of the optimal shortened forebody. This design was obtained after 12 iterations of the optimization loop. Figures 6.3–6.6 show the 1st, 2nd, 3rd, 5th and 12th iterations for each of the flow variables. The initial guess for the parameters were

$$p^0 = \left((M_0^2)^0, P_1, P_2 \right) = (2.0, 0.10, 0.15)$$

and

$$\mathcal{J}(p^0) = 3.2339.$$

The “converged” optimal parameters are

$$p^* = p^{12} = (2.020, 0.294, 0.156)$$

with

$$\mathcal{J}(p^*) = 0.2229.$$

Observe that the cost function was decreased by more than 93%. Figures 6.7–6.10 show a comparison of the flow fields for the optimal shortened forebody simulator and the data. The optimization loops converged rapidly. For example, $\mathcal{J}(p^3) = 0.2334$ and $\mathcal{J}(p^5) = 0.2289$. This is due to the fact that the shock location was found quickly.

Note that although the flows are close, there is a significant error near the forebody. This can also be seen in the plots in Figures 6.11–6.14. It is worthwhile to note that the match is good considering the fact the shortened forebody is constrained to be one half the length of the “real” forebody and only two Bezier parameters are used to model $\Gamma(\cdot)$. It is also important to note that the shock is captured by the optimal design. In particular, observe in Figures 6.3–6.6 how the optimization algorithm “shapes” the shortened forebody so that the optimal shape has a blunt nose. This is necessary in order to generate the correct shock location at the outflow.

7 Conclusions

The numerical experiment above illustrates that the SE method can produce sensitivities suitable for optimization based design. There are a number of interesting theoretical issues that need to be addressed in order to analyze the convergence of this approach. Moreover, one should investigate “fast solvers” for the sensitivity equations (multi-grid, etc.) as well as develop numerical schemes that are not only fast, but produces consistent derivatives when possible.

Finally, we note that we have conducted a number of timing tests which compute sensitivities to compare the SE method with the finite difference method. In particular, we observed that for the problem above (with three design parameters), the SE method needed only 58% of the CPU time required by finite differencing. When twenty design parameters were used, the SE method produced these sensitivities in about 38% of the time required by finite differencing. These early numerical results indicate that considerable computational savings may be possible if one extends and refines the basic SE method presented here.

References

- [1] Beale, D. and Collier, M., *Validation of a Free-jet Technique for Evaluating Inlet-Engine Compatibility*, AIAA Paper 89-2325, AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference, Monterey, CA, July 1989.
- [2] Borggaard, J. T., *On the Presence of Shocks on Domain Optimization of Euler Flows, Sensitivity Calculations for a 2D, Inviscid, Supersonic Forebody Problem*, in Identification and Control of Distributed Parameter Systems, to appear.
- [3] Borggaard, J., Burns, J., Cliff, E. and Gunzburger, M., *Sensitivity Calculations for a 2D, Inviscid, Supersonic Forebody Problem*, in Identification and Control of Distributed Parameter Systems, to appear.
- [4] Cooper, G. and Phares, W., *CFD Applications in an Aerospace Engine Test Facility*, AIAA Paper 90-2003, AIAA/ASME/SAE/ASEE 26th Joint Propulsion Conference, Orlando, FL, July 1990.
- [5] Dennis Jr., J. E. and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1983.
- [6] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press, Inc., San Diego, CA, 1988.
- [7] Frank, P. and Shubin G., *A Comparison of Optimization-Based Approaches for a Model of Computational Aerodynamics Design Problem*, Journal of Computational Physics 98, (1992), pp.74-89.
- [8] Huddleston, D., *Aerodynamic Design Optimization Using Computational Fluid Dynamics*, Ph.D. Dissertation, University of Tennessee, Knoxville, TN, December, 1989.
- [9] Huddleston, D., *Development of a Free-Jet Forebody Simulator Design Optimization Method*, AEDC-TR-90-22, Arnold Engineering Development Center, Arnold AFB, TN, December, 1990.
- [10] Taylor III, A. C., Hou, G. W. and Korivi, V. M., *A Methodology for Determining Aerodynamic Sensitivity Derivatives With Respect to Variation of Geometric Shape*, Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, April 8-10, 1991, Baltimore, MD, AIAA Paper 91-1101.
- [11] Taylor III, A. C., Hou, G. W. and Korivi, V. M., *Sensitivity Analysis, Approximate Analysis and Design Optimization for Internal and External Viscous Flows*, AIAA Paper 91-3083, AIAA Aircraft Design Systems and Operations Meeting, Baltimore, MD, September 1991.

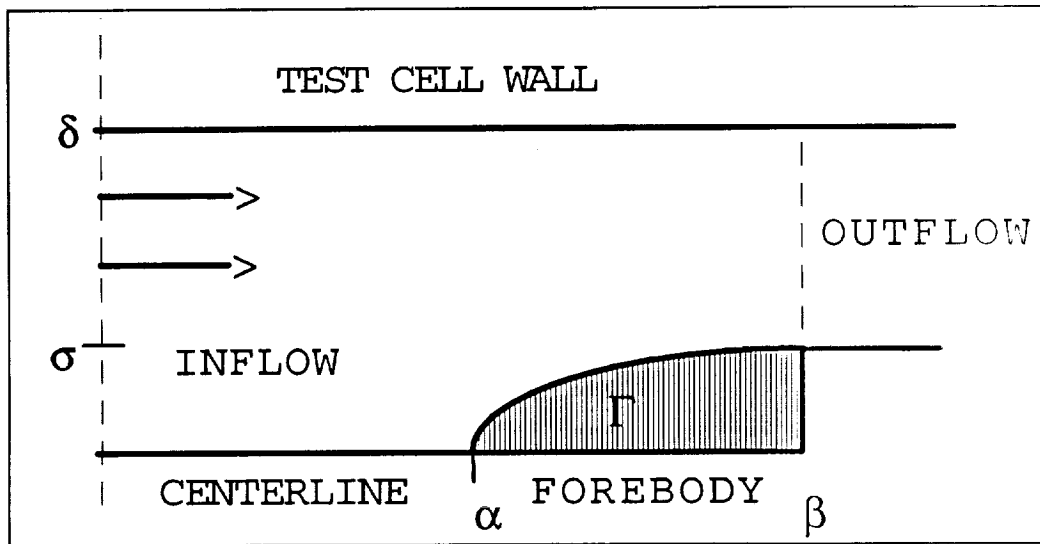


Figure 2.1: 2D Forebody Problem

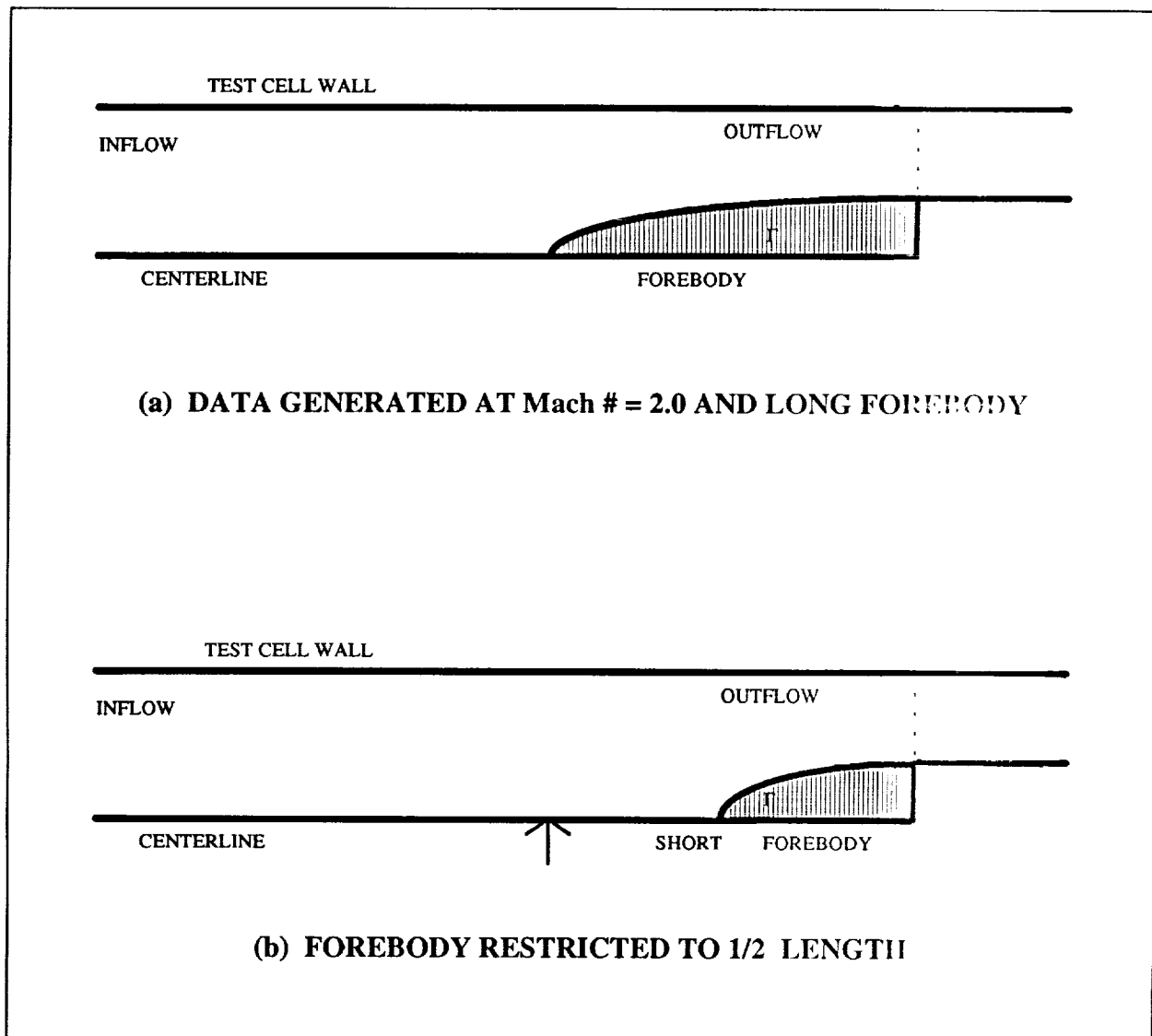


Figure 2.2: A 2D Optimal Forebody Design Problem

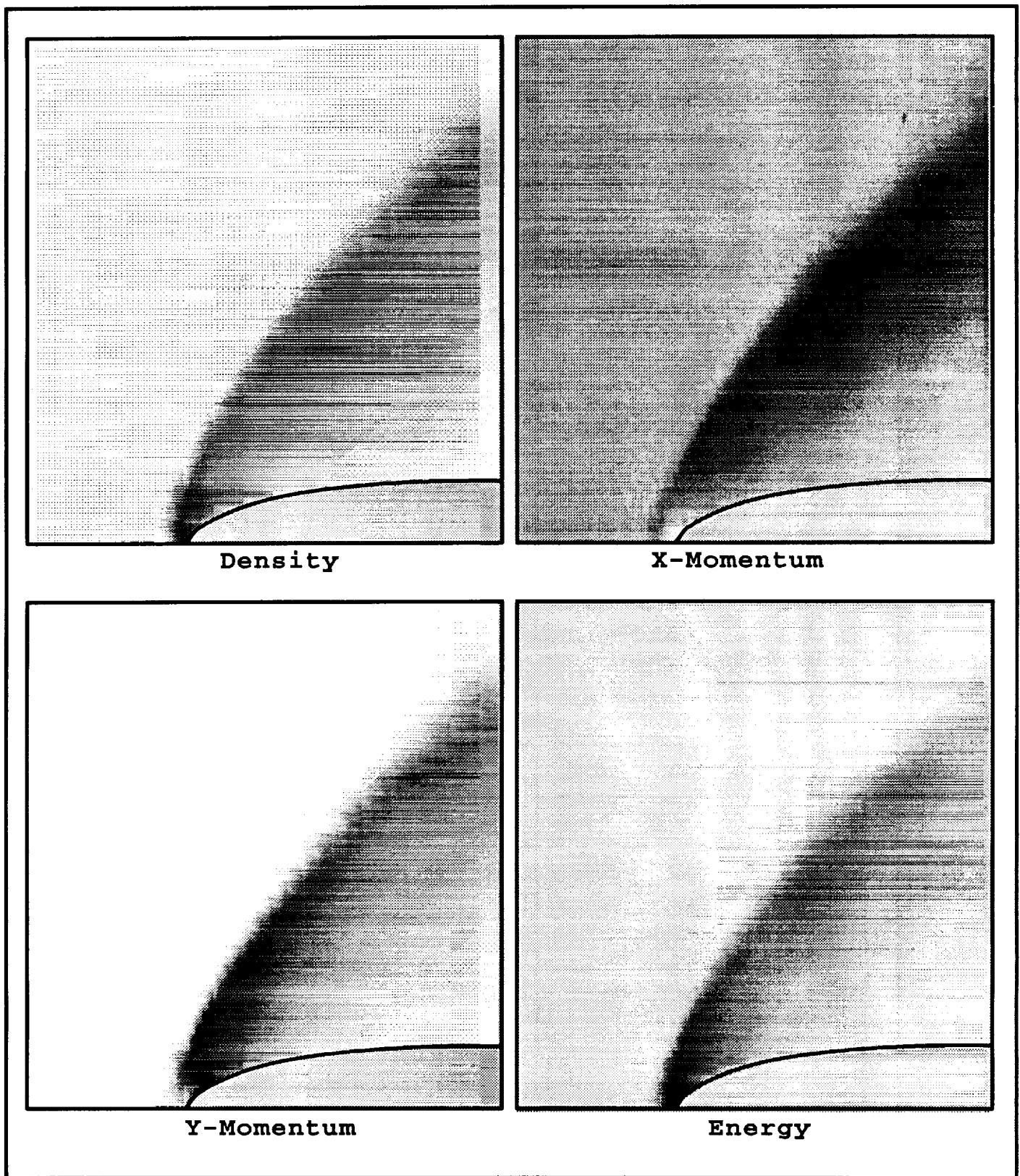


Figure 6.1: Long Forebody (Outflow Data to be Matched)

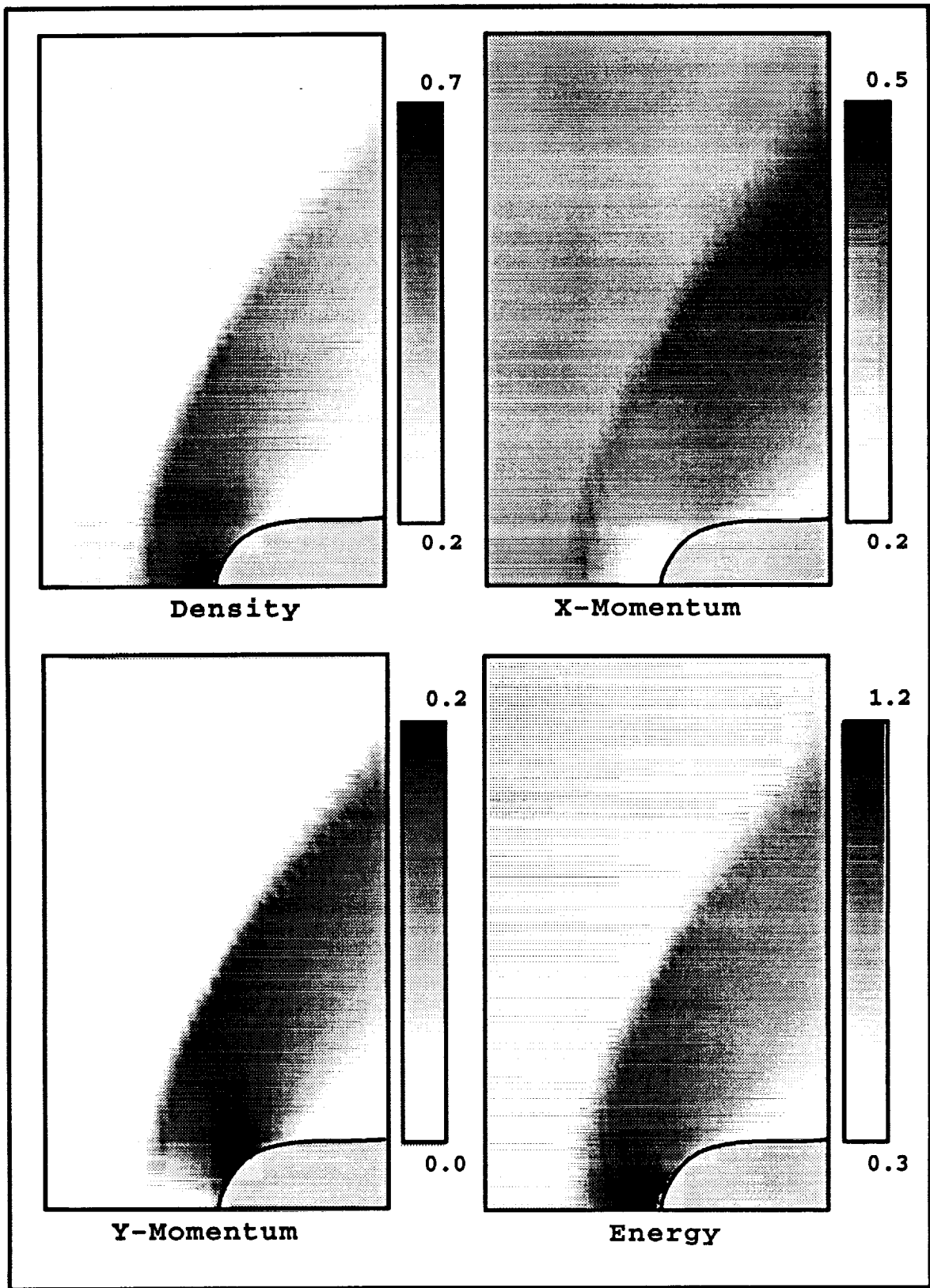
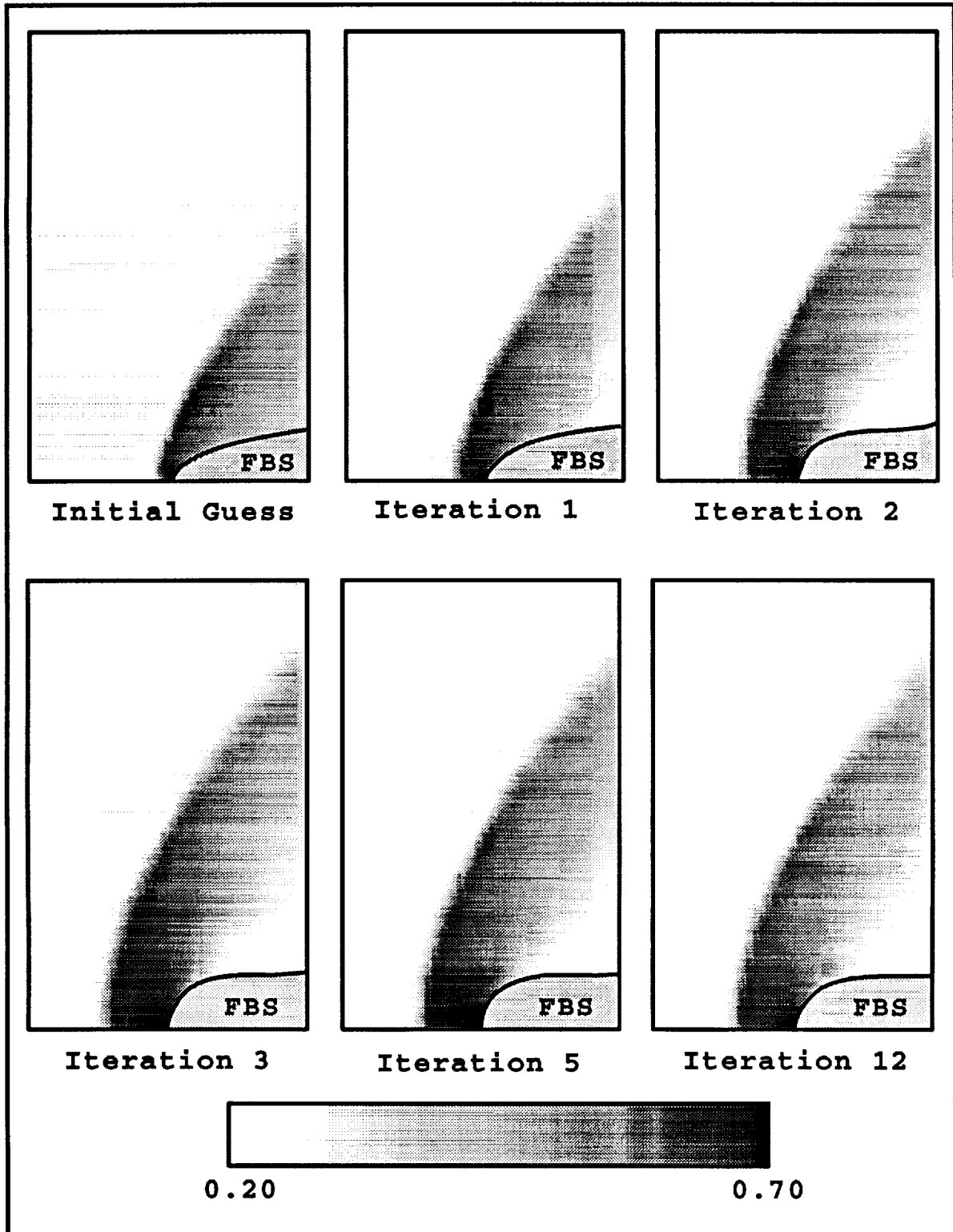
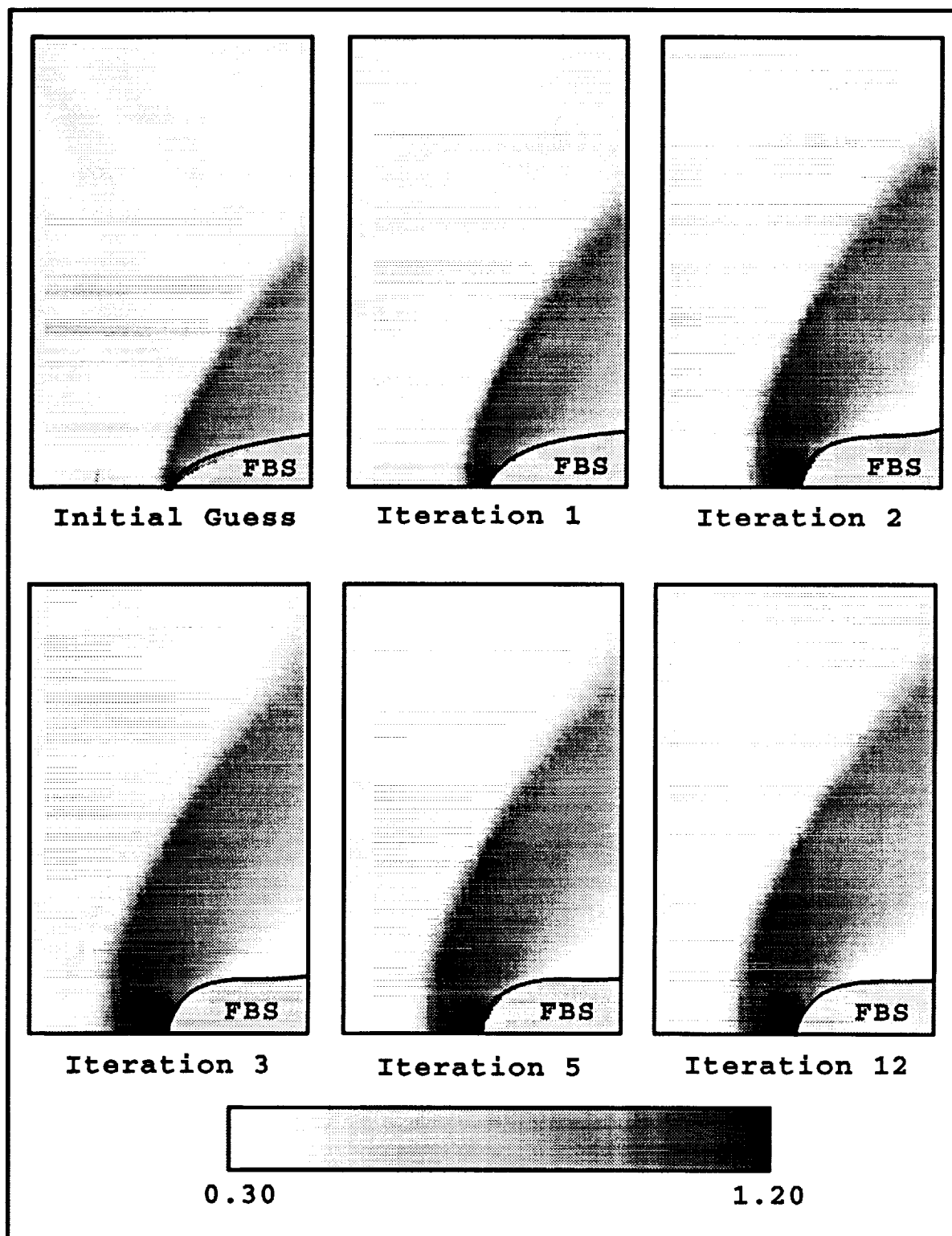


Figure 6.2: Optimal Shortened Forebody Flow



**Figure 6.3: Iteration to Optimal Forebody Design:
Density**



**Figure 6.4: Iteration to Optimal Forebody Design:
Energy**

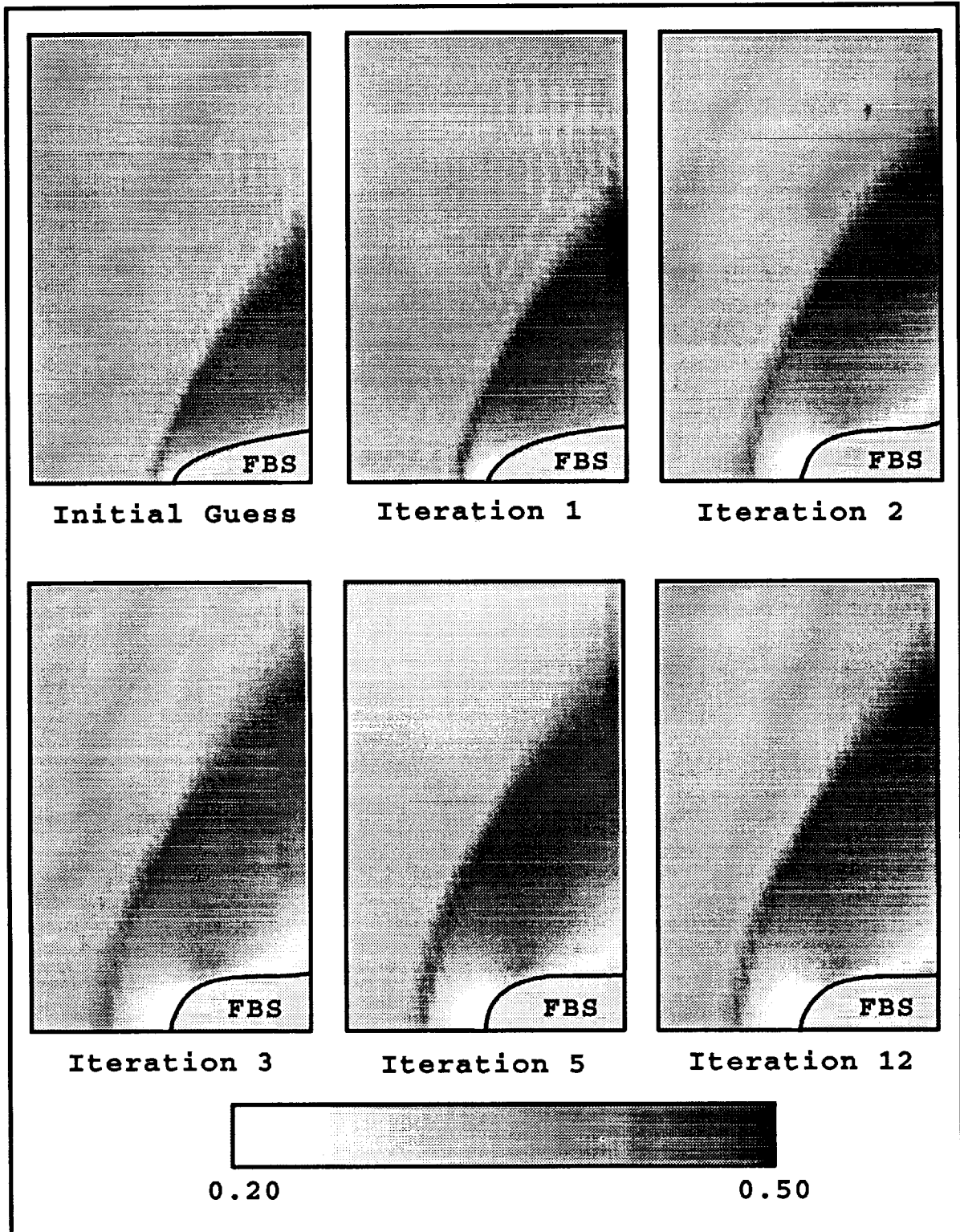


Figure 6.5: Iteration to Optimal Forebody Design:
X-Component of Momentum

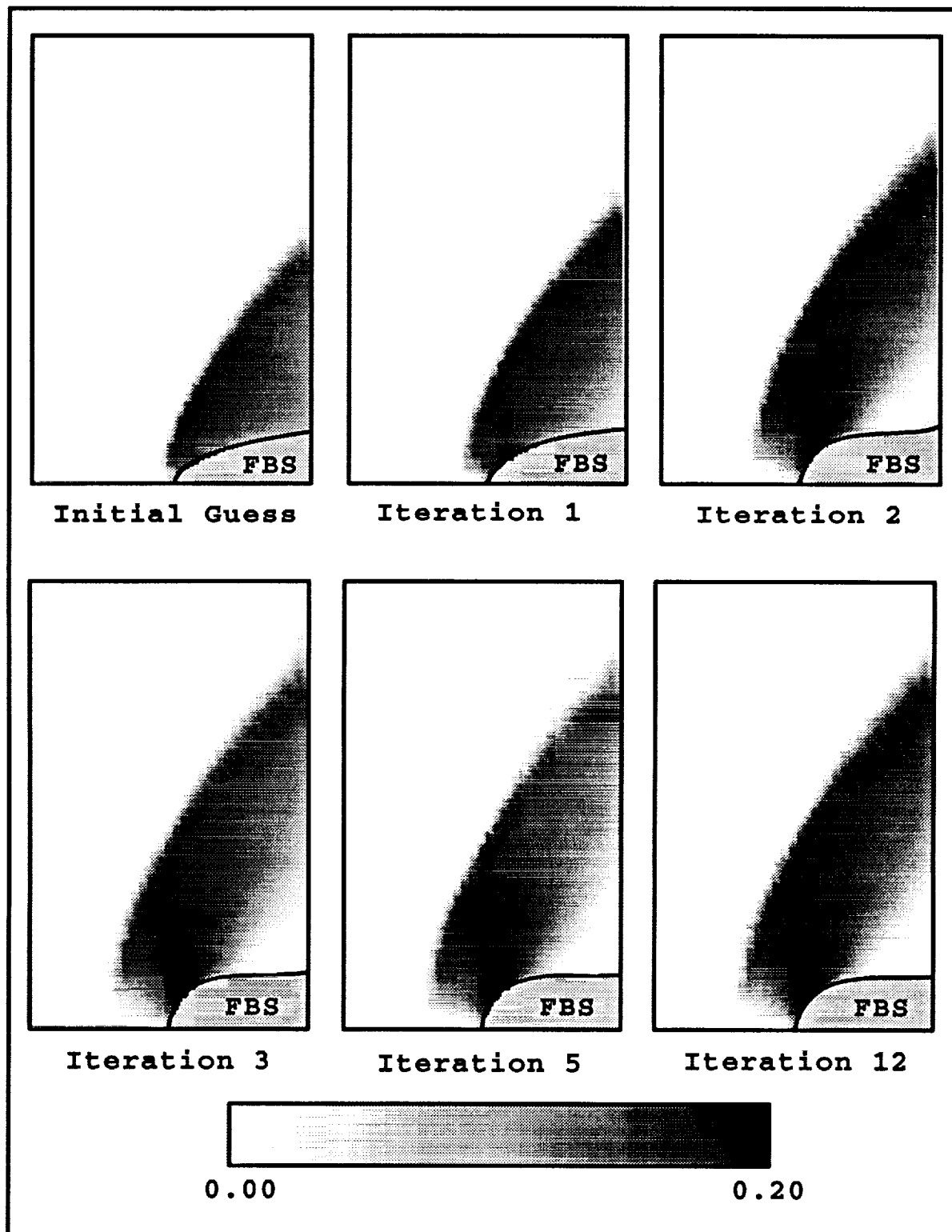


Figure 6.6: Iteration to Optimal Forebody Design:
Y-Component of Momentum

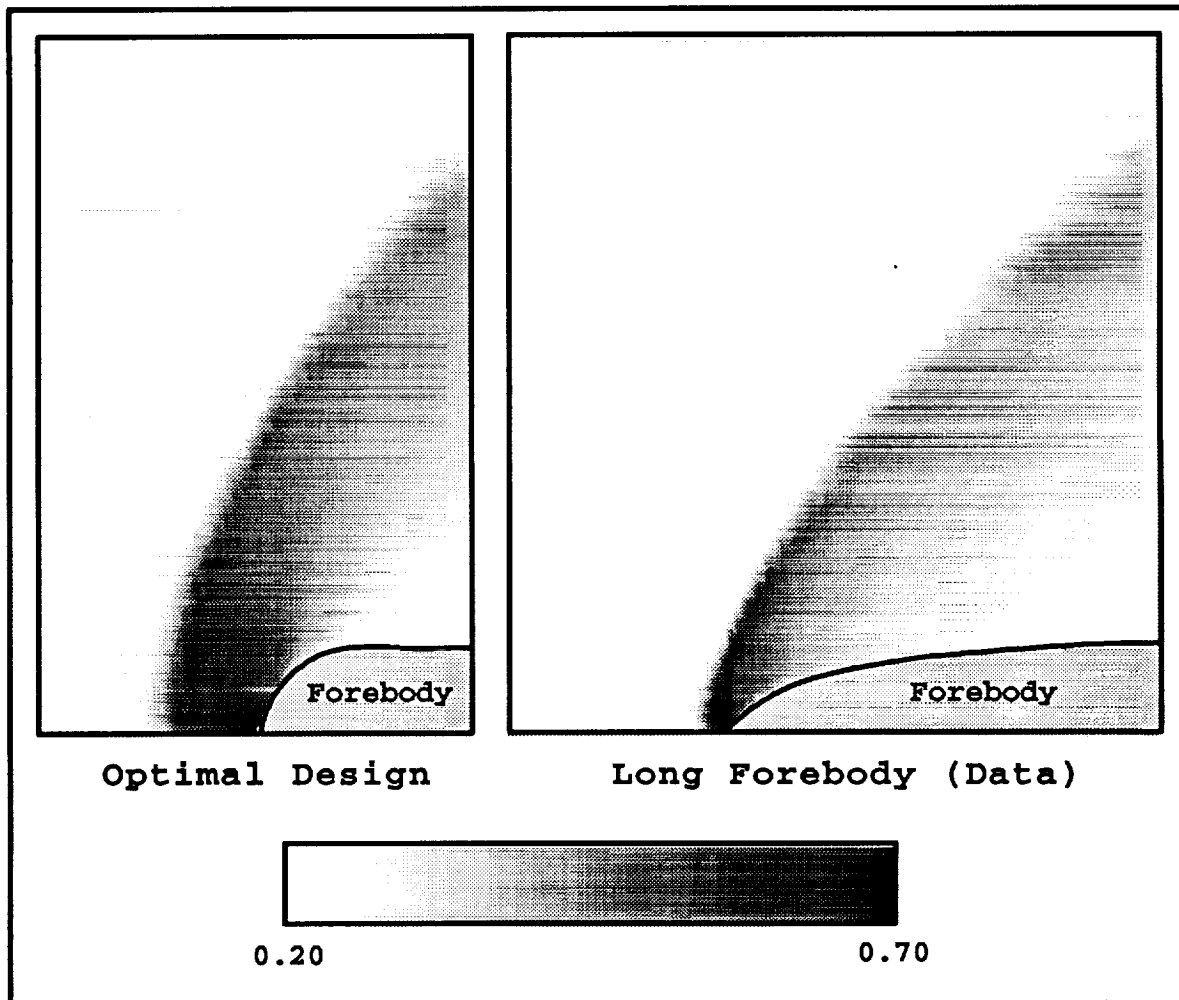


Figure 6.7: Comparison of Optimal Shortened and Long Forebody: Density

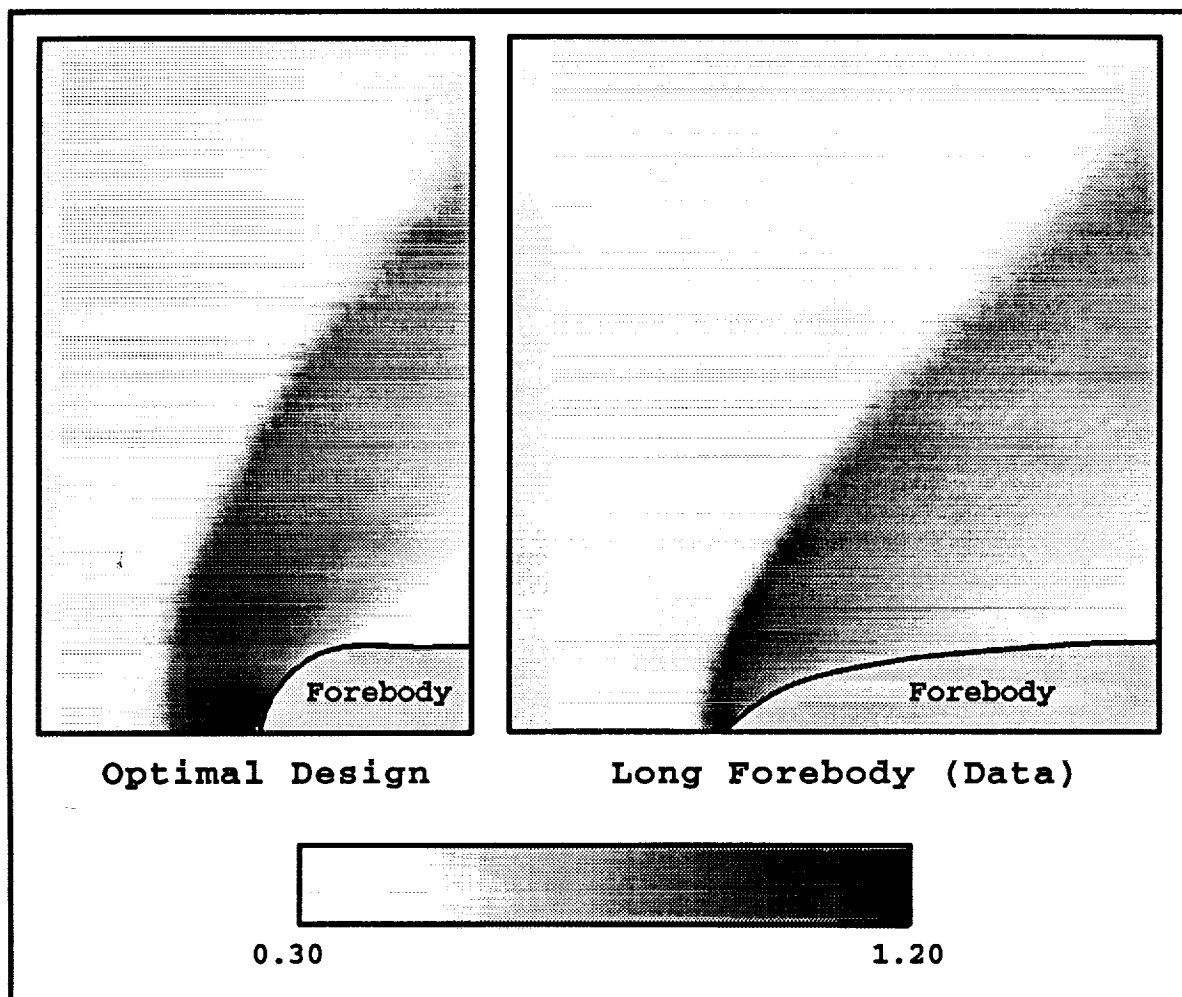


Figure 6.8: Comparison of Optimal Shortened and Long Forebody: Energy

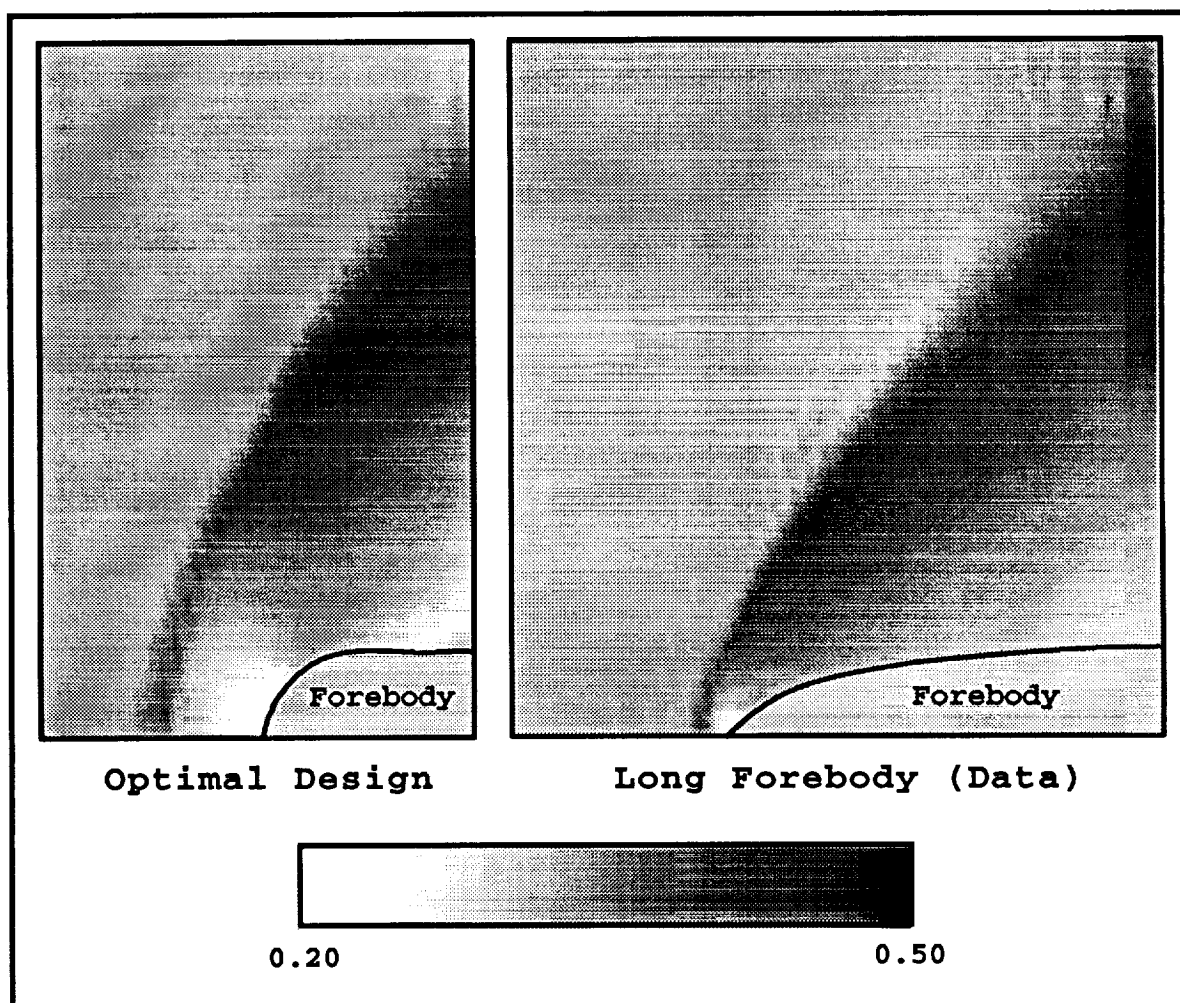


Figure 6.9: Comparison of Optimal Shortened and Long Forebody: X-Component of Momentum

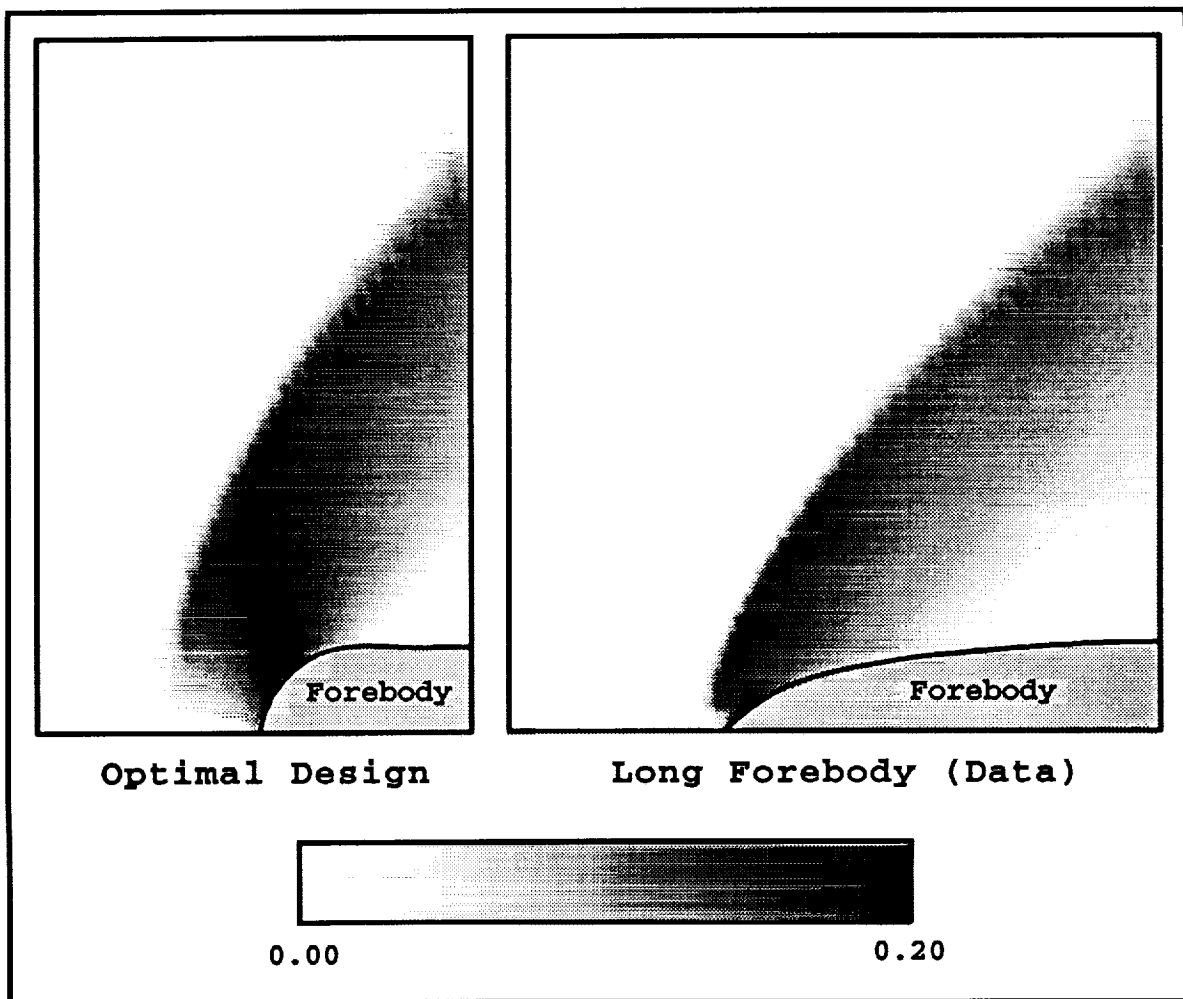
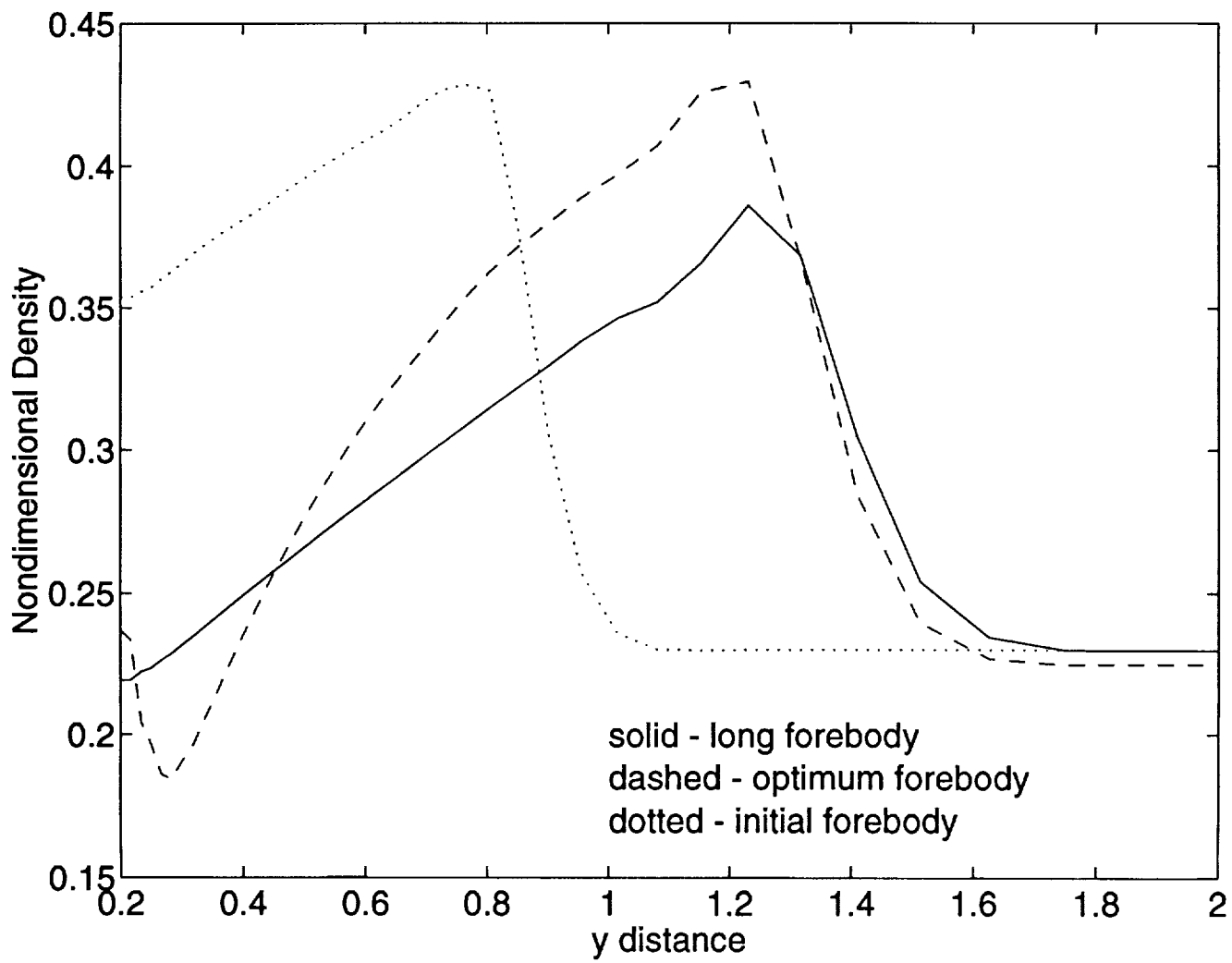
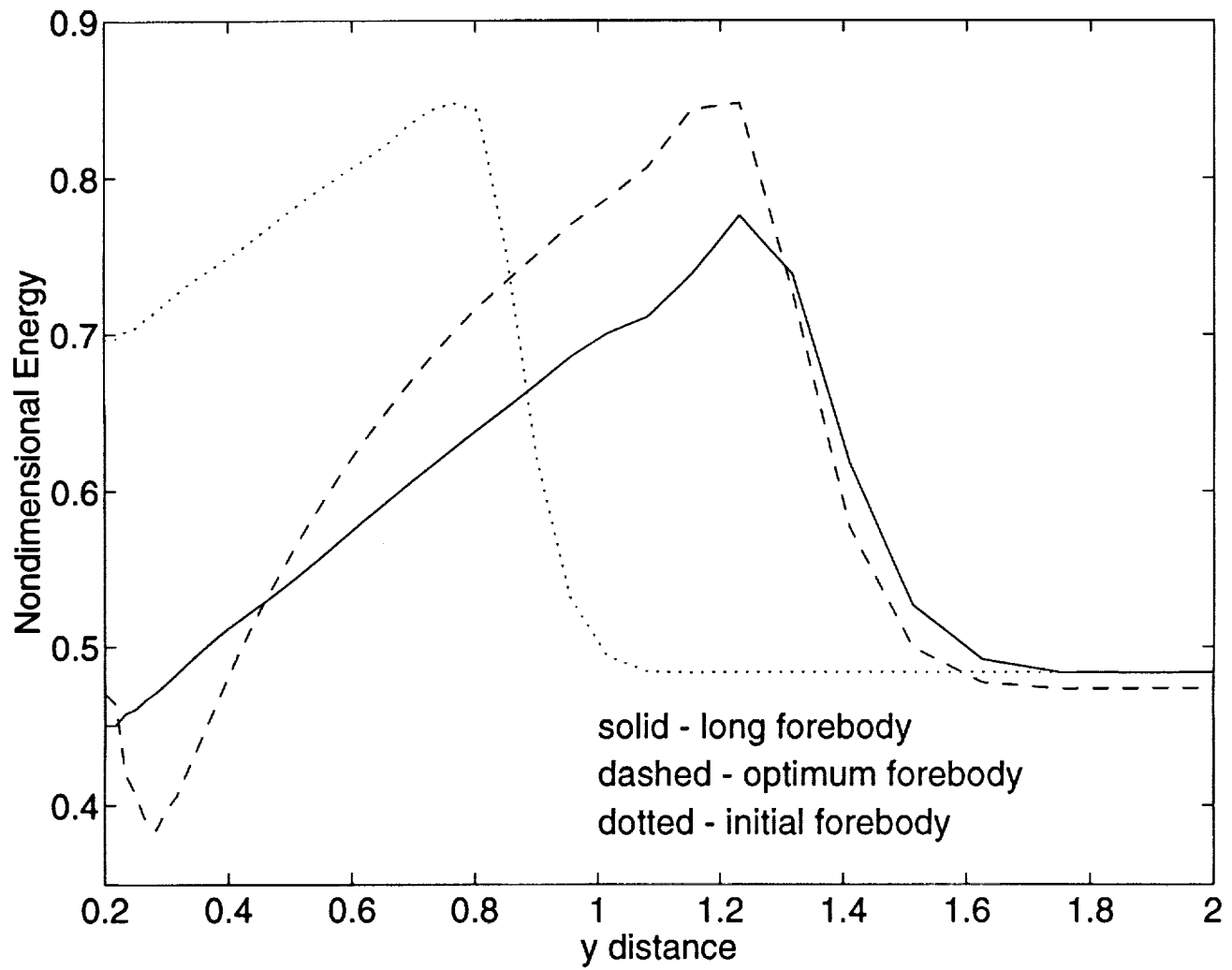
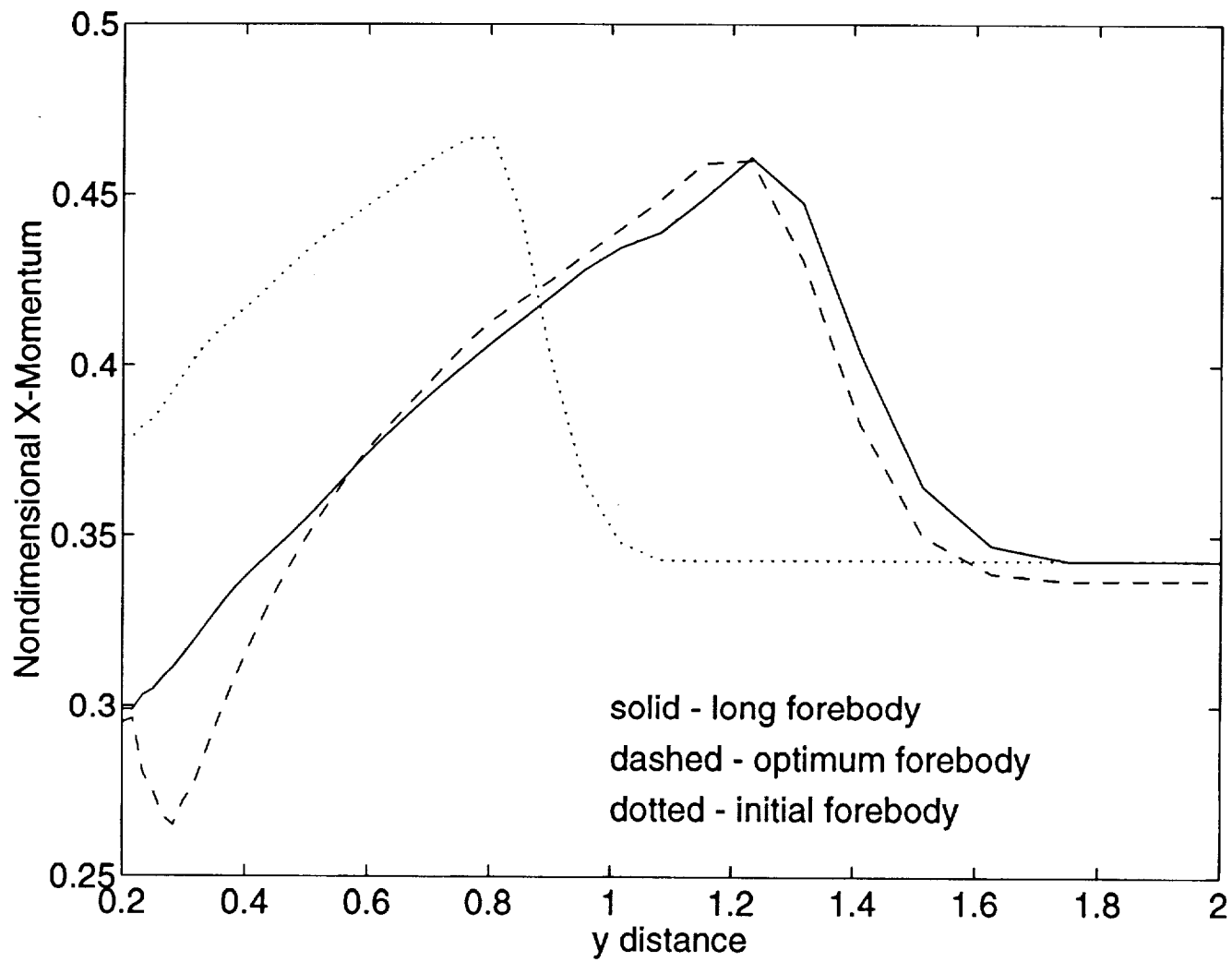
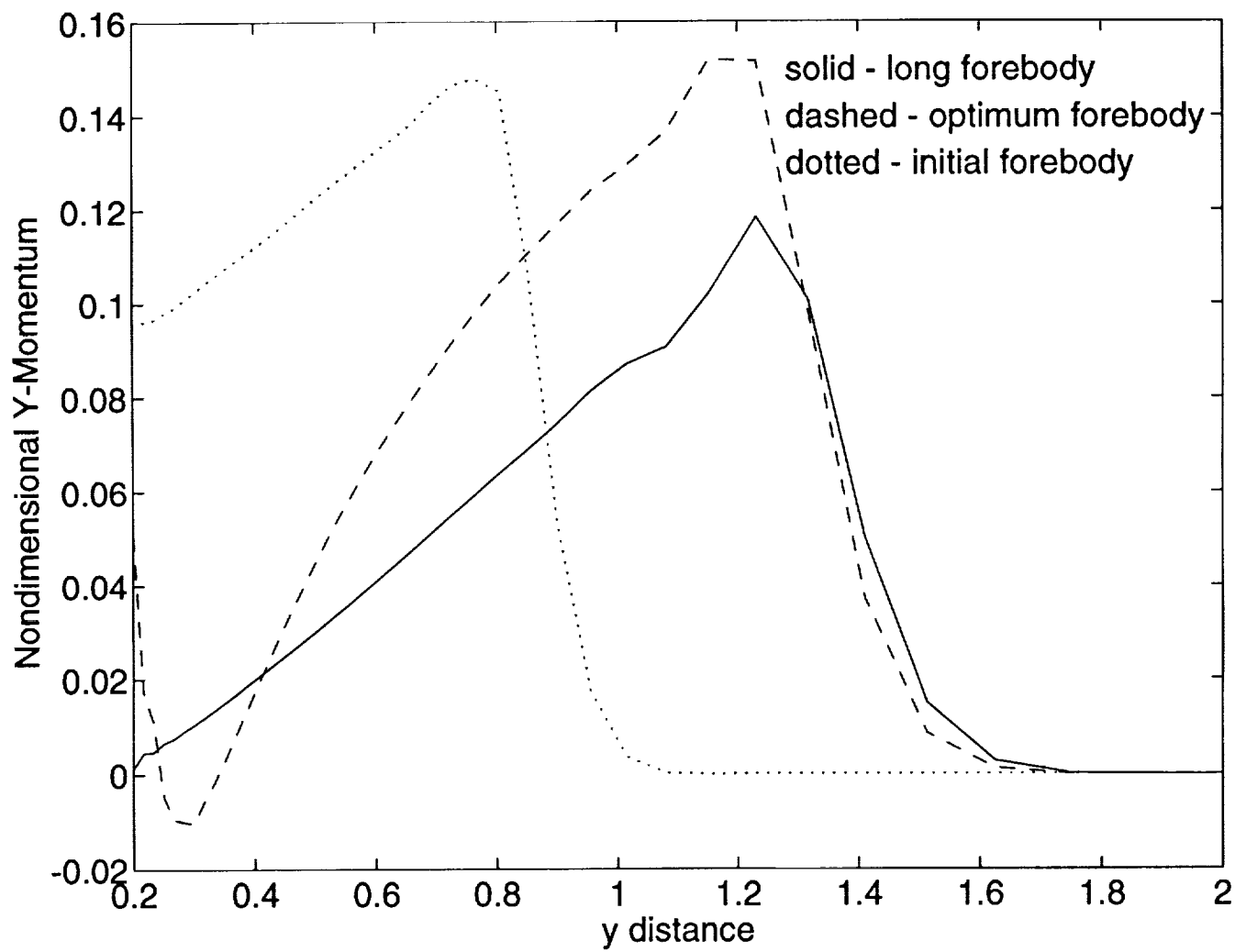


Figure 6.10: Comparison of Optimal Shortened and Long Forebody: Y-Component of Momentum









REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY(Leave blank)	2. REPORT DATE January 1994	3. REPORT TYPE AND DATES COVERED Contractor Report		
4. TITLE AND SUBTITLE A SENSITIVITY EQUATION APPROACH TO SHAPE OPTIMIZATION IN FLUID FLOWS		5. FUNDING NUMBERS C NAS1-19480 C NAS1-18605 WU 505-90-52-01		
6. AUTHOR(S) Jeff Borggaard John Burns				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		8. PERFORMING ORGANIZATION REPORT NUMBER ICASE Report No. 94-8		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-191598 ICASE Report No. 94-8		
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Michael F. Card Final Report To appear in IMA Workshop on Flow Control, Springer, 1993				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified Unlimited Subject Category 64		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) In this paper we apply a sensitivity equation method to shape optimization problems. An algorithm is developed and tested on a problem of designing optimal forebody simulators for a 2D, inviscid supersonic flow. The algorithm uses a BFGS/Trust Region optimization scheme with sensitivities computed by numerically approximating the linear partial differential equations that determine the flow sensitivities. Numerical examples are presented to illustrate the method.				
14. SUBJECT TERMS optimal design; shape optimization; sensitivity analysis		15. NUMBER OF PAGES 34		
		16. PRICE CODE A03		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

