Optimal Control of the Unsteady Navier-Stokes Equations

Zhijin Li

School of Computational Science and Information Technology
Florida State University, Tallahassee, Florida

I. M. Navon

Department of Mathematics and
School of Computational Science and Information Technology
Florida State University, Tallahassee, Florida

M. Y. Hussaini

School of Computational Science and Information Technology
Florida State University, Tallahassee, Florida

and

F.-X. Le Dimet

Laboratoire de Modelization et Calcul
Universite Joseph Fourier, Grenoble Cedex 9, France

Running Title: Control of Navier-Stokes Equations

Corresponding author

Prof. I. M. Navon

Florida State University/CSIT

Dirac Science Library Building

Tallahassee, Florida 32306-4120, Email: navon@csit.fsu.edu

Phone: (850) 644-6560

Fax: (850) 644-0098

A general method based on adjoint formulation is discussed for the optimal control of distributed parameter systems (including boundary parameter) which is especially suitable for large dimensional control problems. Strategies for efficient and robust implementation of the method are described. A set of consistency checks are proposed for the validation of the adjoint code. The method is applied to the problem of controlling vortex shedding behind a cylinder (through suction/ blowing on the cylinder surface) governed by the unsteady two-dimensional incompressible Navier-Stokes equations with time-dependent boundary conditions. Three types of objective functions are considered, with regularization to circumvent ill-posedness. These objective functions involve integration over a space-time domain.

A complete control of vortex shedding is demonstrated for Reynolds numbers up to 110. The optimal values of the suction/blowing parameters are found to be insensitive to initial conditions of the model when the time window of control is larger than the vortex shedding period, the inverse of the Strouhal frequency. Although this condition is necessary for robust control, it is observed that a shorter window of control may suffice to suppress vortex shedding.

Key Words: optimal control, flow control, coding adjoint model, quasi-Newton algorithm, regularization, Navier-Stokes equations, Karman vortex shedding

Mathematical Reviews Index: 49A22; 65K10, 93C20

1. INTRODUCTION

Application of optimal control theory to flow control problems has recently attracted increased attention [1, 6, 15, 22, 11, 23, 2, 19] . A key element of an optimal flow control problem is the minimization of an objective or cost functional which provides a quantitative measure of the desired objective and depends critically on the solution (known as the optimal solution) that satisfies the partial differential equations governing the fluid flow. For instance, the integral of the dissipation function may be employed as an objective functional, the governing PDE's are the Navier-Stokes equations, and their optimal solution represents the flow with minimum drag on a body (e.g., [15, 1]). Minimizing a defined objective functional is necessarily an iterative procedure due to the nonlinearity of the Navier-Stokes equations, and therefore is computationally intensive.

Complete mathematical treatment of optimal control problems and their computer implementations have been carried out in such relatively simple cases as Burgers' equation (e.g.,[3, 5, 6]) and the steady two-dimensional Navier-Stokes equations (e.g., [17, 18, 7, 10, 21]) . The state equations (i.e., the equations governing the flow) are treated as constraints, and all the state variables and control parameters are involved in the minimization process. Constrained minimization algorithms then become applicable such as the general reduced gradient or the gradient projection algorithms [7] and the sequential quadratic programming [10, 21].

The optimal control problems associated with the unsteady Navier-Stokes equations pose a challenge for current computational techniques and computer capabilities. The dimension of the unsteady problem is N times larger than that of the corresponding steady-state problem when the number of time steps is N. For large N, constrained minimization algorithms are computationally unfeasible. Alternative approaches have been employed in some recent attempts at solving such problems numerically (e.g., [23, 22, 2, 19]). Among the alternative approaches are the suboptimal control algorithms that create reduced problems of tractable dimensions. They are essentially of two types. The first type is the method of instantaneous control introduced in [6], where a steady optimal control problem is solved at each discrete time level t_i, and optimal control is used to steer the system from t_i to t_{i+1}. This method was recently applied to the control of Karman vortex shedding behind a circular cylinder by optimizing the blowing and suction on the surface of the cylinder [28]. The second type projects relevant variables to a lower dimensional space, thus reducing the dimension of the control problem, and the optimization is performed with the reduced system [33, 12].

Another strategy to circumvent the problem of computational complexity is to tranform the problem into an unconstrained minimization problem by solving for the state variables in terms of the control parameters, thereby eliminating the constraints. It is well known that all efficient minimization algorithms require computation of the gradient of the objective functional with respect to the control parameters. This gradient information can be directly utilized for efficient unconstrained minimization based on, for example, quasi-Newton types of algorithms. The derivative of the functional can be directly and efficiently computed with respect to the control parameters in an adjoint formulation. This is true for both boundary and distributed controls (see section 2). Such an adjoint technique has been implemented in idealized distributed controls in [19] and boundary controls in [2].

The main difficulty in implementing the adjoint technique is the derivation of the adjoint models [16]. In the literature on flow control, the adjoint equation is derived by applying the method of integration by parts to the discrete Navier-Stokes equations as described in [2]. However, integration by parts is difficult to implement for complex boundary geometries and sophisticated time-stepping schemes. In such cases, it is inevitably necessary to approximate the method of integration by parts, which may lead to inconsistency between the original model and its adjoint.

Due to this limitation, previous investigations based on adjoint models are basically confined to simple rectangular boundaries and relatively simple time-stepping schemes [19, 2].

In the present paper, a flexible and efficient approach for constructing adjoint models is proposed, which does not suffer from the aforementioned limitation. In this approach, the adjoint is constructed directly from the source code of the original discrete nonlinear model, thereby ensuring consistency between the discrete nonlinear model and its adjoint. Boundary conditions at complex geometries and sophisticated time-stepping schemes do not present any particular difficulties in this approach. The consistency between the original model and its adjoint allows us to perform rigorous verification of the correctness of the adjoint and the computed gradient, thus guaranteeing the reliability of this procedure. It is but proper to emphasize that that the source code could be a solver of any partial differential equation set.

Based on the proposed approach, an optimal control algorithm was developed for controlling the Karman vortex street behind a circular cylinder in a uniform stream. Injection and suction at the surface of the cylinder were the control mechanisms to suppress vortex shedding. This optimal control algorithm yielded the optimal distribution of injection/suction that suppressed vortex shedding in a robust way for Reynolds numbers up to at least 110. (Vortex shedding can occur starting at Reynolds number of 40.) Note that vortex shedding is completely suppressed only for lower values of the Reynolds number when other control methods are used [34, 32].

The objective functional was defined as the space-time integral of some physical quantity. The minimization of the objective functional was carried out over a time interval sufficiently large to cover the evolution of a flow pattern (in an uncontrolled situation) that involves all possible unstable wavelengths. It was observed that the time window should be longer than the vortex shedding period in order to attain a robust control, but a complete suppression of vortex shedding is possible even if the time window is only half as large as the vortex shedding period.

The dependence of optimal parameters on the initial conditions was examined. It was noticed that these were rather insensitive to intial conditions if the length of the time window is longer than that of the vortex shedding period. Optimal flow control problems could be ill-posed in the sense of Tikhonov and Arsenin (1977)[37]. The minimization process for the vortex shedding control problem was found to be ill-posed. To deal with the ill-posedness, the objective functional was regularized in a manner similar to that employed by many investigators (e.g., [1, 6, 2]).

This paper is organized as follows. After a general introduction in Section 1, a basic derivation of the general expressions for the gradient of the objective functional in terms of the adjoint models is presented in Section 2. It also includes the construction of discrete adjoint models as well as the description storage strategies. The two-dimensional Navier-Stokes equations and their discretized analogs are described in Section 3. Basic procedures for checking the correctness of the adjoint model, especially the accuracy of computed gradients, are discussed in Section 4. In Section 5, a quasi-Newton minimization algorithm is presented for minimizing the objective functional in the optimal flow control problems under consideration. Numerical experiments are then presented in Section 6. Ill-posedness of vortex shedding control problems and regularization approaches to alleviate this ill-posedness are discussed in Section 7. Finally, Section 8 discusses and summarizes major results obtained in this work.

2. ADJOINT: A GENERAL TECHNIQUE FOR OPTIMAL CONTROL

2.1. Basic assumptions

In the following, boldface letters denote vectors or matrices. We can write a model (the state equation) governing the unsteady fluid flow as

$$\mathbf{x}(t) = \mathbf{M}_{\boldsymbol{\mu}}(t, t_0)\mathbf{x}(t_0), \tag{1}$$

where $\mathbf{M}_{\boldsymbol{\mu}}(t, t_0)$ is a nonlinear operator that depends on the control parameter $\boldsymbol{\mu}$. We focus on a space discretized model such as is obtained by finite difference, finite element or spectral discretization methods. In this case, $\mathbf{M}_{\boldsymbol{\mu}}(t, t_0)$ is a matrix, say, of finite order K, and correspondingly $\mathbf{x}(t)$ is a K-dimensional state vector at time t. In the fluid dynamics context, $\mathbf{x}(t)$ may consist of the discretized flow velocity components and pressure (see Section 3).

By introducing a time discretization, equation (1) may be written as

$$\mathbf{x}_{n+1} = \mathbf{M}_{\boldsymbol{\mu}}(t_{n+1}, t_n)\mathbf{x}_n, \quad n = 0, \cdots, N-1, \tag{2}$$

where N is the total number of time steps in the interval of the model integration and $\mathbf{x}_{n+1} = \mathbf{x}(t_{n+1})$. In boundary control problems, the boundary conditions naturally assume particular importance. The boundary conditions are treated separately, although they can be combined into the state equation (1). We may express the Dirichlet, Neumann and Robin boundary conditions by a single equation

$$\mathbf{x}_{n+1} = \mathbf{C}_{n+1}\mathbf{x}_{n+1} + \mathbf{B}_{n+1}\mathbf{b}, \tag{3}$$

where $\mathbf{C}_{n+1}$ and $\mathbf{B}_{n+1}$ are linear operators. Here, the parameter $\mathbf{b}$ specifies the boundary condition. The entries of $\mathbf{B}_{n+1}$ are either 0 or 1 for all three cases. The entries of $\mathbf{C}_{n+1}$ are also either 0 or 1 for the Dirichlet and Neumann conditions, but some entries of $\mathbf{C}_{n+1}$ depend on the parameter that specifies the Robin condition.

For simplicity, we first define an objective functional of the form

$$J(\mathbf{x}_N). \tag{4}$$

In fact, this objective functional is a function of the initial condition $\mathbf{x}_0$, the model parameters $\boldsymbol{\mu}$ and the boundary parameter $\mathbf{b}$, all of which are control parameters. With these assumptions, the optimal flow control consists in finding a set of parameters, $\mathbf{x}_0$, $\boldsymbol{\mu}$, and $\mathbf{b}$, such that they minimize

$J(\mathbf{x}_N)$ subject to the state equation. This problem is a distributed control problem if it involves $\mathbf{x}_0$ and $\boldsymbol{\mu}$ (which may be distributed over the whole fluid domain) and a boundary control problem if it involves only $\mathbf{b}$.

2.2. Formulation of Gradients

Taking the first variation of the objective function, we obtain

$$\delta J(\mathbf{x}_N) = (J'(\mathbf{x}_N))^T \delta \mathbf{x}_N, \tag{5}$$

where $\delta J(\mathbf{x}_N)$ is the first variation of $J(\mathbf{x}_N)$, and $J'(\mathbf{x}_N)$ the first order derivative of $J(\mathbf{x}_N)$ with respect to x_N. Since $\mathbf{x}_N$ depends on $\mathbf{x}_0$, $\boldsymbol{\mu}$, and $\mathbf{b}$, the $\delta\mathbf{x}_N$ are naturally functions of $\delta\mathbf{x}_0$, $\delta\boldsymbol{\mu}$, and $\delta\mathbf{b}$. The first variation of (2) and the boundary condition yields

$$\delta \mathbf{x}_{n+1} = \mathbf{L}(t_{n+1}, t_n)\delta\mathbf{x}_n + \boldsymbol{\Gamma}_{n+1}\delta\boldsymbol{\mu}, \quad n = 0, \cdots, N-1, \tag{6}$$

$$\delta \mathbf{x}_{n+1} = \mathbf{C}_{n+1}\delta\mathbf{x}_{n+1} + \mathbf{B}_{n+1}\delta\mathbf{b}. \tag{7}$$

Here $\mathbf{L}(t_{n+1}, t_n)$ and $\boldsymbol{\Gamma}_{n+1}$ are the Jacobians associated with $\mathbf{M}(t_{n+1}, t_n)$ with respect to $\mathbf{x}$ and $\boldsymbol{\mu}$ respectively. Eq (6) is generally called the tangent linear model of the nonlinear state equation (2).

Let us denote the propagator or resolvent of the linearized equation as

$$\mathbf{G}(t_j, t_i) = (\mathbf{C}_j\mathbf{L}(t_j, t_{j-1}))(\mathbf{C}_{j-1}\mathbf{L}(t_{j-1}, t_{j-2}))\cdots(\mathbf{C}_i\mathbf{L}(t_{i+1}, t_i)); \tag{8}$$

$$\mathbf{G}(t_j, t_j) = \mathbf{I}; \tag{9}$$

$$j > i, \quad 0 \le i \le N-1, \quad 1 \le j \le N. \tag{10}$$

where I is the identity matrix. The boundary condition (7) has been taken into account in (9) .

We then obtain by time integration of the linearized equation

$$\delta\mathbf{x}_N = \mathbf{G}(t_N, t_0)\delta\mathbf{x}_0 + \sum_{j=1}^{N}\mathbf{G}(t_N, t_j)\boldsymbol{\Gamma}_j\delta\boldsymbol{\mu} + \sum_{j=1}^{N}\mathbf{G}(t_N, t_j)\mathbf{B}_j\delta\mathbf{b}. \tag{11}$$

The substitution of (11) into (5) yields

$$\delta J(\mathbf{x}_N) = [\mathbf{G}^T(t_N,t_0)J'(\mathbf{x}_N)]^T\delta\mathbf{x}_0 + \sum_{j=1}^{N}[\mathbf{\Gamma}_j^T\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N)]^T\delta\boldsymbol{\mu} + \sum_{j=1}^{N}[\mathbf{B}_j^T\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N)]^T\delta\mathbf{b}. \tag{12}$$

The basic principle of variational analysis requires that

$$\delta J(\mathbf{x}_N) = \nabla_{\mathbf{x}_0}J(\mathbf{x}_N)\delta\mathbf{x}_0 + \nabla_{\boldsymbol{\mu}}J(\mathbf{x}_N)\delta\boldsymbol{\mu} + \nabla_b J(\mathbf{x}_N)\delta\mathbf{b} \tag{13}$$

where $\nabla_{\mathbf{x}_0}J(\mathbf{x}_N)$, $\nabla_{\boldsymbol{\mu}}J(\mathbf{x}_N)$ and $\nabla_b J(\mathbf{x}_N)$ are the gradients of $J(\mathbf{x}_N)$ with respect to $\mathbf{x}_0$, $\boldsymbol{\mu}$ and $\mathbf{b}$, respectively. The comparison between (12) and (13) yields the desired formulations of the derivatives

$$\nabla_{\mathbf{x}_0}J(\mathbf{x}_N) = \mathbf{G}^T(t_N,t_0)J'(\mathbf{x}_N), \tag{14}$$

$$\nabla_{\boldsymbol{\mu}}J(\mathbf{x}_N) = \sum_{j=1}^{N}\mathbf{\Gamma}_j^T\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N), \tag{15}$$

$$\nabla_b J(\mathbf{x}_N) = \sum_{j=1}^{N}B_j^T\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N). \tag{16}$$

2.3. Adjoint technique

For the Euclidean inner product, the adjoint of a linear operator $\mathbf{L}$ is $\mathbf{L}^T$. The expressions of derivatives (14), (15) and (16) can be directly computed when $\mathbf{L}^T$ is available. Note that $J(\mathbf{x}_N)$ is usually a simple quadratic function of $\mathbf{x}_N$, and thus the derivative $J'(\mathbf{x}_N)$ (with respect to $\mathbf{x}_N$) can be computed analytically.

Let us now examine how to compute $\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N)$. From the definition of $\mathbf{G}(t_N,t_j)$ in (9), we have

$$\mathbf{G}^T(t_N,t_j)J'(\mathbf{x}_N) = (\mathbf{L}^T(t_j,t_{j-1})\mathbf{C}_j^T)(\mathbf{L}^T(t_{j+1},t_j)\mathbf{C}_{j+1}^T)\cdots(\mathbf{L}^T(t_N,t_{N-1})\mathbf{C}_N^T)J'(\mathbf{x}_N). \tag{17}$$

We note from (17) that $\mathbf{G}^T(t_N, t_j)J'(\mathbf{x}_N)$ is obtained by integrating the adjoint model backward in time from time t_N to t_j, with the final condition $J'(\mathbf{x}_N)$ at time t_N. From the derivative expressions (14), (15) and (16), $\nabla_{\mathbf{x}_0} J(\mathbf{x}_N)$ is thus the immediate result of the time integration of the adjoint model backward from time t_N to t_0, while $\nabla_{\boldsymbol{\mu}} J(\mathbf{x}_N)$ and $\nabla_b J(\mathbf{x}_N)$ are the summation of the product of the adjoint integration result by $\mathbf{\Gamma}^T$ and $\mathbf{B}^T$ at each time step respectively. We now conclude that *the derivatives of the objective functional $J(\mathbf{x}_N)$ with respect to control parameters, including the initial condition $\mathbf{x}_0$, model parameter $\boldsymbol{\mu}$ and boundary parameter $\mathbf{b}$, are computed simultaneously by a single integration of the adjoint model backward in time.*

2.4. Objective functional with a time integration

It is straightforward to extend the previous formulations to cases where the objective functional involves time integration. In the time discretization case, the objective functional assumes the form

$$J = \sum_{j=1}^{N} J_j(\mathbf{x}_j). \tag{18}$$

Using manipulations similar to those in the derivation of (14), (15) and (16), we obtain

$$\nabla_{\mathbf{x}_0} J = \sum_{j=1}^{N} \mathbf{G}^T(t_j, t_0) J_j'(\mathbf{x}_j), \tag{19}$$

$$\nabla_{\boldsymbol{\mu}} J = \sum_{j=1}^{N} \sum_{m=1}^{j} \mathbf{\Gamma}_m^T \mathbf{G}^T(t_j, t_m) J'(\mathbf{x}_j), \tag{20}$$

$$\nabla_{\mathbf{b}} J(\mathbf{x}_N) = \sum_{j=1}^{N} \sum_{m=1}^{j} \mathbf{B}_m^T \mathbf{G}^T(t_j, t_m) J'(\mathbf{x}_j). \tag{21}$$

¿From the rule of linear superposition of solutions of linear equations, we can compute (19), (20) and (21) by the same procedure as that for computing (14), (15) and (16). The only extra operation required is that $J_j'(\mathbf{x}_j)$ be added to the result when the adjoint model is integrated to time t_j. We notice that changing objective functionals does not alter the adjoint model. That is, once the adjoint model is derived, it can be applied to various objective functionals with different $J'(\mathbf{x}_j)$, while $J'(\mathbf{x}_j)$ can be provided in an analytical form.

In the above derivation, the model parameter and boundary values have been purposely treated as being time dependent. The first purpose is to introduce a flexible method for developing adjoint models in the context of optimal flow control; the second is to explore the possibility of finding the optimal open-loop control over a limited time window, which leads to the desired control effect when the optimal control is extended beyond the time window. While a similar derivation procedure can be used to obtain formulations of the derivative of the objective functional with respect to time varying model parameters and boundary values, the algebraic manipulations become more involved. All derivatives can still be calculated by a single integration of the adjoint equation.

2.5. Development of adjoint models

Coding the adjoint model in the context of reverse-mode (as borne out in Eq.(23)) permits a simple implementation of the above algorithm directly at the level of the source code. Also, the gradient calculated thus is consistent with the original nonlinear model at the level of the source code, circumventing difficulties due to the dependence on computational schemes and due to complex boundary geometries .

Coding the adjoint model involves two basic steps. First, we derive the linearized version from the nonlinear forward model, which is simple and mechanical. Second, we derive the adjoint model from the linearized model. The basic idea in deriving the adjoint from the linearized model is straightforward. The linearized model can be viewed as a matrix operator $\mathbf{G}(t_N, t_0)$

$$\mathbf{G}(t_N, t_0) = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_M. \tag{22}$$

Then its adjoint $\mathbf{G}^T(t_N, t_0)$ is

$$\mathbf{G}^T(t_N, t_0) = \mathbf{A}_M^T \mathbf{A}_{M-1}^T \cdots \mathbf{A}_1^T. \tag{23}$$

At the code level, $\mathbf{A}_i$ is a code entity, including a do-loop in the FORTRAN programming language (or a while/for loop in the C programming language), a subroutine (function), or even a code line. Each code entity is actually represented by a matrix operation. Thus, coding the adjoint starts with the last entity in the linearized code, and for each entity, we just transpose its corresponding matrix.

The correctness of the adjoint and linearized model can be rigorously tested. When the linearized model is completely consistent with the nonlinear code, we can always find an appropriately small $\delta\mathbf{x}_0$ that satisfies

$$\mathbf{G}(t_N, t_0)\delta\mathbf{x}_0 = \mathbf{M}_{\boldsymbol{\mu}}(t_N, t_0)(\mathbf{x}_0 + \delta\mathbf{x}_0) - \mathbf{M}_{\boldsymbol{\mu}}(t_N, t_0)\mathbf{x}_0 \tag{24}$$

at half machine precision [26], as shown in section 4.2.

The correctness of the adjoint code can be verified by the relationship

$$< \mathbf{G}(t_N, t_0)\delta\mathbf{x}_0 \ , \ \mathbf{G}(t_N, t_0)\delta\mathbf{x}_0 >=< \mathbf{G}^T(t_N, t_0)\mathbf{G}(t_N, t_0)\delta\mathbf{x}_0 \ , \ \delta\mathbf{x}_0 > . \tag{25}$$

It holds for any $\mathbf{x}_0$ at the machine precision when the adjoint is error free. It is noteworthy that coding the adjoint model at the source code level facilitates the handling of both sophisticated time-stepping schemes and boundary conditions with complex geometries, while this task constitutes a challenge for other adjoint coding methods such as the one employed in [2].

2.6. Storage strategies

A direct consequence of the nonlinearity of the state equation is that $\mathbf{G}(t_N, t_0)$ is a function of the solution of the state equation. Thus, we need, in principle, to have access to the solution of the state equation at all points in space and time when solving the adjoint equation. For a large dimensional model and long time integration, appropriate storage strategies are crucial for rendering this methodology feasible. A simple way to by-pass this problem is to store the solution of the state equation on a disk outside the computer main memory. The totality of the solution of the state equation is written out when the nonlinear model is integrated forward in time, and it is read in when running the adjoint model backward in time.

When the dimension of the problem under question is extremely large, we can partly write out the solution of the state equation. This strategy was introduced by Berggren [2] and Griewank [14] and can be easily implemented in the context of the present adjoint model. We assume that the number of time steps N may be factored as $N = PQ$, where P and Q are positive integers. We then partition N steps into P batches, each consisting of Q steps. The implementation procedure is as follows. First, we run the nonlinear model, storing only the solution at time steps $0, Q, \cdots, (P-1)Q$, as well as the Q time steps in the last batch separately. Thus we store the solution of a total of $P + Q$ steps, rather than PQ. Then, the adjoint model is integrated from N to $N - Q$ backward. Then by rerunning the nonlinear model from $(P-2)Q$ to $(P-1)Q-1$, the totality of Q steps are stored, thus replacing the Q steps for the last batch. In this way, we can run the adjoint model to time t_0. This strategy results in a tremendous memory saving. We can obtain the minimum storage required for a given N, P and Q by minimizing $P + Q$, subject to $N = P + Q$. If $N^{1/2}$ is an integer, the minimum is $P = Q = N^{1/2}$. The total storage is then $2N^{1/2}$. The price for implementing this strategy is equivalent to the extra $1 - \frac{1}{P}$ computations of the state equation. For instance, when $P = 2$, this strategy saves half the storage, while incurring an additonal 50% in CPU time in computing the state equation.

3. THE NAVIER-STOKES EQUATIONS

3.1. Mathematical model

Let Ω denote the flow domain. The flowfield is described by the velocity vector (u, v) and the scalar pressure p and is obtained by solving the following momentum and mass conservation equations (in dimensionless form):

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial t} = \frac{1}{R_e}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) - \frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} \quad \text{in } \Omega, \tag{26}$$

$$\frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = \frac{1}{R_e}(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) - \frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} \quad \text{in } \Omega, \tag{27}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad \text{in } \Omega, \tag{28}$$

subject to the initial condition

$$(u, v)|_{t=0} = (u_0, v_0) \quad \text{in } \Omega. \tag{29}$$

The above equations are nondimensionalized using the cylinder diameter d, the free stream velocity U, $R_e = Ud/\nu$ is the Reynolds number.

We consider optimal control problems for two-dimensional flow around a circular cylinder in a channel. The geometry is represented in Fig. 1. The boundary conditions are specified as follows: In the upper and lower boundary (Γ_1) no-slip conditions are applied. The inflow condition at $x = 0$ (Γ_2) is

$$u = u_{in} \quad \text{and} \quad v = 0, \quad \text{on } \Gamma_2 \tag{30}$$

and the outflow (Γ_3) condition is

$$\frac{\partial u}{\partial x} \quad \text{and} \quad \frac{\partial v}{\partial x} = 0, \quad \text{on } \Gamma_3. \tag{31}$$

On the surface of the cylinder (Γ_4), injection and suction normal to the surface are allowed. The injection and suction are the control parameters in the present study. In this case, the boundary condition on the cylinder surface is

$$u = u_g \quad \text{and} \quad v = v_g \quad \text{on } \Gamma_4. \tag{32}$$

For the uncontrolled flow we use the no-slip boundary condition on the cylinder surface

$$u = 0 \quad \text{and} \quad v = 0 \quad \text{on } \Gamma_4. \tag{33}$$

3.2. Discretization scheme

The discretized analog of the Navier-Stokes equation has been described in Griebel et al. (1998). For example, for the boundary cell marked in Fig. 2, we need to specify $u_{i,j}$, $v_{i,j}$, $u_{i-1,j}$, and $v_{i,j-1}$. We set

$$u_{i,j} = y_a, \quad v_{i,j} = y_b, \quad \text{on } \Gamma_4. \tag{34}$$

These boundary control parameters consist in a vector $\mathbf{y}$, representing suction/injection at the boundary. When the injection/suction on this cell is not used as control parameters, we simply have

$$u_{i,j} = 0, \quad v_{i,j} = 0, \quad \text{on } \Gamma_4. \tag{35}$$

To guarantee that the tangential velocity component at the surface is zero, we then specify

$$u_{i-1,j} = -u_{i-1,j+1}, \quad v_{i,j-1} = -v_{i+1,j-1}. \quad \text{on } \Gamma_4. \tag{36}$$

A semi-implicit method is employed for the discretization in time, which is explicit in the convective terms and implicit in the pressure term. The time step is calculated as

$$\delta t = \tau \min(\frac{R_e}{2}(\frac{1}{\delta x^2} + \frac{1}{\delta y^2})^{-1}, \ \frac{\delta x}{u_{\max}}, \ \frac{\delta y}{v_{\max}}). \tag{37}$$

The factor $\tau \in [0, 1]$ is set here to 0.6.

3.3. Basic experiments

In the following experiments, the model parameters are specified as follows: The channel is 22.0 units in length and 4.1 units in width, with discretization step 0.1 unit; the cylinder measures 1.0 unit in diameter and is situated at a distance of 1.5 units from the left boundary and 1.6 units from the upper boundary (note that the center of the cylinder is not exactly at the middle of the channel); at the inflow boundary, the velocity is set to have a uniform value, i.e., $u = 2$ and $v = 0$. For this set of parameters and without boundary control, the flow pattern displays the expected evolution as the Reynolds number increases. For highly viscous fluids ($R_e < 40.0$), the flow practically remains attached. Figure 3 presents the stream function contours of the steady state at $R_e = 4.0$. At lower viscosities ($R_e > 40.0$), the flow becomes unsteady, and typical Karman vortex streets emerge. Figure 4 shows the evolution of the streaklines roughly over one period for $R_e = 80$.

4. BASIC TESTS OF THE BEHAVIOR OF THE ADJOINT MODEL

4.1. Choice of objective functionals

1. Flow tracking

If the controlled flow is required to be a steady laminar flow then the objective functional can be of tracking type. Let (u_d, v_d) be a prescribed desired velocity field. We want to control the flow so that (u, v) is "close" to (u_d, v_d). It is then natural to minimize the energy norm

$$J_F = \frac{1}{2}\int_{t_1}^{t_2}\int_{\Omega}(|u - u_d|^2 + |v - v_d|^2 d\Omega dt \tag{38}$$

2. Enstrophy minimization

We can expect to control the Karman vortex street by minimizing the enstrophy. The objective functional can be written as

$$J_Z = \frac{1}{2}\int_{t_1}^{t_2}\int_{\Omega}\zeta^2 d\Omega dt \tag{39}$$

where $\boldsymbol{\zeta}$ is the vorticity defined by

$$\boldsymbol{\zeta}(x,y) = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}. \tag{40}$$

3. Viscous drag minimization

An important objective in many applications is the minimization of drag (Gunzburger 1995). For the incompressible flow, the drag on a body can be computed from the integral of the dissipation function

$$J_E = \frac{\nu}{2}\int_{t_1}^{t_2}\int_{\Omega} |(\nabla \mathbf{U}) + (\nabla \mathbf{U})^T|^2 d\Omega dt, \tag{41}$$

where $\mathbf{U}$ is the velocity vector with components u and v, and ν is the kinematic viscosity of the fluid.

These objective functionals depend on variables including initial conditions, boundary condition parameters, and other model parameters, relevant to the problems being considered. At this stage, the optimal control problem can be mathematically described as that of minimizing one of these objective functionals, subject to the Navier-Stokes equations. Intuitively, minimizing these objective functionals requires certain constrained minimization algorithms. Thus, many optimal flow control studies have solved this optimization problem by applying the Lagrange multiplier rule. Then, one obtains an optimality system of partial differential equations whose solution provides optimal states and controls. We know that nonlinear constrained minimization algorithms are computationally more expensive relative to unconstrained ones as explained in the introduction. Also, unconstrained minimization algorithms are relatively well established and robust.

4.2. Performance of the adjoint model

It is crucial that the accuracy of the gradients calculated by the adjoint technique is at the level of machine precision. Inaccuracy may result from three sources: coding errors, non-differentiable functions being used in the original nonlinear model, and round-off errors.

A gradient test method has been well accepted in some disciplines (e.g., [26]). This method is based on the Taylor expansion. For a function $J(\mathbf{y})$ with a continuous second derivative, the Taylor expansion around $\mathbf{y}$ yields

$$J(\mathbf{y}+\delta\mathbf{y}) = J(\mathbf{y}) + J'(\mathbf{y})\delta\mathbf{y} + \frac{1}{2}J''(\boldsymbol{\xi})(\delta\mathbf{y})^2, \tag{42}$$

where $\boldsymbol{\xi}$ lies between $\mathbf{y}$ and $\mathbf{y}+\delta\mathbf{y}$.

Equation (42) may be written as

$$\frac{J(\mathbf{y}+\delta\mathbf{y}) - J(\mathbf{y})}{J'(\mathbf{y})\delta\mathbf{y}} = 1+\epsilon \tag{43}$$

where

$$\epsilon = \frac{J''(\boldsymbol{\xi})\delta\mathbf{y}}{2J'(\mathbf{y})} \tag{44}$$

When the adjoint method provides exact gradients, ϵ tends to zero as $\delta\mathbf{y}$ decreases. As shown by Li and Navon [26], for double precision (15-digit precision), the adjoint model can be assumed to be accurate when a value of ϵ on the order between 10^{-7} and 10^{-8} is reached. We performed this gradient test with the three previously defined objective functionals. The initial condition is the flow at Reynolds number 80 evolving for 20 time units as shown in Fig. 5. Random perturbations are added as boundary parameters. The magnitude of the perturbations is changed by multiplying by a small constant α that varies between 10^{-8} and 1.0. The time window is set to be 1.5 units. Note that $1+\epsilon$ tends to 1 as the magnitude of the perturbation decreases thereby showing consistency. When the perturbation is too small, round-off errors become dominant as is evident in the last 2 rows in the Table I.

Table I. Results of the gradient test for the three objective functionals. $1+\epsilon$ is defined in (43). α is the amplitude of random perturbations of boundary parameters.

α	J_F	J_Z	J_E
10^0	0.82832737	1.08443962	0.80433642
10^{-3}	0.99832118	1.00022408	0.99999749
10^{-4}	0.99980476	0.99991713	0.99997052
10^{-5}	0.99998447	0.99999130	0.99999730
10^{-6}	0.99999849	0.99999981	0.99999980
10^{-7}	0.99999979	0.99999910	1.00000069
10^{-8}	0.99999919	0.99995110	1.00002360

4.3. Computational efficiency of the adjoint model

In section 2, we have generally shown that a single adjoint model integration produces the gradients of the objective functional with respect to all control variables (including the initial conditions, boundary condition parameters, etc.). As for the efficiency of the adjoint technique, the experiments show that the CPU time required by a single adjoint model integration on average is 1.3 times that required by a single original nonlinear model integration. From our experience with a general circulation model (GCM) of the atmosphere which is based on the complete three-dimensional Navier-Stokes equations, this ratio of the CPU time between the adjoint model and original nonlinear model is typical.

5. A QUASI-NEWTON METHOD

Optimal flow control problems solved as unconstrained minimization problems require, in addition to the gradient of the objective functionals, the Hessian matrix that consists of the second order derivatives of the objective functionals. For problems associated with the time-dependent Navier-Stokes equations, the direct computation of the Hessian would be prohibitively expensive both in terms of storage and CPU time. Therefore, the present minimization algorithm employs a quasi-Newton (or secant Newton) method. In this algorithm, the approximated Hessian is computed and updated by using the updated information of descent directions gathered from minimization iterations. The Hessian is constructed by the Davidon-Fletcher-Powell (DFP) quasi-Newton formulation (Vetterling et al. 1993). After obtaining the Newton step $\mathbf{p}$, we update the solution

$$\mathbf{y}_{new} = \mathbf{y}_{old} + \lambda\mathbf{p}, \quad 0 < \lambda \leq 1 \tag{45}$$

The aim of the line search is to find a step size λ so that $J(\mathbf{y}_{old} + \lambda\mathbf{p})$ has decreased sufficiently. Here, $\mathbf{y}$ consists of control parameters and $J(\mathbf{y})$ is any objective functional. The basic criterion for acceptance is

$$J(\mathbf{y}_{new}) \leq J(\mathbf{y}_{old}) + \alpha\nabla J(\mathbf{y}_{new} - \mathbf{y}_{old}), \tag{46}$$

$0 < \alpha < 1$. A value of $\alpha = 10^{-4}$ is usually used. Since the Newton direction is a descent direction, we are guaranteed to decrease $J(\mathbf{y})$ for a sufficiently small λ. A proper choice of λ is based on the norm of the gradients and the permissible range of parameters. Define

$$g(\lambda) \equiv J(\mathbf{y}_{old} + \lambda\mathbf{p}) \tag{47}$$

so that

$$g'(\lambda) = \nabla J \cdot \mathbf{p} \tag{48}$$

The first step λ_1 is chosen as the root of the quadratic,

$$g(\lambda_1) = g(0) + g'(0)\lambda_1 + \frac{1}{2}g''(0)\lambda_1^2 \tag{49}$$

where $g(0)$ and $g'(0)$ are known. Then the next is obtained as the minimum of

$$g(\lambda) = \frac{[g(\lambda_1) - g(0) - g'(0)\lambda_1]}{\lambda_1^2}\lambda^2 + g'(0)\lambda + g(0) \tag{50}$$

which is

$$\lambda = -\frac{g'(0)\lambda_1^2}{[g(\lambda_1) - g(0) - g'(0)\lambda_1]} \tag{51}$$

Since quasi-Newton algorithms still require storing the Hessian matrix, this algorithm is not computationally feasible if the dimension of control problem exceeds a few hundred parameters. In such a case, limited-memory quasi-Newton algorithms [27, 31] and truncated Newton algorithms (e.g., [30, 35]) are suitable. These algorithms require only the storage of a few additional vectors consisting of control parameters, but they constitute only an approximation to quasi-Newton algorithms.

6. NUMERICAL EXPERIMENTS ON VORTEX SHEDDING CONTROL

We conducted various control experiments within the framework of the unsteady Navier-Stokes equations to control vortex shedding in flows past circular cylinders. By optimizing boundary control parameters, we were able to suppress vortex shedding by open-loop control. We employed a regularized objective functional

$$J_{FR} = J_F + \frac{\eta}{2}|\mathbf{y}|^2 = J_F + \eta\Sigma \tag{52}$$

where J_F is defined by (38) for the tracking problem, and η is the regularization parameter (a dimensionless constant), $\mathbf{y}$ is an M-dimensional control-parameter vector (which corresponds to boundary parameters), and Σ is a stabilizing function (which is discussed in a following section).

The desired flow is an unseparated flow similar to the steady flow at low Reynolds number. Here, the time interval is taken to be 1.0 (Case I) and 3.0 (Case II) units, and η is set to 50.0 and 150.0 respectively. The unconstrained minimization process uses the quasi-Newton algorithm described in section 5. The minimization convergence criterion requires that the norm of the gradient scaled by its initial norm should be smaller than 10^{-5}. For both cases, the optimal control methodology yields suction on the major portion of the rear side of the cylinder. That these optimal boundary parameters do suppress the Karman vortex shedding is evident in Fig. 5, which presents the time evolution of the streak lines for the controlled flow.

6.1. Sensitivity to initial conditions

We consider two initial conditions: IC1, the steady state at $Re = 4$ as depicted in Fig. 3; and IC2, the Karman vortex street (the uncontrolled flow evolved over a time of 20 units). Table II lists the values of all the 18 optimal parameters for these two initial conditions, when the time window is 3.0 units. The distribution of boundary parameters is qualitatively and quantitatively similar. This indicates that the optimal boundary parameters is basically insensitive to the initial conditions.

Table II Comparison of optimal boundary parameters

Parameters	1	2	3	4	5	6	7	8	9
IC1	0.000	-0.514	-0.401	-0.446	-0.658	0.552	-0.938	-0.722	-0.773
IC2	0.104	-0.491	-0.512	-0.456	-0.668	0.506	-1.102	-0.911	-0.850

Parameters	10	11	12	13	14	15	16	17	18
IC1	-0.049	1.242	2.208	-0.801	-0.569	-0.429	-0.352	0.116	-0.326
IC2	0.002	1.381	2.602	-0.893	-0.546	-0.409	-0.316	-0.008	-0.209

Table III lists the 18 optimal parameters for the same initial conditions, but for a time window of 1 unit. Substantial differences may be noted between the two sets of parameters, implying thereby sensitivity to initial condition when the time window is relatively small.

Table III Comparison of optimal boundary parameters. The time interval is 1 time unit.

Parameters	1	2	3	4	5	6	7	8	9
IC1	0.265	-0.829	-0.172	0.042	-0.857	0.056	-1.188	-0.759	-0.508
IC2	-0.295	-0.563	0.165	-0.335	-0.622	0.366	-0.720	-0.134	-0.086
Parameters	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
IC1	-0.139	1.051	1.942	-0.516	-0.158	-0.172	-0.126	0.327	0.245
IC2	-0.178	0.900	1.545	-0.214	0.858	-0.013	-0.045	0.228	0.065

6.2. Time window for open-loop optimal control

We conducted several experiments to determine further the effect of the length of time interval. For the case of $Re = 80$ and initial condition IC1, we set the regularization parameter η to 50 for the time window of 1.0 unit, 100 for the time window of 2.0 units, 150 for the time window of 3.0 units, and 200 for the time window of 4.0 units, respectively. Table IV presents the optimal boundary parameters for these cases. It is found that larger regularization parameters tend to yield smaller values for a given control window. The values of the optimal parameters basically become larger as the length of the time window increases. The above-mentioned four cases evolve into a laminar steady flow. However, we have observed differences among these cases when integrating the controlled model starting from various initial conditions. When the initial condition is the steady state for $Re = 4$, the evolution to the final steady state varies. For the larger time window cases, the evolution time is usually shorter. It is interesting to note that some initial conditions do not evolve to a steady state for the optimized parameters for the time window of 1 or 2 units. As an example, Fig. 6 illustrates the streak lines at time 18 units, where the optimized parameters are from the time window of 1.0 unit, and the initial condition is the uncontrolled flow at the time of 20 time units as depicted in Fig. 4. Although no strong vortex shedding occurs, we can observe wave oscillations. On the contrary, irrespective of the initial conditions, the flow evolved to a steady state for optimized parameters corresponding to the cases of time windows of length 3.0 and 4.0 units. Further, the optimal boundary parameters suppress Karman vortex shedding even if the prescribed model parameters are slightly altered. We changed the inflow from a uniform profile with velocity of 2 units to a parabolic profile with the maximum velocity 2.5 located at the center of the channel. The Reynolds number also was increased from 80 to 100. When we apply the optimal boundary parameters as obtained above, vortex shedding does not occur. It is observed that for $Re = 80$ vortex shedding occurs at 2.2 units starting from initial condition as provided in Fig. 4. The result suggests that for a time window longer than the Karman vortex shedding period, we can expect the vortex shedding to be controlled in a robust fashion. When the time window is smaller than the shedding period, it is also possible to suppress Karman vortex shedding for some initial conditions, but the robustness is not guaranteed.

6.3. Convergence of minimizations and computational expense

The computational expense of solving optimal flow control problems is essentially determined by the length of the time window and the convergence rate of the minimization process. We address here these issues. The quasi-Newton algorithm based on the PDF formulation performs well in most cases when the objective functionals are appropriately regularized. Usually, we can obtain a satisfactory optimal solution after about 60 iterations. Figure 7 displays the evolution of the objective functional and the norm of its gradient scaled by its initial value versus the number of iterations for the time windows of 1.0 and 3.0 units, the regularization parameter being 50 and

Table IV Optimal parameters for the time windows of 1.0, 2.0, 3.0 and 4.0 time units respectively. η is the regularization parameter.

Window	T=1	T=2	T=3	T=4
η	50	100	150	200
1	-0.295	-0.001	0.104	0.325
2	-0.563	-0.658	-0.491	-0.442
3	0.165	-0.522	-0.512	-0.449
4	-0.335	-0.516	-0.456	-0.571
5	-0.622	-0.807	-0.668	-0.636
6	0.366	0.502	0.506	0.629
7	-0.720	-1.058	-1.102	-1.117
8	-0.134	-0.528	-0.911	-1.240
9	-0.086	-0.607	-0.850	-1.017
10	-0.179	-0.100	0.023	0.124
11	0.900	1.119	1.381	1.488
12	1.545	2.610	2.602	2.924
13	-0.214	-0.716	-0.893	-1.045
14	0.858	-0.167	-0.546	-0.663
15	-0.013	-0.229	-0.409	-0.528
16	-0.045	-0.296	-0.316	-0.302
17	0.228	0.141	0.084	-0.235
18	0.065	-0.084	-0.209	0.056

150 respectively. For the case of the time window of 3.0 units, the minimization satisfies the convergence criterion at iteration 67 (after 70 objective function computations). The CPU time required on a workstation (SGI/INDIGO) is about 15.0 hours. The case of the time window of 1.0 units requires 61 iterations (62 objective function computations) and the CPU time required is 5.0 hours. The number of iterations required for attaining full convergence depends on the value of the regularization parameter. Generally, the larger the regularization parameter, the fewer iterations are required. When the regularization parameter is too small, the minimization may not even converge. In all the experiments, the major decrease occurs in less than 10 iterations, independent of the value of the regularization parameter. A similar performance of the minimization has also been observed by Berggren (1998). We take the optimal boundary parameters which are obtained after 10 iterations, and employ them to control the flow successfully (Fig. 8).

7. ILL-POSEDNESS AND REGULARIZATION

Let us consider any objective functional J, which represents one of the three objective functionals defined in (38), (39), and (41) respectively. We minimize J with respect to, say, the boundary values y (see section 3). $\mathbf{y}$ is a K-dimensional vector with range in an K-dimensional metric space Y. We seek a $\mathbf{y}$ from an admissible class Y_a in the space Y, such that it minimizes J. The ill-posedness of the minimization can then be mathematically defined as follows. For any $\epsilon > 0$, we assume there exists $\mathbf{y}_1$ such that

$$J_{\mathbf{y}_1} \leq J_{\mathbf{y}_0} + \epsilon. \tag{53}$$

For a given ϵ sufficiently small, the difference $\mathbf{y}_1 - \mathbf{y}_0$ may assume arbitrarily large values. The minimization in such a case is said to be ill-posed [37]. We investigate the behavior of the minimization process of the objective functional without the regularization term. The control window is 3 units. Figure 9 presents the evolution of the objective functional versus the number of iterations.

The minimization process stops at iteration 47 when it can not find a step size that satisfies (46). The objective functional undergoes rapid decrease during the first 10 iterations, and in fact it has decreased by 47% of its intial value. Beyond the initial 10 iterations, the total further decrease is less than 3%. However, the relevant solution undergoes tremendous changes. For example, while the objective functional J_F decreases by only 0.5% after 20 iterations from iteration 21 to 40, the change in the relevant solution is arbitrarily large (Table V). All three objective functionals are found to behave similarly.

Table V The difference between the 'optimal' solutions at iteration 21 and 40; control window 3 units.

Parameters	1	2	3	4	5	6	7	8	9
Difference	0.329	0.099	1.509	1.092	-0.178	0.678	-0.637	-0.340	-0.182
Parameters	10	11	12	13	14	15	16	17	18
Difference	0.121	1.120	0.012	-0.116	0.430	0.626	0.54	-0.075	0.529

This is an example of ill-posedness as defined by Tikhonov and Arsenin [37]. A regularization term is introduced following [37].

$$J_R = J + \eta\Sigma \tag{54}$$

where $\eta > 0$ is a regularization parameter. Σ is called a stabilizing functional. In Tikhonov and Arsenin [37], the exact definition and required properties of a stabilizing functional are given. The stabilizing functional is not unique, and can assume various forms [20]. Also, the determination of the regularization parameter η is not necessarily straightforward.

8. CONCLUSIONS

The optimal control technique is attractive for flow control problems as it does not require 'a priori' knowledge of the flow characteristics. However, it is complex and is very demanding computationally. The approach proposed in this paper addresses these issues to some degree. The numerous experiments for controlling vortex shedding in the wake of a circular cylinder, conducted in the present investigation, bring out the particular merits of the approach. We plan to apply this approach to more sophisticated optimal flow control problems in future.

Questions regarding controllability and the existence of solutions are not discussed in this paper. These issues remain essentially open for the Navier-Stokes equations, particularly in three dimensions, although some advances have been made recently (e.g., [8, 9, 19]). Due to the nonlinearity of the state equation, uniqueness of solutions may not be guaranteed even when the optimal solution exists. From the results presented above, we argue that optimal flow control is feasible and provides solutions of practical usefulness when used in conjunction with regularized objective functionals even prior to addressing the basic questions of controllability, existence and uniqueness of solutions.

We observe that computing the solution of optimal control problems to full optimality is not necessary in practical applications. The numerical experiments presented above indicate that a substantial reduction of the objective functionl may be obtained after performing only a modest number of iterations. The optimal boundary values thus obtained completely suppress vortex shedding after 10 iterations. The minimization process is therefore terminated after 10 iterations, and the CPU time required is just a few hours on SGI/INDIGO workstation.

A key result of the current study is that an open-loop control can suppress vortex shedding in the wake of a circular cylinder in a robust way. The optimal control applied over a limited time window can provide the desired control parameters whose effect extends beyond the extent of the control window. This implies that feedback control may not be necessary for this class of problems. From dynamical system theory (e.g., see [36]), this may possibly apply to other optimal control problems. It is konwn that a dissipative and forced dynamical system associated with the Navier-Stokes equations possesses a limited number of attractors [36, 25]. Each attractor has its attracting basin in phase space. When the major attractor of the system under the optimal control is the desired flow, any initial conditions in its attracting basin will evolve to the desired flow. In such a case, we need only find the optimal control associated with the desired attractor, and the control problem is thus independent of the initial conditions.

REFERENCES

1. F. Abergel and R. Temam, On some control problems in fluid mechanics, *Theoret. Comput. Fluid Dynamics* 1, 303 (1990) .
2. M. Berggren, Numerical solution of a flow-control problem: vorticity reduction by dynamic boundary action, *SIAM J. Sci. Comput.* 19, 829 (1998).
3. M. Berggren, R. Glowinski and J. L. Lions, A computational approach to controllability issues for flow related problems. (I) : Pointwise control of the viscous Burgers equation,*Internat. J. Comput. Fluid Dynamics* 6, 253 (1996).
4. T.R. Bewley, H. Choi, R. Temam, and P. Moin, Optimal feedback control of turbulent channel flow, *Annual Research Briefs-1993, Center for Turbulence Research, Stanford U./NASA Ames*, (1993), p. 12.
5. J.A. Burns and S. Kang, A Control problem for Burgers' equation with bounded input/output.*Tech Report 90-45 , ICASE , NASA Langley Research Center, Hampton, VA*, (1990)
6. H. Choi, R. Temam, P. Moin, and J. Kim, Feedback control for unsteady flow and its application to the stochastic Burgers equation, *J. Fluid Mech.* 253, 509 (1993).
7. M.C. Desai, K. Ito, Optimal control of Navier-Stokes equations, *SIAM J. Control Opt.* 32, 1428 (1994).
8. A.V. Fursikov, A.V., Exact boundary zero controllability of three-dimensional Navier-Stokes equations, *J. Dyn. Control System* 1, 325 (1995).
9. A.V. Fursikov, M.D. Gunzburger, and L.S. Hou, Boundary value problems and optimal boundary control for the Navier-Stokes system: the two-dimensional case, *SIAM J. Control Opt.* 36, 852 (1999).
10. O. Ghattas and J.-H. Bark, Optimal control of two- and three- dimensional Navier-Stokes flows, *J. Comput. Phys.* 136, 23 (1997).
11. R. Glowinski, Kearsley, T. Pan, and J. Periaux, Numerical simulation and optimal shape for viscous flow by a fictitious domain method, *Internat. J. Numer. Methods Fluids* 20, 695 (1995).

12. W.R. Graham, I. Peraire, and K.Y. Tang, Optimal control of vortex shedding using low order models, Part I: Open-loop model development, *Internat. J. Numer. Methods Eng.*, accepted.

13. M. Griebel, T. Dornseifer, and T. Neunhoeffer, *Numerical simulation in fluid dynamics: A practical introduction*, (SIAM, Philadelphia, 1998), p. 217.

14. A. Griewank, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, *Opt. Method and Software* 1, 35 (1990).

15. M.D. Gunzburger, *Flow Control* (Springer, New York, 1995), edited, p. 381.

16. M.D. Gunzburger, Sensitivities, adjoints and flow optimization. *Int. J. Numer. Mech Fluids* 31, 53 (1999).

17. M.D. Gunzburger, L.-S. Hou, and T.P. Svobodny, Analysis and finite element approximations of optimal control problems for the stationary Navier-Stokes equations with distributed and Neumann controls, *Math. Comp.* 57, 123 (1991).

18. M.D. Gunzburger, L.-S. Hou, and T.P. Svobodny, Boundary velocity control of incompressible flow with an application to viscous drag reduction, *SIAM J. Control Optim.* 30, 167-181 (1992).

19. M.D. Gunzburger and S. Manservisi, The velocity tracking problem for Navier-Stokes flows with bounded distributed control, *SIAM J. Control Optim* 37, 1913 (1999).

20. Hansen, P.C., *Rank-deficient and ill-posed problems: Numerical aspects of linear inversion* (SIAM, Philadelphia, 1998), P. 247.

21. L.S. Hou and S.S. Ravindran, Numerical approximation of optimal flow control problems by a penalty method: Error estimates and numerical results, *SIAM J. Sci. Comput* 20, 1753 (1999).

22. L.S. Hou and Y Yan, Dynamics and approximations of a velocity tracking problem for the Navier-Stokes flows with piecewise distributed controls, *SIAM J. Control Optim.* 35, 1847 (1997).

23. R.D. Joslin, M.D. Gunzburger, R.A. Nicolaides, G. Erlebacher, and M.Y. Hussaini, Self-contained automatic methodology for optimal flow control, *AIAA J.* 35, 816 (1997).

24. Y. Leredde, J.M. Lellouche, and I. Dekeyser, On initial, boundary conditions and viscosity coefficient control for Burgers' equation, *Internat. J. Numer. Methods Fluids* 28, 113 (1998).

25. J. P. Li and J.-F. Chou, Existence of the atmosphere attractor, *Sci. China* Ser. D42, 215 (1999).

26. Z. Li, and I. M. Navon, Adjoint Sensitivity of the Earth's Radiation Budget in the NCEP/MRF Model. *J. Geophy. Res.* 103 (D4), 3801 (1998).

27. D.C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Programming*, **45**, 503 (1989).

28. C. Min and H. Choi, Suboptimal feedback control of vortex shedding at low Reynolds numbers, *J. Fluid Mech.* 401, 123 (1999).

29. P. Moin and T.R. Bewley, Feedback control of turbulence, *Appl. Mech. Rev.* 47, part 2, S2 (1994).

30. S.G. Nash, Preconditioning of Truncated-Newton Methods, *SIAM J. Sci. Stat. Comput*, **6**, 599 (1985).

31. Nocedal, J., Updating quasi-Newton matrices with limited storage, *Math. of Computation* **35**, 773 (1980).

32. D.S. Park, D.M., Ladd, and E.W. Hendricks, Feedback control of von Karman vortex shedding behind a circular cylinder at low Reynolds numbers, *Phys. Fluid* 6, 2390 (1994).

33. J.S. Peterson, The reduced basis for incompressible viscous flow calculations, *SIAM J. Sci. Stat. Comput.* 10, 777 (1989).

34. K. Roussopoulos, K., Feedback control of vortex shedding at low Reynolds numbers, *J. Fluid Mech.* 248, 267 (1993).

35. T. Schlick and A. Fogelson, TNPACK - A truncated Newton package for large-scale problems: I. Algorithms and usage. *ACM Transactions on Mathematical Software*, 18, 46 (1992).

36. R. Temam, *Infinite-dimensional dynamical systems in mechanics and physics* (Springer-Verlag, New York. 1988), p. 500.

37. A.V. Tikhonov and V. Arsenin, 1977: *Solution of ill-posed problems* (Winston and Sons, Washington DC, 1977), p. 224.

38. S. Twomey *Introduction to the mathematics of inversion in remote sensing and indirect measurements* (Elsevier, Amsterdam, 1977), p. 243.

39. W.T. Vetterling, S. A. Teukolsky, and W. H. Press, *Numerical Recipes in C : The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1993), p. 325.

Figure Captions

Figure 1. The geometry of the computational domain Ω. Only the left half part is shown.

Figure 2. A schematic illustration of boundary cells and boundary values.

Figure 3. Streamfunction of the uncontrolled steady state for $R_e = 4.0$.

Figure 4. Evolution of streaklines during one vortex shedding period about 2.0 time units starting at the time of 17.8 time units. The Reynolds number is 80.0. The flow displays well developed Karman vortex street at the time 16.0 time units with the intial condition depicted in Figure 3.

Figure 5. Evolution of streak lines for the controlled flow.

Figure 6. Evolution of streak lines for the controlled flow. The optimized injection and suction at the surface of the cylinder are obtained by minimizing J_F, with the time window starting at $t = 0.$ with the state depicted in Figure 4. Then the model is integrated with the initial condition at $t = 20$ depicted in Figure 4.

Figure 7. Evolution of both the objective functional and its gradient norm with minimization iteration numbers. The objective functional J_F is used. The time window are 1.0 units (a) and 3.0 units (b).

Figure 8. Steady states of streak lines of the controlled flow with the optimal injection/suction obtained after 10 minimization iterations.

Figure 9. Same as in Fig. 7b, but without regularization. The minimization stops after 47 iterations since the minimization can not find a sufficient descent step leading to a sufficient decrease.