J. Fluid Mech. (1993), *vol.* 253, *pp.* 509–543

Feedback control for unsteady flow and its application to the stochastic Burgers equation

By HAECHEON CHOI, ROGER TEMAM†, PARVIZ MOIN AND JOHN KIM

Center for Turbulence Research, Stanford University, Stanford CA 94305, USA *and* NASA Ames Research Center, Moffett Field, CA 94035, USA

(Received 2 February 1992 and in revised form 1 February 1993)

Mathematical methods of control theory are applied to the problem of control of fluid flow with the long-range objective of developing effective methods for the control of turbulent flows. The procedure of how to cast the problem of controlling turbulence into a problem in optimal control theory is presented using model problems through the formalism and language of control theory. Then we present a suboptimal control and feedback procedure for general stationary and time-dependent problems using methods of calculus of variations through the adjoint state and gradient algorithms. This suboptimal feedback control procedure is applied to the stochastic Burgers equation. Two types of controls are investigated: distributed and boundary controls. The control inputs are the momentum forcing for the distributed control and the boundary velocity for the boundary control. Costs to be minimized are defined as the sum of the mean-square velocity gradient inside the domain for the distributed control or the square velocity gradient at the wall for the boundary control; and in both cases a term was added to account for the implementation cost. Several cases of both controls have been numerically simulated to investigate the performances of the control algorithm. Most cases considered show significant reductions of the costs. Another version of the feedback procedure more effective for practical implementation has been considered and implemented, and the application of this algorithm also shows significant reductions of the costs. Finally, dependence of the control algorithm on the time-discretization method is discussed.

1. Introduction

The potential benefits of managing and controlling turbulent flows that occur in various engineering applications are known to be significant. It is recognized that organized structures in turbulent flows play an important role in turbulent transport (Cantwell 1981). Therefore, attempts to control turbulent flows in engineering applications have focused on manipulation of coherent structures.

Many strategies for controlling turbulent flows have been investigated to achieve different goals such as drag reduction and heat and mass transfer augmentation. There are, in general, many ways of reducing the skin-friction by passive means: riblets, large eddy break-up (LEBU) devices, compliant walls, polymer addition, etc. (for a succinct summary of this subject, see Bushnell & McGinley 1989). Among them, surface-mounted longitudinal grooves in turbulent boundary layers are most successful in

† Permanent address: Université de Paris-Sud, Laboratoire d'Analyse Numérique, Bâtiment 425, 91405 Orsay, France

reducing net drag, in spite of a substantial increase in the surface area (Bechert & Bartenwerfer 1989; Choi, Moin & Kim 1992; Walsh 1983).

Choi *et al.* (1992) used the direct numerical simulation technique to explore concepts for active control of turbulent channel flow with the goal of drag reduction using selective blowing and suction at the wall. The drag reduction (about 20%) was accompanied by significant reductions in the intensity of the wall-layer structures and reductions in the magnitude of Reynolds stresses throughout the flow. Experimental efforts using active (or feedback) control devices to control turbulence are described in Bushnell & McGinley (1989).

When certain aspects of the physics of a problem are well-known, such as the existence of organized patterns, one can devise a scheme to manipulate these patterns, or at least impede or amplify their formation by preassigned kinematic modifications. However, when physics of a phenomenon is not known or is very complicated, it is tempting to appeal to the more systematic but less intuitive methods of control theory. This is the objective of this work: to provide a framework for systematic control of turbulent flows.

The issue of minimizing turbulence in an evolutionary Navier–Stokes flow was addressed from the point of view of optimal control by Abergel & Temam (1990). They derived theoretical results for various physical situations. However, the application of their optimal control algorithm to the unsteady three-dimensional Navier–Stokes equations is not practical due to the great complexity of the algorithm.

In fact the problem of controlling turbulent flows using control theory is extremely difficult. We encounter here two major difficulties which have been addressed by two different segments of the scientific community: the control of nonlinear systems, studied mostly in the control community, and the problems related to the resolution of two- or three-dimensional flows in the presence of turbulence and complicated geometries, studied by the fluid mechanics community.

While the control of linear systems is fairly well understood, the control of nonlinear systems remains the subject of active research at this time, even in finite and small dimensions. For such applications as the control of flight or the control of industrial processes, the objective has been to improve the control processes based on a simplified linear description of the process. The basic theoretical as well as practical issue is the determination of efficient nonlinear feedback controllers. However, as is well known in the fluid mechanics community, nonlinearity leads to complex and often chaotic behaviours and linearization of the governing equations produces approximations that are only valid for a limited time.

Nonlinear control theory has been studied by Lions (1971), in his early work on distributed systems, i.e. in infinite dimension; he addresses the question of existence of an optimal control and the derivation of necessary conditions. Nonlinear control theory has been addressed more recently by numerous authors from the point of view of H^{∞} theory; see, among others, Foias & Tannenbaum (1989) and for the infinite-dimensional case Barbu (1992). Feedbacks of nonlinear distributed systems are addressed for instance by Banach & Baumann (1990), Byrnes & Gilliam (1991), and Kang, Ito & Burns (1991) for the Burgers equation.

The control of fluid flow itself is a rapidly developing subject which has been already addressed by several authors. Beside the work mentioned at the beginning of this introduction, emanating mostly from the fluid mechanics community, more theoretically oriented work can be found in a series of articles by Gunzburger, Hou & Svobodny (1990, 1991, 1992); in a series of articles by Sritharan and co-authors (Sritharan 1991*a*, *b*, 1992; Sritharan *et al.* 1991); and in Abergel & Temam (1990, 1992); see also the references above for the Burgers equation and the book edited by Sritharan (1992) and the proceedings of the IMA Conference edited by Gunzburger (1992).

Let us summarize for the more mathematically oriented reader the theoretical and numerical work done on control of flows. Gunzburger *et al.* consider various optimal control problems (open loop) in fluid mechanics and study mathematical and numerical problems such as the existence of optimal controls, necessary optimality conditions of the first order, the discretization of these problems by finite elements, convergence and error estimates for the discrete problems. Sritharan and co-authors consider also mathematical and numerical problems for control of flows. For open-loop problems they provide the open-loop control using an appropriate version of the Maximum Principle of Pontryagin. They study theoretical questions concerning closed-loop problems in relation with the Hamilton–Jacobi–Bellman equation. This approach provides in principle the feedback law corresponding to the optimal control but necessitates the solution of hyperbolic equations in infinite (or large) dimensions. In the articles mentioned above concerning the optimal control of the Burgers equation, the authors introduce a feedback law for a related linear problem. The solution of the Burgers equation supplemented with this feedback forcing is then studied; it is shown that damping is enhanced and this is numerically confirmed for solutions of these equations which display a discontinuity (shock). Finally the work of Abergel & Temam (1990, 1992), already referred to, concerns theoretical and numerical problems for the control of turbulence: for several physically relevant situations the problem is set as an optimal control problem, existence of optimal control is proved; necessary conditions of optimality are derived; gradient-type algorithms are described which require the classic technique of control theory (in particular adjoint state and adjoint equations) for their effective implementation.

The work that we present here departs from the previous works. Instead of searching for an optimal control, we address the more practical problem of trying to reduce the cost function through a procedure which could be efficiently implemented, i.e. we favour effectiveness over optimality. The determination of the cost function is a part of the modelling of the control problem. As it is explained later, the cost is a weighted sum of the cost generated by the flow that we want to reduce (e.g. the drag force), and the cost of the work necessary to implement the control. From the point of view of control theory the method that we present here is a suboptimal procedure based on the determination, at each instant of time, of the best control among an *a priori* chosen class of feedback controls. For cases in which off-line optimal designs can be constructed, the implementation of real-time feedback controllers usually requires much more computational power than off-line optimal designs. However, it is quite difficult to implement a simple off-line optimal controller for unsteady flows, particularly for turbulent flows. The optimal control procedures suggested by Abergel & Temam (1990) require the iterative solution of the Navier–Stokes equations and their adjoint on the global time period; such computations are out of reach at this time (see also Choi *et al.* 1992). Accordingly, the implementation of off-line and on-line optimal controllers for unsteady flows, especially for turbulent flows, is very difficult. Hence, as compared to optimal control, our suboptimal feedback procedure does not require excessive computer resources because it only requires information at each instant of time. At this stage, we do not have any theoretical justification of our method from the point of view of control theory, except the good numerical results.

As a first step towards the solution of the much more difficult problem of controlling the Navier–Stokes equations, we consider here the control of systems governed by the

Burgers equation. The Burgers equation has been studied extensively both theoretically and numerically. It describes the formulation and decay of weak waves in a compressible fluid as well as being a one-dimensional model of the Navier–Stokes equations. Chambers *et al.* (1988) showed that the statistics of the solutions of the Burgers equation subject to random forcing qualitatively resemble those of the velocity fluctuations normal to the wall in the direct numerical simulation of channel flow by Kim, Moin & Moser (1987).

The objective of this study is to develop a feedback control method of minimizing a cost function, and to apply that method to the Burgers equation as a first step towards application to fluid mechanics problems. The extension of the feedback-control algorithm to the Navier–Stokes equations will necessitate more delicate developments and more extensive computer resources and will be addressed in the future. Section 2 is partly expository and directed to fluid mechanicians not familiar with control theory. It shows how to cast the problem of controlling turbulence in a channel flow into a problem in optimal control theory, and introduces the formalism and language of control theory. Section 3 describes our feedback control algorithm for general stationary and time-dependent problems. Also, §3 presents the method for the purpose of suboptimal feedback laws when an analytical representation of feedback laws, i.e. an explicit functional dependence between the quantity being measured by a sensor and the quantity being controlled by an actuator, is difficult. Then in §4 we apply this method to the distributed and boundary controls of the stochastic Burgers equation. The results of numerous computations with or without control are presented and discussed. A computational issue together with practical (or physical) implementation of the present control algorithm is also addressed in §4. Our method depends partly on the time-discretization method for the state equation, namely the Burgers equation. Hence we study in this section the effect of the time-discretization method on the control algorithms and present the results of the practical control algorithm. Conclusions and discussions are given in §5. Appendices A, B and C contain some technical details on the implementation of the control algorithms.

2. Introduction to control theory: some model problems in flow control

Keeping in mind that turbulent flows are time dependent, we will distinguish between stationary and time-dependent flows and start with the case of stationary flows.

2.1. *Stationary channel flow*

Consider a stationary channel flow, where x is the streamwise direction, z is the spanwise direction, and the walls are at $y = \pm 1$. The mass flux is prescribed and is equal to M. Periodicity of velocities and pressure is assumed in the z-direction, and periodicity of velocities with an unknown drop of pressure is assumed in the x direction. Let $u = (u_1, u_2, u_3)$ denote the velocity vector of the fluid, and assume that the flow is controlled through blowing and suction at the wall, i.e. through the wall-normal velocity at the wall

$$\phi = u_2|_w, \tag{2.1}$$

where $\iint \phi \,\mathrm{d}x\,\mathrm{d}z = 0$ is imposed so that the mass flux M is constant. It can be shown that the stationary Navier–Stokes equations reduce to a functional equation for u (see e.g. Temam 1984, 1991) involving ϕ:

$$\nu A u + R(u, \phi) = 0. \tag{2.2}$$

Here $\nu > 0$ is the kinematic viscosity, A is the so-called Stokes operator, and R corresponds to the inertial and boundary terms and is a function of u and ϕ; R actually depends on M although the dependence is not made explicit here.

A typical optimal control problem for equation (2.2) is the following: find the best ϕ such that some observation $\gamma = Cu$ achieves some desired value γ_d or is at least as close as possible to γ_d, where C is a general linear or nonlinear operator which may involve integrals of u and/or derivatives of u. In the language of control theory: u is the *state* of the system and (2.2) is the *state equation*; ϕ is the *control*; γ is the *observation*.

The *cost function* could be, for instance, the function $J = J(\phi)$†

$$J(\phi) = \tfrac{1}{2}l\|\phi\|^2 + \tfrac{1}{2}m\|Cu-\gamma_d\|^2. \tag{2.3}$$

Here $\frac{1}{2}m\|Cu-\gamma_d\|^2$ ($m > 0$, $\|Cu-\gamma_d\| = L_2$ norm of $Cu-\gamma_d$) accounts for the cost of γ being different from γ_d; $\frac{1}{2}l\|\phi\|^2$ ($l \geqslant 0$, $\|\phi\| = L_2$ norm of ϕ) is the cost of implementing the control itself; l/m is small or zero for cheap controls and large for expensive controls. For example, high blowing and suction flow rates are reflected in high values of $\frac{1}{2}l\|\phi\|^2$. High values of l/m may also be used to empirically account for indirect costs such as expensive equipment for realizing fast actuator response. Values of l and m in the cost function are dimensional quantities and are usually prescribed from a parametric study. Keep in mind, however, that (2.2) is an idealized problem which, owing to the absence of turbulence, would only make sense physically for very viscous fluids.

The mathematical formulation of the problem is the following: find ϕ which minimizes J subject to (2.2), i.e. in the notation used in the optimization theory,‡

$$\mathrm{Inf}_\phi\, J(\phi). \tag{2.4}$$

The control ϕ can be unrestricted or restricted to some admissible set of controls $\mathscr{U}_{ad}$ due to physical and technological limitations.

The methods of calculus of variations indicate that a problem such as (2.4) possesses at least one solution and give us some characterizations on the best ϕ through the adjoint state and some algorithms to reach the best (optimal) control. Feedback theory involves constructing ϕ as a function of u or some observation of u. Although feedback schemes are mainly relevant to time-dependent problems, we can formulate such a scheme here.

Searching for the best feedback in a prescribed class of feedbacks, we reduce the problem to a parameter optimization one. For instance, without advocating such a choice, we could look for

$$\phi = E + Fu, \tag{2.5}$$

where E and F are a scalar function on the wall and an operator to be determined, respectively. Now problem (2.4) with (2.5) substituted into (2.2) becomes: find E, F which minimize $J(\phi) = \tilde{J}(E,F)$ subject to (2.2) and (2.5):

$$\mathrm{Inf}_{E,F}\, \tilde{J}(E,F). \tag{2.6}$$

† u is a function of ϕ through (2.2), $u = u(\phi)$. Hence the second term of J is also a function of ϕ. Note that u is the traditional notation for the control in control theory, and it is also commonly used for the velocity in fluid mechanics. We use the latter convention here.

‡ In optimization theory $\mathrm{Inf}_\phi\, J(\phi)$ denotes equivalently either the problem of minimizing $J(\phi)$ in the class of ϕ values or the actual value of the corresponding infimum.

Instead of (2.5) more general shape functions $\theta_0(u), \ldots, \theta_r(u)$ could be considered with

$$\phi = \sum_{i=0}^{r} \alpha_i \theta_i(u). \tag{2.7}$$

The practical importance of these shape functions is discussed in §§3.1, 4.3 and 4.4. Note that as in (2.5), the θ_i are functionals of u and not simple pointwise functions. They may involve complex or non-local operations such as differentiation or integration of u. The α_i are determined through a control algorithm and thus are *a posteriori* functions of u.

2.2. *Time-dependent channel flow*

The *state equation* is the Navier–Stokes equations including the boundary condition (2.1) and other boundary conditions. We infer from the mathematical theory of the Navier–Stokes equations (see e.g. Ladyzhenskaya 1963 and Temam 1984, 1991) that these conditions and equations amount to an evolution equation of infinite dimension for the velocity field $u = u(x, t)$. It reads (compare to (2.2))

$$\frac{\partial u}{\partial t} + \nu A u + R(u, \phi) = 0. \tag{2.8}$$

Here u is the velocity vector field; again, R accounts for the inertial and boundary terms and depends on the mass flux M, although the dependence on M is not made explicit here.

The drag is essentially measured on average by $D = D(u)$:

$$D = \iint \left[\left. \frac{\partial u_1}{\partial x_2} \right|_{x_2=-1} - \left. \frac{\partial u_1}{\partial x_2} \right|_{x_2=1} \right] \mathrm{d}x_1 \, \mathrm{d}x_3. \tag{2.9}$$

Here $x_1 = x$, $x_2 = y$, $x_3 = z$, and $x_2 = y = \pm 1$ are the walls. The choice of the cost function is at our disposal and depends on the costs that we want to reduce. If we choose to reduce a time average of the drag as expressed by (2.9), then a plausible cost function could be

$$J(\phi) = \tfrac{1}{2} l \frac{1}{T} \int_0^T \iint_w |\phi|^2 \, \mathrm{d}x_1 \, \mathrm{d}x_3 \, \mathrm{d}t + \tfrac{1}{2} m \frac{1}{T} \int_0^T |D|^2 \, \mathrm{d}t, \tag{2.10}$$

where D is a function of ϕ through u which itself is a function of ϕ. A control problem like (2.4) can be posed: find $\phi = \phi(x_1, x_3, t)$ which minimizes J subject to (2.8) and (2.10):

$$\mathrm{Inf}_{\phi} \, J(\phi). \tag{2.11}$$

The methods of control theory and calculus of variation (Lions 1971), as developed in Abergel & Temam (1990), prove the existence of an optimal control (the best ϕ) and produce an algorithm for its determination. However, in their present form, and especially for turbulent flows, these classical methods require the iterative solution of the Navier–Stokes equations and their adjoint (see §3) on the whole and large interval $(0, T)$; such computations are out of reach at this time (Choi *et al.* 1992). Furthermore, optimal control of unsteady flows depends on the initial distribution of velocities $u|_{t=0}$, although one would hope that the effects of initial velocities dissipate as T becomes large.

If (2.8) were linear, the optimal control would be given by a linear feedback law:

$$\phi = E + Fu, \tag{2.12}$$

where F is an operator that is the solution of a Riccati-type equation, and E is easily determined. When (2.8) is nonlinear, to the best of our knowledge there is no general method for constructing the feedback law corresponding to the optimal control even for finite and small dimensions (as in flight control), not to mention high- or infinite-dimensional problems. There are also no general nonlinear estimators and the solution of a control problem could require in principle the repeated solution of the state equation with different forcing terms.

We describe below some empirical and not yet fully mathematically justified procedures proposed to address these problems and overcome these difficulties.

3. Suboptimal control and feedback procedures

In this section, we present a systematic approach to the mathematical formulation of the problem of minimizing a cost function using feedback control and parameter optimization procedures. We consider first the stationary case and then the more important and more relevant case of time-dependent problems.

3.1. *Stationary problem*

Equations (2.2), (2.3) and (2.4) define an optimal control problem which can be satisfactorily solved by a gradient algorithm (although a conjugate gradient method would be better, we now restrict ourselves to a gradient algorithm for simplicity).

The gradient algorithm consists of computing the Fréchet derivative†

$$\frac{\mathscr{D}J}{\mathscr{D}\phi} \tag{3.1}$$

and using the following iterative process for the cost minimization:

$$\phi^{k+1} - \phi^k = -\rho \frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^k), \tag{3.2}$$

where ϕ^k is a member of a sequence of controls and ρ is the parameter of descent whose optimal value can be found either by a trial-and-error procedure or by relevant theoretical studies (see e.g. Luenberger 1973). By Taylor's formula and (3.2),

$$\left.\begin{aligned} J(\phi^{k+1}) &\approx J(\phi^k) + \frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^k)(\phi^{k+1} - \phi^k), \\ J(\phi^{k+1}) &\approx J(\phi^k) - \rho \left| \frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^k) \right|^2, \end{aligned}\right\} \tag{3.3}$$

so that the sequence $J(\phi^k)$ is clearly decreasing. With the same methods as in Abergel & Temam (1990) relying on optimization theory, we can prove that the sequence ϕ^k will converge to an optimal control for suitable ρ if the initial value ϕ^0 is chosen sufficiently close to an optimal state.

† When it exists, the Fréchet differential of J in the direction of $\hat{\phi}$ is defined by (Finlayson 1972)

$$\frac{\mathscr{D}J}{\mathscr{D}\phi}\hat{\phi} = \lim_{\epsilon \to 0} \frac{J(\phi + \epsilon\hat{\phi}) - J(\phi)}{\epsilon}.$$

The introduction of the adjoint state and adjoint state equation produces a convenient way to compute the Fréchet derivative (3.1). Let

$$\eta = \frac{\mathscr{D}u}{\mathscr{D}\phi}\hat{\phi}, \tag{3.4}$$

where the right-hand side of (3.4) is the Fréchet differential of u with respect to ϕ applied to a test function $\hat{\phi}$ (of the same type as ϕ). Then, by Fréchet differentiation of (2.2), we promptly see that η is solution of the equation

$$\nu A\eta + \frac{\mathscr{D}R}{\mathscr{D}u}(u,\phi)\,\eta + \frac{\mathscr{D}R}{\mathscr{D}\phi}(u,\phi)\,\hat{\phi} = 0. \tag{3.5}$$

By Fréchet differentiation of the functional J in (2.3) and using (3.4) we obtain

$$\frac{\mathscr{D}J}{\mathscr{D}\phi}\hat{\phi} = l\langle \phi, \hat{\phi}\rangle + m\langle Cu(\phi) - \gamma_d, C\eta\rangle,$$

where $\langle \cdot, \cdot \rangle$ indicates the inner product. Define the adjoint state ζ through the following adjoint state equation (see e.g. Luenberger 1973)

$$\nu A^*\zeta + \left(\frac{\mathscr{D}R}{\mathscr{D}u}(u,\phi)\right)^* \zeta = C^*(Cu(\phi) - \gamma_d). \tag{3.6}$$

In (3.6) and hereafter asterisks indicate adjoint operators with respect to the inner product under consideration $\langle \cdot, \cdot \rangle$. Then

$$\begin{aligned}
\langle Cu(\phi) - \gamma_d, C\eta\rangle &= \langle C^*(Cu(\phi) - \gamma_d), \eta\rangle \\
&= \left\langle \nu A^*\zeta + \left(\frac{\mathscr{D}R}{\mathscr{D}u}(u,\phi)\right)^* \zeta, \eta \right\rangle \\
&= \left\langle \zeta, \nu A\eta + \left(\frac{\mathscr{D}R}{\mathscr{D}u}(u,\phi)\right)\eta \right\rangle \\
&= -\left\langle \zeta, \frac{\mathscr{D}R}{\mathscr{D}\phi}(u,\phi)\,\hat{\phi} \right\rangle \quad \text{(by (3.5))} \\
&= -\left\langle \left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u,\phi)\right)^* \zeta, \hat{\phi} \right\rangle.
\end{aligned}$$

Hence, we get
$$\frac{\mathscr{D}J}{\mathscr{D}\phi}\hat{\phi} = l\langle \phi, \hat{\phi}\rangle - m\left\langle \left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u,\phi)\right)^* \zeta, \hat{\phi} \right\rangle.$$

Since $\hat{\phi}$ is an arbitrary test function, we deduce that

$$\frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^k) = l\phi^k - m\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u^k,\phi^k)\right)^* \zeta^k \tag{3.7}$$

and we are in a position to implement the gradient algorithm (3.2) Note that the solution of the adjoint equation is used to obtain the Fréchet derivative in (3.2). Once ϕ^k is known, compute the adjoint state ζ^k by solving (3.6) with $\phi = \phi^k$ and $u = u^k$. Obtain ϕ^{k+1} from (3.2) using (3.7). Then compute u^{k+1} by solving the state equation (2.2) with $\phi = \phi^{k+1}$, and continue until convergence.

3.1.1. *Suboptimal feedback laws*

The feedback procedure described above results in the determination of ϕ as an implicit function of u in the entire domain: indeed, in order to update the control input ϕ, one has to solve the adjoint state equation (3.6), in which data on u may be needed for the entire domain. However, for the practical implementation of the control algorithm such a detailed knowledge of u is not available. Of course, in some control problems, in particular in the linear case, the construction of a dynamical model of the actual system is quite possible using appropriate filters or state estimators (e.g. Kalman filters and compensators), so one may not necessarily need the knowledge of u in the entire domain. Given limited data, such state estimators aim to provide important dynamic behaviours of the actual systems. However, it is quite difficult to find such a dynamical model in nonlinear unsteady flows, especially in turbulent flows. At the end of §4, we present a method to avoid a detailed knowledge of u for the boundary control of the stochastic Burgers equation.

If the control ϕ is explicitly determined from measurable quantities, it will be much easier to construct feedback control devices. Such *feedback laws* are pursued in this subsection. Physically a feedback law amounts to a formula relating the quantity being measured by a sensor such as pressure or stresses or temperature to the quantity being controlled by an actuator such as blowing rate at the boundary.

In the previous subsection, we indicated how to determine ϕ which minimizes the cost. For most nonlinear control problems, ϕ has an implicit dependence on u, i.e. ϕ cannot be represented as an explicit function of u. However, it may be possible, from physical intuition or experience, that feedback laws for ϕ, $\phi = \phi(u)$, can be constructed which produce a reduction of costs. For example, in the drag reduction study by Choi *et al.* (1992), control relations such as $\phi = -u_2|_{y^+\approx 10}$ and $\phi = \alpha_1 \partial/\partial x_3(\partial u_3/\partial x_2)|_w$ give a significant reduction of drag, where ϕ is the blowing and suction at the wall, u_2 and u_3 are the normal and spanwise velocities, respectively, and α_1 is a constant. Here, we present a quite general framework which should be able to accommodate even the experimental experience gained from the physical observation of unsteady flows.

Suboptimal feedback laws can be implemented in a similar manner by looking for the best feedback,

$$\phi = \phi(u), \tag{3.8}$$

in a particular class of functions corresponding to a suitable approximation of (3.8)

$$\phi = \alpha_0 + \alpha_1 \theta_1(u), \tag{3.9}$$

where $\theta_1(u)$ is prescribed from physical intuition or experience, and α_0 and α_1 are determined through a control algorithm and thus have an implicit dependence on u, i.e. $\alpha_0 = \alpha_0(u)$ and $\alpha_1 = \alpha_1(u)$. In fact, (3.9) contains two elements of (2.7) ($\theta_0(u) = 1$). As we have already mentioned, $\theta_1(u)$ can be a simple function of u or it can be a complicated functional involving integrals or derivatives of u. For example, $\theta_1(u)$ can be the velocity derivative at the wall which can be measured directly in the physical implementation of the control algorithm and, thus, $\theta_1(u)$ represents the quantity that is measured using suitable sensors in the control process. Note that, when $\theta_1 = 0$ (no input from the physical experience), the optimization problem is the initial control problem (2.4). Hence, the optimization problem with $\theta_1 = 0$ produces the optimal control but it does not provide an explicit feedback law. The feedback law in (3.9) can be obtained from control theory for linear optimal or suboptimal problems (see (2.12)). For nonlinear control problems, however, if the function $\theta_1(u)$ cannot be determined by control theory, it can be specified by physical intuition. The proper choice of θ_1 may produce simple expressions for α_0 and α_1 (e.g. constant values). Such a feedback will be certainly easy to implement. One of our objectives, therefore, is to inquire whether one can determine α_0 and α_1 as explicit functions of measurable quantities which can be prescribed *a priori* in practical implementations.

For a feedback of this type, set $e = \{\alpha_0, \alpha_1\}$. The cost function J is chosen to be a function $\tilde{J}$ of α_0 and α_1 through (2.3) and (3.9)†:

$$\tilde{J}(e) = \tfrac{1}{2}l\|e\|^2 + \tfrac{1}{2}m\|Cu-\gamma_d\|^2, \tag{3.10}$$

where $\|e\|^2 = \|\alpha_0\|^2 + \|\alpha_1\|^2$.

The analogue of the gradient algorithm (3.2) consists of constructing two sequences α_0^k, α_1^k, recursively defined by

$$\alpha_0^{k+1} - \alpha_0^k = -\rho_0 \frac{\mathscr{D}\tilde{J}}{\mathscr{D}\alpha_0}(\alpha_0^k, \alpha_1^k), \quad \alpha_1^{k+1} - \alpha_1^k = -\rho_1 \frac{\mathscr{D}\tilde{J}}{\mathscr{D}\alpha_1}(\alpha_0^k, \alpha_1^k). \tag{3.11}$$

Note that the parameters of descent $\rho > 0$ are chosen differently in the two equations (3.11).

The introduction of the adjoint state and adjoint state equation again produces a convenient way to compute the Fréchet derivatives in (3.11). Let us define η first, using the Fréchet differential in the direction of $\hat{e} = \{\hat{\alpha}_0, \hat{\alpha}_1\}$

$$\eta = \frac{\mathscr{D}u}{\mathscr{D}e}\hat{e} = \frac{\mathscr{D}u}{\mathscr{D}\phi}\frac{\mathscr{D}\phi}{\mathscr{D}e}\hat{e}, \tag{3.12}$$

where

$$\begin{aligned} \frac{\mathscr{D}\phi}{\mathscr{D}e}\hat{e} &= \frac{\mathscr{D}\phi}{\mathscr{D}\alpha_0}\hat{\alpha}_0 + \frac{\mathscr{D}\phi}{\mathscr{D}\alpha_1}\hat{\alpha}_1 \\ &= \hat{\alpha}_0 + \hat{\alpha}_1\theta_1 + \alpha_1\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\frac{\mathscr{D}u}{\mathscr{D}e}\hat{e} \\ &= \hat{\alpha}_0 + \hat{\alpha}_1\theta_1 + \alpha_1\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\eta. \end{aligned}$$

Then, by Fréchet differentiation of (2.2), we promptly see that η is the solution of the equation

$$\nu A\eta + \frac{\mathscr{D}R}{\mathscr{D}u}\eta + \frac{\mathscr{D}R}{\mathscr{D}\phi}\left(\hat{\alpha}_0 + \hat{\alpha}_1\theta_1 + \alpha_1\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\eta\right) = 0,$$

or

$$\left(\nu A + \frac{\mathscr{D}R}{\mathscr{D}u} + \frac{\mathscr{D}R}{\mathscr{D}\phi}\alpha_1\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\right)\eta + \frac{\mathscr{D}R}{\mathscr{D}\phi}(\hat{\alpha}_0 + \hat{\alpha}_1\theta_1) = 0. \tag{3.13}$$

By Fréchet differentiation of (3.10) using (3.12) we obtain

$$\frac{\mathscr{D}\tilde{J}}{\mathscr{D}e}\hat{e} = l\langle\alpha_0, \hat{\alpha}_0\rangle + l\langle\alpha_1, \hat{\alpha}_1\rangle + m\langle Cu-\gamma_d, C\eta\rangle.$$

Define now the adjoint state ζ through the following adjoint state equation:

$$\left(\nu A + \frac{\mathscr{D}R}{\mathscr{D}u} + \frac{\mathscr{D}R}{\mathscr{D}\phi}\alpha_1\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\right)^* \zeta = C^*(Cu-\gamma_d). \tag{3.14}$$

† We implicitly assume that $\tilde{J}(e)$ is finite. If this is not true for all e in the considered class of feedbacks, we discard those for which $\tilde{J}(e)$ is infinite. Note that this difficulty does not arise in the discretized problem.

Then by the same procedure as before we get

$$\frac{\mathscr{D}\tilde{J}}{\mathscr{D}e}\hat{e} = l\langle\alpha_0, \hat{\alpha}_0\rangle + l\langle\alpha_1, \hat{\alpha}_1\rangle - m\left\langle\zeta, \frac{\mathscr{D}R}{\mathscr{D}\phi}(\hat{\alpha}_0 + \hat{\alpha}_1\theta_1)\right\rangle$$

$$= \left\langle l\alpha_0 - m\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}\right)^* \zeta, \hat{\alpha}_0\right\rangle + \left\langle l\alpha_1 - m\theta_1\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}\right)^* \zeta, \hat{\alpha}_1\right\rangle. \quad (3.15)$$

Since $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are arbitrary test functions, we deduce that

$$\left.\begin{aligned}\frac{\mathscr{D}\tilde{J}}{\mathscr{D}\alpha_0}(\alpha_0^k, \alpha_1^k) &= l\alpha_0^k - m\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u^k, \phi^k)\right)^* \zeta^k,\\ \frac{\mathscr{D}\tilde{J}}{\mathscr{D}\alpha_1}(\alpha_0^k, \alpha_1^k) &= l\alpha_1^k - m\theta_1(u^k)\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u^k, \phi^k)\right)^* \zeta^k\end{aligned}\right\} \quad (3.16)$$

and we are in a position to implement the gradient algorithm (3.11). Note that the solution of the adjoint equation is used to obtain the Fréchet derivatives in (3.11). Once α_0^k and α_1^k are known, compute the adjoint state ζ^k by solving (3.14) with $\phi = \phi^k$ and $u = u^k$. Obtain α_0^{k+1} and α_1^{k+1} from (3.11) using (3.16). Then compute u^{k+1} by solving the state equation (2.2) with ϕ^{k+1} given by (3.9), and continue until convergence.

3.2. *Time-dependent problem*

We now consider the case of time-dependent problems. The suboptimal procedure that we propose in this case consists of the following:

(i) discretize the state equation in time;

(ii) at each instant of time, the discretized equation is a stationary one to which the above procedure is applied, while the cost function is an instantaneous version of (2.10) (i.e. no time averaging, see (3.20)).

This procedure means that, at each instant of time, we are directing the flow in a direction that produces the decay of the instantaneous cost function. Of course, there is no reason to believe that the controls will be optimal, or even that the cost will actually decay in the long range. However, the numerical experiments conducted in the case of the stochastic Burgers equation and the other model problems (not reported here) show that indeed the cost function decreases significantly without being monotonically decreasing all the time (see §4 for the Burgers equation).

Consider the evolution state equation (2.8); again, this could be the original Navier–Stokes equations for channel flow. For step (i) we consider here the Crank–Nicolson method:

$$\frac{u^n - u^{n-1}}{\Delta t} + \tfrac{1}{2}\nu(Au^n + Au^{n-1}) + \tfrac{1}{2}(R(u^n, \phi^n) + R(u^{n-1}, \phi^{n-1})) = 0, \quad (3.17)$$

which we rewrite as

$$\mathscr{A}u + \mathscr{R}^n(u, \phi) = 0, \quad (3.18)$$

with $u = u^n$, $\phi = \phi^n$, and

$$\left.\begin{aligned}\mathscr{A}u &= u^n + \tfrac{1}{2}\nu\Delta t\, Au^n,\\ \mathscr{R}^n(u^n, \phi^n) &= -u^{n-1} + \tfrac{1}{2}\Delta t(\nu Au^{n-1} + R(u^n, \phi^n) + R(u^{n-1}, \phi^{n-1})).\end{aligned}\right\} \quad (3.19)$$

At each step n, the cost function J is still given by (2.3):

$$J^n = J(\phi^n) = \tfrac{1}{2}l\|\phi^n\|^2 + \tfrac{1}{2}m\|Cu^n - \gamma_d\|^2, \quad (3.20)$$

with u^n a function of ϕ^n ($= u^n(\phi^n)$) through (3.17)–(3.19). Note that for a sufficiently small Δt there exists a unique solution u^n to (3.17). Hence the difficulty of non-uniqueness of solution for (2.2) does not arise for (3.18).

The adjoint state is defined as in (3.4)–(3.6):

$$\eta^n = \frac{\mathscr{D}u^n}{\mathscr{D}\phi}\hat{\phi}, \tag{3.21}$$

$$\mathscr{A}\eta^n + \frac{\mathscr{D}\mathscr{R}}{\mathscr{D}u}(u^n,\phi^n)\,\eta^n + \frac{\mathscr{D}\mathscr{R}}{\mathscr{D}\phi}(u^n,\phi^n)\,\hat{\phi} = 0, \tag{3.22}$$

$$\mathscr{A}^*\zeta^n + \left(\frac{\mathscr{D}\mathscr{R}}{\mathscr{D}u}(u^n,\phi^n)\right)^*\zeta = C^*(Cu^n-\gamma_d). \tag{3.23}$$

The gradient algorithm (3.2) now reads

$$\phi^{n,k+1}-\phi^{n,k} = -\rho\frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^{n,k}), \tag{3.24}$$

where $\phi^{n,k}$ is a member of a sequence of controls at a given time step n, ρ is the parameter of descent, and k is the iteration index at each time step. By Taylor's formula, as in (3.3), for all n,k,

$$J(\phi^{n,k+1}) \leqslant J(\phi^{n,k}),$$

and as $k\to\infty$, $\phi^{n,k}$ converges to ϕ^n which achieves the minimum of J^n. It is not necessarily true that the minimum of J^n decreases as n increases, i.e. for all n,

$$J^n \leqslant J^{n-1}. \tag{3.25}$$

Using the property of the adjoint operator as has been done in previous subsection, we obtain

$$\frac{\mathscr{D}J}{\mathscr{D}\phi}(\phi^{n,k}) = l\phi^{n,k}-\tfrac{1}{2}m\Delta t\left(\frac{\mathscr{D}R}{\mathscr{D}\phi}(u^{n,k},\phi^{n,k})\right)^*\zeta^{n,k}, \tag{3.26}$$

and we are in a position to implement the gradient algorithm (3.24). Note that the solution of the adjoint equation is used to obtain the Fréchet derivative in (3.24). Once $\phi^{n,k}$ is known, compute the adjoint state $\zeta^{n,k}$ by solving (3.23) with $\phi^n=\phi^{n,k}$ and $u^n = u^{n,k}$. Obtain $\phi^{n,k+1}$ from (3.24) using (3.26). Then compute $u^{n,k+1}$ by solving the state equation (3.18) with $\phi^n = \phi^{n,k+1}$, and continue until convergence.

Suboptimal feedback laws for the time-dependent problem can be implemented in a manner similar to that described in §3.1.1.

4. Application to the stochastic Burgers equation

As a first step towards application to the problems in fluid mechanics, the feedback control procedures described in §3 are applied to the Burgers equation subject to random forcing. This equation contains nonlinear convection and diffusion terms and its solution exhibits a chaotic nature; these qualities make it a natural model for the more complicated Navier–Stokes equations. We first specify the form of the Burgers equation that we consider (§4.1). Then we show how to implement our feedback procedure for distributed and boundary control problems (§4.2) and present and discuss the results of our numerical experiments (§4.3). The form of the effective

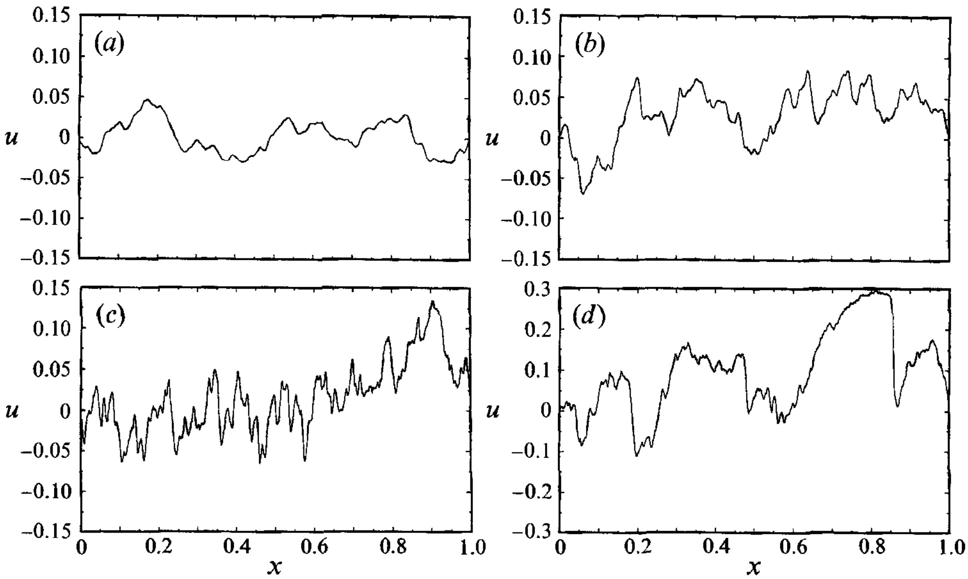

FIGURE 1. Instantaneous velocity profiles. (*a*) $Re = 500$, $\Delta t_r = 0.1$; (*b*) $Re = 1500$, $\Delta t_r = 0.1$; (*c*) $Re = 4500$, $\Delta t_r = 0.1$; (*d*) $Re = 4500$, $\Delta t_r = 1$.

feedback law is discussed in §4.4. We also discuss several implementation issues (discretization of the equation and practical implementation of the feedback procedure) in §4.5.

4.1. *The Burgers equation with random forcing*

Consider the randomly forced Burgers equation with no-slip boundary conditions

$$\left.\begin{aligned}&\frac{\partial \tilde{u}}{\partial \tilde{t}}+\frac{\partial}{\partial \tilde{x}}\frac{\tilde{u}^2}{2}=\nu\frac{\partial^2\tilde{u}}{\partial \tilde{x}^2}+\tilde{\chi}(\tilde{x},\tilde{t}), \quad 0<\tilde{x}<L,\\ &\tilde{u}(\tilde{x}=0)=\tilde{u}(\tilde{x}=L)=0,\end{aligned}\right\} \tag{4.1}$$

where $\tilde{u}$ is the velocity, ν the kinematic viscosity, $\tilde{\chi}$ the random forcing, and L the length of the computational domain. In the absence of forcing ($\tilde{\chi}=0$), the solutions of (4.1) decay to zero from any bounded initial data.

The forcing function $\tilde{\chi}$ is a white noise random process in $\tilde{x}$ with zero mean (see Chambers *et al.* 1988; Bensoussan & Temam 1972, 1973). The mean-square value of the dimensional forcing, σ^2, defines a velocity scale $U=(\sigma L)^{\frac{1}{2}}$. The Burgers equation in non-dimensional form using U and L as the typical velocity and length reads

$$\left.\begin{aligned}&\frac{\partial u}{\partial t}+\frac{\partial}{\partial x}\frac{u^2}{2}=\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}+\chi(x,t), \quad 0<x<1,\\ &u(x=0)=u(x=1)=0,\end{aligned}\right\} \tag{4.2}$$

where u, x, t and χ are dimensionless quantities, Re is the Reynolds number UL/ν, and

$$\langle\chi\rangle_x=0, \quad \langle\chi^2\rangle_x=1. \tag{4.3}$$

Here $\langle\cdot\rangle_x$ denotes the average value over space. A Crank–Nicolson method in time and second-order centred differences in space are used to discretize (4.2). A Newton iterative method is used to solve the discretized nonlinear equation.

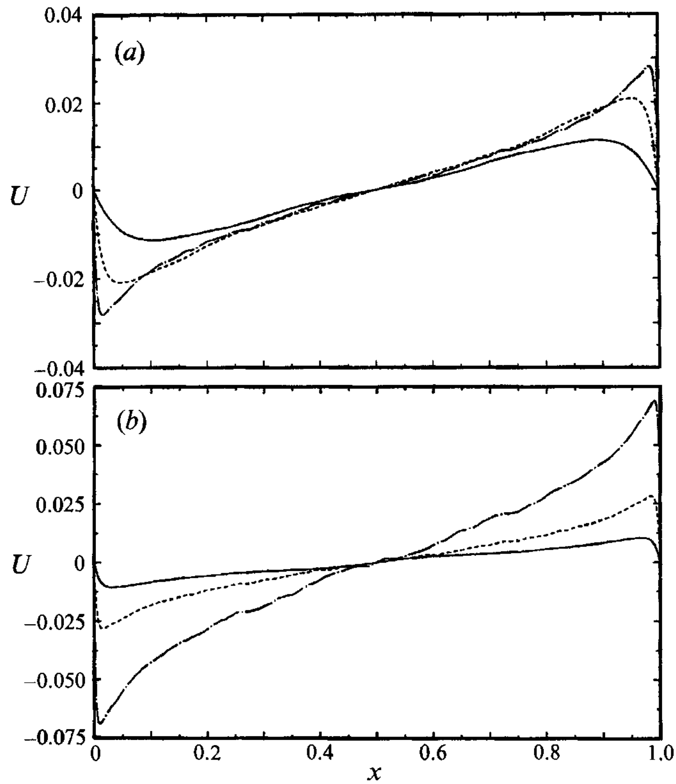

FIGURE 2. Mean-velocity profiles. (*a*) $\Delta t_r = 0.1$: ——, $Re = 500$; ----, $Re = 1500$; —·—, $Re = 4500$; (*b*) $Re = 4500$: ——, $\Delta t_r = 0.01$; ----, $\Delta t_r = 0.1$; —·—, $\Delta t_r = 1$.

In the following I denotes the number of grid points, Δt the computational time step, $\Delta x = 1/(I-1)$, $u_i^n \approx u((i-1)\Delta x, n\Delta t)$, $\chi_i^n \approx \chi((i-1)\Delta x, n\Delta t)$, and $i = 2, ..., I-1$. At each instant of time $n\Delta t$, the χ_i^n are uncorrelated random variables; χ_i^n is constant on a time interval $(p\Delta t_r, (p+1)\Delta t_r)$, where p is an integer, and if $n\Delta t$ and $n'\Delta t$ belong to two different such intervals, all the $\chi_j^{n'}$ are independent of all the χ_i^n $(n' > n)$, with $\langle \chi_i^n \rangle_x = 0$ and $\langle (\chi_i^n)^2 \rangle_x = 1$.

The solution of the Burgers equation with random forcing (equation (4.2)) depends not only on the Reynolds number Re, but also on the mean-square value $\langle \chi^2 \rangle_x$ and the timescale Δt_r of the random forcing. For all calculations presented here, $\langle \chi \rangle_x = 0$, $\langle \chi^2 \rangle_x = 1$, and $\Delta t = 0.001$, which is the largest time step which accurately predicts the small-scale motion with the values of Re and Δt_r used in our calculations. A uniform computational mesh of 2048 point is used in x ($\Delta x = 1/2047$). Three different Reynolds numbers, $Re = 500, 1500, 4500$, and three different timescales, $\Delta t_r = 0.01, 0.1, 1$, are investigated. Instantaneous velocity fields for these Reynolds numbers and two of the timescales are shown in figure 1. Strong thin internal shocks can be seen for high Re and large Δt_r. Figure 2 shows the effect of the Reynolds number Re and the timescale Δt_r on the mean velocity. The magnitudes of the mean velocity and the mean velocity gradient clearly increase with increasing Re and Δt_r. Owing to the convective nature of the solution of the Burgers equation, the wall-layer thickness gets thinner as the Reynolds number increases (figure 2*a*, also see Chambers *et al.* 1988). The mean velocity gradient near the centreline, however, is not much changed with the Reynolds number. On the contrary, increasing Δt_r with a fixed Reynolds number significantly increases the mean velocity gradient near the centreline as well as near the wall layer (figure 2*b*). Wall-layer thickness, however, is not affected by increasing Δt_r.

4.2. *Feedback control procedures*

Two types of feedback controls are investigated: distributed and boundary controls. Distributed control by body forces corresponds to the unrealistic case where volume forces are applied throughout the fluid; however it turns out to be a good introduction to more interesting and more complicated situations. It also contains all of the basic features of general control problems. For boundary control the control is the boundary velocity, which is more practical in fluid mechanics and can be implemented in real situations.

4.2.1. *Distributed control (formulation)*

The non-dimensionalized Burgers equation with distributed control is

$$\left.\begin{aligned}&\frac{\partial u}{\partial t}+\frac{\partial}{\partial x}\frac{u^2}{2}=\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}+\chi(x,t)+f(x,t), \quad 0<x<1,\\&u(x,t=0)=u_0(x),\\&u(x=0,t)=0, \quad u(x=1,t)=0.\end{aligned}\right\} \tag{4.4}$$

Here χ is the random forcing and u_0 the initial data, an instantaneous solution of the Burgers equation with random forcing χ and $f=0$ (equation (4.2)). The control input forcing f is of the form

$$f=\alpha_0+\alpha_1\,\theta_1(u). \tag{4.5}$$

Here $\theta_1(u)$ is prescribed from physical intuition or experience (see §§3.1.1 and 4.3), and α_0 and α_1 are determined through our control algorithm and thus have an implicit dependence on u. Note that α_0 and α_1 are not constant in time and space and they are continuously updated with the change of u.

At each instant of time, the cost function considered is

$$J^n=J(e^n)=\tfrac{1}{2}l_d\|e^n\|^2+\tfrac{1}{2}m_d\int_0^1\left(\frac{\partial u^n}{\partial x}\right)^2 \mathrm{d}x, \tag{4.6}$$

where $\|e\|^2=\|\alpha_0\|^2+\|\alpha_1\|^2$. Here we want to reduce the mean-square velocity gradient inside the domain at the expense of the control input. The choice of l_d and m_d (or more precisely of the ratio l_d/m_d) is an engineering problem which is not addressed here, although we do consider in our computations several values of this ratio. The detailed procedure of distributed control by body forces is described in Appendix A.

4.2.2. *Boundary control (formulation)*

The non-dimensionalized Burgers equation with boundary control is

$$\left.\begin{aligned}&\frac{\partial u}{\partial t}+\frac{\partial}{\partial x}\frac{u^2}{2}=\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}+\chi(x,t), \quad 0<x<1,\\&u(x,t=0)=u_0(x),\\&u(x=0,t)=\psi_0(t), \quad u(x=1,t)=\psi_1(t).\end{aligned}\right\} \tag{4.7}$$

Here χ is the random forcing and u_0 is the initial data, which is an instantaneous solution of the Burgers equation with random forcing χ and $\psi_0 = \psi_1 = 0$ (equation (4.2)). The control input velocities at the boundary, ψ_0 and ψ_1, are of the form,

$$\psi_0 = \alpha_{0,0} + \alpha_{1,0}\theta_{1,0}(u), \quad \psi_1 = \alpha_{0,1} + \alpha_{1,1}\theta_{1,1}(u). \tag{4.8}$$

We denote $\alpha_0 = \{\alpha_{0,0}, \alpha_{0,1}\}$, $\alpha_1 = \{\alpha_{1,0}, \alpha_{1,1}\}$, and $\theta_1 = \{\theta_{1,0}, \theta_{1,1}\}$. Here the $\theta_1(u)$ are prescribed from physical intuition or experience (see §§3.1.1 and 4.3), and the α_0 and α_1 are determined through a control algorithm and thus have an implicit dependence on u. Note that the α_0 and α_1 are not constant in time and they are continuously updated with the change of u.

At each instant of time, the instantaneous cost function considered is

$$J^n = J(e^n) = \tfrac{1}{2}l_b\|e^n\|^2 + \tfrac{1}{2}m_b\left[\left(\frac{\partial u^n}{\partial x}\right)^2_{x=0} + \left(\frac{\partial u^n}{\partial x}\right)^2_{x=1}\right], \tag{4.9}$$

where $\|e\|^2 = \|\alpha_0\|^2 + \|\alpha_1\|^2$. Here we want to reduce the wall velocity gradient at the expense of the control input. The detailed procedure of boundary control by boundary velocities is described in Appendix B.

4.2.3. *Numerical algorithm*

From the previous subsections, we can write for both control problems the numerical algorithm of minimizing the cost function J^n:

Step 1: Start with an initial condition u_0 or a solution of the previous time step $u^{n,k-1} = u^{n-1}$. Choose initial $\alpha_0^{n,k-1}$ and $\alpha_1^{n,k-1}$.
Step 2: Solve the adjoint equation with $u^{n,k-1}$, and $f^{n,k-1}$ or $\psi^{n,k-1}$ to obtain $\zeta^{n,k-1}$.
Step 3: Update $\alpha_0^{n,k}$ and $\alpha_1^{n,k}$.
Step 4: Solve the discretized Burgers equation with $f^{n,k}$ or $\psi^{n,k}$ to obtain $u^{n,k}$.
Step 5: Iterate Steps 2–4 until $u^{n,k}$ converges.
Step 6: When converged, $u^n = u^{n,k}$.

4.3. *Results of numerical simulation*

Values of l and m in the cost function ((4.6) and (4.9)) are dimensional quantities and are usually prescribed from a physical setting of the actual systems (see §2.1 and above). Also the problem can be scaled by l/m. Once l and m are given, we need to find the best ρ (the parameter of descent). We do it by trial and error instead of by theoretical procedures (see e.g. Luenberger 1973). Note that conjugate gradient methods can be used to eliminate the unnecessary trial-and-error procedure and also to get faster convergence.

As mentioned in detail in §3.1.1, for many cases feedback laws are in general hard to obtain from a mathematical approach, and one must resort to physical intuition or experience for a better result or for a simple feedback law. The reasons for using the formulations (4.5) and (4.8) rather than using a simple formulation ($f = f(u)$ or $\psi = \psi(u)$) are two-fold. First, the formulations (4.5) and (4.8) can accommodate experience and physical observations. Second, they can accommodate a good feedback law if we prescribe a proper $\theta_1(u)$.

We consider two cases for functions θ_1 in (4.5) and (4.8): $\theta_1 = 0$ and $\theta_1 \neq 0$. (Here, for simplicity we let the functions, $\theta_{1,0}$ and $\theta_{1,1}$ in (4.8), be equal to a same function now denoted θ_1; hence both ends, $x = 0$ and $x = 1$, play the same role.) We call the former *non-prescribed feedback control* because in that case the control input itself (f or ψ) does not have any information from physical experience and is determined directly from a mathematical approach. Note that non-prescribed feedback control still yields a relation between u and ϕ (f or ψ) because the α_0 are continuously updated

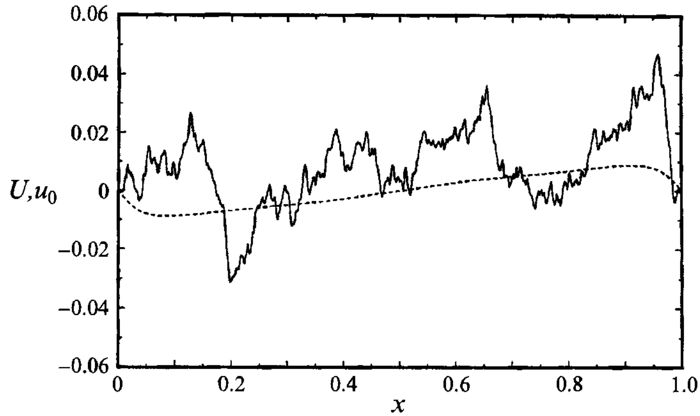

FIGURE 3. Velocity profiles: ----, time-averaged mean velocity U with no control and $Re = 1500$, $\Delta t_r = 0.01$; ——, initial velocity u_0 for distributed and boundary controls.

from our optimization procedures which need information on u (see §4.4 on *actual feedback laws*). Since there are two different parameters ρ_0, ρ_1 in the case $\theta_1 \neq 0$, many trial-and-error iterations are needed to find the best ρ_0 and ρ_1. Non-prescribed feedback control ($\theta_1 = 0$) is first investigated to find the best ρ_0; the best is ρ_1 is then searched while holding $\alpha_0 = 0$. For most cases considered in the following, non-prescribed feedback control is better at finding a local minimum of the cost function in the sense that non-prescribed feedback control gives faster and more stable convergence. However, we believe that this is not always the case since it depends on the function type θ_1 and the cost J as well as on the gradient algorithm. In some other situations or with a different choice of the shape function (the class of feedback laws), the results may turn out to be better. Most of the results presented below were obtained for the non-prescribed feedback control case; some of the cases with $\theta_1 \neq 0$ are also presented.

In this section, we only discuss the results of the cases with $Re = 1500$ and $\Delta t_r = 0.01$. Cases with different Reynolds number ($Re = 4500$) and different $\Delta t_r (= 0.1)$ have also been tested, and the results show the same trend. Figure 3 shows the time-averaged mean velocity U of the no-control case with $Re = 1500$ and $\Delta t_r = 0.01$ and the initial velocity u_0 used in the following controls.

4.3.1. *Numerical results for distributed control*

Two types of shape function θ_1 (equation (4.5)) were investigated: $\theta_1 = 0$ (non-prescribed feedback control), and $\theta_1 = u$. Concerning the parameters l_d and m_d, we have chosen quite arbitrarily to study the following cases: (i) $l_d = 1$, $m_d = 1$; (ii) $l_d = 1$, $m_d = 2047 (= 1/\Delta x)$; (iii) $l_d = 1$, $m_d = 4.2 \times 10^6 (= 1/\Delta x^2)$; (iv) $l_d = 0$, $m_d = 1$. Results with control were compared to those with no control. A set of random values of the momentum forcing χ was stored and used for both the control and no-control cases for the accurate estimation of the parameter ρ. Case (i) showed almost no change of the cost when control was applied. When the ratio of the weight parameters, m_d/l_d, is small, the input cost becomes so expensive that there is no means of reducing the total cost by addition of controls.

With non-prescribed feedback control ($\theta_1 = 0$), optimal values ρ_0 of 1, 0.001 and 10000 were found for cases (ii), (iii), and (iv), respectively. Effects of the parameter of descent ρ_0 on the cost function and its convergence are shown in figure 4. Larger cost

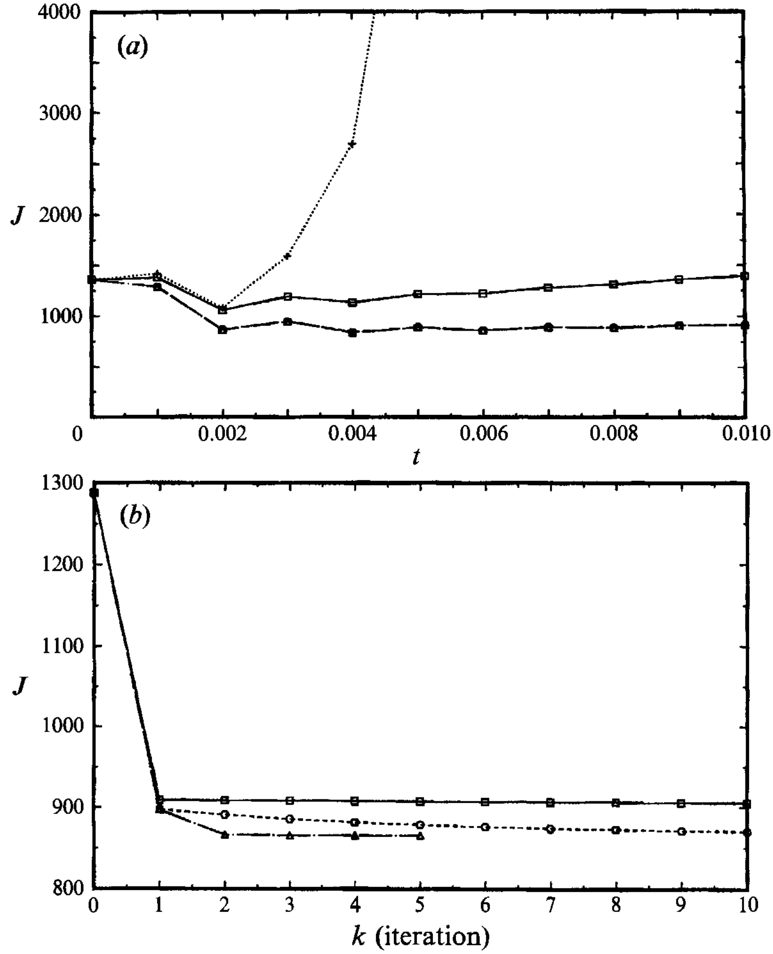

FIGURE 4. Variation of the cost and its convergence with respect to the parameter of descent ρ_0 for case (ii). (*a*) □, No control; ○, $\rho_0 = 0.1$; △, $\rho_0 = 1$; +, $\rho_0 = 1.69$; (*b*) at $t = 0.002$, □, $\rho_0 = 0.01$; ○, $\rho_0 = 0.1$; △, $\rho_0 = 1$.

J or divergence of the solution was obtained for ρ_0 larger than the optimal ρ_0, and slower convergence was achieved for smaller ρ_0. Case (ii) needed about six iterations to converge (figure 4*b*). Cases (iii) and (iv) need many more iterations (about 50) to converge. From the practical point of view, however, it is not necessary to get a converged solution since a few iterations give a significant reduction of the cost function.

Figure 5 shows the time history of the cost, energy ($= \int_0^1 \frac{1}{2}u^2\,\mathrm{d}x$), wall velocity gradient, and momentum forcings (random input χ and control input f) at $x = 0.5$ for case (ii). Figure 6 shows the same for case (iii). Results of case (iv) are essentially the same as those of case (iii). It can be seen that the distributed control significantly reduces the cost as well as the energy inside the domain (see figures 6*b* and 7*d*).

Since one of the effects of control is to cancel or attenuate the effects of random forcing, it is interesting to investigate a case with no random forcing and no control ($\chi = f = 0$) and the same initial velocity u_0, and compare the results to those with control. Temporal evolution of cost and energy, and velocity profiles at $t = 2$ with no random forcing and no control ($\chi = f = 0$) are compared with control cases (ii) and (iii)

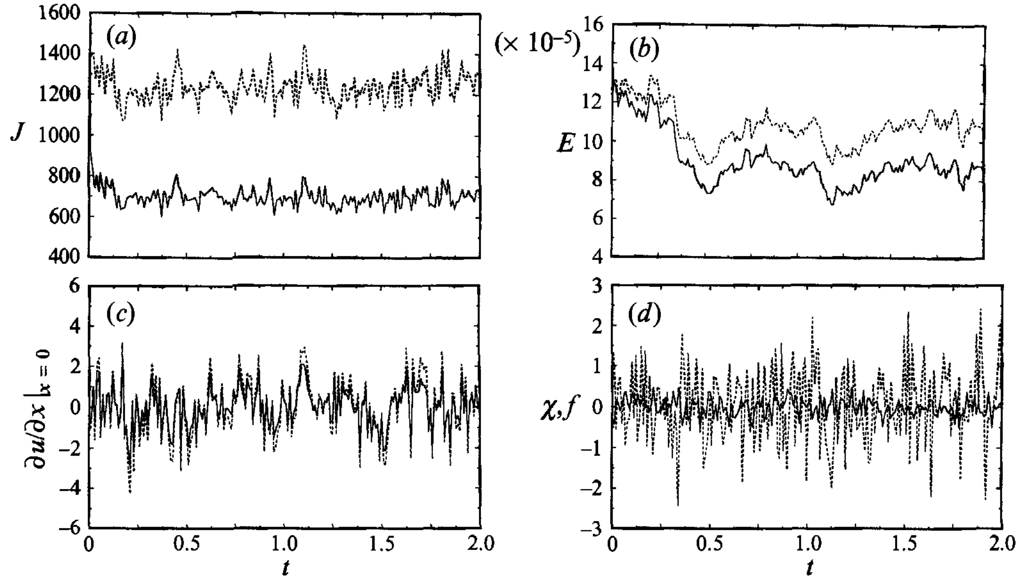

FIGURE 5. Time history of flow parameters for case (ii): ——, with control; ----, without control. (*a*) Cost; (*b*) energy inside the domain; (*c*) wall velocity gradient $\partial u/\partial x(x=0)$; (*d*) momentum forcings at $x = 0.5$: ——, control forcing f; ----, random forcing χ.

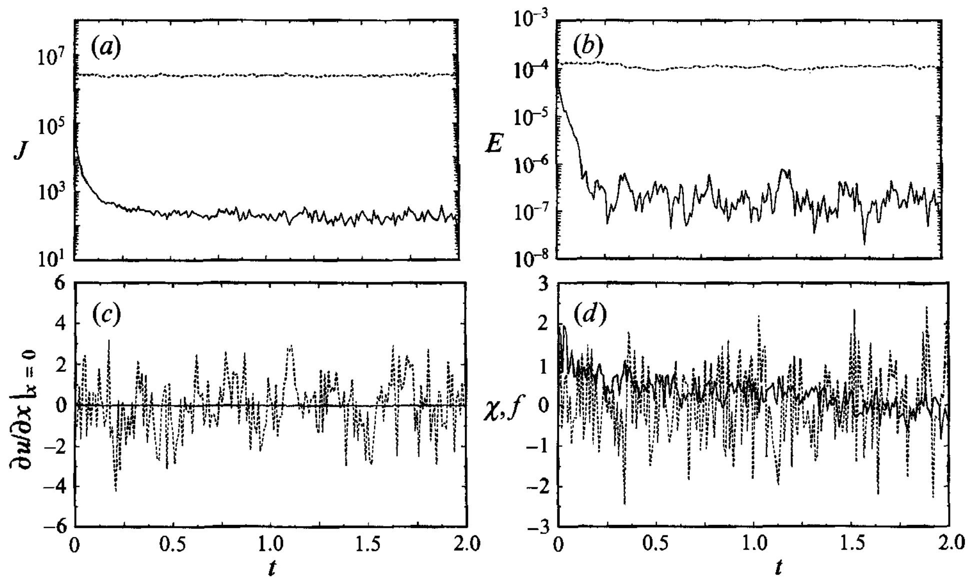

FIGURE 6. Time history of flow parameters for case (iii): ——, with control; ----, without control. (*a*) Cost; (*b*) energy inside the domain; (*c*) wall velocity gradient $\partial u/\partial x(x=0)$; (*d*) momentum forcings at $x = 0.5$: ——, control forcing f; ----, random forcing χ.

in figure 7. Energy and costs rapidly decrease and the velocity profile is smoothed when there is no random forcing. The control simply attenuates the effect of the random forcing for case (ii). Case (iii), however, shows that the control can do more than cancel the effects of random forcing.

The case $\theta_1 = u$ was tested next. By setting $\alpha_0 = 0$ in case (ii), the best ρ_1 was searched for. $\rho_1 = 2$ gave a larger cost and $\rho_1 \leqslant 1$ gave no change of cost as compared to the no-

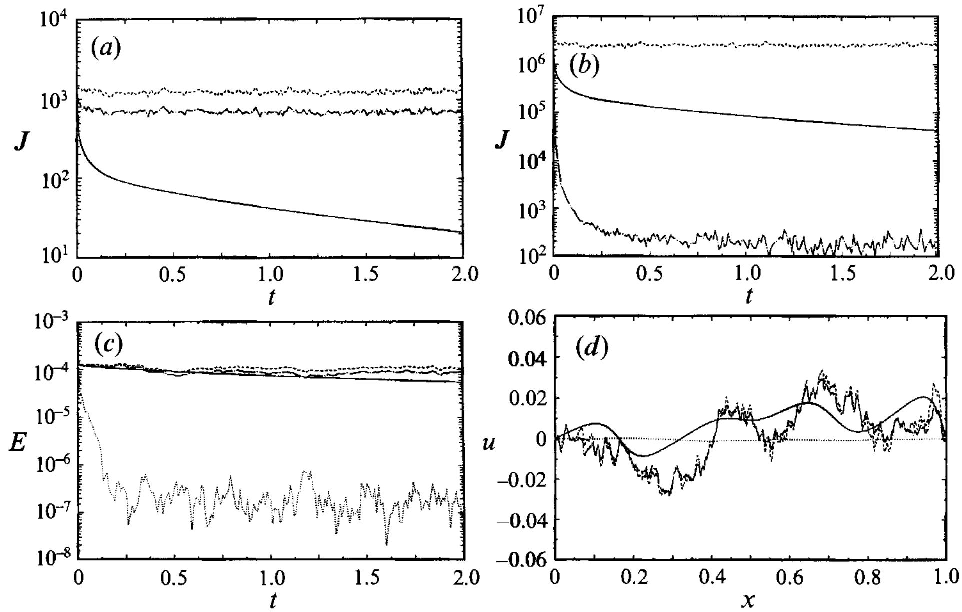

FIGURE 7. Temporal evolution of flow parameters and velocity profiles with and without random forcing and control. (*a*) Cost with $l_a = 1$ and $m_a = 1/\Delta x$: ——, $\chi = f = 0$; ----, $f = 0$ and $\chi \neq 0$; —·—, control (case ii); (*b*) cost with $l_a = 1$ and $m_a = 1/\Delta x^2$: ——, $\chi = f = 0$; ——, $f = 0$ and $\chi \neq 0$; —·—, control (case iii); (*c*) energy inside the domain: ——, $\chi = f = 0$; ——, $f = 0$ and $\chi \neq 0$; —·—, control (case ii); ······, control (case iii); (*d*) instantaneous velocity profiles at $t = 2$: ——, $\chi = f = 0$; ----, $f = 0$ and $\chi \neq 0$; —·—, control (case ii); ······, control (case iii).

control case. The sensitivity of the cost function with respect to the control variables is a good indicator of the effectiveness of the control variables in reducing the cost. As equation (A 7) suggests, the sensitivity is a function of not only the time step (n) but also the iteration (k). Convergence of the sensitivity with fixed n is strongly dependent on the parameter ρ, but the temporal evolution of the sensitivity at the first iteration is nearly independent of ρ. The magnitudes of the sensitivities of the cost functions for cases (ii) and (iii) were measured with the parameters $\rho_0 = \rho_1 = 1$ for case (ii), and $\rho_0 = \rho_1 = 0.001$ for case (iii). The sensitivity with respect to α_1 was at least two orders of magnitude smaller than that with respect to α_0 (figure 8).

4.3.2. *Numerical results for boundary control*

For simplicity, the functions $\theta_{1,0}$ and $\theta_{1,1}$ in (4.8) are taken to be the same so that both ends, $x = 0$ and 1, play the same role ($\theta_{1,0} = \theta_{1,1} = \theta_1$).

Three types of such shape function θ_1 were investigated†: $\theta_1 = 0$ (non-prescribed feedback control), $\theta_1 = \partial u/\partial x$, and $\theta_1 = (\partial u/\partial x)/[1 + (\partial u/\partial x)^2]$. Concerning the parameters l_b and m_b, we have chosen quite arbitrarily to study the following cases: (i′) $l_b = 1$, $m_b = 2.5 \times 10^{-7} (= \Delta x^2)$; (ii′) $l_b = 1$, $m_b = 5 \times 10^{-4} (= \Delta x)$; (iii′) $l_b = 1$, $m_b = 1$; (iv′) $l_b = 0$, $m_b = 1$.

Results with control were compared to those with no control. A set of random values of the momentum forcing χ was stored and used for both the control and no-control

† Again these choices were based on physical intuition but, as explained hereafter, the results are not satisfactory. The last function was chosen to limit the magnitude of θ_1 to less than one; in this way one can expect equation (4.8) to be a more accurate truncation of the Taylor series expansion of $\psi(\theta_1)$.

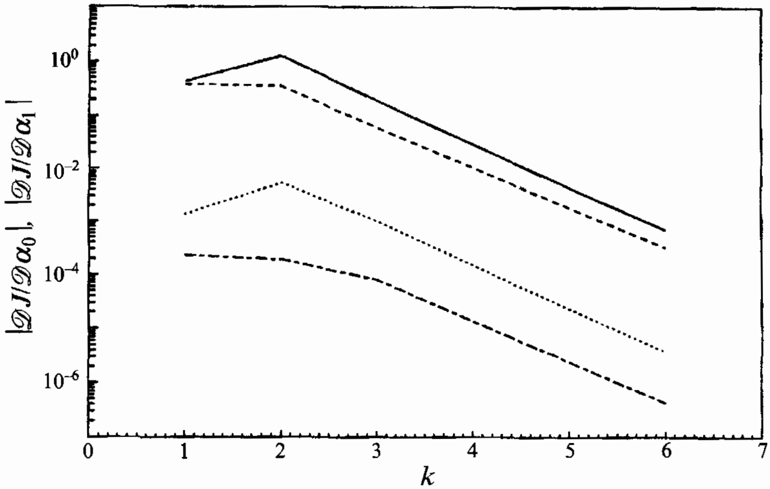

FIGURE 8. Convergence of the sensitivity of the cost function at $t = 0.001$ for case (ii) and $\theta_1(u) = u$: ——, $|\mathscr{D}J/\mathscr{D}\alpha_0|_{x=0.5}$; ----, $|\mathscr{D}J/\mathscr{D}\alpha_0|_{x=0.01}$; ······, $|\mathscr{D}J/\mathscr{D}\alpha_1|_{x=0.5}$; —·—, $|\mathscr{D}J/\mathscr{D}\alpha_1|_{x=0.01}$.

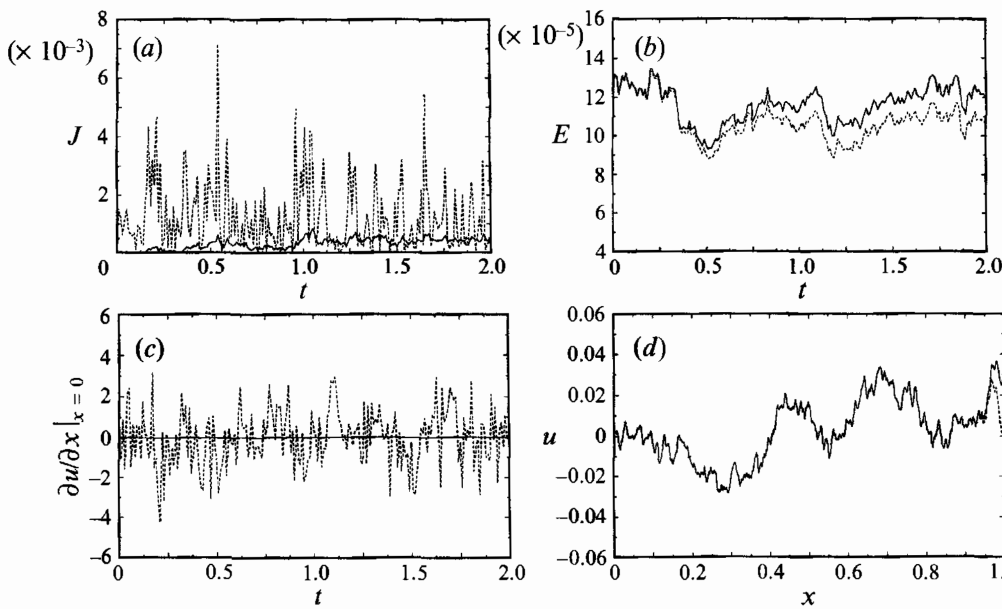

FIGURE 9. Time history of flow parameters, and velocity profile for case (ii′): ——, with control; ----, without control. (*a*) Cost; (*b*) energy inside the domain; (*c*) wall velocity gradient $\partial u/\partial x(x = 0)$; (*d*) velocity profile at $t = 2$.

cases for the accurate estimation of the parameter ρ. All cases except (i′) showed a significant reduction of the cost when control was applied. Case (i′) showed no reduction of the cost since m_b was very small relative to l_b. With non-prescribed feedback control, optimal values ρ_0 of 0.001, 10^{-6} and 10^{-6} were found for cases (ii′), (iii′), and (iv′), respectively. Effects of ρ_0 on the cost function and the sensitivity are the same as those described in the distributed control subsection. Cases (ii′), (iii′) and (iv′) needed about seven iterations to converge.

Figure 9 shows the time history of the cost, energy, wall velocity gradient, and velocity profile at $t = 2$ for case (ii′). Figure 10 shows the costs of cases (iii′) and (iv′).

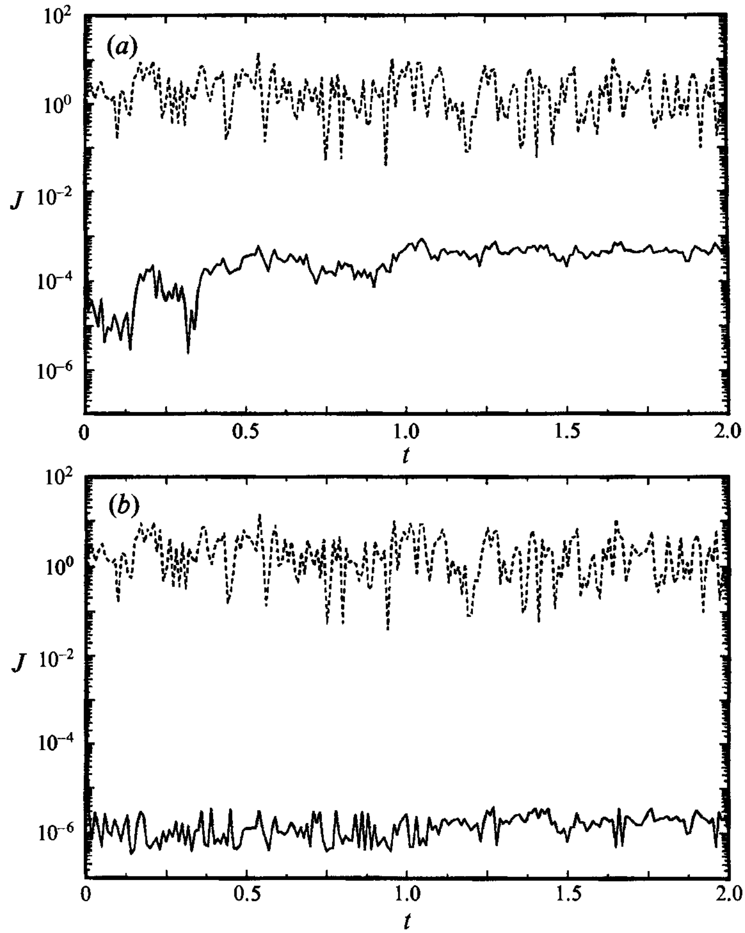

FIGURE 10. Time history of the cost: ——, with control; ----, without control. (*a*) Case (iii′); (*b*) case (iv′).

Temporal evolutions of the energy, wall velocity gradient, and velocity profiles of cases (iii′) and (iv′) are quite similar to case (ii′). In contrast to distributed control, the energy inside the domain is changed little though the cost and $\partial u/\partial x$ at the wall are reduced significantly. The velocity profile at $t = 2$ shows that boundary control modifies the flow field only near the wall. A longer time integration up to $t = 100$ has been completed for case (iii′). Again only the wall region is modified.

One may think that a fixed non-zero wall velocity (passive control) may reduce the cost or $\partial u/\partial x$ at the wall and may even give more of a reduction than does boundary control. Several fixed non-zero wall velocities, $u(x=0) = -u_s$, $u(x=1) = u_s$, were tested in order to study the effect of a slip velocity at the wall, where u_s was taken from 0 to 0.1 (for the relative scale of the velocity, see figure 3). All cases considered showed a negligible reduction or a significant increase of the cost. We also used the time-averaged value of the control wall velocities obtained from the feedback control algorithm as a fixed wall velocity. The cost reduction was again negligible.

As described at the beginning of this subsection, we tested two more shape functions: $\theta_1 = \partial u/\partial x$ and $\theta_1 = (\partial u/\partial x)/(1+(\partial u/\partial x)^2)$. By setting $\alpha_0 = 0$ in case (ii′), the best ρ_1 was searched for. For the function $\theta_1 = \partial u/\partial x$, the simple gradient algorithm was unstable and did not converge. For the function $\theta_1 = (\partial u/\partial x)/(1+(\partial u/\partial x)^2)$, $\rho_1 = 0.001$ gave a reduction of the cost but converged more slowly than the non-prescribed feedback control case. Larger values of ρ_1 give either a larger cost or divergence of the solution. The sensitivity of the cost function with respect to the control variables was measured with $\rho_0 = \rho_1 = 0.001$. The sensitivity with respect to α_1 was about an order of magnitude smaller than that with respect to α_0.

4.4. *Remarks on the feedback laws*

Physically a feedback law amounts to a formula relating the quantity being measured by a sensor to the quantity being controlled by an actuator. Here some comments are made on the choice of the feedback laws. As indicated before, our control is optimal among a prescribed class of feedback laws: see e.g. (2.5), (2.7) and (3.9) in the stationary case and similarly (4.5) and (4.8) in the time-dependent case. Our choices of the feedback laws, i.e. the functionals θ_i or $\theta_{i,j}$ in (2.7), (4.5) and (4.8), were fully arbitrary because of the lack of similar results in the literature. Surprisingly the most efficient feedback laws were 'the non-prescribed ones', i.e. those where the control ϕ does not explicitly depend on the state u. Hence at first sight there is no feedback, but this is not actually true since the feedback is hidden in the history of the flow. Concerning the feedback laws we would like to emphasize here the phase diagrams of the solutions using non-prescribed feedback control for cases (ii′) and (iii′) in figure 11. Here one of the controls (wall velocity) is plotted against one of the observations directly related to the cost function, namely the velocity gradient $\partial u/\partial x$ at the wall. These phase diagrams can be considered as the actual feedback laws resulting from our method. A linear feedback law is achieved for case (ii′), whereas no apparent relation is found for case (iii′). For cases (iii′) and (iv′), the wall velocity gradients are nearly zero irrespective of the magnitude of control wall velocities (figure 11*b*). Therefore, a linear feedback law between wall velocity and wall velocity gradient is not found for cases (iii′) and (iv′). A feedback law for case (ii′) is deduced from figure 11:

$$u^n(x=0) = 0.533\frac{\partial u}{\partial x}\bigg|^{n-1}(x=0), \quad u^n(x=1) = -0.533\frac{\partial u}{\partial x}\bigg|^{n-1}(x=1), \qquad (4.10)$$

where $t = n\,\Delta t$. A natural and puzzling question resulting from the feedback laws in (4.10) was whether we could by pass our procedure and directly implement a boundary condition of the type (4.10)

$$u(x=0) = \lambda\frac{\partial u}{\partial x}(x=0), \quad u(x=1) = \lambda'\frac{\partial u}{\partial x}(x=1), \qquad (4.11)$$

with λ and λ' equal or close to ± 0.533 as in (4.10). We found that numerical instabilities were developing: once the wall velocity obtained from (4.11) deviates slightly from the value used in actual control procedures, the wall velocity gradient at the next time step is significantly increased compared to the feedback control case. Hence the actual boundary condition resulting from control algorithm is indispensable for stability.

4.5. *Further remarks on implementation issues*

We make here two further remarks on implementation issues: one is related to the effect of the time-discretization method and the other is related to some aspects of the gradient algorithm.

4.5.1. *Remarks on the time-discretization method*

It is clear that, in the time-dependent case, the time discretization of the evolution

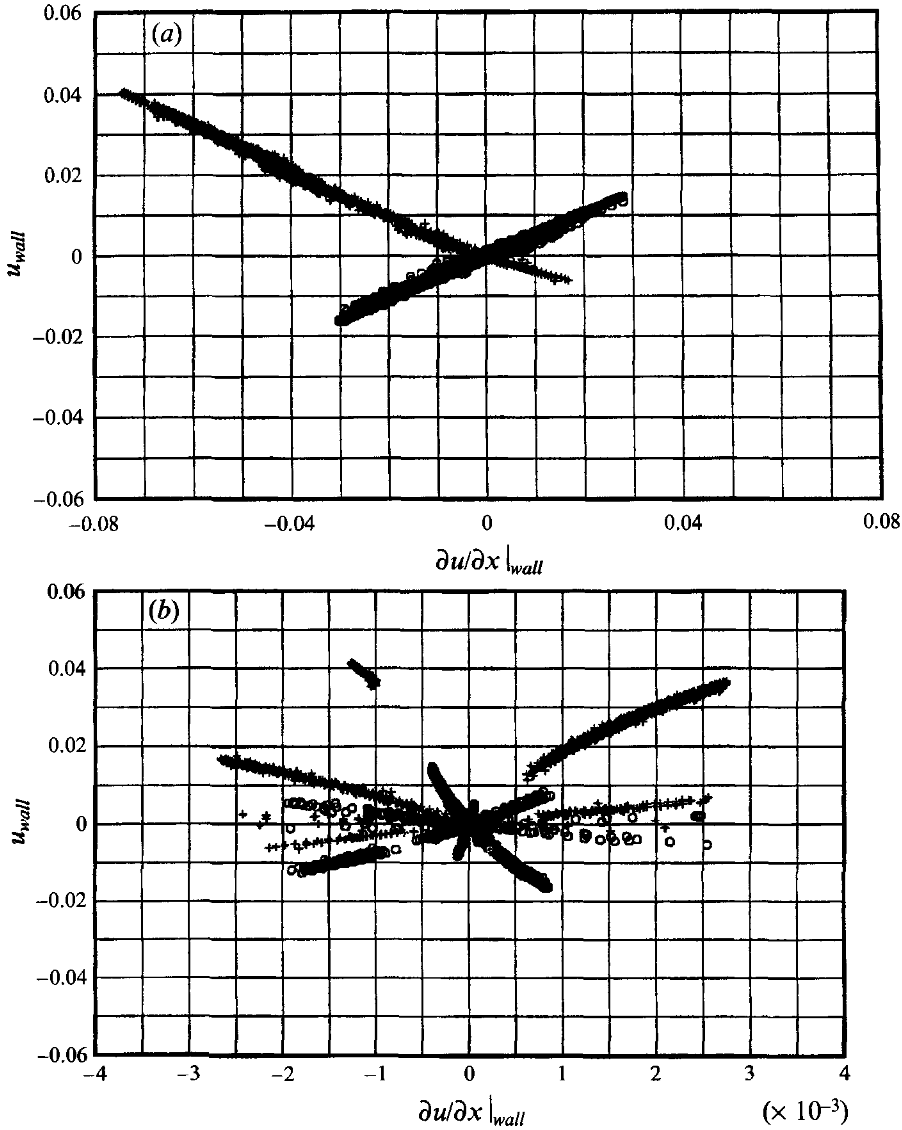

FIGURE 11. Phase diagram of control velocity and velocity gradient at the wall for the time interval $0 < t < 2$: ○, at $x = 0$; +, at $x = 1$. (*a*) Case (ii′); (*b*) case (iii′). The phase diagram of case (iv′) is nearly same as that of case (iii′). Note that the wall velocity gradients of case (iii′) are much smaller than those of case (ii′).

equation strongly affects (3.18) and therefore our whole procedure. In the most extreme case where a fully explicit scheme is used, our method cannot be implemented at all. Hence we have found it important to test our method with various classical forms of time discretization. We have already presented the control procedure resulting from a fully implicit discretization of the Burgers equation using a Crank–Nicolson scheme (see (3.17)). We have also tested our procedure in the case of boundary control using another fully implicit method (implicit Euler) and a semi-implicit method (Adams–Bashforth for the nonlinear term and Crank–Nicolson for the viscous term).

In the case of the boundary control using a fully implicit method, the adjoint equation (B 4) contains all the interior velocities as well as the boundary velocities (or

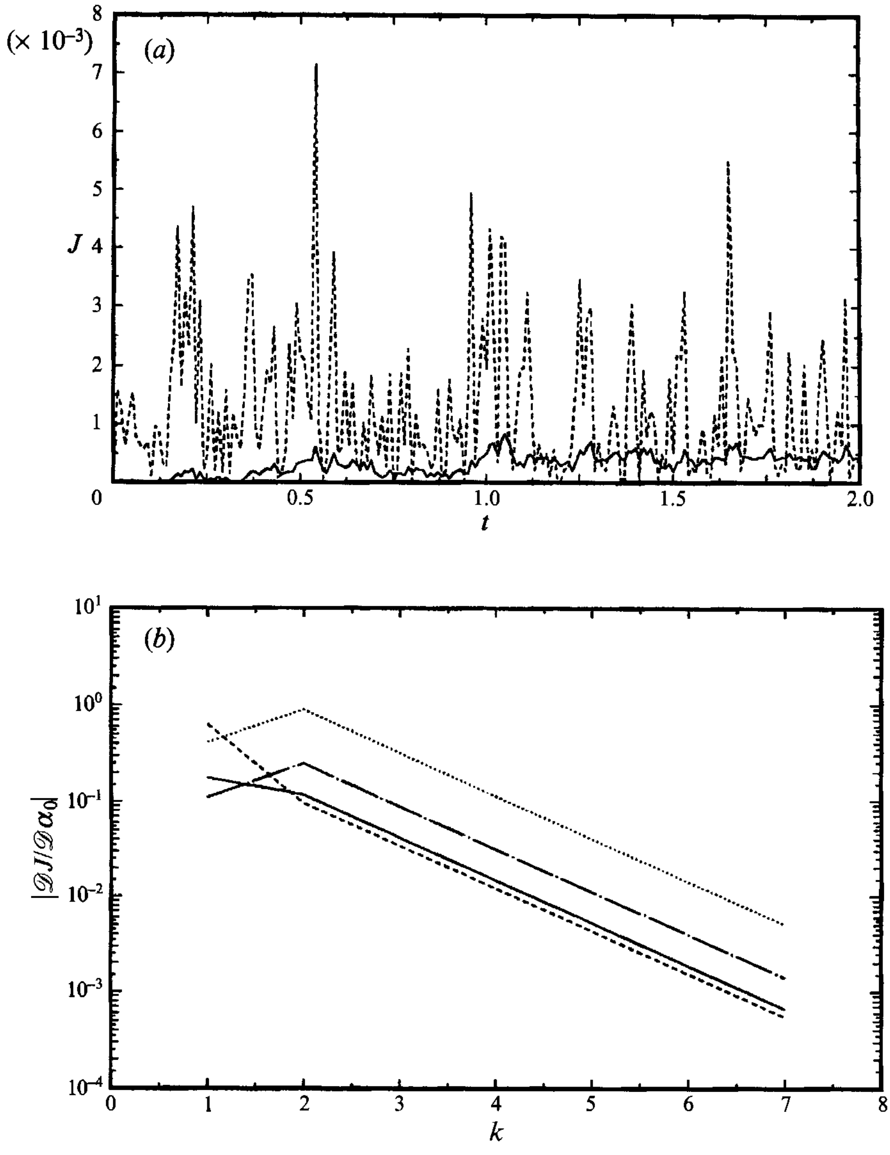

FIGURE 12. Variation of the cost and sensitivity with respect to time-discretization methods for case (ii′). (*a*) Time history of the cost: ——, with control; ----, without control using a semi-implicit method. The result of using a fully implicit method is the same as that of using a semi-implicit method. (*b*) Convergence of the sensitivity at $t = 0.001$: with a semi-implicit method, ——, $|\mathscr{D}J/\mathscr{D}\alpha_{0,0}|$; ----, $|\mathscr{D}J/\mathscr{D}\alpha_{0,1}|$; with a fully implicit method, —·—, $|\mathscr{D}J/\mathscr{D}\alpha_{0,0}|$; ······, $|\mathscr{D}J/\mathscr{D}\alpha_{0,1}|$.

boundary velocity gradients). Hence, a full knowledge of the flow field would be required for the implementation of the control algorithm, which is highly impractical for physical implementation. On the other hand, in the case of the boundary control using a semi-implicit method (Adams–Bashforth for the nonlinear term and Crank–Nicolson for the viscous term), one can circumvent this problem. The feedback procedure for the Burgers equation using this semi-implicit method is presented in Appendix C. The resulting adjoint equation (C 2) does not contain any velocities except the wall velocity gradients. Figure 12 shows the time history of the cost and

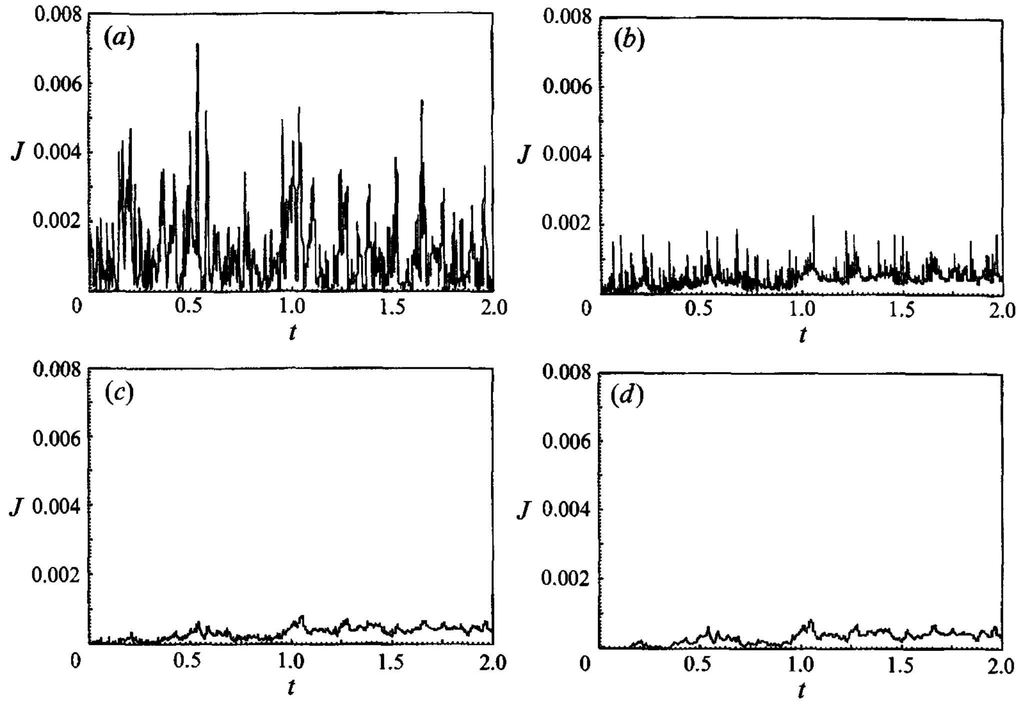

FIGURE 13. Time history of the cost for various iterations k for case (ii′): (*a*) without control; (*b–d*) with control: (*b*) $k = 1$; (*c*) $k = 2$; (*d*) $k = 7$. $\Delta t_r = 0.01$, $\Delta t = 0.001$, and $\rho = 0.001$ are used.

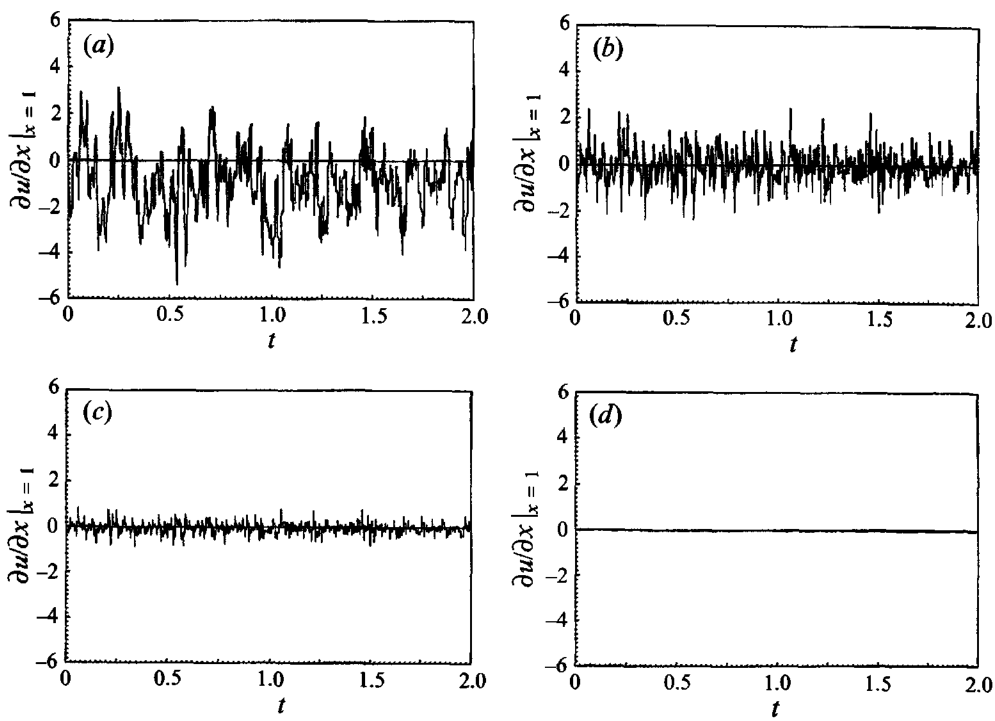

FIGURE 14. Time history of the wall velocity gradient for various iterations k for case (ii′): (*a*) without control; (*b–d*) with control: (*b*) $k = 1$; (*c*) $k = 2$; (*d*) $k = 7$. $\Delta t_r = 0.01$, $\Delta t = 0.001$, and $\rho = 0.001$ are used.

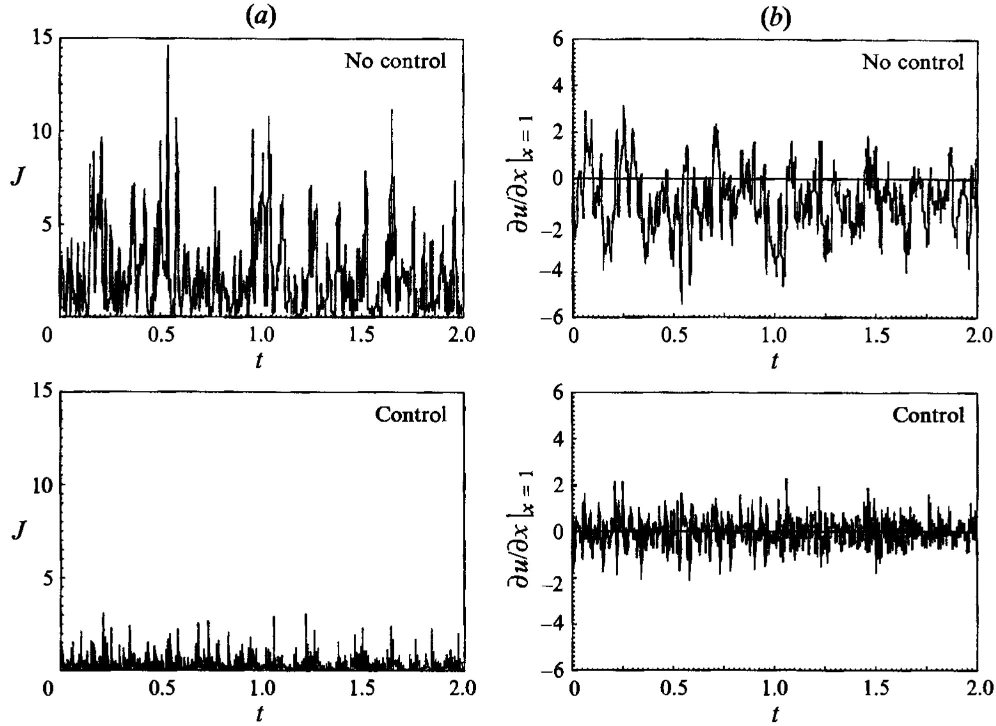

FIGURE 15. Time history of the cost and wall velocity gradient with $k = 1$ for case (iv′): (*a*) cost; (*b*) wall velocity gradient. $\Delta t_r = 0.01$, $\Delta t = 0.001$, and $\rho = 8\times10^{-7}$ are used. Mean values of the cost in cases with and without controls are 0.291 and 2.247, respectively.

Case	No control	$k=1$	$k=2$	$k=7$
$J\times10^3$	1.098	0.4409	0.3183	0.3053
Reduction of J	—	60 %	71 %	72 %

TABLE 1. Variation of the cost with respect to the iteration k for case (ii′).

convergence of the sensitivity for case (ii′) ($l_b = 1, m_b = \Delta x$). Essentially the same result is obtained as with a fully implicit method; the convergence behaviour is only slightly changed.

4.5.2. *Practical implementation of the control algorithm*

From the practical point of view, sensors and actuators must be placed at the wall. In this respect, the boundary control is more realistic than the distributed control. As mentioned above, when the semi-implicit method is used, the resulting adjoint equation (C 2) does not contain any velocities except the wall velocity gradients, while all interior velocities should be measured when a fully implicit method is used. Hence, for the practical implementation of the feedback control algorithm, one has to resort to the semi-implicit method.

Also, it should be pointed out that all previous computations were carried out until the cost reached a minimum at each time step n, which required about seven iterations at each time step. However, in practical situations, the number of iterations k should be limited to one. This is because in a physical setting $\partial u/\partial x$ at the boundary at a given

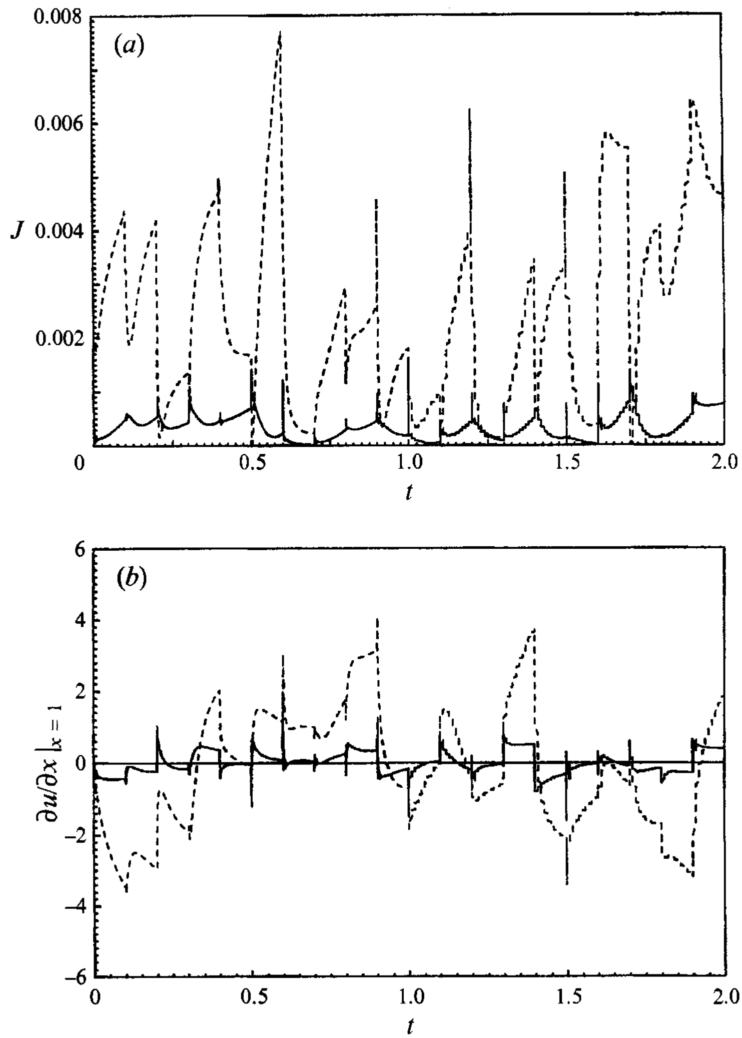

FIGURE 16. Time history of the cost and wall velocity gradient with $k = 1$ for case (ii′): ——, with control; ----, without control. (a) Cost; (b) wall velocity gradient. $\Delta t_r = 0.1$, $\Delta t = 0.001$, and $\rho = 0.001$ are used. Mean values of the cost in cases with and without controls are 0.000325 and 0.00246, respectively.

instant is used to obtain the control input velocity which leads to a new velocity field at the next instant in time with the corresponding $\partial u/\partial x$ at the boundary. Physically one cannot use the new data on $\partial u/\partial x$ at the boundary to go back in time and refine the input velocity.

In this section, we investigate the effectiveness of the control algorithm with the iteration k set equal to one, the Reynolds number, $Re = 1500$, and the computational time step, $\Delta t = 0.001$. Figures 13 and 14 show the time history of the cost and wall velocity gradient for various iterations k for case (ii′) ($l_b = 1$, $m_b = \Delta x$). Here, the timescale of the random forcing, Δt_r, is set to be 0.01. Reductions of the cost as well as the wall velocity gradient are accomplished with a few iterations. Table 1 shows the mean value of the cost and percentage reduction of the cost. The cost is significantly reduced with $k = 1$, and the reduced cost with $k = 2$ is nearly the same as that with $k = 7$, the maximum reduction of the cost. Figure 15 shows the time history of the cost and wall velocity gradient with $k = 1$ for case (iv′) ($l_b = 0$, $m_b = 1$). Again, a cost reduction of 87% is found with only one iteration. Note that in case (iv′) the control input cost is not included into the total cost, i.e. a cheap control.

In order to investigate the effect of the random-forcing timescale on the efficiency of the control scheme, a larger timescale, $\Delta t_r = 0.1$ was used. Time history of the cost and wall velocity gradient with $k = 1$ for case (ii′) ($l_b = 1$, $m_b = \Delta x$) is shown in figure 16. The effect of random forcing on the wall velocity gradient is clearly seen. A cost reduction of 87% is obtained by the control, which is clearly larger than in the case of $\Delta t_r = 0.01$, where a cost reduction of 60% was obtained. The control scheme with $k = 1$ adjusts to sudden changes in the flow in a short time to create a cost reduction. Here, the random-forcing timescale is a flow timescale, and the computational time step is considered the control timescale. Clearly, it should be much easier to get a cost reduction if the flow timescale is much larger than the control timescale. The numerical experiments with two different Δt_r show that this is indeed the case.

5. Conclusions and discussion

Some avenues for the application of the mathematical methods of control theory to the problem of control of fluid flow have been presented. The problem of controlling turbulence was considered and posed as a problem in optimal control theory using the methods, formalism and language of control theory. We have presented a new suboptimal control and feedback procedure, which applies to fairly general cost functions and fairly general time-dependent equations including in particular stochastic equations. This procedure was not strictly justified but did produce good numerical results and is fairly simple.

Feedback control procedures were applied to the stochastic Burgers equation. Two types of controls were investigated: distributed and boundary controls. Even though distributed control by body forces is rather unrealistic, it turns out to be a good introduction to more complicated situations. For boundary control, the control is the boundary velocity, which is more practical and can be implemented in real situations.

Several case studies of both types of controls have been completed to investigate the performance of the control algorithm: the reduction of cost, convergence of the gradient algorithm, dependence of the sensitivity of the cost function with respect to the control variables, effect of the parameter of descent, and choice of the form of the feedback law. Most cases considered showed a significant reduction of cost.

The role of the preassigned form of the feedback law was discussed in §§3.1.1 and 4.4. The feedback procedures considered in the present study depend on the time-discretization method used. One would hope that the control results are insensitive to such numerical considerations. Numerical experiments show that this may indeed be the case.

The semi-implicit method seems most promising for future and more involved applications. Indeed, for boundary control using a fully implicit method the adjoint equation, (B 4), contains all interior velocities as well as the boundary velocities (or boundary velocity gradients) so that the feedback control algorithm may not be practical: that is, a full knowledge of the flow field would be required for the implementation of the control algorithm when there exists no successful state estimator for the system. However using the semi-implicit method, one can circumvent this problem. The resulting adjoint equation (C 2) with the semi-implicit method does not contain any velocities except the wall velocity gradients. Practical implementation of the control algorithm developed here may be possible with the restriction of only one iteration ($k=1$) at each instant of time. The effectiveness of the control algorithm with one iteration was investigated: cases considered still showed a significant reduction of cost.

The result of the application of this semi-implicit scheme to the Navier–Stokes equations cannot be predicted at this time even though the scheme was very successful for the Burgers equation. The mathematical analysis of this topic and the application of feedback control to the Navier–Stokes equations are in progress and will be reported in the future.

Financial support from the Air Force Office of Scientific Research under Contract No. AFOSR-89-0411 is gratefully acknowledged. We also thank Mr Thomas Bewley for useful comments on a draft of this manuscript.

Appendix A. Distributed control of the Burgers equation

The Burgers equation with distributed control is shown in (4.4). Crank–Nicolson in time and second-order centred difference in space are used to discretize it. The analogue of (3.18), then, reads

$$\mathscr{A}u+\mathscr{R}^n(u,f)=0, \tag{A 1}$$

with $u=u^n$, $f=f^n$, and

$$\mathscr{A}u=u_i^n-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}(u_{i+1}^n-2u_i^n+u_{i-1}^n),$$

$$\mathscr{R}^n(u,f)=\frac{1}{8}\frac{\Delta t}{\Delta x}({u_{i+1}^n}^2-{u_{i-1}^n}^2)-\tfrac{1}{2}\Delta t f_i^n-u_i^{n-1}+\frac{1}{8}\frac{\Delta t}{\Delta x}({u_{i+1}^{n-1}}^2-{u_{i-1}^{n-1}}^2)-\tfrac{1}{2}\Delta t f_i^{n-1}$$
$$-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}(u_{i+1}^{n-1}-2u_i^{n-1}+u_{i-1}^{n-1})-\tfrac{1}{2}\Delta t(\chi_i^n+\chi_i^{n-1}),$$

where $f_i^n=\alpha_{0,i}^n+\alpha_{1,i}^n\theta_{1,i}^n$, and $i=2,\dots,I-1$. From (4.6), the cost function becomes

$$J(e^n)=\tfrac{1}{2}l_d\sum_{i=2}^{I-1}({\alpha_{0,i}^n}^2+{\alpha_{1,i}^n}^2)+\tfrac{1}{2}m_d\sum_{i=1}^{I-1}\frac{(u_{i+1}^n-u_i^n)^2}{\Delta x}. \tag{A 2}$$

From (A 1) and (A 2), we find

$$(\mathscr{D}\mathscr{R}/\mathscr{D}u)\,\eta^n=\tfrac{1}{4}(\Delta t/\Delta x)\,(u_{i+1}^n\eta_{i+1}^n-u_{i-1}^n\eta_{i-1}^n),$$
$$(\mathscr{D}\mathscr{R}/\mathscr{D}f)\,\alpha_1^n(\mathscr{D}\theta_1/\mathscr{D}u)\,\eta^n=-\tfrac{1}{2}\Delta t\,\alpha_{1,i}^n\kappa_{1,i}^n\eta_i^n,$$
$$(\mathscr{D}\mathscr{R}/\mathscr{D}f)\,(\hat{\alpha}_0^n+\hat{\alpha}_1^n\theta_1^n)=-\tfrac{1}{2}\Delta t(\hat{\alpha}_{0,i}^n+\hat{\alpha}_{1,i}^n\theta_{1,i}^n),$$
$$\|Cu^n-\gamma_d\|^2=\sum_{i=1}^{I-1}\frac{(u_{i+1}^n-u_i^n)^2}{\Delta x},$$
$$\langle C^*(Cu^n-\gamma_d),\eta^n\rangle=\sum_{i=2}^{I-1}\frac{1}{\Delta x}(-u_{i+1}^n+2u_i^n-u_{i-1}^n)\,\eta_i^n,$$

where $\kappa_1=\mathscr{D}\theta_1/\mathscr{D}u$, and $i=2,\dots,I-1$. For distributed control, θ_1 is usually taken to be u. In that case, $\kappa_1=1$. The analogue of (3.13), then, reads

$$\left(-\frac{1}{4}\frac{\Delta t}{\Delta x}u_{i-1}^n-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\eta_{i-1}^n+\left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}-\tfrac{1}{2}\Delta t\,\alpha_{1,i}^n\kappa_{1,i}^n\right)\eta_i^n$$
$$+\left(\frac{1}{4}\frac{\Delta t}{\Delta x}u_{i+1}^n-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\eta_{i+1}^n=\tfrac{1}{2}\Delta t(\hat{\alpha}_{0,i}^n+\hat{\alpha}_{1,i}^n\theta_{1,i}^n), \tag{A 3}$$

$$\eta_1^n=\eta^n(x=0)=0,\quad \eta_I^n=\eta^n(x=1)=0,$$

where $i=2,\dots,I-1$.

Using the property of the discrete adjoint operator (e.g. $\mathscr{A}^*_{ij} = \mathscr{A}_{ji}$) the analogue of (3.14) is

$$\left(\frac{1}{4}\frac{\Delta t}{\Delta x}u_i^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i-1}^n + \left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2} - \tfrac{1}{2}\Delta t\,\alpha_{1,i}^n\kappa_{1,i}^n\right)\zeta_i^n$$
$$+\left(-\frac{1}{4}\frac{\Delta t}{\Delta x}u_i^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i+1}^n = \frac{1}{\Delta x}(-u_{i+1}^n + 2u_i^n - u_{i-1}^n), \quad \text{(A 4)}$$
$$\zeta_1^n = \zeta^n(x=0) = 0, \quad \zeta_I^n = \zeta^n(x=1) = 0,$$

where $i = 2, \ldots, I-1$.

The directional derivative of J (equation (3.15)) is

$$\frac{\mathscr{D}J}{\mathscr{D}e}\hat{e} = \sum_{i=2}^{I-1}(l_d\,\alpha_{0,i}^n + \tfrac{1}{2}m_d\,\Delta t\,\zeta_i^n)\,\hat{\alpha}_{0,i}^n + \sum_{i=2}^{I-1}(l_d\,\alpha_{1,i}^n + \tfrac{1}{2}m_d\,\Delta t\,\theta_{1,i}^n\,\zeta_i^n)\,\hat{\alpha}_{1,i}^n. \quad \text{(A 5)}$$

The analogue of the gradient algorithm equation (3.11) consists of constructing two sequences $\alpha_0^{n,k}$, $\alpha_1^{n,k}$, recursively defined by

$$\left.\begin{aligned}\alpha_{0,i}^{n,k+1} &= \alpha_{0,i}^{n,k} - \rho_0(l_d\,\alpha_{0,i}^{n,k} + \tfrac{1}{2}m_d\,\Delta t\,\zeta_i^{n,k}),\\ \alpha_{1,i}^{n,k+1} &= \alpha_{1,i}^{n,k} - \rho_1(l_d\,\alpha_{1,i}^{n,k} + \tfrac{1}{2}m_d\,\Delta t\,\theta_{1,i}^{n,k}\,\zeta_i^{n,k}),\end{aligned}\right\} \quad \text{(A 6)}$$

where k is the iteration index, $\rho_0, \rho_1 > 0$, and $i = 2, \ldots, I-1$.

The sensitivity of the cost function with respect to the control variables can be described by

$$\left.\begin{aligned}(\mathscr{D}J/\mathscr{D}\alpha_{0,i})(\alpha_{0,i}^{n,k}, \alpha_{1,i}^{n,k}) &= l_d\,\alpha_{0,i}^{n,k} + \tfrac{1}{2}m_d\,\Delta t\,\zeta_i^{n,k},\\ (\mathscr{D}J/\mathscr{D}\alpha_{1,i})(\alpha_{0,i}^{n,k}, \alpha_{1,i}^{n,k}) &= l_d\,\alpha_{1,i}^{n,k} + \tfrac{1}{2}m_d\,\Delta t\,\theta_{1,i}^{n,k}\,\zeta_i^{n,k},\end{aligned}\right\} \quad \text{(A 7)}$$

where $i = 2, \ldots, I-1$.

Appendix B. Boundary control of the Burgers equation

The Burgers equation with boundary control is shown in (4.7). Crank–Nicolson in time and second-order centred difference in space are used to discretize (4.7). The analogue of (3.18), then, reads

$$\mathscr{A}u + \mathscr{R}^n(u, \psi) = 0, \quad \text{(B 1)}$$

with $u = u^n$, $\psi = \psi^n$, and

$$\mathscr{A}u = \begin{cases} u_i^n - \dfrac{1}{2}\dfrac{1}{Re}\dfrac{\Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) & \text{for } i = 2, \ldots, I-1;\\ u_1^n & \text{for } i = 1;\\ u_I^n & \text{for } i = I,\end{cases}$$

$$\mathscr{R}^n(u,\psi) = \begin{cases} \dfrac{1}{8}\dfrac{\Delta t}{\Delta x}({u_{i+1}^n}^2 - {u_{i-1}^n}^2) - u_i^{n-1} + \dfrac{1}{8}\dfrac{\Delta t}{\Delta x}({u_{i+1}^{n-1}}^2 - {u_{i-1}^{n-1}}^2) \\ \quad -\dfrac{1}{2}\dfrac{1}{Re}\dfrac{\Delta t}{\Delta x^2}(u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}) - \tfrac{1}{2}\Delta t(\chi_i^n + \chi_i^{n-1}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } i = 2, \ldots, I-1;\\ -\psi_0^n \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\; \text{for } i = 1;\\ -\psi_1^n \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\; \text{for } i = I,\end{cases}$$

where $\psi_j^n = \alpha_{0,j}^n + \alpha_{1,j}^n \theta_{1,j}^n$, and $j = 0, 1$. From (4.9), the cost function becomes

$$J(e^n) = \tfrac{1}{2} l_b ({\alpha_{0,0}^n}^2 + {\alpha_{0,1}^n}^2 + {\alpha_{1,0}^n}^2 + {\alpha_{1,1}^n}^2) + \tfrac{1}{2} m_b \left[\left(\frac{u_2^n - u_1^n}{\Delta x} \right)^2 + \left(\frac{u_I^n - u_{I-1}^n}{\Delta x} \right)^2 \right]. \quad \text{(B 2)}$$

From (B 1) and (B 2), we find

$$\frac{\mathscr{D}\mathscr{R}}{\mathscr{D}u}(u^n, \psi^n)\,\eta^n = \begin{cases} \dfrac{1}{4}\dfrac{\Delta t}{\Delta x}(u_{i+1}^n \eta_{i+1}^n - u_{i-1}^n \eta_{i-1}^n) & \text{for} \quad i = 2, \dots, I-1; \\ 0 & \text{for} \quad i = 1, I, \end{cases}$$

$$\frac{\mathscr{D}\mathscr{R}}{\mathscr{D}\psi} \alpha_1^n \frac{\mathscr{D}\theta_1}{\mathscr{D}u} \eta^n = \begin{cases} 0 & \text{for} \quad i = 2, \dots, I-1; \\ -\alpha_{1,0}^n \kappa_{1,0}^n \dfrac{\partial \eta^n}{\partial x}(x=0) & \text{for} \quad i = 1; \\ -\alpha_{1,1}^n \kappa_{1,1}^n \dfrac{\partial \eta^n}{\partial x}(x=1) & \text{for} \quad i = I, \end{cases}$$

$$\frac{\mathscr{D}\mathscr{R}}{\mathscr{D}\psi}(\hat{\alpha}_0^n + \hat{\alpha}_1^n \theta_1^n) = \begin{cases} 0 & \text{for} \quad i = 2, \dots, I-1; \\ -(\hat{\alpha}_{0,0}^n + \hat{\alpha}_{1,0}^n \theta_{1,0}^n) & \text{for} \quad i = 1; \\ -(\hat{\alpha}_{0,1}^n + \hat{\alpha}_{1,1}^n \theta_{1,1}^n) & \text{for} \quad i = I, \end{cases}$$

$$\|Cu^n - \gamma_d\|^2 = \left(\frac{u_2^n - u_1^n}{\Delta x} \right)^2 + \left(\frac{u_I^n - u_{I-1}^n}{\Delta x} \right)^2,$$

$$\begin{aligned} \langle C^*(Cu^n - \gamma_d), \eta \rangle &= \langle Cu^n - \gamma_d, C\eta \rangle \\ &= \frac{1}{\Delta x^2}[(u_2^n - u_1^n)(\eta_2^n - \eta_1^n) + (u_I^n - u_{I-1}^n)(\eta_I^n - \eta_{I-1}^n)]. \end{aligned}$$

Here we assumed that $\theta_1 = \theta_1(\partial u/\partial x)$. Hence

$$\frac{\mathscr{D}\theta_1}{\mathscr{D}u}\eta = \frac{\mathscr{D}\theta_1}{\mathscr{D}(\partial u/\partial x)} \frac{\mathscr{D}(\partial u/\partial x)}{\mathscr{D}u}\eta = \kappa_1 \frac{\partial \eta}{\partial x},$$

where $\kappa_1 = \mathscr{D}\theta_1/\mathscr{D}(\partial u/\partial x)$. The analogue of (3.13), then, reads

$$\left(-\frac{1}{4}\frac{\Delta t}{\Delta x} u_{i-1}^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2} \right) \eta_{i-1}^n + \left(1 + \frac{1}{Re}\frac{\Delta t}{\Delta x^2} \right) \eta_i^n + \left(\frac{1}{4}\frac{\Delta t}{\Delta x} u_{i+1}^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2} \right) \eta_{i+1}^n = 0, \quad \text{(B 3)}$$

$$\eta_1^n = \eta^n(x=0) = \hat{\alpha}_{0,0}^n + \hat{\alpha}_{1,0}^n \theta_{1,0}^n + \alpha_{1,0}^n \kappa_{1,0}^n \frac{\partial \eta^n}{\partial x}(x=0),$$

$$\eta_I^n = \eta^n(x=1) = \hat{\alpha}_{0,1}^n + \hat{\alpha}_{1,1}^n \theta_{1,1}^n + \alpha_{1,1}^n \kappa_{1,1}^n \frac{\partial \eta^n}{\partial x}(x=1),$$

where $i = 2, \dots, I-1$.

Using the property of the discrete adjoint operator (e.g. $\mathscr{A}^*_{ij} = \mathscr{A}_{ji}$), the analogue of (3.14) is

$$\left(\frac{1}{4}\frac{\Delta t}{\Delta x}u_i^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i-1}^n + \left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_i^n + \left(-\frac{1}{4}\frac{\Delta t}{\Delta x}u_i^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i+1}^n = 0 \quad \text{for} \quad i = 3, \dots, I-2, \tag{B 4a}$$

$$\left(1+\frac{1}{\Delta x}\alpha_{1,0}^n \kappa_{1,0}^n\right)\zeta_1^n + \left(-\frac{1}{4}\frac{\Delta t}{\Delta x}u_1^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_2^n = \frac{1}{\Delta x^2}(-u_2^n + u_1^n) \quad \text{for} \quad i=1, \tag{B 4b}$$

$$\left(-\frac{1}{\Delta x}\alpha_{1,0}^n \kappa_{1,0}^n\right)\zeta_1^n + \left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_2^n + \left(-\frac{1}{4}\frac{\Delta t}{\Delta x}u_2^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_3^n = -\frac{1}{\Delta x^2}(-u_2^n + u_1^n) \quad \text{for} \quad i=2, \tag{B 4c}$$

$$\left(\frac{1}{4}\frac{\Delta t}{\Delta x}u_{I-1}^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-2}^n + \left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-1}^n + \left(\frac{1}{\Delta x}\alpha_{1,1}^n \kappa_{1,1}^n\right)\zeta_I^n = \frac{1}{\Delta x^2}(-u_I^n + u_{I-1}^n) \quad \text{for} \quad i=I-1, \tag{B 4d}$$

$$\left(\frac{1}{4}\frac{\Delta t}{\Delta x}u_I^n - \frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-1}^n + \left(1-\frac{1}{\Delta x}\alpha_{1,1}^n \kappa_{1,1}^n\right)\zeta_I^n = -\frac{1}{\Delta x^2}(-u_I^n + u_{I-1}^n) \quad \text{for} \quad i=I. \tag{B 4e}$$

The directional derivative of J (equation (3.15)) is

$$\frac{\mathscr{D}J}{\mathscr{D}e}\hat{e} = (l_b \alpha_{0,0}^n + m_b \zeta_1^n)\hat{\alpha}_{0,0}^n + (l_b \alpha_{0,1}^n + m_b \zeta_I^n)\hat{\alpha}_{0,1}^n + (l_b \alpha_{1,0}^n + m_b \zeta_1^n \theta_{1,0}^n)\hat{\alpha}_{1,0}^n + (l_b \alpha_{1,1}^n + m_b \zeta_I^n \theta_{1,1}^n)\hat{\alpha}_{1,1}^n. \tag{B 5}$$

The analogue of the gradient algorithm equation (3.11) consists of constructing two sequences $\alpha_0^{n,k}, \alpha_1^{n,k}$, recursively defined by

$$\left.\begin{aligned} \alpha_{0,0}^{n,k+1} &= \alpha_{0,0}^{n,k} - \rho_0(l_b \alpha_{0,0}^{n,k} + m_b \zeta_1^{n,k}), & \alpha_{0,1}^{n,k+1} &= \alpha_{0,1}^{n,k} - \rho_0(l_b \alpha_{0,1}^{n,k} + m_b \zeta_I^{n,k}), \\ \alpha_{1,0}^{n,k+1} &= \alpha_{1,0}^{n,k} - \rho_1(l_b \alpha_{1,0}^{n,k} + m_b \zeta_1^{n,k}\theta_{1,0}^{n,k}), & \alpha_{1,1}^{n,k+1} &= \alpha_{1,1}^{n,k} - \rho_1(l_b \alpha_{1,1}^{n,k} + m_b \zeta_I^{n,k}\theta_{1,1}^{n,k}), \end{aligned}\right\} \tag{B 6}$$

where k is the iteration index, and $\rho_0, \rho_1 > 0$.

The sensitivity of the cost function with respect to the control variables can be described by

$$\left.\begin{aligned} \frac{\mathscr{D}J}{\mathscr{D}\alpha_{0,0}}(\alpha_{0,j}^{n,k}, \alpha_{1,j}^{n,k}) &= l_b \alpha_{0,0}^{n,k} + m_b \zeta_1^{n,k}, & \frac{\mathscr{D}J}{\mathscr{D}\alpha_{0,1}}(\alpha_{0,j}^{n,k}, \alpha_{1,j}^{n,k}) &= l_b \alpha_{0,1}^{n,k} + m_b \zeta_I^{n,k}, \\ \frac{\mathscr{D}J}{\mathscr{D}\alpha_{1,0}}(\alpha_{0,j}^{n,k}, \alpha_{1,j}^{n,k}) &= l_b \alpha_{1,0}^{n,k} + m_b \zeta_1^{n,k}\theta_{1,0}^{n,k}, & \frac{\mathscr{D}J}{\mathscr{D}\alpha_{1,1}}(\alpha_{0,j}^{n,k}, \alpha_{1,j}^{n,k}) &= l_b \alpha_{1,1}^{n,k} + m_b \zeta_I^{n,k}\theta_{1,1}^{n,k}, \end{aligned}\right\} \tag{B 7}$$

where $j = 0, 1$.

Appendix C. Dependence of the control algorithm on the time-discretization method

In this Appendix, a semi-implicit method is used to discretize the Burgers equation with boundary control (equation (4.7)): Adams–Bashforth scheme for the nonlinear term and Crank–Nicolson scheme for the viscous term. The equations corresponding to (B 3) and (B 4) become

$$\left.\begin{aligned}
&\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\eta_{i-1}^n+\left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\eta_i^n+\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\eta_{i+1}^n=0,\\
&\eta_1^n=\eta^n(x=0)=\hat{\alpha}_{0,0}^n+\hat{\alpha}_{1,0}^n\theta_{1,0}^n+\alpha_{1,0}^n\kappa_{1,0}^n\frac{\partial\eta^n}{\partial x}(x=0),\\
&\eta_I^n=\eta^n(x=1)=\hat{\alpha}_{0,1}^n+\hat{\alpha}_{1,1}^n\theta_{1,1}^n+\alpha_{1,1}^n\kappa_{1,1}^n\frac{\partial\eta^n}{\partial x}(x=1),
\end{aligned}\right\}\quad (C\,1)$$

where $i=2,\ldots,I-1$, and

$$\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i-1}^n+\left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_i^n+\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{i+1}^n=0 \quad \text{for} \quad i=3,\ldots,I-2, \quad (C\,2a)$$

$$\left(1+\frac{1}{\Delta x}\alpha_{1,0}^n\kappa_{1,0}^n\right)\zeta_1^n+\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_2^n=\frac{1}{\Delta x^2}(-u_2^n+u_1^n) \quad \text{for} \quad i=1, \quad (C\,2b)$$

$$\left(-\frac{1}{\Delta x}\alpha_{1,0}^n\kappa_{1,0}^n\right)\zeta_1^n+\left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_2^n+\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_3^n$$
$$=-\frac{1}{\Delta x^2}(-u_2^n+u_1^n) \quad \text{for} \quad i=2, \quad (C\,2c)$$

$$\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-2}^n+\left(1+\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-1}^n+\left(\frac{1}{\Delta x}\alpha_{1,1}^n\kappa_{1,1}^n\right)\zeta_I^n$$
$$=\frac{1}{\Delta x^2}(-u_I^n+u_{I-1}^n) \quad \text{for} \quad i=I-1, \quad (C\,2d)$$

$$\left(-\frac{1}{2}\frac{1}{Re}\frac{\Delta t}{\Delta x^2}\right)\zeta_{I-1}^n+\left(1-\frac{1}{\Delta x}\alpha_{1,1}^n\kappa_{1,1}^n\right)\zeta_I^n=-\frac{1}{\Delta x^2}(-u_I^n+u_{I-1}^n) \quad \text{for} \quad i=I.(C\,2e)$$

Note that only the wall velocity gradients are needed to solve the adjoint equation (C 2). This fact is, of course, of crucial importance for the physical implementation of the control algorithm.

REFERENCES

ABERGEL, F. & TEMAM, R. 1990 On some control problems in fluid mechanics. *Theor. Comput. Fluid Dyn.* 1, 303.

ABERGEL, F. & TEMAM, R. 1992 Optimal control of turbulent flows. In *Optimal Control of Viscous Flows* (ed. S. Sritharan). SIAM (to appear).

BANACH, A. S. & BAUMANN, W. T. 1990 Gain-scheduled control of nonlinear partial differential equations. In *Proc. 29th IEEE Conf. on Decision and Control*, pp. 387–392.

BARBU, V. 1992 H_∞ boundary control with state feedback; the hyperbolic case. Preprint.

BECHERT, D. W. & BARTENWERFER, M. 1989 The viscous flow on surfaces with longitudinal ribs. *J. Fluid Mech.* **206**, 105.

BENSOUSSAN, A. & TEMAM, R. 1972 Equations aux dérivées partielles stochatiques non linéaires (I). *Israel J. Maths* **11**, 95.

BENSOUSSAN, A. & TEMAM, R. 1973 Equations stochastiques du type Navier–Stokes. *J. Funct. Anal.* **13**, 195.

BUSHNELL, D. M. & MCGINLEY, C. B. 1989 Turbulence control in wall flows. *Ann. Rev. Fluid Mech.* **21**, 1.

BYRNES, C. I. & GILLIAM, D. S. 1991 Boundary feedback design for nonlinear distributed parameter systems. In *Proc. 30th IEEE Conf. on Decision and Control*, pp. 2340–2342.

CANTWELL, B. J. 1981 Organized motion in turbulent flow. *Ann. Rev. Fluid Mech.* **13**, 457.

CHAMBERS, D. H., ADRIAN, R. J., MOIN, P., STEWART, D. S. & SUNG, H. J. 1988 Karhunen–Loéve expansion of Burgers' model of turbulence. *Phys. Fluids* **31**, 2573.

CHOI, H., MOIN, P. & KIM, J. 1992 Turbulent drag reduction: studies of feedback control and flow over riblets. *Rep.* TF-55. Department of Mechanical Engineering, Stanford University, Stanford, CA.

FINLAYSON, B. A. 1972 *The Method of Weighted Residuals and Variational Principles*. Academic.

FOIAS, C. & TANNENBAUM, A. 1989 Weighted optimization theory for nonlinear systems. *SIAM J. Control Optim.* **27**, 842.

GUNZBURGER, M. D. (ed.) 1992 *Proc. Conference on Control of Flows*. Inst. for Math. and Its Appl., University of Minnesota, November 1992.

GUNZBURGER, M. D., HOU, L. & SVOBODNY, T. P. 1990 A numerical method for drag minimization via the suction and injection of mass through the boundary. In *Stabilization of Flexible Structures* (ed. J. P. Zolesio). Springer.

GUNZBURGER, M. D., HOU, L. & SVOBODNY, T. P. 1991 Analysis and finite element approximations of optimal control problems for the stationary Navier–Stokes equations with Dirichlet controls. *Math. Model. Numer. Anal.* **25**, 711.

GUNZBURGER, M. D., HOU, L. & SVOBODNY, T. P. 1992 Boundary velocity control of incompressible flow with an application to viscous drag reduction. *SIAM J. Control Optim.* **30**, 167.

KANG, S., ITO, K. & BURNS, J. A. 1991 Unbounded observation and boundary control problems for Burgers equation. In *Proc. 30th IEEE Conf. on Decision and Control*, pp. 2687–2692.

KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.* **177**, 133.

LADYZHENSAKAYA, O. A. 1963 *The Mathematical Theory of Viscous Incompressible Flow*. Gordon and Breach.

LIONS, J. L. 1971 *Optimal Control of Systems Governed by Partial Differential Equations*. Springer.

LUENBERGER, D. 1973 *Introduction to linear and nonlinear programming*. Addison-Wesley.

SRITHARAN, S. S. 1991*a* Dynamic programming of the Navier–Stokes equations. *Systems Control Lett.* **16**, 299.

SRITHARAN, S. S. 1991*b* On the nonsmooth verification technique for the dynamic programming of viscous flows. *IMA Preprint* 850.

SRITHARAN, S. S. 1992 Optimal control theory of viscous flows. In *Optimal Control of Viscous Flows* (ed. S. Sritharan). SIAM (to appear).

SRITHARAN, S. S. (ed.) 1992 *Optimal Control of Viscous Flows*. SIAM (to appear).

SRITHARAN, S. S., OU, Y.-R., BURNS, J. A., LADD, D., HENDRICKS, E. & NOSSIER, N. 1991 Active control of viscous flow past a cylinder: theory, computation and experiment. *Bull. Am. Phys. Soc.* **36**, 2626.

TEMAM, R. 1984 *Navier–Stokes Equations*, 3rd revised edn. North-Holland.

TEMAM, R. 1991 Navier–Stokes equations: theory and approximations, chapter I. *Preprint* 91-16, Institute for Scientific Computing and Applied Mathematics, Bloomington, Indiana. Also to appear in the *Handbook of Numerical Analysis* (ed. P. G. Ciarlet & J. L. Lions). North-Holland.

WALSH, M. J. 1983 Riblets as a viscous drag reduction technique. *AIAA J.* **21**, 485.