INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 53–78 (1999)

SENSITIVITIES, ADJOINTS AND FLOW OPTIMIZATION

MAX D. GUNZBURGER*
Department of Mathematics, Iowa State University, Ames, IA 50011-2064, USA

SUMMARY

Several issues related to the calculation of flow sensitivities and the solution of flow optimization problems are considered. For the latter, one-shot Lagrange multiplier methods are presented, as well as sensitivity- and adjoint-based iterative algorithms. A sample application of each method to a specific flow optimization problem is provided. Then some difficulties associated with the practical implementation of the methods are discussed. Particular emphasis is placed on the effect of flow discontinuities on approximate sensitivities and adjoints. A discussion of these issues is given in the context of the Reimann problem for which exact information is known. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: flow sensitivity; adjoint-based iterative algorithms; Reimann problem

1. INTRODUCTION

As in any optimization setting, for a flow control or optimization problem we have the following ingredients:

- state variables: ϕ
- controls or design parameters: g
- constraints, e.g. state equations: $F(\phi, g) = 0$
- objective functional: $\mathscr{J}(\phi, g)$

which are necessary to pose the optimization problem

$$\text{minimize } \mathscr{J}(\phi, g) \text{ subject to } F(\phi, g) = 0. \tag{1}$$

In this paper, various issues concerned with the solution of such problems are discussed in the fluid dynamics setting. Of course, by now, there is a vast amount of literature devoted to solving such problems; no attempt is made at citing any here since we are not concerned with advancing any particular methodology, but rather we are interested in pointing out various issues that are common to most, if not all, approaches.

Three methods for solving problem (1) are considered. The first method is a *Lagrange multiplier* or *adjoint equation* method, sometimes also referred to as an *all-at-once* or *one-shot* method, which is defined as follows: solve the coupled system

$$F(\phi, g) = 0, \qquad \lambda^* \frac{\partial F}{\partial \phi} = -\frac{\partial \mathscr{J}}{\partial \phi} \quad \text{and} \quad \lambda^* \frac{\partial F}{\partial g} = -\frac{\partial \mathscr{J}}{\partial g}. \tag{2}$$

[*] Correspondence to: Department of Mathematics, Iowa State University, Ames, IA 50011-2064, USA.

CCC 0271–2091/99/170053–26$17.50
Copyright © 1999 John Wiley & Sons, Ltd.

for the state variable ϕ, control g, and adjoint, costate or Lagrange multiplier variable λ. The first equation is the *state equation*, the second is referred to as the *adjoint equation*, and the third as the *optimality condition*. The coupled system of three equations is referred to as the *optimality system*.

The second method for solving (1) is an *adjoint-based iterative algorithm*, which is defined as follows:

1. guess g:
2. determine $\phi(g)$ from $F(\phi, g) = 0$;
3. determine the *adjoint variable* λ from $\lambda^*(\partial F/\partial\phi) = -\partial\mathcal{J}/\partial\phi$;
4. determine $\mathrm{d}\mathcal{J}/\mathrm{d}g = \partial\mathcal{J}/\partial g + \lambda^*(\partial F/\partial g)$;
5. use $\mathrm{d}\mathcal{J}/\mathrm{d}g$ to determine a new g;
6. repeat until satisfied.

The third method for solving (1) is a *sensitivity-based iterative algorithm*, which is defined as follows:

1. guess g;
2. determine $\phi(g)$ from $F(\phi, g) = 0$;
3. determine the sensitivity $\mathrm{d}\phi/\mathrm{d}g$ from $(\partial F/\partial\phi)(\mathrm{d}\phi/\mathrm{d}g) = -\partial F/\partial g$;
4. determine $\mathrm{d}\mathcal{J}/\mathrm{d}g = \partial\mathcal{J}/\partial g + (\partial\mathcal{J}/\partial\phi)(\mathrm{d}\phi/\mathrm{d}g)$;
5. use $\mathrm{d}\mathcal{J}/\mathrm{d}g$ to determine a new g;
6. repeat until satisfied.

To implement the second and third methods, start with an existing simulation code, i.e. one that solves for an approximation to ϕ for given g, then build in a capability to compute sensitivities and/or adjoints and a capability to compute $\mathrm{d}\mathcal{J}/\mathrm{d}g$, then apply a 'good' optimization algorithm to update g, and then grind away and solve problems! To implement the first method, again start with an existing simulation code and then build in the capability to solve it, as a coupled system, with the other two equations in (2).

In the setting of fluids problems, the ingredients in flow control problems can take the form of *constraints*, such as the potential flow equations, the Euler equations, the Navier–Stokes equations, equations for reacting flows, etc. Example *controls* are the body force, the velocity on the boundary, the initial velocity, the shape of domain, heat or concentration fluxes, etc. Example *objectives* are drag minimization, lift enhancement, velocity matching, temperature matching, enhancing mixing, etc. In the next three sections, examples of each of these approaches are presented. Some difficulties that arise in the practical implementation of algorithms for flow control and optimization, and in the determination of sensitivities and adjoints, are then discussed.

2. EXAMPLE FOR THE COUPLED ADJOINT EQUATION METHOD

As an example of the application of the coupled adjoint equation or Lagrange multiplier method, the *suppression of instabilities in boundary layer flows by injection/suction control and stress matching* [1] is considered.

The state equations. i.e. the constraints, are given by the Navier–Stokes system for the velocity components u and v and the pressure p:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+\frac{\partial p}{\partial x}-\nu\frac{\partial}{\partial x}\left(2\frac{\partial u}{\partial x}\right)-\nu\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)=0 \quad \text{in } (0, T)\times\Omega, \tag{3}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+\frac{\partial p}{\partial y}-\nu\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)-\nu\frac{\partial}{\partial y}\left(2\frac{\partial v}{\partial y}\right)=0 \quad \text{in } (0, T)\times\Omega, \tag{4}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \quad \text{in } (0, T)\times\Omega, \tag{5}$$

along with the initial and boundary data

$$\begin{cases} (u, v)\big|_{t=0}=(u_0, v_0) \quad \text{in } \Omega, \\ (u, v)\big|_{\Gamma_a}=\begin{cases}(g_1, g_2) & \text{in } (T_0, T_1) \\ (0, 0) & \text{in } (0, T_0) \text{ and } (T_1, T)\text{'}\end{cases} \\ (u, v)\big|_{\Gamma_i}=(u_i, v_i) \quad \text{and} \quad (u, v)\big|_{\Gamma_w}=(0, 0), \\ u\big|_{\Gamma_e}=U_\infty \quad \text{and} \quad p-2\nu\left.\frac{\partial v}{\partial y}\right|_{\Gamma_e}=P_\infty. \end{cases} \tag{6}$$

See Figure 1 for a sketch of the problem configuration.

The *control variables* are g_1 and g_2, i.e. the velocity components along a section Γ_a of the wall. Thus, Γ_a is the injection/suction orifice along the wall and (T_0, T_1) is the time interval over which control is effected. All other data is assumed to be given. It is require that $g_1|_{t=T_0}=g_{10}(x)$ and $g_2|_{t=T_0}=g_{20}(x)$ for given functions $g_{10}(x)$ and $g_{20}(x)$. No outflow conditions are specified on Γ_0; a 'buffer zone' technique is used near the outflow [2].

The *objective functional* is given by

$$\begin{aligned}&\mathcal{J}(u, v, p, g_1, g_2)\\ &=\frac{\alpha_1}{2}\int_{T_a}^{T_b}\int_{\Gamma_s}\left|\nu\frac{\partial u}{\partial y}-\tau_s\right|^2 \mathrm{d}\Gamma\,\mathrm{d}t+\frac{\alpha_2}{2}\int_{T_a}^{T_b}\int_{\Gamma_s}\left|-p+2\nu\frac{\partial v}{\partial y}-\tau_n\right|^2 \mathrm{d}\Gamma\,\mathrm{d}t\\ &\quad+\frac{\beta_1}{2}\int_{T_0}^{T_1}\int_{\Gamma_a}\left(\left|\frac{\partial g_1}{\partial t}\right|^2+|g_1|^2\right)\mathrm{d}\Gamma\,\mathrm{d}t+\frac{\beta_2}{2}\int_{T_0}^{T_1}\int_{\Gamma_a}\left(\left|\frac{\partial g_2}{\partial t}\right|^2+|g_2|^2\right)\mathrm{d}\Gamma\,\mathrm{d}t.\end{aligned}$$

The first two terms try to match the tangential and normal stresses along the boundary segment Γ_s and over the time interval (T_a, T_b) to given functions $\tau_s(t, x)$ and $\tau_n(t, x)$ respectively. The last two terms are penalty terms that limit the size and time oscillations of the control functions g_1 and g_2. The constants α_1, α_2, β_1 and β_2 are chosen to adjust the relative importance of the four terms in $\mathcal{J}(\cdot)$.

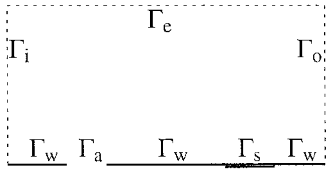

Figure 1. The computational domain for the boundary layer instability suppression problem.

The *optimization problem* is then defined by:

> given τ_s, τ_n, Γ_a, Γ_s, T_0, T_0, T_a and T_b, find u, v, p, g_1 and g_2 such that the functional $\mathcal{J}(u, v, p, g_1, g_2)$ is minimized subject to (u, v, p, g_1, g_2) satisfying the Navier–Stokes system.

The *optimality system* is given by the Navier–Stokes system (3)–(6), the adjoint system and the optimality conditions. The adjoint system for the adjoint velocity $(\hat u, \hat v)$ and the adjoint pressure $\hat p$ is given by

$$-\frac{\partial \hat v}{\partial t} + \hat u\frac{\partial u}{\partial y} + \hat v\frac{\partial v}{\partial y} - u\frac{\partial \hat v}{\partial x} - v\frac{\partial \hat v}{\partial y} - \frac{\partial \hat p}{\partial y} - \nu\frac{\partial}{\partial x}\left(\frac{\partial \hat u}{\partial y} + \frac{\partial \hat v}{\partial x}\right) - \nu\frac{\partial}{\partial y}\left(2\frac{\partial \hat v}{\partial y}\right) = 0, \tag{7}$$

$$-\frac{\partial \hat u}{\partial t} + \hat u\frac{\partial u}{\partial x} + \hat v\frac{\partial v}{\partial x} - u\frac{\partial \hat u}{\partial x} - v\frac{\partial \hat u}{\partial y} - \frac{\partial \hat p}{\partial x} - \nu\frac{\partial}{\partial x}\left(2\frac{\partial \hat u}{\partial x}\right) - \nu\frac{\partial}{\partial y}\left(\frac{\partial \hat u}{\partial y} + \frac{\partial \hat v}{\partial x}\right) = 0, \tag{8}$$

$$\frac{\partial \hat u}{\partial x} + \frac{\partial \hat v}{\partial y} = 0, \tag{9}$$

on $(0, T) \times \Omega$,

$$\begin{cases} (\hat u, \hat v)\big|_{t=T} = (0, 0) \quad \text{in } \Omega, \qquad (\hat u, \hat v)\big|_{\Gamma_0} = (0, 0) \quad \text{on } (0, T). \\[2ex] \hat v = \begin{cases} \alpha_2\left(-p + 2\nu\dfrac{\partial v}{\partial y} - \tau_b\right) & \text{on } (T_a, T_b) \times \Gamma_s \\ 0 \quad \text{otherwise along } \Gamma_w \text{ and } \Gamma_a \end{cases}, \\[4ex] \hat u = \begin{cases} \alpha_1\left(\nu\dfrac{\partial u}{\partial y} - \tau_a\right) & \text{on } (T_a, T_b) \times \Gamma_s \\ 0 \quad \text{otherwise along } \Gamma_w \text{ and } \Gamma_a \end{cases}, \\[4ex] \hat u = 0 \quad \text{on } (0, T) \times \Gamma_e, \\[1ex] \hat p + 2\nu\dfrac{\partial \hat v}{\partial y} + v\hat v = 0 \quad \text{on } (0, T) \times \Gamma_e. \end{cases} \tag{10}$$

Note that the adjoint equations are properly posed backwards-in-time so that terminal conditions (instead of initial conditions) are appropriate. Furthermore, the inflow boundary for the adjoint equations is the outflow boundary for the Navier–Stokes systems and conversely. No boundary conditions are imposed on the adjoint variables $(\hat u, \hat v, \hat p)$ at its 'outflow' boundary Γ_i; a 'buffer zone' technique is used near there.

The optimality conditions are given by

$$\begin{cases} \dfrac{\partial^2 g_1}{\partial t^2} - g_1 = \dfrac{1}{\beta_1}\left(\nu\dfrac{\partial \hat u}{\partial y}\right) \quad \text{on } (T_0, T_1) \times \Gamma_a, \\[3ex] g_1\big|_{t=T_0} = g_{10}(x) \quad \text{on } \Gamma_a, \qquad \dfrac{\partial g_1}{\partial t}\bigg|_{t=T_1} = 0 \quad \text{on } \Gamma_a, \end{cases} \tag{11}$$

$$\begin{cases} \dfrac{\partial^2 g_2}{\partial t^2} - g_2 = \dfrac{1}{\beta_2}\left(\hat{p} + 2\nu \dfrac{\partial \hat{v}}{\partial y}\right) \quad \text{on } (T_0, T_1) \times \Gamma_a, \\ g_2\big|_{t=T_0} = g_{20}(x) \quad \text{on } \Gamma_a, \qquad \left.\dfrac{\partial g_2}{\partial t}\right|_{t=T1} = 0 \quad \text{on } \Gamma_a. \end{cases} \tag{12}$$

These are two-point boundary value problems *in time* that hold at each point in Γ_a.

If one solves the coupled system (3)–(12) all-at-once, the optimal solution may be obtained without iteration. However, this is a formidable system, involving more than twice the unknowns of the Navier–Stokes system itself. Note that in (11)–(12), one needs to solve differential equations along the boundary. Even more onerous, the adjoint system (7)–(10) is well-posed as a backward in time problem with terminal conditions (at $t = T$) instead of initial conditions. As a result, solving this system together with the Navier–Stokes system (3)–(6) means that one cannot march in time, i.e. one needs to solve in space–time in a coupled manner. This is, of course, impossible in practice. Thus, in practice, the optimality system (3)–(12) is solved by a method such as that of Section 3 [1], or by solving local-in-time quasi-optimal control problems [3]. Extensive computational results for this problem may be found elsewhere [1].

3. EXAMPLE OF AN ADJOINT-BASED ITERATIVE ALGORITHM

As an example of the application of an adjoint-based iterative method, a *velocity matching by body force control for the Navier–Stokes equations* [4] is considered.

The *constraint equations* are the Navier–Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = f \quad \text{and} \quad \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \times (0, T),$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma \times (0, T), \quad \text{and} \quad \mathbf{u}\big|_{t=0} = \mathbf{u}_0 \quad \text{in } \Omega.$$

The *control* (in this academic example) is the body force $\mathbf{f}$ and the *objective functional* is given by

$$\mathcal{J}(\mathbf{u}) = \frac{\alpha}{2}\int_0^T \int_\Omega |\mathbf{u} - \mathbf{U}|^2 \, \mathrm{d}\Omega \, \mathrm{d}t + \frac{\gamma}{2}\int_\Omega |\mathbf{u}(T) - \mathbf{U}(T)|^2 \, \mathrm{d}\Omega.$$

The constants α and γ are chosen to adjust the relative importance of the two terms in $\mathcal{J}(\,\cdot\,)$. The inclusion of the second term has been found to be useful in achieving good matching near the final time $t = T$. The *optimization problem* is given by:

> given T, Ω, $\mathbf{u}_0$ and $\mathbf{U}$, minimize $\mathcal{J}(\mathbf{u}(\mathbf{f}))$ over $\mathbf{f}$ subject to $\mathbf{u}$ and $\mathbf{f}$ satisfying the Navier–Stokes system.

Thus, the goal of the optimization is to match the flow velocity $\mathbf{u}$ to a given velocity field $\mathbf{U}$.

In practice, one solves a discretized version of this problem; here, finite element spatial discretizations are considered. First, subdivide $(0, T)$ into N intervals of length Δt and then introduce stable and optimally accurate finite element spaces $\mathbf{X}_h$ and S_h for the velocity and pressure respectively, satisfying the boundary condition $\mathbf{u} = \mathbf{0}$. A finite element space $\mathbf{Y}_h$ is chosen for the control, which may be chosen to be essentially the same as $\mathbf{X}_h$. The approximate solution is denoted by $\{(\mathbf{u}_h^n, p_h^n, \mathbf{f}_h^n)\}_{n=1}^N$. The discrete functional is given by

Copyright © 1999 John Wiley & Sons, Ltd.

$$\mathcal{J}_h^N = \frac{\alpha \Delta t}{2} \sum_{n=1}^{N} \|\mathbf{u}_h^n - \mathbf{U}^n\|^2 + \frac{\gamma}{2} \|\mathbf{u}_h^N - \mathbf{U}^N\|^2 + \frac{\beta \Delta t}{2} \sum_{n=1}^{N} \|\mathbf{f}_h^n\|^2,$$

where it has been regularized through the addition of a penalty term with penalty parameter β. The discrete constraint equations are the fully discrete Navier–Stokes system: for $n = 1, 2, \ldots, N$,

$$\frac{1}{\Delta t} (\mathbf{u}_h^n, \mathbf{w}_h) + a(\mathbf{u}_h^n, \mathbf{w}_h) + c(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{w}_h) + b(\mathbf{w}_h, \mathbf{p}_h^n) = (\mathbf{f}_h^n, \mathbf{w}_h) + \frac{1}{\Delta t} (\mathbf{u}_h^{n-1}, \mathbf{w}_h) \quad \forall \mathbf{w}_h \in \mathbf{X}_h,$$

$$b(\mathbf{w}^n, r_h) = 0 \quad \forall r_h \in S_h,$$

where $\mathbf{w}_h^0 = \Pi_{\mathbf{X}_h} \mathbf{w}_0$ and $\Pi_{\mathbf{X}_h}$ is an appropriate projection of the initial data onto the velocity finite element space. The discrete optimization problem is then defined by:

> given T, Ω, $\mathbf{u}_0$ and $\mathbf{U}$, minimize $\mathcal{J}_h^N(\mathbf{u}_h^n(\mathbf{f}_h^n))$ with respect to $\mathbf{f}_h^n$ subject to $\mathbf{u}_h^n$ and $\mathbf{f}_h^n$ satisfying the discrete Navier–Stokes system.

In order evaluate the gradient of the functional, we introduce the discrete adjoint system: starting with $\mu_h^N = -(\gamma/\beta)(\mathbf{u}_h^N - \mathbf{U}^N)$, for $n = N, N-1, \ldots, 1$,

$$\frac{1}{\Delta t} (\mu_h^{n-1}, \omega_h) + a(\omega_h, \mu_h^{n-1}) + c(\omega_h, \mathbf{u}_h^n, \mu_h^{n-1}) + c(\mathbf{u}_h^n, \omega_h, \mu_h^{n-1}) + b(\omega_h, \pi_h^{n-1})$$

$$= \frac{1}{\Delta t} (\mu_h^n, \omega_h) + \alpha(\mathbf{u}_h^n - \mathbf{U}^n, \omega_h), \quad \forall \omega_h \in \mathbf{X}_h,$$

$$b(\mu_h^{n-1}, \sigma_h) = 0 \quad \forall \sigma_h \in S_h.$$

In this example, a simple gradient method is chosen for the solution of the optimiziation problem. The method is given as follows:

(a) initialization:
(i) choose a tolerance τ and $\mathbf{f}_h^n(0)$ and set $k = 0$ and $\epsilon = 1$;
(ii) solve for $\mathbf{u}_h^n(0)$ from the discrete Navier–Stokes equations with right-hand-side $\mathbf{f}_h(0)$
(iii) evaluate $\mathcal{J}_h^N(0) = \mathcal{J}_h^N(\mathbf{u}_h^n(0), \mathbf{f}_h^n(0))$;
(b) main loop:
(iv) set $k = k+1$;
(v) solve for $\mu_h^n(k)$ from the adjoint equations using $\mathbf{u}_h^n(k-1)$ for $\mathbf{u}_h^n$;
(vi) set $\mathbf{f}_h^n(k) = \mathbf{f}_h^n(k-1) - \epsilon(\beta \mathbf{f}_h^n(k-1) + \mu_n^h(k))$;
(vii) solve for $\mathbf{u}_h^n(k)$ from the discrete Navier–Stokes equations with right-hand-side $\mathbf{f}_h^n(k)$;
(viii) evaluate $\mathcal{J}_h^N(k) = \mathcal{J}_h^N(\mathbf{u}_h^n(k), \mathbf{f}_h^n(k))$;
(ix) if $\mathcal{J}_h^N(k) > \mathcal{J}_h^N(k-1)$, set $\epsilon = 0.5\epsilon$ and go to (vi); otherwise continue;
(x) if $|\mathcal{J}_h^N(k) - \mathcal{J}_h^N(k-1)|/|\mathcal{J}_h^N(k)| > \tau$, set $\epsilon = 1.5\epsilon$ and go to (iv); otherwise stop.

The bulk of the computational costs are found in the backward-in-time solution of the discrete adjoint system in step (v) and the forward-in-time solution of the discrete Navier–Stokes system in step (vii). Step (vi) is a simple update in the direction of the negative gradient of the functional, which in this example is given by $\mathbf{f}_h^n + (1/\beta)\mu_h^{n-1}$.

Copyright © 1999 John Wiley & Sons, Ltd.

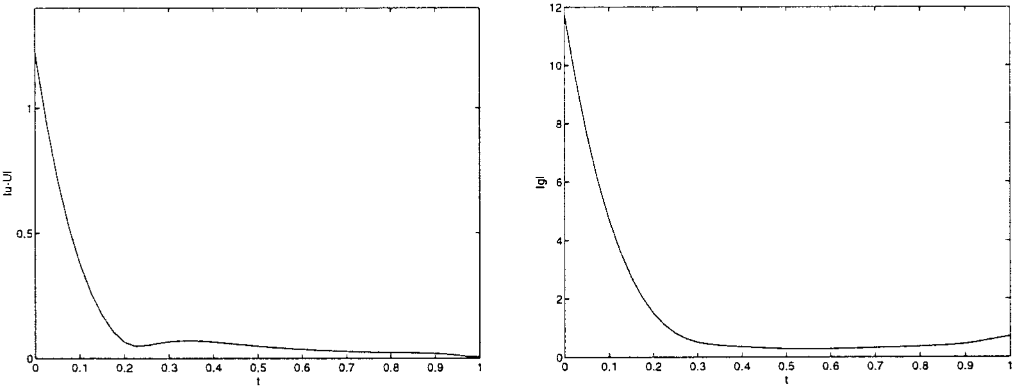

Figure 2. Error $\|\mathbf{u}-\mathbf{U}\|$ (left and control norm $\|\mathbf{f}\|$ (right) vs. time.

3.1. A computational example

The algorithm described in the previous section was implemented on the following test example. The flow domain is the unit square $\Omega=(0, 1)\times(0, 1)$. It is assumed that the time interval [0, 1] is divided in equal intervals of time $\Delta t=1/N$. The Taylor–Hood pair of finite element spaces are used on a rectangular mesh, i.e. the finite element spaces are chosen to be continuous piecewise biquadratic polynomials for the velocity and continuous piecewise bilinear polynomials for the pressure. The mesh size is h and calculations with varying mesh sizes have been performed to show convergence. All the vector plots are normalized by the maximum values in order to enhance the clarity of the plots.

The target velocity $\mathbf{U}$ for this test is a solution of the Stokes system with zero initial velocity and body force

$$\mathbf{F}(x, y)=\begin{pmatrix} a(0.4, x, y)-\mathrm{e}^{-2t}\,a(0.6, x, y)\\ b(0.4, x, y)-\mathrm{e}^{-2t}\,b(0.6, x, y)\end{pmatrix},$$

where

$$a(z, x, y)=10\,\frac{\mathrm{d}\psi(z, x, y)}{\mathrm{d}y}, \qquad b(z, x, y)=-10\,\frac{\mathrm{d}\psi(z, x, y)}{\mathrm{d}x},$$

and

$$\psi(z, x, y)=(1-\cos(4\pi zx))(1-x)^2(1-\cos(4\pi zy))(1-y)^2.$$

The resulting target velocity is a superposition of two flows: a vortex at the centre of the domain with large radius and another vortex with small radius centred in the lower-left corner. Each of these flows prevails at different times of the evolution. We set $\alpha=1$, $\beta=\frac{1}{5000}$, $\gamma=0.5$, $\Delta t=0.025$ and $h=\frac{1}{16}$. The initial velocity for the controlled flow is

$$\mathbf{u}_0(x, y)=\begin{pmatrix} (\cos(2\pi x)-1)\ \sin(2\pi y)\\ -(\cos(2\pi y)-1)\ \sin(2\pi x)\end{pmatrix}.$$

Figure 2 shows the error $\|\mathbf{u}-\mathbf{U}\|$ between the controlled flow $\mathbf{u}$ and the target flow $\mathbf{U}$ and the norm of the control $\|\mathbf{f}\|$. It can be seen that the error reduces. The evolution is given in Figure 3. The target flow is on the left and the controlled flow is on the right. It can be seen that by $t=0.5$, we reach a near perfect match. For times later than that, the plots of the controlled

Copyright © 1999 John Wiley & Sons, Ltd.

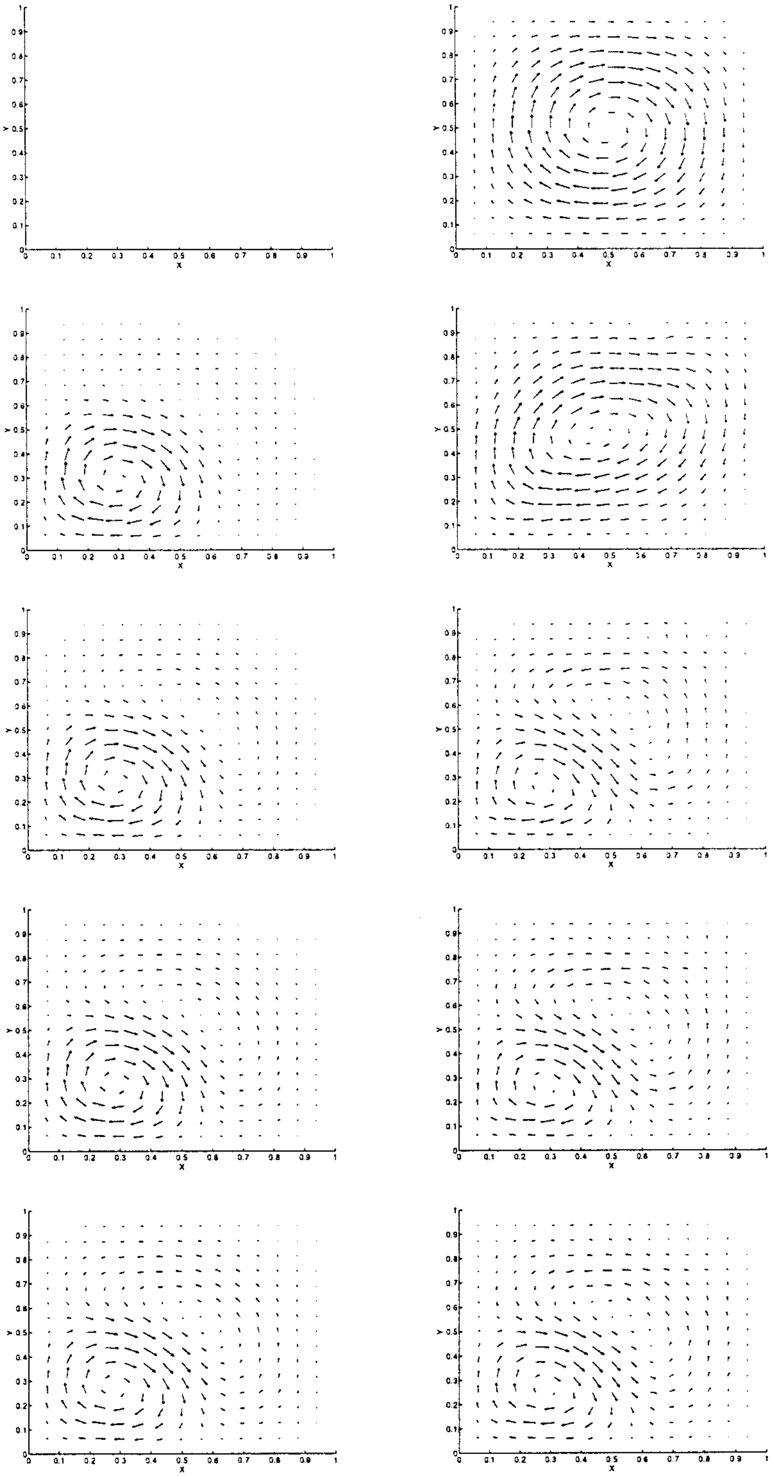

Figure 3. Desired (left) and controlled (right) velocity fields at time levels $t = 0$, 0.1, 0.3, 04, 0.5.

flow and the target flow are indistinguishable; thus, we do not present them here. From Figure 2, we see that the control works hard at the beginning in order to steer the controlled flow to the target flow and subsequently remains flat. Near $t=T$, the control strength again increases in order to reduce the error near the final time.

4. EXAMPLE OF A SENSITIVITY-BASED ITERATIVE ALGORITHM

As an example of the application of a sensitivity-based iterative method, a *shape optimization problem for steady Euler flows*, i.e. an optimal design problem [5–8], is considered.

The *constraint equations* are the two-dimensional steady Euler equations that are written in the form

$$\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\mathbf{0} \quad \text{in } \Omega,$$

where Ω denotes a two-dimensional flow domain. $\mathbf{F}(\mathbf{Q})$ and $\mathbf{G}(\mathbf{Q})$ are flux functions, and $\mathbf{Q}$ is a vector of conservative variables, e.g. $\mathbf{Q}=(\rho, m_1, m_2, e)^T$. The inflow condition

$$\mathbf{Q}=\begin{pmatrix}\beta\\ \alpha_0\\ 0\\ \sigma\end{pmatrix} \quad \text{on } \Gamma_0$$

and the no-penetration condition

$$m_1n_1+m_2n_2=0 \quad \text{on } \Gamma,$$

are imposed, where Γ denotes an obstacle in the flow and Γ_0 the inflow boundary.

The obstacle Γ is determined by a finite number of design parameters $\alpha_1, \ldots, \alpha_K$, through the relation

$$y=\Phi(x;\ \alpha_1, \ldots, \alpha_K).$$

In terms of this representation for Γ, the no-penetration condition can be written as

$$-m_1(x, y=\Phi(x;\ \alpha_1, \ldots, \alpha_K);\ \alpha_1, \ldots, \alpha_K)\frac{\partial \Phi}{\partial x}(x;\ \alpha_1, \ldots, \alpha_K)$$
$$+m_2(x, y=\Phi(x;\ \alpha_1, \ldots, \alpha_K);\ \alpha_1, \ldots, \alpha_K)=0.$$

The shape design parameters enter into this condition in two ways: the components of the momentum depend on the design parameters and the condition is applied on the obstacle Γ, whose shape depends explicitly on the design parameters. It is important to keep track of both types of dependencies when one is interested in sensitivities.

The *objective functional* is introduced

$$\mathscr{J}(\mathbf{Q})=\frac{1}{2}\int_{\Gamma_1}|\mathbf{Q}-\hat{\mathbf{Q}}|^2\,\mathrm{d}\Gamma,$$

where Γ_1 denotes a part of the boundary of the flow domain, e.g. the outflow boundary, and $\hat{\mathbf{Q}}$ is a given flow field.

Copyright © 1999 John Wiley & Sons, Ltd.

The *optimization problem* is given by:

> given $\hat{\mathbf{Q}}$ and Γ_1, find parameters α_k, $k=0,\ldots,K$, so that the functional $\mathcal{J}(\mathbf{Q})$ is minimized, subject to the flow field $\mathbf{Q}$ satisfying the Euler equations plus boundary conditions.

Thus, the goal of the optimization is to have the flow field match a given flow field $\hat{\mathbf{Q}}$ along Γ_1 as well as possible. The controls are the parameters that determine the shape of Γ, the obstacle and the inflow x component of the momentum.

4.1. Sensitivity equations

For $k=1,\ldots,K$, let $\mathbf{Q}_k=\partial\mathbf{Q}/\partial\alpha_k$ denote the sensitivity with respect to the shape design parameters α_k If we formally differentiate the Euler equations with respect to α_k we obtain

$$\frac{\partial}{\partial x}\left(\frac{\partial\mathbf{F}}{\partial\mathbf{Q}}\cdot\mathbf{Q}_k\right)+\frac{\partial}{\partial y}\left(\frac{\partial\mathbf{G}}{\partial\mathbf{Q}}\cdot\mathbf{Q}_k\right)=\mathbf{0},\quad k=1,\ldots,K.$$

The inflow conditions and inflow boundary are independent of α_k for $k\geq 1$, so that the derivative of the inflow condition with respect to α_k is given by

$$\mathbf{Q}_k=\mathbf{0},\quad k=1,\ldots,K,\quad \text{on } \Gamma_0.$$

The derivative of the no-penetration condition with respect to α_k is given by

$$-(m_1)_k\frac{\partial\Phi}{\partial x}+(m_2)_k=\left(\frac{\partial m_1}{\partial y}\frac{\partial\Phi}{\partial x}-\frac{\partial m_2}{\partial y}\right)\Phi_k+m_1\frac{\partial\Phi_k}{\partial x},\quad k=1,\ldots,K,$$

$$\text{on } y=\Phi(x;\ \alpha_1,\ldots,\alpha_K),$$

where, for each $k=1,\ldots,K$,

$$(m_1)_k=\frac{\partial m_1}{\partial\alpha_k},\quad (m_2)_k=\frac{\partial m_2}{\partial\alpha_k}\quad\text{and}\quad \Phi_k=\frac{\partial\Phi}{\partial\alpha_k}.$$

If the state $\mathbf{Q}$ is given, then the system for the sensitivity equations is linear in $\mathbf{Q}_k$. The left-hand-side operators are independent of k; the only dependence on k is through the appearance of Φ_k on the right-hand-side of the differentiated no-penetration condition. This means that, if $\mathbf{Q}$ is given, a discretization of the sensitivity system will yield a linear system for an approximation of $\mathbf{Q}_k$ having a coefficient matrix that is independent of k and is therefore the same for all sensitivities $\mathbf{Q}_k$, $k=1,\ldots,K$. However, for each k, the right-hand-side would be different.

The inflow x-momentum component data α_0 is a *value parameter* that is also at our disposal in the optimization process. Let $\mathbf{Q}_0=(\rho_0,(m_0)_1,(m_0)_2,e_0)^T=\partial\mathbf{Q}_0/\partial\alpha_0$, i.e. the sensitivity with respect to the inflow x-momentum component. Differentiating the Euler system with respect to α_0 yields the sensitivity system

$$\frac{\partial}{\partial x}\left(\frac{\partial\mathbf{F}}{\partial\mathbf{Q}}\cdot\mathbf{Q}_0\right)+\frac{\partial}{\partial y}\left(\frac{\partial\mathbf{G}}{\partial\mathbf{Q}}\cdot\mathbf{Q}_0\right)=\mathbf{0},\quad k=1,\ldots,K,$$

$$\mathbf{Q}_0=(0,1,0,0)^T\quad\text{on } \Gamma_0,$$

$$(m_0)_1n_1+(m_0)_2n_2=0\quad\text{on } \Gamma.$$

Let us compare the shape and value sensitivity equations. The differential equations for both the shape and value sensitivities are the same. The forms of the corresponding boundary conditions, i.e. the left-hand-sides, are also the same. What is different is the data in the

Copyright © 1999 John Wiley & Sons, Ltd.

boundary conditions. Note the appearance of the derivatives $\partial m_1/\partial y$ and $\partial m_2/\partial y$, i.e. derivatives of the flow field evaluated on the boundary segment Γ_1 on the right-hand-side of the shape sensitivity system. This is generic to sensitivity equations in shape optimization problems and leads to inaccuracies. Note that the right-hand-sides of the value sensitivity system do not involve derivatives of the flow field.

4.2. The optimization algorithm

The derivative of the functional with respect to a parameter α_k is given by

$$\frac{\partial \mathcal{J}}{\partial \alpha_k} = \int_{\Gamma_1} \left((\rho - \hat{\rho}) \frac{\partial \rho}{\partial \alpha_k} + (m_1 - \hat{m}_1) \frac{\partial m_1}{\partial \alpha_k} + (m_2 - \hat{m}_2) \frac{\partial m_2}{\partial \alpha_k} + (e - \hat{e}) \frac{\partial e}{\partial \alpha_k} \right) \mathrm{d}\Gamma,$$
$$k = 0, \ldots, K.$$

The flow sensitivities $\partial \rho/\partial \alpha_k$, $\partial m_1/\partial \alpha_k$, $\partial m_2/\partial \alpha_k$ and $\partial e/\partial \alpha_k$ are determined by solving the appropriate sensitivity equations.

The optimization algorithm employed is given as follows:

- Start with an initial guess $\alpha_0^{(0)}, \alpha_1^{(0)}, \ldots, \alpha_K^{(0)}$ for the design parameters;
- For $n = 0, 1, 2, \ldots,$
  1. solve the Euler system to obtain the corresponding flow field $\mathbf{Q}^{(n)}$;
  2. solve the sensitivity systems to obtain the sensitivities $\mathbf{Q}_0^{(n)}$ and $\mathbf{Q}_k^{(n)}$, $k = 1, \ldots, K$;
  3. Compute $(\mathrm{d}\mathcal{J}/\mathrm{d}\alpha_k)(\phi^{(n)}, g^{(n)})$, $k = 0, \ldots, K$;
  4. use the results of steps 1, 2 and 3 to compute steps $\delta\alpha_k^{(n)}$, $k = 0, \ldots, K$;
  5. set $\alpha_k^{(n+1)} = \alpha_k^{(n)} + \delta\alpha_k^{(n)}$, $k = 0, \ldots, K$;
  6. test for convergence and either stop or return to step 1.

Once one chooses an optimization method for effecting the update of step 4, the optimization algorithm is completely defined. We have found a BFGS-quasi-Newton method [9] to be particularly effective in this regard.

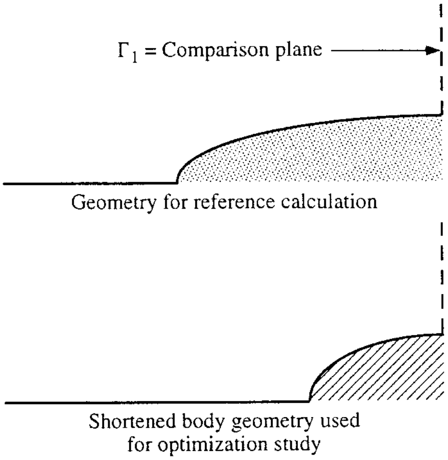
Γ_1 = Comparison plane
Geometry for reference calculation
Shortened body geometry used for optimization study

Figure 4. Body for the reference flow (top) and a candidate body for the optimal flow (bottom).

Copyright © 1999 John Wiley & Sons, Ltd.

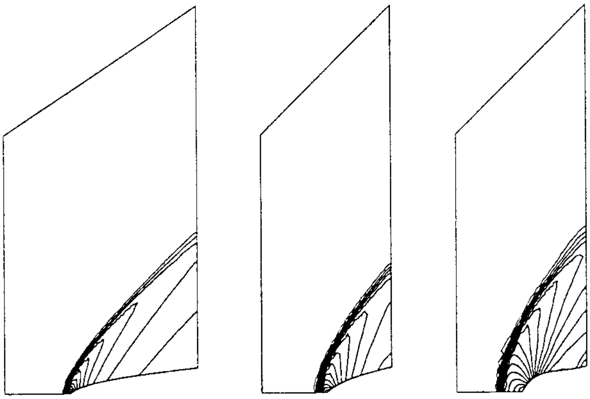

Figure 5. Pressure contours for target flow (left), initial flow for the optimization (middle), and optimal flow (right).

4.3. A computational example

A computational example of the use of sensitivities in Euler flow optimization [5–8] is now given. Consider the supersonic flow around a body as depicted in the top sketch of Figure 4. For a given body shape and free stream Mach number, this flow is used, which of course, is determined computationally, as the reference flow $\hat{\mathbf{Q}}$. In the objective functional Γ_1, the part of the boundary along which we wish to match the flow to the given flow $\hat{\mathbf{Q}}$, is the exit plane as indicated in Figure 4. The goal of the optimization is to try to match the reference flow along the exit plane with the flow around a body half as long as the reference body, as depicted in the bottom sketch of Figure 4. The thickness of the body at the exit plane is the same as that of the reference body. The design parameters are the free stream Mach number and two Bezier parameters that determine the shape of the body. The optimization is initiated with a body that is simply the squeezed body of the reference flow, i.e. if $y = f(x - L)$ denotes the body for the reference flow, where L denotes the position of the exit plane, then the initial body for the optimization is given by $y = f(2(x - L))$. The free stream Mach number for the reference flow and the initial flow is 2. The body is determined by four Bezier parameters, two of which are fixed so that the body has the correct intercepts along the horizontal axis and the exit plane. The two free Bezier parameters are initially set to 0.2 and 0.3; although these numbers may not mean anything to the reader, we give them to show the changes that occur through the optimization process. The flow is computed using Aerosoft Corporation $\mathscr{U}$GASP unstructured mesh code, which is based on [10]. Sensitivities were computed by a modification of the $\mathscr{U}$GASP code that approximately solves the continuous sensitivity equations.

In Figure 5, the pressure contours for the target flow, the initial flow for the optimization, and the optimal flow are given. The final values of the design parameters are: free stream Mach number 1.98; Bezier parameters 0.415 and 0.279. It can be seen that at least one of these parameters changed appreciable during the optimization process, i.e. the starting guesses for the parameters in the optimization were not all ‘close’ to their optimal values. From Figure 5 it can be seen that the shock position for the initial flow is incorrect; the optimal flow has the shock in the right place. The correct placement of the shock at the exit plane is accomplished by a body half the length of the reference body by making the optimal body blunter. This is done automatically by the optimization process. The matching of the continuous portions of the target and optimal flows along the exit plane is not perfect due to the small number of parameters used in the optimization.

Copyright © 1999 John Wiley & Sons, Ltd.

In Figure 6, a graph of the final x velocity sensitivity (determined by two different approaches) plotted along the exit plane is given. Note, for future reference, the large spike in the sensitivity; this spike is located at the intersection of the shock wave and the exit plane.

5. SOME DIFFICULTIES IN PRACTICAL FLOW OPTIMIZATION

There are many difficulties that one encounters in the practical implementation of flow control and optimization algorithms. Among these are the appearance of spurious (numerical) local minima [11,12]. Perhaps foremost among the difficulties is the need to decide between the differentiate-then-discretize or the discretize-then-differentiate approaches. In the former case, one derives sensitivity or adjoint equations at the continuous level, and then discretizes the result to obtain approximate sensitivity or adjoint equations; in the latter case, one discretes the flow equations first and then differentiates these to obtain approximate sensitivity or adjoint equations. The two sets of approximate sensitivity or adjoint equations are not in general the same. The relative merits of the two approaches are discussed at length elsewhere, so it will be dwelled upon here. Instead, difficulties associated with inaccurate data in boundary conditions for shape sensitivities are briefly discussed, then difficulties that arise from discontinuities in the flow field will be discussed. Although the ensuing discussion is in the context of sensitivities, almost everything said holds just as well for adjoints.

5.1. Inaccurate data in boundary conditions

Consider the Navier–Stokes problem

$$-\nu\Delta\mathbf{u}+\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p=\mathbf{f} \quad \text{and} \quad \nabla\cdot\mathbf{u}=0 \quad \text{in } \Omega,$$

$$\mathbf{u}=\mathbf{0} \quad \text{on } \Gamma_1=\{y=\beta\phi(x)\}, \quad \text{and} \quad \mathbf{u}=\alpha\mathbf{g} \quad \text{on } \Gamma_2.$$

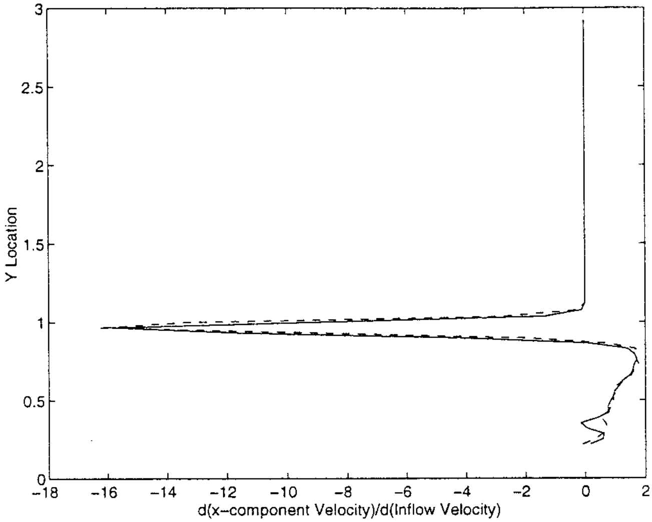

Figure 6. Sensitivity of the x component of the velocity at the exit plane computed by finite difference quotients (—) and by the sensitivity equation method (---).

Copyright © 1999 John Wiley & Sons, Ltd.

The part of the boundary Γ_2 is fixed and the boundary data contains a parameter α that is at our disposal; another part of the boundary Γ_1 is determined by the equation $y=\beta\phi(x)$ so that its specification contains a parameter β. Thus, α is an example of a *value* design parameter or control and β is an example of a *shape* design parameter or control.

The sensitivities are denoted by

$$\mathbf{u}_\alpha=\frac{\partial \mathbf{u}}{\partial \alpha},\quad p_\alpha=\frac{\partial p}{\partial \alpha},\quad \mathbf{u}_\beta=\frac{\partial \mathbf{u}}{\partial \beta}\quad\text{and}\quad p_\beta=\frac{\partial p}{\partial \beta}.$$

The equations for the sensitivities with respect to *valve* parameter α are

$$-\nu\Delta\mathbf{u}_\alpha+\mathbf{u}_\alpha\cdot\nabla\mathbf{u}+\mathbf{u}\cdot\nabla\mathbf{u}_\alpha+\nabla p_\alpha=\mathbf{0}\quad\text{and}\quad\nabla\cdot\mathbf{u}_\alpha=0\quad\text{in }\Omega;$$

$$\mathbf{u}_\alpha=\mathbf{0}\quad\text{on }\Gamma_1\quad\text{and}\quad\mathbf{u}_\alpha=\mathbf{g}\quad\text{on }\Gamma_2. \tag{13}$$

The equations for the sensitivities with respect to *shape* parameter, β are

$$-\nu\Delta\mathbf{u}_\beta+\mathbf{u}_\beta\cdot\nabla\mathbf{u}+\mathbf{u}\cdot\nabla\mathbf{u}_\beta+\nabla p_\beta=\mathbf{0}\quad\text{and}\quad\nabla\cdot\mathbf{u}_\beta=\mathbf{0}\quad\text{in }\Omega;$$

$$\mathbf{u}_\beta=-\phi(x)\frac{\partial\mathbf{u}}{\partial y}\quad\text{on }\Gamma_1\quad\text{and}\quad\mathbf{u}_\beta=\mathbf{0}\quad\text{on }\Gamma_2. \tag{14}$$

For the system (14), note that we have that $\mathbf{u}=\mathbf{u}(x,y;\alpha,\beta)$ so that we have $\mathbf{u}(x,\beta\phi(x);\alpha,\beta)=\mathbf{0}$ on Γ_1. Note that along Γ_1, $\mathbf{u}$ depends on β in two ways. If we differentiate with respect to β, then the chain rule yields the boundary condition along Γ_1 in (14).

The operators on the left-hand-side of the systems (13) and (14) are identical. The only difference between the two systems is the data on the right-hand-sides. For the first system, the data is given exclusively in terms of the data for the flow system. However, for the second system, the data depends additionally on a derivative of the flow solution, specifically $\partial\mathbf{u}/\partial y$ evaluated along the boundary segment Γ_1. In practice, where one deals with discrete equations and approximate solutions, there are three sources of error in the right-hand-side of the second system: first, only approximations of the flow components are known; second, approximate differentiation rules must be used to approximate the derivatives of the flow components; and third, the approximate derivatives must be evaluated at the boundary, e.g. only one-sided difference quotients can be used to approximate derivatives. These difficulties arise in all sensitivity calculations involving shape parameters and arise in both the discretize-then-differentiate and differentiate-then-discretize approaches.

5.2. Discontinuous flows, e.g. shock waves

One cannot differentiate the flow equations across a shock wave to obtain sensitivity or adjoint equations. A naive determination of sensitivity equations by both the discretize-then-differentiate and differentiate-then-discretize approaches yield sensitivities containing spikes near the shock location (see Figure 6). These spikes are approximations to δ functions. Despite these spikes, naively computed sensitivities do rather well in an optimization setting. However, improved performance of optimzation algorithms may be achievable if one could compute one-sided sensitivities having no spikes. This difficulty is discussed at greater length in the next section in the context of the Reimann problem for which an exact solution is known.

Copyright © 1999 John Wiley & Sons, Ltd.

6. SENSITIVITIES FOR THE REIMANN PROBLEM

In order to discuss in more detail the difficulties that arise in sensitivity calculations for discontinuous flow, a setting for which we can determine the sensitivities exactly, namely the sensitivity of Riemann flow with respect to the initial high pressure p_4, is chosen. The sensitivities are calculated by using finite difference quotients of the discrete flow solution, automatic differentiation of the discrete flow equations, both of which are discretize-then-differentiate approaches, and the sensitivity equation method in which one differentiates the continuous flow equations before differentiating them to obtain approximate sensitivity equations. The difficulties using three numerical schemes of Lax–Wendroff, Godunov and Roe-type to calculate the flow are illustrated; these methods have well-known properties with respect to flow fidelity, e.g. shock resolution.

The finite difference quotient approximation of the sensitivity with respect to p_4 is given by

$$(\mathbf{Q}^h)_{p_4} \approx \frac{\mathbf{Q}^h(p_4 + \Delta p_4) - \mathbf{Q}^h(p_4)}{\Delta p_4};$$

we use $\Delta p_4 = 0.1$ in the computational experiments. The automatic differentiation sensitivity is calculated using the software package ADIFOR [13–15] on the discrete flow equations. The sensitivity equation method differentiates the continuous Euler equations and initial conditions with respect to p_4 to derive a new set of continuous equations, the sensitivity equations that can then be solved using one of the numerical methods.

The governing equations are the one-dimensional Euler equations

$$\mathbf{Q}_t + \mathbf{F}(\mathbf{Q})_x = \mathbf{0},$$

where the conservative variables $\mathbf{Q}$ and flux functions $\mathbf{F}(\mathbf{Q})$ are respectively given by

$$\mathbf{Q}(x\ t) = \begin{bmatrix} \rho \\ rm \\ e \end{bmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{Q}) = \begin{bmatrix} m \\ p + (m^2/\rho) \\ (m/\rho)(e+p) \end{bmatrix}.$$

The initial condition is

$$\mathbf{Q}(x, 0) = \mathbf{Q}_4 \quad \text{for } x < c, \quad \text{and} \quad \mathbf{Q}(x, 0) = \mathbf{Q}_1 \quad \text{for } x > c,$$

which represent two states at rest with different pressures and densities separate by a diaphragm located at $x = c$; it is assumed that the state on the left is at higher pressure, i.e. $p_4 > p_1$. The primitive variables $\mathbf{U} = (p, \rho, u)^T$ are related to the conservative variables by $m = \rho u$ and $e = p/(\gamma - 1) + \rho u^2$.

The exact solution of the Riemann problem is given by

$$\mathbf{U} = \mathbf{U}_4 \quad \text{for } x < -a_4 t + c,$$

$$\mathbf{U} = \mathbf{U}_R \quad \text{for } -a_4 t + c \le x \le \left(\frac{\gamma+1}{2} u_3 - a_4\right) t + c,$$

$$\mathbf{U} = \mathbf{U}_3 \quad \text{for } \left(\frac{\gamma+1}{2} u_3 - a_4\right) t + c \le x \le u_2 t + c,$$

$$\mathbf{U} = \mathbf{U}_2 \quad \text{for } u_2 t + c < x < a_1\left(\frac{\gamma_1 - 1}{2\gamma_1} + \frac{\gamma_1 + 1}{2\gamma_1}\frac{p_2}{p_1}\right)^{1/2} t + c,$$

Copyright © 1999 John Wiley & Sons, Ltd.

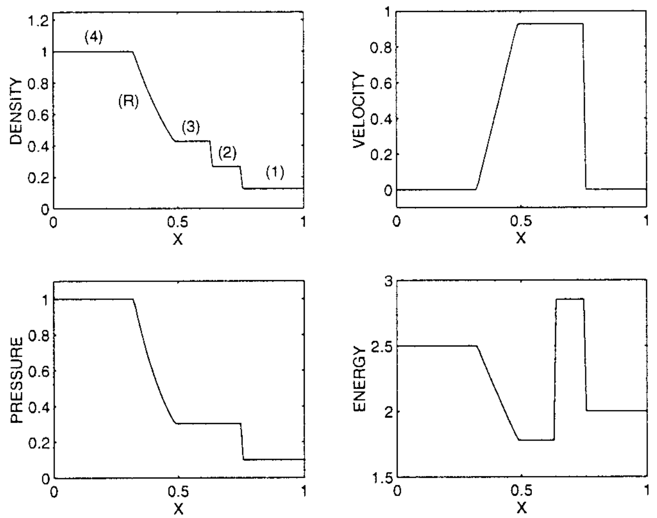

Figure 7. Exact solution of the one-dimensional Riemann problem at $t = 0.148$.

$$\mathbf{U} = \mathbf{U}_1 \quad \text{for } a_1\left(\frac{\gamma_1 - 1}{2\gamma_1} + \frac{\gamma_1 + 1}{2\gamma_1}\frac{p_2}{p_1}\right)^{1/2} t + c \le x,$$

where $a_i^2 = \gamma p_i/\rho_i$, $i = 1$ or 4, and p_2 is given implicitly as the solution of

$$\frac{p_4}{p_1} = \frac{p_2}{p_1}\left(1 - \frac{(\gamma - 1)(a_1/a_4)(p_2/p_1 - 1)}{\sqrt{2\gamma}\sqrt{2\gamma + (\gamma + 1)(p_2/p_1 - 1)}}\right)^{-2\gamma/(\gamma - 1)}. \tag{15}$$

The remaining variables are given by

$$\rho_2 = \rho_1 \frac{p_2}{p_1}\left(1 + \frac{\gamma - 1}{\gamma + 2}\frac{p_1}{p_2}\right) \Big/ \left(1 + \frac{\gamma - 1}{\gamma + 1}\frac{p_2}{p_1}\right),$$

$$u_2 = u_3 = a_1\left(\frac{p_2}{p_1} - 1\right)\sqrt{2/\gamma}\left((\gamma + 1)\frac{p_2}{p_1} + (\gamma - 1)\right)^{-1/2},$$

$$\rho_3 = \rho_4\left(\frac{p_3}{p_4}\right)^{1/\gamma}, \qquad p_3 = p_2,$$

$$p_R = p_4\left(1 - \frac{\gamma - 1}{2}\frac{u_R}{a_4}\right)^{2\gamma/(\gamma - 1)}, \qquad \rho_R = \rho_4\left(1 - \frac{\gamma - 1}{2}\frac{u_R}{a_4}\right)^{2/(\gamma - 1)},$$

$$u_R = \left(\frac{u_3 - u_4}{[(\gamma + 1)/2]u_3}\right)\left(\frac{x - c}{t}\right) + \frac{a_4 u_3 - (a_4 - [(\gamma + 1)/2]u_3)u_4}{[(\gamma + 1)/2]u_3}.$$

The solution has five distinct regions: left and right constant flow regions in which the solutions are given by the high- and low-pressure initial states; two other constant flow regions between the shock and the contact discontinuity and the contact discontinuity and the rarefaction wave; and a rarefaction wave adjacent to the left high-pressure region (see Figure 7).

One cannot differentiate the flow (with respect to space, time or design parameters) across the shock wave or the contact discontinuity. If one differentiates across these phenomena, δ functions result. Although the flow is continuous at the edges of the rarefaction weave, it is not

Copyright © 1999 John Wiley & Sons, Ltd.

differentiable there either. Differentiation across the edges of that wave results in jump discontinuities in the sensitivities. However, the flow solution can be differentiated within each of the five regions.

6.1. Exact flow sensitivities in continuous regions of the flow

The sensitivity of the flow **U** is denoted with respect to p_4 by

$$\mathbf{U}' = \begin{pmatrix} \dfrac{\partial p}{\partial p_4} \\ \dfrac{\partial \rho}{\partial p_4} \\ \dfrac{\partial u}{\partial p_4} \end{pmatrix} = \begin{pmatrix} p' \\ \rho' \\ u' \end{pmatrix}.$$

Differentiation of the flow (in continuous regions) with respect to the design parameter p_4 yields the exact sensitivities $\mathbf{U}' = \mathbf{U}'_4 = (1, 0, 0)^T$ in the high-pressure region $\mathbf{U}' = \mathbf{U}'_R$ in the rarefaction wave, $\mathbf{U}' = \mathbf{U}'_3$ in the region between the rarefaction wave and the contact discontinuity, $\mathbf{U}' = \mathbf{U}'_2$ in the region between the contact discontinuity and the shock wave, and $\mathbf{U}' = \mathbf{U}'_1$ in the low-pressure region. A linear equation for p'_2 is found by differentiating the implicit relation (15) for p_2 with respect to p_4. The other variables are given by explicit formulae that are too complicated to list here. However, Figure 8 plots the exact sensitivities of the flow with respect to p_4 within the regions in which flow field is smooth. Note the jumps occurring in the sensitivities at the shock, the contact discontinuity and the two edges of the rarefaction wave.

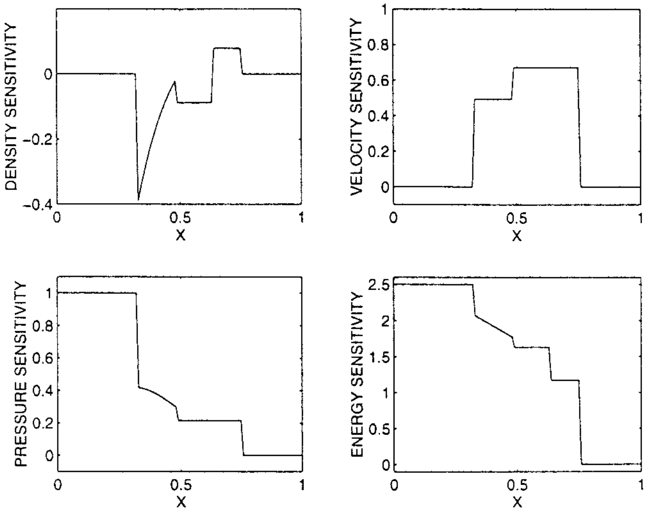

Figure 8. Exact continuous sensitivity with respect to p_4 at $t = 0.148$.

Copyright © 1999 John Wiley & Sons, Ltd.

6.2. Continuous sensitivity equations

The continuous sensitivities are denoted by $\mathbf{S}=(\rho', m', e')^T$, where

$$\rho'=\frac{\partial\rho}{\partial p_4},\quad m'=\frac{\partial m}{\partial p_4}\quad\text{and}\quad e'=\frac{\partial e}{\partial p_4}.$$

The continuous equations for the sensitivity equations are derived by differentiating the flow equations with respect to p_4 to obtain

$$\mathbf{S}_t+\frac{\partial\mathbf{G}(\mathbf{Q},\mathbf{S})}{\partial x}=\mathbf{0},$$

where the sensitivity flux function $\mathbf{G}(\,\cdot\,,\cdot\,)$ is given by

$$\mathbf{G}(\mathbf{Q},\mathbf{S})=\begin{bmatrix} m' \\ \dfrac{2mm'}{\rho}-\dfrac{m^2}{\rho^2}\rho'+p' \\ \left(\dfrac{m'}{\rho}-\dfrac{m}{\rho^2}\rho'\right)(e+P)+\left(\dfrac{m}{\rho}\right)(e'+p') \end{bmatrix},$$

where $m'=\rho'u+\rho u'$ and $e'=[P'/(\gamma-1)]+\frac{1}{2}\rho'u^2+\rho uu'$. The initial conditions for the sensitivity $\mathbf{S}$ are found by differentiating the initial conditions for the flow with respect to p_4, i.e.

$$\mathbf{S}(x,0)=\begin{bmatrix} 0 \\ 0 \\ 1/(\gamma-1) \end{bmatrix}\quad\text{for } x<c,\quad\text{and}\quad \mathbf{S}(x,0)=\mathbf{0}\quad\text{for } x>c.$$

The continuous sensitivity equation only holds inside each of the five regions wherein the flow field is smooth. They do not hold across the shock wave, the contact discontinuity or even at the two edges of the rarefaction wave, since the flux function $\mathbf{F}(\mathbf{Q})$ is non-differentiable at these locations. It is important to note that the continuous sensitivity equations are linear, hyperbolic equations with discontinuous coefficients.

The three numerical methods used in the simulations are:

- the two-step Lax–Wendroff finite difference with the post-processing artificial viscosity term of Lapidus added to the approximate solution;
- the Godunov finite difference scheme, again with the post-processing artificial viscosity term of Lapidus added to the approximate solution;
- a Roe-type scheme [16].

The Roe scheme for the sensitivity equations requires the development of a new numerical scheme [5]. The non-conservation law form of the continuous sensitivity equations are

$$\mathbf{S}_t+A(\mathbf{U})\mathbf{S}_x=-\frac{\partial\mathbf{G}(\mathbf{Q},\mathbf{S})}{\partial\mathbf{Q}}\mathbf{Q}_x,$$

where $A(\mathbf{U})$ is the same Jacobian as encountered in the flow equation. One can solve the sensitivity equation by a 'Strang splitting' approach in which one alternates between solving the homogeneous equation

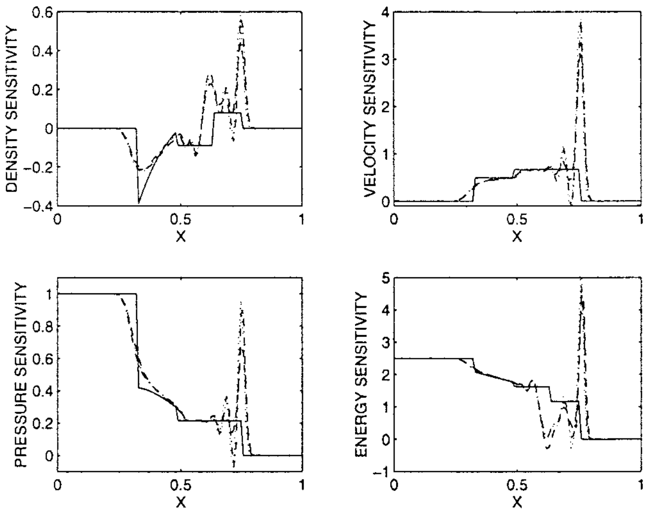

Figure 9. Comparison of three different sensitivity calculation methods using the Lax–Wendroff numerical scheme vs. the exact sensitivity. (· · ·) ADIFOR sensitivity; (·-·) finite difference sensitivity; (---) sensitivity equation method sensitivity; (—) exact sensitivity.

$$\mathbf{S}_t + A(\mathbf{U})\mathbf{S}_x = 0,$$

and the equation

$$\mathbf{S}_t = -\frac{\partial \mathbf{G}(\mathbf{Q}, \mathbf{S})}{\partial \mathbf{Q}}\mathbf{Q}_x.$$

6.3. *Computational results*

For a specific value of the parameter p_4, we calculate sensitivities for the Reimann problem using finite difference quotients, ADIFOR, and the sensitivity equation method for each of three numerical schemes described. For the finite difference quotient approach, both the flow at value of the parameter p_4 and at the perturbed value $p_4 + \Delta p_4$ are solved using the same numerical scheme and then the sensitivities are approximated by a finite difference quotient. For the automatic differentiation sensitivities, the ADIFOR package is applied to the code of each numerical scheme to derive new FORTRAN codes that will calculate both the flow solution and the automatic differentiation flow sensitivity.

For Figures 9–11, the sensitivities are calculated using $\Delta x = 0.01$ and $\Delta t = 0.001$ in the Lax–Wendroff, Godunov and Roe schemes respectively. The grid spacing can have an effect on the accuracy of the sensitivity calculations and, of course, the flow solution calculation. To show the effect of grid size, the Lax–Wendroff sensitivities are calculated for two coarser meshes: the results are given in Figures 12 and 13 for which the grid spacing are ($\Delta t = 0.004$, $\Delta x = 0.04$) and ($\Delta t = 0.002$, $\Delta x = 0.02$) respectively. Clearly, there is substantial agreement between all three sensitivity calculation methods for all three numerical schemes. Therefore, the method or numerical scheme used to calculate sensitivities may not be that important insofar as accuracy is concerned and the easiest and cheapest method should be used. However, none of the methods give good agreement with the exact sensitivities. Certainly, for all combinations of sensitivity calculation method and numerical discretization scheme, the accuracy of the computed sensitivities are nowhere near as good as that for the approximate flow field using the same numerical method.

Copyright © 1999 John Wiley & Sons, Ltd.

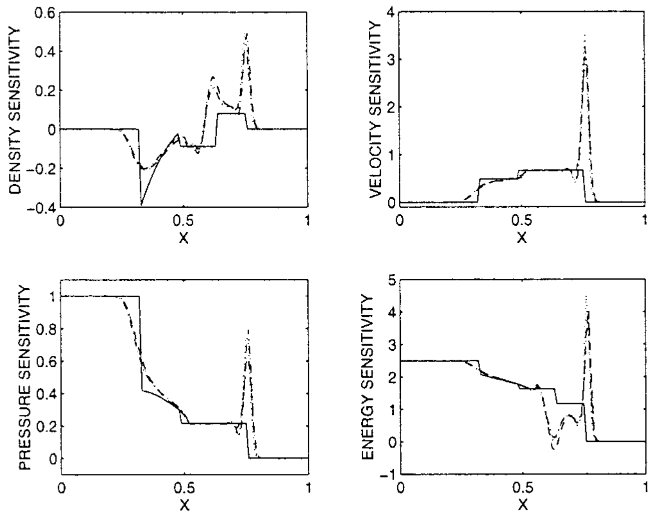

Figure 10. Comparison of three different sensitivity calculation methods using the Godunov numerical scheme vs. the exact sensitivity. (· · ·) ADIFOR sensitivity; (·-·) finite difference sensitivity; (---) sensitivity equation method sensitivity; (—) exact sensitivity.

Shock waves, contact discontinuities, rarefaction waves and the grid spacing have a large impact on the accuracy of the sensitivity calculations. All the sensitivity calculation methods give virtually indistinguishable results for the medium and fine grids. Consider, for example, Figures 9–11. In each of these figures, it is difficult to distinguish between the three different methods except in the regions where large spikes occur. There are differences in the sensitivities if one chooses different grid sizes or numerical method, but if the grid size and numerical method remain fixed, there is little difference between the different sensitivity calculation methods. For example, the sensitivities calculated using the Roe scheme on a fine grid, see

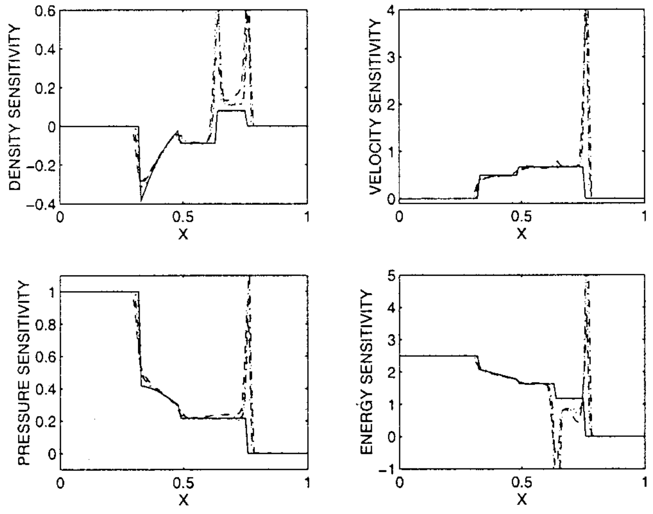

Figure 11. Comparison of three different sensitivity calculation methods using the Roe numerical scheme vs. the exact sensitivity. (· · ·) ADIFOR sensitivity; (·-·) finite difference sensitivity; (---) sensitivity equation method sensitivity; (—) exact sensitivity.

Copyright © 1999 John Wiley & Sons, Ltd.

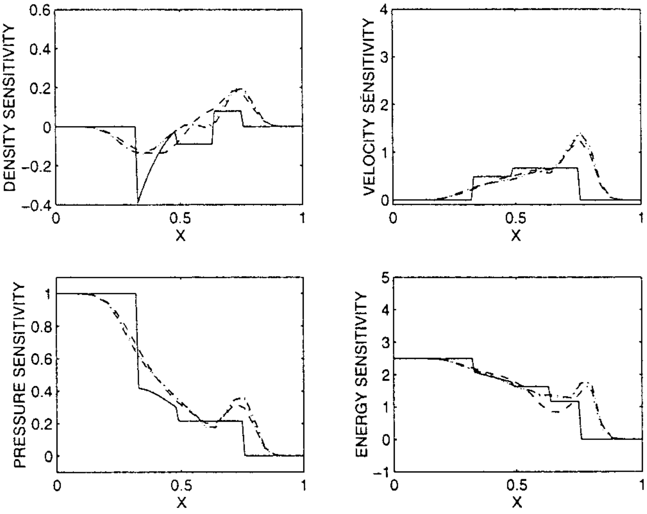

Figure 12. Comparison of the three different sensitivity calculation methods using the Lax–Wendroff numerical scheme on a coarse grid vs. the exact sensitivity. (· · ·) ADIFOR sensitivity; (·-·) finite difference sensitivity; (---) sensitivity equation method sensitivity; (—) exact sensitivity.

Figure 11, are a better approximation to the exact sensitivities than the sensitivities calculated using the Lax–Wendroff method on either a fine grid (Figure 9) or medium grid (Figure 13) but each of the three sensitivity calculation methods yield almost identical results in each of the above figures.

The results for the fine and medium grids are quite interesting; they are very close. This implies that the inconsistencies in the different sensitivity methodologies arise mainly in the area around the spikes and are negligible elsewhere. These spikes are due largely to errors in the sensitivity approximation methods. Hence, it is reasonable to expect that if the spikes can

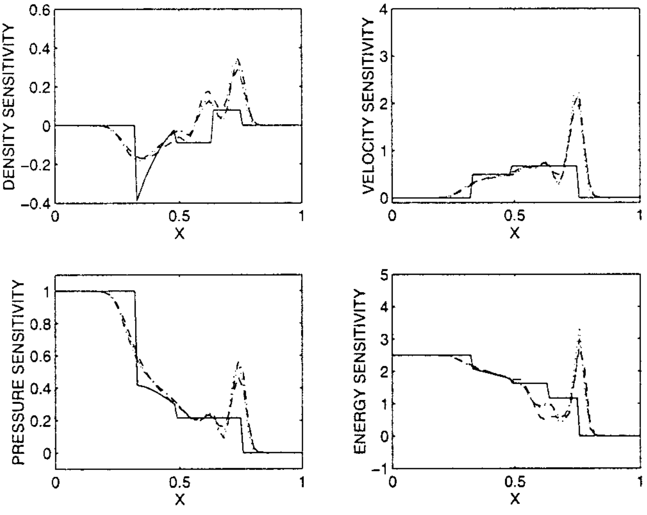

Figure 13. Comparison of the three different sensitivity calculation methods using the Lax–Wendroff numerical scheme on a medium grid vs. the exact sensitivity. (· · ·) ADIFOR sensitivity; (·-·) finite difference sensitivity; (---) sensitivity equation method sensitivity; (—) exact sensitivity.

Copyright © 1999 John Wiley & Sons, Ltd.

be eliminated, then the different sensitivity methodologies can become interchangeable in many applications.

As far as the numerical methods are concerned, the Roe scheme does the best job of approximating the exact continuous sensitivities followed by the Godunov scheme and the Lax–Wendroff scheme. This is as expected, since the Roe scheme does the best job of approximating the exact flow solution followed again by the Godunov and Lax–Wendroff schemes. It is interesting to note that even though the approximate flow solution using the Roe scheme on a fine grid is very close to the exact flow solution, the sensitivities calculated using the Roe scheme on a fine grid are not very close to the exact continuous sensitivities in the region between the shock wave and the contact discontinuity; see Figure 11. This is due largely to the spikes arising at both the shock wave and the contact discontinuity.

The final observation is that all the different sensitivity calculation methods incurred problems at the shock wave, the contact discontinuity, and even the rarefaction wave. At the shock wave and contact discontinuity, large spikes arose as the methods attempted to approximate the δ function at that point. If the grid was coarse, then the spikes were small humps and much of the flow sensitivity information was effectively lost. If the grid was refined, then the spikes became thinner but grew in magnitude. The flow sensitivities of the rarefaction wave involve jump discontinuities at both ends. In all the methods these jumps were not captured very accurately unless the grid was refined and a better numerical scheme, such as the Roe scheme, was used. An interesting thing is that the discontinuities in flow sensitivities are not captured with the same accuracy as that for discontinuities in the flow solution, even when using equivalent methods and grids.

6.4. Flow sensitivities and discontinuities

6.4.1. Flow sensitivities and shocks. Suppose that the flow is characterized by a single parameter α; then, in one dimension, the flow field may be viewed as a function of x, t and α, i.e. $\mathbf{Q} = \mathbf{Q}(x, t; \alpha)$. In a flow simulation, one would pick a particular value α_0 for the parameter and then proceed to determine $\mathbf{Q}$ as a function of x and t, i.e. $\mathbf{Q}(x, t; \alpha_0)$. The location of the shock wave is denoted by $x_{\rm s}$. In general, if the value of the parameter is changed, the location of the shock wave changes so that the shock location is a function of the parameter, i.e. at least locally we can write $x_s = x_s(t; \alpha)$. See Figure 14 for a sketch at a particular instant of time. For a fixed value of the parameter $\alpha = \alpha_0$, the flow variable $\mathbf{Q}(x, t_0; \alpha_0)$ is a discontinuous function of x, and certainly cannot be differentiated with respect to x at $x_s(t_0; \alpha_0)$. Likewise, if the value of x is fixed at, say at $x = x_0$, then for at least some x_0, $\mathbf{Q}(x_0, t_0; \alpha)$, viewed as a function of α, is discontinuous and hence not differentiable with respect to α at $x_{\rm s}(t_0; \alpha) = x_0$.

Flow sensitivity derivatives do not exist at shock waves in much the same way that spatial and temporal flow derivatives do not exist at those locations. In the calculation of flow sensitivities by all three numerical approaches, the presence of shock waves was ignored, e.g. the same algorithm was applied at points in $(x, t; \alpha)$ space where a shock wave is present as was applied at points where the flow is smooth. As a result, approximate flow sensitivities with large spikes at the shock were obtained; these spikes approximate the δ function that the exact flow sensitivities contain at that location.

To see how these spikes arise, consider the finite difference quotient approach to approximating sensitivities. Suppose the flow is determined at (x_1, t_1) for two values α_1 and $(\alpha_1 + \Delta\alpha)$ of the parameter. We approximate the flow sensitivity at (x_1, t_1) by

$$\frac{\partial \mathbf{Q}(x_1, t_1;\ \alpha_1)}{\partial \alpha} \approx \frac{\mathbf{Q}(x_1, t_1;\ \alpha_1 + \Delta\alpha) - \mathbf{Q}(x_1, t_1;\ \alpha_1)}{\Delta\alpha}.$$

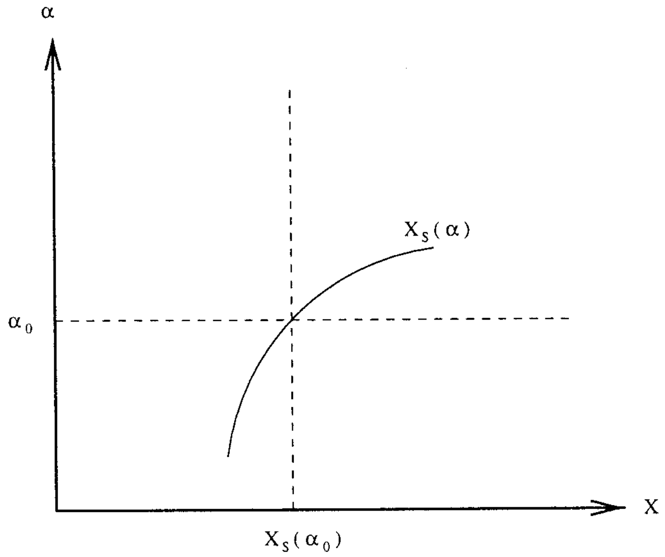

Figure 14. A shock wave in (x, α)-space. Flow variables are discontinuous when one crosses the shock wave $x_s(\alpha)$ at the point $(x_s(\alpha_0), \alpha_0)$ along either the horizontal or vertical dashed lines.

If the flow is smooth and $\Delta\alpha$ is small, then both the numerator and the denominator are small, and in fact, $\Delta\alpha \to 0$, both approach zero in such a way that their ratio converges to $\partial\mathbf{Q}/\partial\alpha(x_1, t_1; \alpha_1)$. Now, suppose that the points $(x_1, t_1; \alpha_1 + \Delta\alpha)$ and $(x_1, t; \alpha)$ in $(x, t; \alpha)$-space lie on opposite sides of the shock wave as depicted in Figure 15; then, even if $\Delta\alpha$ is small, the numerator is relatively large. Hence, the solution of the finite difference sensitivity is large and a spike occurs when differencing across a shock.

In the sensitivity equation approach, the same spikes develop. Again, suppose one parameter α determines the flow so that $\mathbf{Q} = \mathbf{Q}(x, t; \alpha)$. The equation for the sensitivity $\mathbf{S} = \partial\mathbf{Q}/\partial\alpha$ is

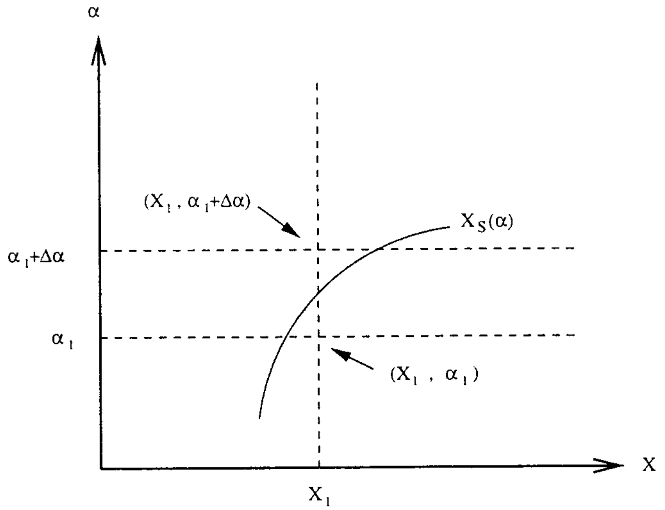

Figure 15. The points (x_1, α_1) and $(x_1, \alpha_1 + \Delta\alpha)$ are on opposite sides of the shock wave in the (x, t_0, α)-plane. The variables at the two points differ significantly even if the two points are close together, i.e. even if $\Delta\alpha$ is small.

Copyright © 1999 John Wiley & Sons, Ltd.

obtained by differentiating the flow equation with respect to α. In full detail, this process goes as follows:

$$\mathbf{0}=\frac{\partial}{\partial\alpha}\left(\frac{\partial\mathbf{U}}{\partial t}+\frac{\partial\mathbf{F}(\mathbf{Q})}{\partial x}\right)=\frac{\partial^2\mathbf{Q}}{\partial\alpha\,\partial t}+\frac{\partial^2\mathbf{F}(\mathbf{Q})}{\partial\alpha\,\partial x}=\frac{\partial}{\partial t}\left(\frac{\partial\mathbf{Q}}{\partial\alpha}\right)+\frac{\partial}{\partial x}\left(\frac{\partial\mathbf{F}(\mathbf{Q})}{\partial\alpha}\right)=\frac{\partial\mathbf{S}}{\partial t}+\frac{\partial}{\partial x}\,\mathbf{G}(\mathbf{Q},\mathbf{S}),$$

where the flux function $\mathbf{G}(\mathbf{Q},\mathbf{S})$ is defined by

$$\mathbf{G}(\mathbf{Q},\mathbf{S})=\frac{\partial\mathbf{F}(\mathbf{Q})}{\partial\alpha}=\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{Q}}(\mathbf{Q})\mathbf{S}.$$

In this derivation, it is important to notice that the order of differentiation has been interchanged. In smooth regions of the flow, there is no difficulty in interchanging derivatives; however, at a shock wave the derivation is not allowable due to the non-existence of the appropriate derivatives at that location. One cannot use the flux function **G** as defined above or one should at least realize that the derivation has a certain amount of error built in and deal with this in the numerical scheme.

It can also be seen that the same spikes arise in the ADIFOR automatic differentiation sensitivities. These spikes arise using the automatic differentiation method because even though the exact solution is discontinuous, the discrete solution is ‘continuous’, e.g. shock waves are smeared out. Hence, ADIFOR has no problem differentiating the function and the result is spikes in the neighbourhood of the discontinuity.

6.4.2. Flow sensitivities, contact discontinuities, and rarefaction waves. Contact discontinuities are another source of error in the sensitivity calculations. Across a contact discontinuity, the velocity u and the pressure p are continuous, while the density ρ, and hence other conserved quantities, experience a jump discontinuity. Thus, for the conserved quantities, the same problems arise as for shock waves and spikes are present in the sensitivities at a contact discontinuity.

Unlike the case for shock waves and contact discontinuities, the flow field is continuous not only within the rarefaction wave but also across its edges. However, the flow is not differentiable across those edges. As a result, the flow sensitivities experience a jump discontinuity at both edges of the rarefaction wave. Numerical schemes, even if they are designed to do a good job capturing shock-type discontinuities, have difficulty capturing the essentially different jumps in the sensitivities occurring at the edges of the rarefaction wave. This can be clearly seen in the computational results.

7. SENSITIVITY-BASED OPTIMIZATION FOR THE REIMANN PROBLEM

We now examine, in the context of the Riemann problem, some issues directly related to flow optimization. The example optimization problem considered is to match, as well as possible, at a specified time t_0, the flow to a desired flow by altering the initial conditions of the problem. The initial conditions that can be altered are the pressure and density to the left and right of the diaphragm. In this study, the single design parameter is chosen to be the initial left, or high, pressure p_4. The initial left density, the initial right pressure and the initial right density are kept fixed throughout the optimization procedure. The cost functional is given by

$$\mathscr{J}(\mathbf{Q})=\frac{1}{2}\int_\Gamma|\mathbf{Q}(x,t_0;\,p_4)-\hat{\mathbf{Q}}(x,t_0)|^2\,\mathrm{d}x,$$

where $\hat{\mathbf{Q}}(x, t_0)$ is the target flow, $\mathbf{Q}(t_0; p_4)$ is the candidate flow, and Γ is the computational domain. In the cost functional, both the target and candidate flows have been evaluated at a specific time $t = t_0$. The flow matching optimization problem is then defined as follows:

Problem 1
Find the initial left pressure p_4 such that $\mathscr{J}(\mathbf{Q})$ is minimized, where $\mathbf{Q}(x, t; p_4)$ is a solution of the one-dimensional Euler equations.

The gradient of the functional is simply its derivative with respect to p_4, i.e.

$$\frac{\mathrm{d}\mathscr{J}}{\mathrm{d}p_4} = \int_\Gamma (\mathbf{Q}(x, t_0;\ p_4) - \hat{\mathbf{Q}}(x, t_0)) \cdot \frac{\partial \mathbf{Q}(x, t_0;\ p_4)}{\partial p_4}\, \mathrm{d}x.$$

The candidate flow for any value of the design parameter p_4 must satisfy the one-dimensional Euler equations.

In practice, one often actually minimizes a discrete approximation to the cost functional. For example, the discrete functional

$$\mathscr{J}^h(\mathbf{Q}^h) = \frac{1}{2} \sum_{j=0}^{J} \Delta x |\mathbf{Q}_j^h(t_0;\ p_4) - \hat{\mathbf{Q}}(x_j,\ t_0)|^2$$

is obtained from $\mathscr{J}$ by a simple composite one-point quadrature rule. For the discrete functional, Γ is subdivided into J intervals of equal length Δx; $\mathbf{Q}_j^h(t_0;\ p_4)$ denotes an approximation to the exact flow $\mathbf{Q}(x_j, t_0;\ p_4)$, where $x_j = j\Delta x$. The target flow $\hat{\mathbf{Q}}(x_j, t_0)$ is assumed known and thus does not have to be approximated at the grid points. We then arrive at the following discrete optimization problem:

Problem 2
Find the initial left pressure p_4 such that $\mathscr{J}^h(\mathbf{Q}^h)$ is minimized, where $\mathbf{Q}^h(x, t; p_4)$ is a solution of an approximation to the flow equations.

The approximation $\mathbf{Q}^h(x, t; p_4)$ can be determined by a favourite CFD algorithm. The gradient of the discrete functional with respect to the design parameter p_4 is given by

$$\frac{\mathrm{d}\mathscr{J}^h}{\mathrm{d}p_4} = \sum_{j=0}^{J} \Delta x (\mathbf{Q}_j^h(t_0;\ p_4) - \hat{\mathbf{Q}}(x_j,\ t_0)) \frac{\partial}{\partial p_4} \mathbf{Q}_j^h(t_0;\ p_4).$$

If the sensitivity of the approximate flow solution $\mathbf{Q}_j^h(t_0;\ p_4)$ is determined exactly, as is the case, in principle, for the automatic differentiation approach, then the gradient of the discrete cost functional can also be determined exactly. In this sense, such an approach leads to a 'consistent gradient', i.e. it is the true gradient of the discrete functional. In fact, if one takes the view, as is often done, that Problem 2 is the optimization problem one really wants to solve, then we have the gradient one really wants. Thus, if the exact sensitivity of the approximate flow is available, then, up to round-off error. Problem 2 can be solved exactly as well; of course. Problem 1 is not solved exactly. The consistency of the gradient in the sense discussed here is viewed by some as an important advantage of the discretize-then-differentiate approach. However, in general, this does not mean that optimizations are without difficulties.

If one takes the differentiate-then-discretize approach, then one does not get a consistent gradient. To see this, first consider the exact gradient; it can be approximated by

$$\left(\frac{\mathrm{d}\mathscr{J}}{\mathrm{d}p_4}\right)^h = \sum_{j=0}^{J} \Delta x (\mathbf{Q}_j^h(t_0;\ p_4) - \hat{\mathbf{Q}}(x_j,\ t_0)) \left(\frac{\partial \mathbf{Q}}{\partial p_4}\right)_j^h (t_0;\ p_4),$$

where $(\partial \mathbf{Q}/\partial p_4)_j^h$ denotes an approximation to the exact sensitivity $(\partial \mathbf{Q}/\partial p_4)$ at the grid point x_j, which may be determined from a discretization of the sensitivity equation. A functional gradient determined by the differentiate-then-discretize is not the true gradient of anything. Furthermore, neither Problem 1 or 2 can be solved exactly by this approach. In this sense, the differentiate-then-discretize approach yields 'inconsistent gradients'. It is possible that this inconsistency can cause difficulties in the solution of Problem 2; however, should they arise, they are usually easy to overcome [11].

We do not have space here to report in detail about numerical experiments connected with the Reimann flow optimization problem [5]. A brief summary of the results is as follows. Optimization seems to be quite robust in the presence of the inaccuracies in sensitivities or adjoints as discussed in the previous sections. Certainly the optimizers can quickly find the position of singularities, e.g. shock waves, in the flow. The spikes indicate large sensitivities near a shock wave or other discontinuities in the flow. However, it seems the accuracy of sensitivities or adjoints does become important when the optimizer works hard as it attempts to find the smooth part of the optimal flow. This is borne out in Reimann problem experiments where we have seen that the performance of the optimizer iteration is improved if one uses the exact sensitivities available for such problems and the presence of spikes, even sharp ones, slow down the convergence.

REFERENCES

1. R Joslin, M. Gunzburger, R. Nicolaides, G. Erlebacher and M. Hussaini, 'An automated methodology for optimal flow control with an application to transition delay', *AIAA J.*, **35**, 816–824 (1997).
2. C. Streett and M. Macaraeg, 'Spectral multi-domain methods for large-scale fluid dynamics simulations', *Appl. Numer. Math.*, **6**, 123–140 (1989).
3. L. Hou and Y. Yan, 'Dynamics and approximations of a velocity tracking problem for incompressible flows with piecewise distributed controls', *SIAM J. Cont. Optim.*, **35**, 1847–1885 (1997).
4. M. Gunzburger and S. Manservisi, 'Analysis and approximation of the velocity tracking problem for Navier–Stokes flows with distributed control', to appear in *SIAM J. Numer. Anal.*
5. J. Appel, 'Sensitivity calculations for conservation laws with application to discontinuous fluid flows', *Ph.D. Thesis*, Virginia Technical, Blacksburg, 1997.
6. J. Appel. A. Godfrey, M. Gunzburger and E. Cliff, 'Optimization-based design in high-speed flows', *CFD for Design and Optimization*, vol. FED-232, ASME, New York, 1995, pp. 61–68.
7. J. Borggaard, 'The sensitivity equation method for optimal design', *Ph.D. Thesis*, Virginia Technical, Blacksburg, 1994.
8. J. Borggaard, J. Burns, E. Cliff and M. Gunzburger, 'Sensitivity calculations of a 2D, inviscid, supersonic forebody problem', in H.T. Banks, K. Ito and R. Fabiano (eds.), *Identification and Control of Systems Governed by Partial Differential Equations*, SIAM, Philadelphia, 1993.
9. D. Gay, 'Algorithm 611 subroutines for unconstrained minimization using a model/trust region approach', *ACM Trans. Math. Soft.*, **9**, 503–524 (1983).
10. M. Applebaum and R. Walters, 'UCFD, an unstructured computational fluid dynamics package', Technical Report, Dept. of Aerospace Engng., Virginia Tech., Blacksburg, 1995.
11. J. Burkardt and M. Gunzburger, 'Sensitivity discrepancy for geometric parameters', *CFD for Design and Optmization*, vol. FED-232, ASME, New York, 1995, pp. 9–15.
12. J. Burkardt, M. Gunzburger and J. Peterson, 'Discretization of cost and sensitivities in shape optimization', in *Computation and Control IV*, Birkhaüser, Boston, 1995, pp. 43–56.
13. C. Bischof, A. Carle, P. Khadermi, A. Mauer and P. Hovland, 'ADIFOR 2.0 users guide', *Technical Report CRPC-95516-S*, Center for Research on Parallel Computation, 1995.
14. C. Bischof, P. Khademi, A. Mauer and A. Carle, 'Adifor 2.0: automatic differentiation of Fortran 77 programs', *Comp. Sci. Eng.*, **3**, 18–32 (1996).
15. C. Bischof, W. Jones, A. Mauer and J. Samareh, 'Application of automatic differentiation to 3D volume grid generation software', *CFD for Design and Optimization*, vol. FED-232, ASME, New York, 1995, pp. 17–22.
16. R. LeVeque, *Numerical Methods for Conservation Laws*, Birkhaüser, Basel, 1991.

Copyright © 1999 John Wiley & Sons, Ltd.