# Controllability Issues for Flow-Related Models: A Computational Approach

Martin Berggren
Roland Glowinski

December 1994

TR94-47

# Controllability Issues for Flow-Related Models: A Computational Approach

M. Berggren* R. Glowinski†

December 2, 1994

**Abstract**

We discuss the numerical solution of some controllability problems for time-dependent flow models. The emphasis is on algorithmic aspects, discretization issues, and memory-saving devices. Numerical results are presented for controllability studies involving the viscous Burgers equation, an advection-diffusion equation, and the unsteady Stokes equations.

## 1 Introduction

The main goal of this expository article is to discuss the numerical solution of flow-related controllability problems. Since these problems are quite difficult both from a mathematical and a computational points of view for general flow models, our strategy is to develop expertise first for simple models like the one-dimensional viscous Burgers equation (Section 2), a linear advection-diffusion equation (Section 3), and the unsteady Stokes equations (Section 4). The corresponding approximate controllability problems are solved by a combination of penalty techniques, finite element space approximations, finite difference time discretizations, and either direct or iterative methods for solving the discrete controllability problems.

It is our opinion that the models and methods discussed here are necessary preliminary steps to address more complicated and realistic flow problems, like those modelled, for example, by the Navier-Stokes equations for incompressible viscous flow. Indeed, despite their basic difficulties, flow control problems have motivated a large body of literature; concerning the problems considered in this article, let us mention [1], [3], [6], [7], [8], [9], [11], [13], [18], [23], [24], [25], [27], [28], [29], [36], [38], [39], [42], [43], [44], [46], [47], [51], [53]. The problems considered here are clearly related to some which have been treated in the above references. However, in the present work a particular attention has been given to time-dependent problems and to algorithmic aspects, including storage-saving devices (Appendix A) and direct solution methods, such as the one based on the Singular Value Decomposition (Appendix B).

*Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892.

†Department of Mathematics, University of Houston, Houston, TX 77204-3476.

# 2 Pointwise Control of the Burgers Equation

## 2.1 Generalities

*Nonlinear advection* and *viscous diffusion* are two important features of many flow models. The simplest equation with both these features is the *viscous Burgers equation*, namely

$$y_t - \nu y_{xx} + y y_x = f \qquad \text{in } (0,1)\times(0,T). \tag{2.1}$$

In (2.1), $y(x,t)$ can be interpreted as a *velocity* at a point $x$ and at time $t$; we shall assume from now on that

$$\{x,t\} \in (0,1)\times(0,T) = Q,$$

where $0 < T < +\infty$. In (2.1), $\nu > 0$ is a *viscosity parameter* and $f$ a density of *external forces*. Equation (2.1) can be used in the modeling of *weak shock waves* when the flow of interest is a perturbation of a uniform sonic gas flow (see [45]). Indeed, the simplicity of the Burgers equation has made it natural starting point in the investigation of flow control problems as shown by recent articles such as [1], [4], [7], [8], [13], [20]. This section is another contribution in that direction.

## 2.2 Problem Formulation

The kind of problems that will be discussed here in connection with equation (2.1) are *pointwise control problems*, where $m$ functions $v_m(t)$, $m = 1,\dots,M$ will "force" the equation at the points $a_m \in (0,1)$, $m = 1,\dots,M$. Completing Burgers equation with *initial* and *boundary conditions*, the *state equation* under consideration becomes

$$y_t - \nu y_{xx} + y y_x = f + \sum_{m=1}^{M} v_m \delta(x - a_m) \qquad \text{in } Q, \tag{2.2}_1$$

$$y_x(0,t) = 0, \quad y(1,t) = 0, \qquad t \in (0,T), \tag{2.2}_2$$

$$y(0) = y_0. \tag{2.2}_3$$

In (2.2), $x \mapsto \delta(x - a_m)$ denotes the *Dirac measure* at $a_m$ and $f$ now denotes a density of possible forcing in addition to the control.

A remark concerning notation: Throughout the rest of the paper, $y(t)$ will be used to denote the function $x \mapsto y(x,t)$. Also, the following notations will be used:

$$(y,z) = \int_0^1 y(x)z(x)\,dx, \qquad \forall y,z \in L^2(0,1),$$

$$||y||_{L^2(0,1)} = (y,y)^{1/2}, \qquad \forall y \in L^2(0,1),$$

$$||\mathbf{v}||_{L^2(0,T;\mathbb{R}^M)} = \left(\sum_{m=1}^{M}\int_0^T |v_m(t)|^2\,dt\right)^{1/2}, \quad \forall \mathbf{v} = \{v_m\}_{m=1}^M \in L^2(0,T;\mathbb{R}^M).$$

A *variational formulation* of the state equation (2.2) is given by

$$\begin{cases} y(t) \in V_0 \text{ a.e. on } (0,T);\ \forall z \in V_0 \text{ we have} \\ (y_t, z) + \nu\,(y_x, z_x) + (y_x y, z) \\ \qquad = (f, z) + \sum_{m=1}^{M} v_m\,(t)\, z\,(a_m) \qquad\qquad \text{a.e. on } (0,T), \end{cases} \tag{2.3}_1$$

$$y(0) = y_0, \tag{2.3}_2$$

where

$$V_0 = \left\{ z \mid z \in H^1\,(0,1), z(1) = 0 \right\}.$$

Given a *target function* $y_T (\in L^2(0,1))$, the aim is to find functions $\mathbf{v} = \{v_k\}_{k=1}^{M}$ such that $y(T)$ is close to $y_T$ at a minimal cost for the control function. To make this precise, we define the space of *admissible controls* $\mathscr{U}$ by

$$\mathscr{U} = L^2(0,T;\mathbb{R}^M)$$

Next, let the *cost function* $J$ be defined by

$$J(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|_{\mathscr{U}}^2 + \frac{k}{2}\|y(T) - y_T\|_{L^2(0,1)}^2, \tag{2.4}$$

where, in (2.4), $y$ is a function of $\mathbf{v}$ through (2.3) and $k$ a *"large" positive* parameter. Finally, the *control problem* that we consider is defined by

$$\begin{aligned} &\textit{Find } \mathbf{u} \in \mathscr{U} \textit{ such that} \\ &J(\mathbf{u}) \le J(\mathbf{v}), \quad \forall \mathbf{v} \in \mathscr{U}. \end{aligned} \tag{2.5}$$

In this way, the closeness to the target $y_T$ will be forced—in a least-squares sense—by *penalty*, with $k$ being the penalty parameter. The value of $k$ determines the relative importance between the cost of the control and the distance to the target.

The numerical solution of the control problem (2.5) was considered in [13] (and reported in [18]). There are some substantial differences between the methods discussed in the above references and those in this article. For example, we rely here on conjugate gradient algorithms instead of quasi-Newton's, and we use "more" explicit time discretization schemes. Another difference is that we also consider the case where the "supports" of the controllers, the $a_m$'s, are *unknown*.

In order to handle those cases when the $a_m$'s are unknown, we introduce the (convex) set

$$\tilde{\mathscr{U}} = \mathscr{U} \times (0,1)^M$$

and the augmented cost function $\tilde{J} : \tilde{\mathscr{U}} \to \mathbb{R}$ defined with obvious notation by

$$\tilde{J}(\mathbf{v}, \mathbf{b}) = \frac{1}{2}\|\mathbf{v}\|_{\mathscr{U}}^2 + \frac{k}{2}\|y(T) - y_T\|_{L^2(0,1)}^2 + \phi(\mathbf{b}), \tag{2.6}$$

where $\phi : \mathbb{R}^M \to \mathbb{R}^+$ is an auxiliary function whose relevance will be discussed below. The new control problem is defined then by

$$\begin{aligned} &\textit{Find } \{\mathbf{u}, \mathbf{a}\} \in \tilde{\mathscr{U}} \textit{ such that} \\ &\tilde{J}(\mathbf{u}, \mathbf{a}) \le \tilde{J}(\mathbf{v}, \mathbf{b}), \quad \forall \{\mathbf{v}, \mathbf{b}\} \in \tilde{\mathscr{U}}. \end{aligned} \tag{2.7}$$

**Remark 2.1:** We could have allowed some of the $b_m$'s to take the value 0; from the variational formulation $(2.3)_1$, this would have been equivalent to replace the homogeneous Neumann boundary condition $y_x(0,t) = 0$ in $(2.2)_2$ by

$$y_x(0,t) = \sum_{m \in \mathcal{J}} v_m(t),$$

where $\mathcal{J}$ is the subset of $\{1, \ldots, M\}$ consisting of those integers $m$ such that $b_m = 0$.

**Remark 2.2:** A crucial difference between problems (2.5) and (2.7) is that in the second case, the set $\tilde{\mathcal{U}}$ of the admissible controls is not a vector space. This can complicate the *iterative* solution of problem (2.7). However, since $\tilde{\mathcal{U}}$ is *convex*, it is fairly easy to choose $\phi$ in (2.6) so that $\mathbf{b}$ will stay in $(0,1)^M$; we shall return to that point in Section 2.6.

## 2.3 Gradient Calculations

Most minimization algorithms use some information on the gradient of the cost function. In the case of the functional $\tilde{J}$, an expression for the gradient of $\tilde{J}$ can be found easily using the *perturbation technique* described below.

Consider thus the solution $y$ to the state equation (2.2), (2.3) for a specific control $\{\mathbf{v}, \mathbf{b}\} \in \tilde{\mathcal{U}}$. Differentiating $\tilde{J}$ at $\{\mathbf{v}, \mathbf{b}\}$ with respect to a variation, $\{\delta\mathbf{v}, \delta\mathbf{b}\}$ of the control yields

$$\begin{aligned} \delta\tilde{J}(\mathbf{v},\mathbf{b}) &= \int_0^T \frac{\partial \tilde{J}}{\partial \mathbf{v}}(\mathbf{v},\mathbf{b}) \cdot \delta\mathbf{v}\, dt + \frac{\partial \tilde{J}}{\partial \mathbf{b}}(\mathbf{v},\mathbf{b}) \cdot \delta\mathbf{b} \\ &= \int_0^T \mathbf{v} \cdot \delta\mathbf{v}\, dt + k \int_0^1 (y(T) - y_T)\delta y(T)\, dx + \frac{d\phi}{d\mathbf{b}}(\mathbf{b}) \cdot \delta\mathbf{b}, \end{aligned} \tag{2.8}$$

where we have used the dot-product notation for the canonical scalar product in $\mathbb{R}^M$. Consider now (2.3) and $p \in L^2(0,T;V_0)$ such that $p_t \in L^2(0,T;V_0')$ ($V_0'$: dual space of $V_0$); we have then $p \in \mathscr{C}_0([0,T];L^2(0,1))$ [12, Ch. XVIII]. Taking $z = p(t)$ in (2.3) and integrating over $(0,T)$ we obtain

$$\begin{aligned} & y(0) = y_0, \\ & \left\{ \begin{aligned} & \int_0^T < y_t, p > \, dt + \nu \int_0^T dt \int_0^1 y_x p_x \, dx + \int_0^T dt \int_0^1 y_x y p \, dx \\ & \qquad = \int_0^T dt \int_0^1 f p \, dx + \sum_{m=1}^M \int_0^T p(b_m,t) v_m(t)\, dt, \end{aligned} \right. \end{aligned} \tag{2.9}$$

where $< \cdot, \cdot >$ denotes the *duality pairing* between $V_0'$ and $V_0$ (if $y_t$ is *smooth enough*, $< y_t, p >$ reduces to $\int_0^1 y_t p \, dx$).

Differentiating (2.9) yields

$$\begin{aligned} & \delta y(0) = 0, \\ & \left\{ \begin{aligned} & \int_0^T < \delta y_t, p > \, dt + \nu \int_0^T dt \int_0^1 \delta y_x p_x \, dx + \int_0^T dt \int_0^1 (y\delta y)_x p \, dx \\ & \qquad = \sum_{m=1}^M \int_0^T [p(b_m,t)\delta v_m(t) + p_x(b_m,t) v_m(t) \delta b_m] \, dt. \end{aligned} \right. \end{aligned} \tag{2.10}$$

Integrating by parts in time, and using the fact that

$$\delta y \in L^2(0,T;V_0) \cap \mathscr{C}^0([0,T];L^2(0,1)),$$
$$\delta y_t \in L^2(0,T;V_0'),$$

it follows from (2.10) that

$$\begin{aligned} &\int_0^1 p(T)\delta y(T)\,dx - \int_0^T < p_t, \delta y > \, dt \\ &\quad + \nu \int_0^T dt \int_0^1 p_x \delta y_x \, dx + \int_0^T dt \int_0^1 (y\delta y)_x p \, dx \\ &\quad = \sum_{m=1}^M \int_0^T [p(b_m,t)\delta v_m(t) + p_x(b_m,t)v_m(t)\delta b_m] \, dt. \end{aligned} \tag{2.11}$$

Let us assume now that $p$ is the solution of the following *adjoint equation*,

$$\begin{aligned} &p(T) = k(y_T - y(T)), \\ &- < p_t, z > + \nu \int_0^1 p_x z_x \, dx + \int_0^1 p(yz)_x \, dx = 0, \quad \forall z \in V_0, \text{ a.e. on } (0,T). \end{aligned} \tag{2.12}$$

Then (2.11) reduces to

$$\begin{aligned} &-k \int_0^1 (y(T) - y_T)\, \delta y(T)\, dx \\ &\quad = \int_0^T \left( \sum_{m=1}^M p(b_m,t)\delta v_m(t) \right) dt + \sum_{m=1}^M \left( \int_0^T p_x(b_m,t)v_m(t)\,dt \right) \delta b_m. \end{aligned} \tag{2.13}$$

Combining (2.13) and (2.8), we obtain

$$\begin{aligned} &\int_0^T \frac{\partial \tilde{J}}{\partial \mathbf{v}}(\mathbf{v},\mathbf{b}) \cdot \delta \mathbf{v}\, dt + \frac{\partial \tilde{J}}{\partial \mathbf{b}}(\mathbf{v},\mathbf{b}) \cdot \delta \mathbf{b} \\ &\quad = \int_0^T \left[ \sum_{m=1}^M (v_m(t) - p(b_m,t))\, \delta v_m(t) \right] dt \\ &\quad\quad + \sum_{m=1}^M \left[ \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \int_0^T p_x(b_m,t)v_m(t)\,dt \right] \delta b_m. \end{aligned} \tag{2.14}$$

Let us consider $\{\mathbf{w},\mathbf{c}\} \in \mathscr{U} \times \mathbb{R}$; it follows then from (2.14) that the derivative of $\tilde{J}$ at $\{\mathbf{v},\mathbf{b}\}$ in the $\{\mathbf{w},\mathbf{c}\}$-direction is given by

$$\begin{aligned} &\int_0^T \frac{\partial \tilde{J}}{\partial \mathbf{v}}(\mathbf{v},\mathbf{b}) \cdot \mathbf{w}\, dt + \frac{\partial \tilde{J}}{\partial \mathbf{b}}(\mathbf{v},\mathbf{b}) \cdot \mathbf{c} \\ &\quad = \int_0^T \left[ \sum_{m=1}^M (v_m(t) - p(b_m,t))\, w_m(t) \right] dt \\ &\quad\quad + \sum_{m=1}^M \left[ \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \int_0^T p_x(b_m,t)v_m(t)\,dt \right] c_m, \end{aligned} \tag{2.15}$$

where $p$ is obtained from $\{\mathbf{v},\mathbf{b}\}$ and $y$ via the solution of the adjoint equation (2.12).

**Remark 2.3:** The adjoint system (2.12) can also be written as

$$\begin{aligned} p(T) &= k(y_T - y(T)), \\ -p_t - \nu p_{xx} - y p_x &= 0 \qquad\qquad \text{in } Q, \\ \nu p_x(0,t) + p(0,t)y(0,t) = 0, \quad p(1,t) &= 0 \qquad \text{a.e. on } (0,T). \end{aligned} \tag{2.16}$$

**Remark 2.4:** One approach to solve the minimization problems (2.5) or (2.7) would be to directly discretize equations (2.2), (2.16) and then (2.4) (or (2.6)) and (2.15), and use the discrete cost function and gradient obtained in this manner in the minimization algorithm. However, quite large values of the parameter $k$ in the cost function is usually needed to closely approximate a given target function. This makes the minimization problem badly conditioned and an accurate expression for the gradient is needed. Unfortunately, there is no guarantee that a directly discretized adjoint equation will produce an accurate gradient of the *discrete* cost function. A proper (and safer) way to proceed is to first select discretizations of the state equation and of the cost function, and then derive the adjoint equation and the gradient associated with the discrete problem.

## 2.4 The Semi-Discrete Control Problem

*Time discretization* being considered first, we divide the time interval $(0,T)$ into $N$ subintervals of equal length $\Delta t = T/N$. We approximate then $\mathscr{U}$ by

$$\tilde{\mathscr{U}}^{\Delta t} = \mathbb{R}^{N\times M} \times (0,1)^M.$$

A typical element of $\tilde{\mathscr{U}}^{\Delta t}$ is $\{\mathbf{v}, \mathbf{b}\}$ with

$$\mathbf{v} = \left\{ \{v_m^n\}_{n=1}^N \right\}_{m=1}^M, \quad \mathbf{b} = \{b_m\}_{m=1}^M.$$

The semi-discrete analogue of problem (2.7) is then

$$\begin{aligned} &\textit{Find } \{\mathbf{u}^{\Delta t}, \mathbf{a}^{\Delta t}\} \in \tilde{\mathscr{U}}^{\Delta t} \textit{ such that} \\ &\tilde{J}^{\Delta t}(\mathbf{u}^{\Delta t}, \mathbf{a}^{\Delta t}) \le \tilde{J}^{\Delta t}(\mathbf{v}, \mathbf{b}), \qquad \forall \{\mathbf{v}, \mathbf{b}\} \in \tilde{\mathscr{U}}^{\Delta t}, \end{aligned} \tag{2.17}$$

where, in (2.17), the functional $\tilde{J}^{\Delta t}$ is defined by

$$\tilde{J}^{\Delta t}(\mathbf{v}, \mathbf{b}) = \frac{\Delta t}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} |v_m^n|^2 + \frac{k}{2} \|y^N - y_T\|^2_{L^2(0,1)} + \phi(\mathbf{b}), \tag{2.18}$$

with $y^N$ obtained from $\{\mathbf{v}, \mathbf{b}\}$ via the solution of the following semi-discrete state

equation:

$$y^0 = y_0;$$

*for* $n = 1, \ldots, N$, $y^n$ *is obtained from* $y^{n-1}$ *through the solution of the elliptic problem*

$$\begin{cases} y^n \in V_0;\ \forall z \in V_0 \text{ we have} \\ \displaystyle\int_0^1 \frac{y^n - y^{n-1}}{\Delta t} z\,dx + \nu \int_0^1 y_x^n z_x\,dx + \int_0^1 y_x^{n-1} y^{n-1} z\,dx \\ \qquad = \displaystyle\int_0^1 f^n z\,dx + \sum_{m=1}^{M} v_m^n z(b_m). \end{cases} \tag{2.19}$$

A few comments have to be made about the chosen discretization; we note in particular that the diffusion term is treated *implicitly* while the advection term is treated *explicitly*. This implies that the fully discrete analogue of the elliptic problem in (2.19) will be equivalent to a linear system associated with a matrix which is symmetric, positive definite and independent of $n$. This matrix will be *Cholesky factorized* once and for all, followed by just the solution of two triangular systems at each time step for the corresponding right-hand side. The semi-implicit nature of the time-stepping scheme implies a problem-dependent limit on the size of $\Delta t$, in particular for advection-dominated problems.

To compute the gradient of $\tilde{J}^{\Delta t}$, we can proceed as in Section 2.3. Differentiating (2.18), we obtain

$$\begin{aligned} \delta \tilde{J}^{\Delta t}(\mathbf{v}, \mathbf{b}) &= \frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{v}}(\mathbf{v}, \mathbf{b}) \cdot \delta \mathbf{v} + \frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{b}}(\mathbf{v}, \mathbf{b}) \cdot \delta \mathbf{b} \\ &= \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} v_m^n \delta v_m + k \int_0^1 (y^N - y_T) \delta y^N\,dx + \frac{d\phi}{d\mathbf{b}}(\mathbf{b}) \cdot \delta \mathbf{b}, \end{aligned} \tag{2.20}$$

and differentiating (2.19) yields

$$\delta y^0 = 0; \tag{2.21}_1$$

*and for* $n = 1, \ldots, N$

$$\begin{cases} \delta y^n \in V_0;\ \forall z \in V_0 \text{ we have} \\ \displaystyle\int_0^1 \frac{\delta y^n - \delta y^{n-1}}{\Delta t} z\,dx + \nu \int_0^1 \delta y_x^n z_x\,dx \\ \qquad + \displaystyle\int_0^1 \left(\delta y_x^{n-1} y^{n-1} + y_x^{n-1} \delta y^{n-1}\right) z\,dx \\ \qquad = \displaystyle\sum_{m=1}^{M} \left[\delta v_m^n z(b_m) + v_m^n z_x(b_m) \delta b_m\right]. \end{cases} \tag{2.21}_2$$

Consider now $\{p^n\}_{n=1}^N \subset V_0^N$; it follows then from $(2.21)_2$ that

$$\begin{aligned}
&\Delta t \sum_{n=1}^{N} \int_0^1 \frac{\delta y^n - \delta y^{n-1}}{\Delta t} p^n \, dx + \nu \Delta t \sum_{n=1}^{N} \int_0^1 \delta y_x^n, p_x^n \, dx \\
&\quad + \Delta t \sum_{n=1}^{N} \int_0^1 \left( \delta y_x^{n-1} y^{n-1} + y_x^{n-1} \delta y^{n-1} \right) p^n \, dx \\
&\quad = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ \delta v_m^n p^n(b_m) + v_m^n p_x^n(b_m) \delta b_m \right].
\end{aligned} \tag{2.22}$$

Some algebraic manipulation in the left-hand side of (2.22) yields

$$\begin{aligned}
&\int_0^1 p^{N+1} \delta y^N \, dx + \Delta t \sum_{n=1}^{N} \int_0^1 \frac{p^n - p^{n+1}}{\Delta t} \delta y^n \, dx + \nu \Delta t \sum_{n=1}^{N} \int_0^1 p_x^n \delta y_x^n \, dx \\
&\quad + \Delta t \sum_{n=1}^{N-1} \int_0^1 \left( y_x^n \delta y^n + y \delta y_x^n \right) p^{n+1} \, dx \\
&\quad = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ p^n(b_m) \delta v_m^n + v_m^n p_x^n(b_m) \delta b_m \right].
\end{aligned} \tag{2.23}$$

Suppose now that to $\{p^n\}_{n=1}^N$ we add $p^{N+1}$ so that $\{p^n\}_{n=1}^{N+1}$ satisfies

$$p^{N+1} = k(y_T - y^N) \tag{2.24}_1$$

$$\begin{cases} p^N \in V_0 \text{ such that, } \forall z \in V_0, \\ \displaystyle\int_0^1 \frac{p^N - p^{N+1}}{\Delta t} z \, dx + \nu \int_0^1 p_x^N z_x \, dx = 0, \end{cases} \tag{2.24}_2$$

and for $n = N-1, \ldots, 1$:

$$\begin{cases} p^n \in V_0 \text{ such that, } \forall z \in V_0, \\ \displaystyle\int_0^1 \frac{p^n - p^{n+1}}{\Delta t} z \, dx + \nu \int_0^1 p_x^n z_x \, dx + \int_0^1 p^{n+1} \left( y^n z \right)_x \, dx = 0, \end{cases} \tag{2.24}_3$$

Taking $z = \delta y^n$ in (2.24) for $n = 1, \ldots, N$, and combining with (2.23), we obtain

$$k \int_0^1 \left( y_T - y^N \right) \delta y^N \, dx = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ p^n(b_m) \delta v_m^n + v_m^n p_x^n(b_m) \delta b_m \right]. \tag{2.25}$$

Finally, combining (2.20) and (2.25), we obtain

$$\begin{aligned}
&\frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{v}}(\mathbf{v}, \mathbf{b}) \cdot \delta \mathbf{v} + \frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{b}}(\mathbf{v}, \mathbf{b}) \cdot \delta \mathbf{b} \\
&\quad = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ v_m^n - p^n(b_m) \right] \delta v_m^n + \sum_{m=1}^{M} \left[ \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \Delta t \sum_{n=1}^{N} v_m^n p_x^n(b_m) \right] \delta b_m.
\end{aligned}$$

We have thus proved that, $\forall \{\mathbf{v}, \mathbf{b}\}, \{\mathbf{w}, \mathbf{c}\} \in \tilde{\mathscr{U}}^{\Delta t}$,

$$\begin{aligned} &\frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{v}}(\mathbf{v}, \mathbf{b}) \cdot \mathbf{w} + \frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{b}}(\mathbf{v}, \mathbf{b}) \cdot \mathbf{c} \\ &\quad = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ v_m^n - p^n(b_m) \right] w_m^n + \sum_{m=1}^{M} \left[ \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \Delta t \sum_{n=1}^{N} v_m^n p_x^n(b_m) \right] c_m. \end{aligned} \tag{2.26}$$

If the control positions are *fixed*, (2.26) reduces to

$$\frac{\partial \tilde{J}^{\Delta t}}{\partial \mathbf{v}}(\mathbf{v}) \cdot \mathbf{w} = \Delta t \sum_{n=1}^{N} \sum_{m=1}^{M} \left[ v_m^n - p^n(b_m) \right] w_m^n, \quad \forall \mathbf{v}, \mathbf{w}.$$

**Remark 2.5:** We note that the initialization of the discrete adjoint equation (2.24) is made at time step $N+1$. This is typical of discrete adjoint equations and will be observed again in the following paragraphs of this article. We also observe that in (2.24), step $N$ is different from the other ones. This is typical of those situations where one term in the state equation is treated explicitly. Other observations are that the discretization of the adjoint equation reflects the one of the state equation, namely implicit treatment of the diffusion and explicit treatment of the advection. The fact that the discrete advective term contains $y^n$ is clearly a consequence of the nonlinearity. This does not cause any difficulty other than the necessity of storing $\{y^n\}_{n=1}^N$. At first glance, it seems like the values of $y^n$ have to be stored at *every* time step. Actually, as shown in Appendix A, storage requirements can be dramatically reduced at the cost of an extra solution of the state equation. This will be a critical issue for problems with multi-dimensional state equation.

**Remark 2.6:** If a *finite element* method is used for the *space approximation*, using piecewise-linear functions to approximate the space $V_0$ is a most natural choice. A difficulty with this choice is that the gradient $\partial \tilde{J}^{\Delta t}/\partial \mathbf{b}$ will be discontinuous, since derivatives of $p$ appears in $\partial \tilde{J}^{\Delta t}/\partial \mathbf{b}$ (see (2.26)). On the other hand, usual descent methods require the cost function to be at least $\mathscr{C}^1$. One obvious way to obtain a continuous $\partial \tilde{J}^{\Delta t}/\partial \mathbf{b}$ is to use $\mathscr{C}^1$ approximations of $p$. This would unnecessarily complicate the numerical methodology; a simpler method compatible with $\mathscr{C}^0$ approximation of $V_0$ will be discussed in Section 2.6.

## 2.5 The Discrete Control Problem (I): Fixed Control Positions

### 2.5.1 Space Discretization and Related Properties

For the space discretization, we introduce an integer $I$, then $h = 1/I$ and $x_i = ih$ for $i = 0, \ldots, I$; we denote by $K_i$ the interval $[x_{i-1}, x_i]$. We approximate then $V_0$ by

$$V_{0h} = \left\{ z \mid z \in \mathscr{C}^0[0,1],\, z(1) = 0,\, z|_{K_i} \in P_1,\, i = 1, \ldots, I \right\}, \tag{2.27}$$

where $P_1$ is the space of polynomials of degree $\leq 1$. As a vector basis for $V_{0h}$ we consider $\mathcal{B}_h = \{\varphi_i\}_{i=0}^I$ where $\varphi_i$ is the usual "hat" functions defined by

$$\varphi_i \in V_{0h},\, \varphi_i(x_j) = \delta_{ij},\, \forall j = 0, \ldots, I+1$$

($\delta_{ij}$ is the Kronecker delta[1]).

The discrete control problem considered in this section is

$$\begin{aligned} &\textit{Find } \mathbf{u}_h^{\Delta t} \in \mathbb{R}^{N\times M} \textit{ such that} \\ &J_h^k(\mathbf{u}_h^{\Delta t}) \le J_h^k(\mathbf{v}), \qquad \forall \mathbf{v} \in \mathbb{R}^{N\times M}, \end{aligned} \tag{2.28}$$

where

$$J_h^k(\mathbf{v}) = \frac{\Delta t}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}|v_m^n|^2 + \frac{k}{2}\|y_h^N - y_T\|^2_{L^2(0,1)}, \tag{2.29}$$

and where $y_h^N$ is obtained from the solution of the fully discretized state equation

$$\begin{cases} y_h^0 \in V_{0h}, \textit{ such that} \\ (y_h^0, \varphi_i) = (y_0, \varphi_i), \quad \forall i = 0, \dots, I; \end{cases} \tag{2.30}_1$$

*for $n = 1, \dots, N$, $y_h^n$ is obtained from $y_h^{n-1}$ through the solution of the discrete elliptic problem*

$$\begin{cases} y_h^n \in V_{0h};\ \forall i = 0, \dots, I, \textit{ we have} \\ (\dfrac{y_h^n - y_h^{n-1}}{\Delta t}, \varphi_i) + \nu(\dfrac{dy_h^n}{dx}, \dfrac{d\varphi_i}{dx}) + (y_h^{n-1}\dfrac{dy_h^{n-1}}{dx}, \varphi_i) \\ \qquad = (f^n, \varphi_i) + \displaystyle\sum_{m=1}^{M} v_m^n \varphi_i(a_m). \end{cases} \tag{2.30}_2$$

The fully discrete adjoint system corresponding to (2.29), (2.30) is given by

$$\begin{cases} p_h^{N+1} \in V_{0h}, \textit{ such that} \\ (p_h^{N+1}, \varphi_i) = k(y_T - y_h^N, \varphi_i), \quad \forall i = 0, \dots, I; \end{cases} \tag{2.31}_1$$

$$\begin{cases} p_h^N \in V_{0h};\ \forall i = 0, \dots, I, \textit{ we have} \\ (\dfrac{p_h^N - p_h^{N+1}}{\Delta t}, \varphi_i) + \nu(\dfrac{dp_h^N}{dx}, \dfrac{d\varphi_i}{dx}) = 0; \end{cases} \tag{2.31}_2$$

*for $n = N-1, \dots, 1$, we obtain $p_h^n$ through the solution of*

$$\begin{cases} p_h^n \in V_{0h};\ \forall i = 0, \dots, I, \textit{ we have} \\ (\dfrac{p_h^n - p_h^{n+1}}{k}, \varphi_i) + \nu(\dfrac{dp_h^n}{dx}, \dfrac{d\varphi_i}{dx}) + (\dfrac{dy_h^n}{dx}p_h^{n+1}, \varphi_i) \\ \qquad + (y_h^n p_h^{n+1}, \dfrac{d\varphi_i}{dx}) = 0. \end{cases} \tag{2.31}_3$$

Then the derivative of $J_h^{\Delta t}$ at $\mathbf{v}$ in the $\mathbf{w}$-direction will be

$$\frac{\partial J_h^{\Delta t}}{\partial \mathbf{v}}(\mathbf{v}) \cdot \mathbf{w} = \Delta t \sum_{n=1}^{N}\sum_{m=1}^{M}\left[v_m^n - p_h^n(a_m)\right] w_m^n. \tag{2.32}$$

---

[1] $\delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j \end{cases}$

### 2.5.2 Implementation Details

In the numerical experiments, the initial condition $y_0$, the target $y_T$, and the forcing term $f$, are, for simplicity, replaced by piecewise linear approximations, $y_{0h}$, $y_{Th}$, and $f_h$. This changes the first line in (2.30) and (2.31) to a simple identity. The state, $\{y_h^n\}$, and the adjoint state, $\{p_h^n\}$, are expanded in the basis $\mathcal{B}_h$, and the integrals obtained are computed exactly using the Simpson rule. This gives a system of equations with a constant, symmetric, positive definite, tridiagonal matrix on the left hand side. The Cholesky factorization of this matrix is done once and for all requiring the storage of a bidiagonal matrix. At each time step, the state is then computed by solving two bidiagonal linear systems.

The *Fletcher-Reeves* version of the conjugate gradient algorithm [16] is used to solve the minimization problem (2.28). (The *Polak-Ribière* version [41] was also tested but was found to be less efficient.) The unknowns are in $\mathbb{R}^{N\times M}$ and the scalar product is the canonical one. The algorithm can shortly be described as follows (see for instance Polak [41] for a more extensive discussion): Starting with an initial guess, $\mathbf{u}_0 \in \mathbb{R}^{N\times M}$, successive approximations of the solution to the minimization problem, are found by setting, for $m \geq 0$,

$$\mathbf{u}_{m+1} = \mathbf{u}_m - \rho_m \mathbf{w}_m.$$

The first *search direction*, $\mathbf{w}_0$, is the gradient of the cost function at step 0, $\mathbf{g}_0$. Search directions for $m > 0$, that is, $\mathbf{w}_m$, are found by linear combinations of the previous search direction, $\mathbf{w}_{m-1}$, and the present gradient, $\mathbf{g}_m$,

$$\mathbf{w}_m = \mathbf{g}_m + \gamma_m \mathbf{w}_{m-1},$$

where

$$\gamma_m = \frac{\mathbf{g}_m \cdot \mathbf{g}_m}{\mathbf{g}_{m-1} \cdot \mathbf{g}_{m-1}}.$$

(This choice of $\gamma_m$ constitutes the Fletcher-Reeves version.) The *step length*, $\rho_m$, should be chosen to minimize the function $\rho \mapsto J(\mathbf{u}^m - \rho \mathbf{w}^m)$. Since this *line search* problem is nonlinear, it is in itself a nontrivial and computationally expensive task. However, as has been shown by Al-Baali [2], the line search does not need to be done exactly for the conjugate gradient method to converge if certain conditions are satisfied. Here, a *cubic backtracking* strategy is used for the selection of the step length as described by Dennis and Schnabel [14, Ch. 6].

The computer programs were implemented in C with double precision IEEE arithmetic and all runs were performed on SUN Sparc 2 or Sparc 10 workstations. The stopping criterion in the conjugate gradient algorithm was

$$\frac{\mathbf{g}_n \cdot \mathbf{g}_n}{\mathbf{g}_0 \cdot \mathbf{g}_0} \leq \epsilon^2, \tag{2.33}$$

where $\epsilon = 10^{-5}$. Initial guess was $\mathbf{u}_0 = 0$.

### 2.5.3 Numerical Results

Here, the following test problem is considered:

$$T = 1, \quad I = 128, \quad N = 256, \quad \nu = 10^{-2}, \quad k = 8;$$

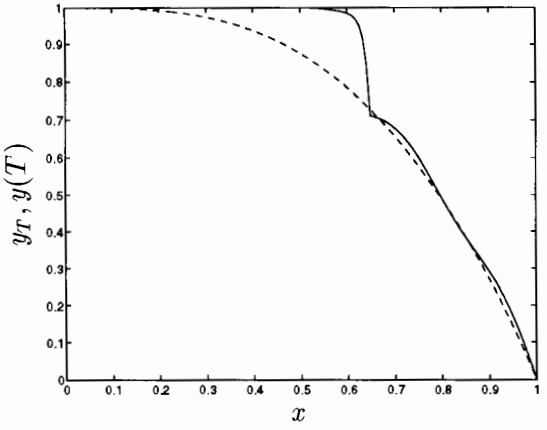


Figure 1: The target state (dashed) and the computed final state (solid) for $a = 2/3$; $\|y(T) - y_T\|/\|y_T\| = 0.091$, 47 iterations.

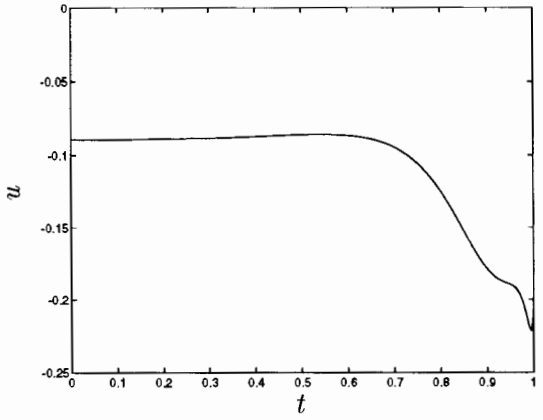


Figure 2: The computed optimal control for $a = 2/3$; $\|v\| = 0.11$.

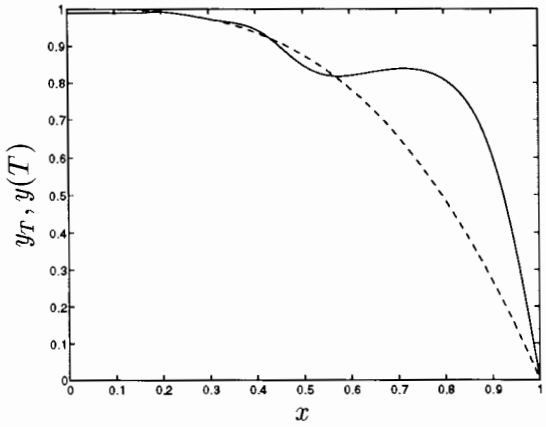


Figure 3: The target state (dashed) and the computed final state (solid) for $a = 1/5$; $\|y(T) - y_T\|/\|y_T\| = 0.20$, 89 iterations.

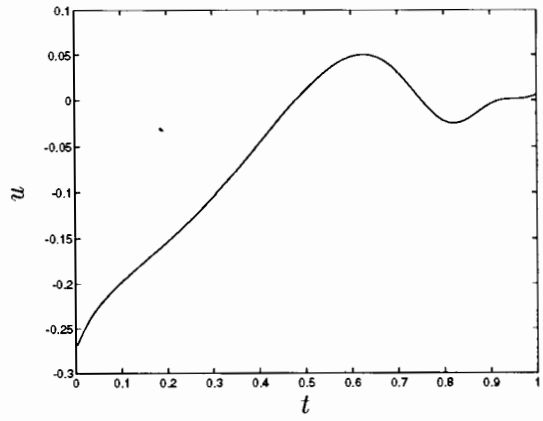


Figure 4: The computed optimal control for $a = 1/5$; $\|v\| = 0.11$.

$$
\begin{aligned}
f(x,t) &= \begin{cases} 1, & (x,t) \in (0,1/2) \times (0,T); \\ 2(1-x), & (x,t) \in [1/2,1) \times (0,T), \end{cases} \\
y(0) &= 0 \qquad\qquad x \in (0,1), \\
y_T &= 1 - x^3 \qquad\qquad x \in (0,1).
\end{aligned}
$$

This test problem was also considered by Dean and Gubernatis [13][2]. The target state satisfies the boundary conditions but is not necessarily reachable.

First, several runs were performed with a single control point at different control positions. Figures 1 and 3 show the target function, $y_T$, and the final state, $y(T)$, when $a = 2/3$ and $a = 1/5$ respectively[3], while Figures 2 and 4 show the computed

[2]Dean and Gubernatis used $I = N = 60$.

[3]To be precise: The control points were put on the grid points *nearest* to these values.

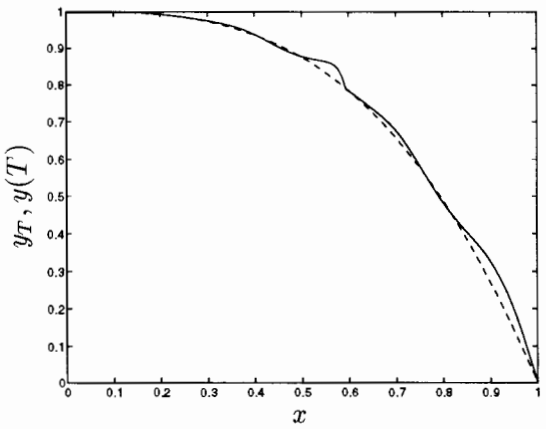


Figure 5: The target state (dashed) and the computed final state (solid) with two control points at $a_1 = 1/5$ and $a_2 = 3/5$; $\|y(T) - y_T\|/\|y_T\| = 0.025$, 86 iterations.

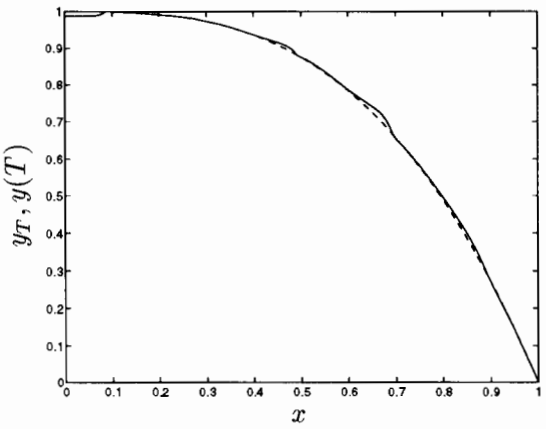


Figure 6: The target state (dashed) and the computed final state (solid) with five control points, $a_1 = 0.1, \ldots, a_5 = 0.9$, evenly spaced; $\|y(T) - y_T\|/\|y_T\| = 0.0085$, 82 iterations.

optimal control. The norms refers to the $L^2$-norm of the discrete entities. For $a = 2/3$, a good fit *downstream* (i.e., for larger $x$) from the control point can be noticed, while the solution seems to be close to uncontrollable upstream. The positive sign of the solution implies that the convection is directed towards increasing $x$'s, which is why it seems reasonable that the state is at least locally controllable in that direction. The only way of controlling the system upstream is through the diffusion term, which is small compared to the convection term for small $\nu$. For the case $a = 1/5$, there are clearly problems with controllability far downstream of the solution. (Recall that there is a distributed, uncontrolled forcing, $f$, which affects the solution.)

Figure 5 shows the target and the final state when two control points are used; one at $a = 1/5$, and one at $a = 3/5$. This gives a significantly better result. Figure 6 shows the result when five control points are used, placed at 0.1, 0.3, 0.5, 0.7, and 0.9, giving a very good fit. The dimension of the minimization problem is $N \times M$, but the computational time is not much dependent on the number of control points. For example, the CPU time (user time, SPARC 10) for the case of one control point at 1/5 was about 22 seconds compared to 27 seconds with five control points. Thus, the time-consuming part is the solution of the state and adjoint equation and not the manipulations of the control vectors.

### 2.5.4 Comparison with the Results Reported by Dean and Gubernatis

Dean and Gubernatis [13] report results for a run with a single control point at 3/5 (Figure 3 in [13]), and for a run with two control points, one at 1/5 and one at 2/3 (Figure 4 in [13]). The latter problem is solved with a relaxation technique, solving first for one control point at 3/5 and then, keeping that control fixed, solving for another control point at 1/5. The result for a single control point is very close to the one achieved here for $a = 2/3$ (compare Figure 1 with Figure 3 in [13]). For two control

points—compare Figure 5 with Figure 4 in [13]—the results look similar upstream of the second control point. There is some discrepancy downstream though, with somewhat better fit here. This is natural, since here the two controls are solved for simultaneously.

## 2.6 The Discrete Control Problem (II): Unknown Control Positions

As already mentioned in Section 2.4, there are some additional complications arising in the discrete case when the control positions are treated as unknown.

Recall (from Section 2.4) that the semi-discrete control problem is

$$\begin{aligned} &\textit{Find } \{\mathbf{u},\mathbf{a}\} \in \mathbb{R}^{N\times M} \times (0,1)^M \textit{ such that} \\ &J^{\Delta t}(\mathbf{u},\mathbf{a}) \le J^{\Delta t}(\mathbf{v},\mathbf{b}), \quad \forall\, \{\mathbf{v},\mathbf{b}\} \in \mathbb{R}^{N\times M} \times (0,1)^M, \end{aligned} \tag{2.34}$$

where

$$J^{\Delta t}(\mathbf{v},\mathbf{b}) = \frac{1}{2}\Delta t \sum_{n=1}^{N}\sum_{m=1}^{M} (v_m^n)^2 + \frac{k}{2}\|y^N - y_T\|^2_{L^2(0,1)} + \phi(\mathbf{b}). \tag{2.35}$$

The partial derivatives of $J^{\Delta t}$ with respect to the control position are, from (2.26),

$$\frac{\partial J^{\Delta t}}{\partial b_m}(\mathbf{v},\mathbf{b}) = \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \Delta t \sum_{n=1}^{N} v_m^n(t) p_x^n(b_m), \qquad m = 1,\ldots.M, \tag{2.36}$$

with $p$ given by (2.24). A problem with this gradient is, as pointed out before, that it will be discontinuous when the space approximation is done with piecewise linear elements. So, an expression for the gradient is needed that does not contain derivatives of the adjoint state. This is possible to obtain by systematic use of the fact that

$$\int_0^1 z(x)\delta(b-x)\,dx = z(b) = \lim_{H\downarrow 0} \frac{1}{H}\int_{b-H/2}^{b+H/2} z(x)\,dx.$$

The basic idea is to approximate the pointwise controls of the system with terms like the last one in the expression above—but with a finite $H$. That is, the control will be distributed over a segment of length $H$ instead of acting pointwise. Let $H > 0$ be given, and replace the state equation (2.19) with

$$y^0 = y_0; \tag{2.37}_1$$

*for* $n = 1,\ldots,N$, $y^n$ *is obtained from* $y^{n-1}$ *through the solution of the elliptic problem*

$$\begin{cases} y^n \in V_0;\ \forall z \in V_0 \textit{ we have} \\ \displaystyle\int_0^1 \frac{y^n - y^{n-1}}{\Delta t} z\,dx + \nu\int_0^1 y_x^n z_x\,dx + \int_0^1 y_x^{n-1} y^{n-1} z\,dx \\ \qquad\qquad = \displaystyle\int_0^1 f^n z\,dx + \sum_{m=1}^{M} \frac{v_m^n}{H}\int_{b_m-H/2}^{b_m+H/2} z(\xi)\,d\xi. \end{cases} \tag{2.37}_2$$

A straightforward calculation, analogous with the one in Section 2.4, yields

$$\begin{aligned} &m = 1, \ldots, M, \\ &\frac{\partial J^{\Delta t}}{\partial b_m}(\mathbf{v}, \mathbf{b}) = \frac{\partial \phi}{\partial b_m}(\mathbf{b}) - \frac{\Delta t}{H} \sum_{n=1}^{N} v_m^n \left[ p^n(b_m + H/2) - p^n(b_m - H/2) \right], \end{aligned} \tag{2.38}$$

where $\{p^n\}_{n=1}^{N+1}$ is the solution to the same adjoint equation as before, that is, equation (2.24). Comparing with (2.36), we note that $p_x^n(b_m)$ has been replaced by a difference approximation consistent with the approximated pointwise control.

Now perform a finite element approximation with piecewise linear functions. We get the fully discrete state equation by considering (2.37) with $V_0$ replaced by the space $V_{0h}$ defined as in (2.27), and with $(2.37)_1$ replaced by

$$\int_0^1 y^0 z \, dx = \int_0^1 y_0 z \, dx, \quad \forall z \in V_{0h}. \tag{2.37}'_1$$

The fully discrete state equation and the cost function (2.35) yield a discrete adjoint system defined as in (2.24) but with $V_0$ replaced by $V_{0h}$ and with $(2.24)_1$ replaced by

$$\int_0^1 p^{N+1} z \, dx = k \int_0^1 (y_T - y^N) z \, dx, \quad \forall z \in V_{0h}. \tag{2.24}'_1$$

As already discussed in Section 2.2, the reason for including the auxiliary function $\phi$ in the cost function is that the control positions may not stay inside the domain $(0, 1)$ during the iteration process. This function was chosen to be the $\mathscr{C}^1$ function

$$\phi(\mathbf{b}) = \sum_{m=1}^{M} \left\{ \theta(-b_m) b_m^2 + \theta(b_m - 1) \left[ 1 - b_m \right]^2 \right\}, \tag{2.39}$$

where $\theta$ is the Heaviside function[4]. To allow the positions $b_m$ to attain any real value, we extend the functions in $V_{0h}$ by zero outside of $(0, 1)$. Then for those $b_m \notin (-H/2, 1 + H/2)$, we have

$$\int_{b_m - H/2}^{b_m + H/2} z(\xi) \, d\xi = 0, \quad \forall z \in V_{0h},$$

and corresponding terms in the sum on the right-hand side of $(2.37)_2$ will thus be zero. The function $\phi$ has a nonzero derivative only outside of $[0, 1]$, and its contribution to the gradient of $J^{\Delta t}$ will act as a "soft wall" leading a temporary $b_m \notin (0, 1)$ back to the proper region as the iteration process continues. This means that the constraint $(0, 1)$ can be dropped in the minimization problem (2.34), giving the following problem to implement:

$$\begin{aligned} &\textit{Find } \{\mathbf{u}, \mathbf{a}\} \in \mathbb{R}^{N \times M} \times \mathbb{R}^M \textit{ such that} \\ &J^{\Delta t}(\mathbf{u}, \mathbf{a}) \leq J^{\Delta t}(\mathbf{v}, \mathbf{b}), \quad \forall \{\mathbf{v}, \mathbf{b}\} \in \mathbb{R}^{N \times M} \times \mathbb{R}^M, \end{aligned} \tag{2.40}$$

[4] $$\theta(x) = \begin{cases} 1 & x > 0, \\ 0 & x < 0. \end{cases}$$

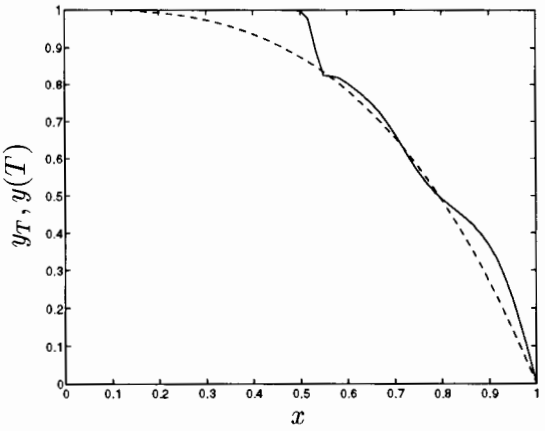


Figure 7: The target state, $y_T$, (dashed) and the computed final state, $y(T)$ (solid). Test problem from Section 2.5.3. Both position and control as unknown simultaneously; $\|y(T)-y_T\|/\|y_T\| = 0.062$; 312 iterations.

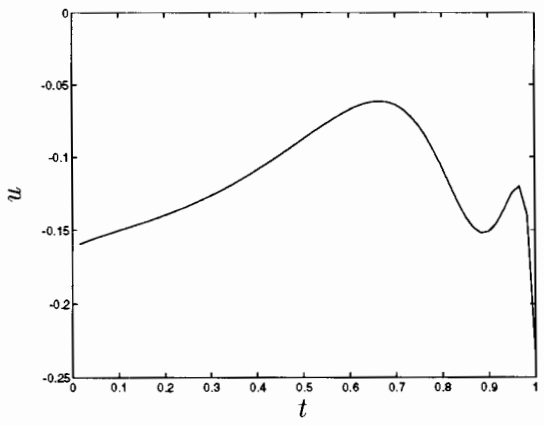


Figure 8: The computed optimal control. Both position and control as unknown simultaneously. Starting values: $u \equiv 0$, $a = 0.1$. Final values: $\|u\| = 0.12$; $a = 0.54$.

or if the controls, $\mathbf{v}$, are fixed:

$$\begin{aligned} &\textit{Find } \mathbf{a} \in \mathbb{R}^M \textit{ such that} \\ &J^{\Delta t}(\mathbf{a}) \leq J^{\Delta t}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^M. \end{aligned} \tag{2.41}$$

We discuss now some numerical results for unknown position and control. Again, consider the problem in Section 2.5.3. Recall that using a single control point at $a = 2/3$, it was hard to control the solution upstream of the control point (see Figure 1), while with $a = 1/5$ it was hard to control the solution close to the right boundary (see Figure 3). It seems natural that the optimal placement of the single control point should be somewhere in between these positions. To verify this, we solved the control problem with both the control and the position as unknown. The results[5] with the starting guess $u \equiv 0$, $a = 0.1$ are shown in Figures 7 and 8.

The optimal position turned out to be $a = 0.54$. Several experiments using different starting positions gave the same result. It should be pointed out that treating both control and position as unknown seems to lead to a more badly-conditioned problem than if one of them is fixed. The convergence rate was slow, and the convergence criterion had to be relaxed somewhat—above, $\epsilon = 5 \times 10^{-4}$ was used—but the run reported above still needed about 300 iterations to converge.

We considered also the following test problem in this context:

*Problem 2*

$$T = 1, \quad I = 60, \quad N = 120, \quad \nu = 10^{-2}, \quad k = 8;$$

[5] For all experiments in this section, $I = 60$ and $N = 120$ was used.

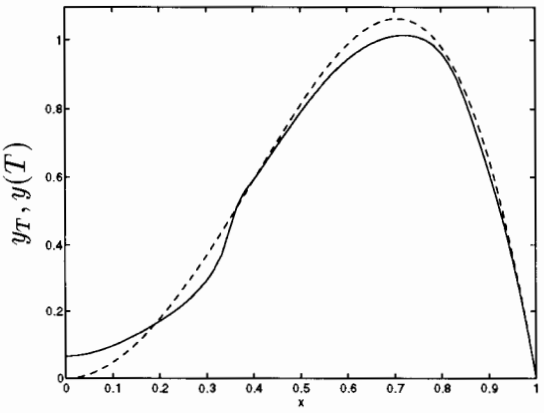


Figure 9: The target state, $y_T$, (dashed) and the computed final state, $y(T)$ (solid) for Test Problem 2. Both position and control as unknown simultaneously; $\|y(T) - y_T\|/\|y_T\| = 0.059$.

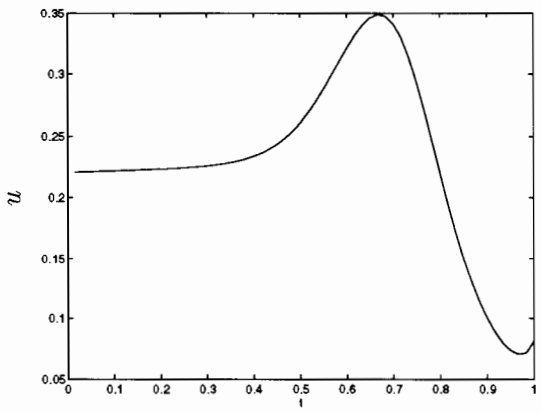


Figure 10: The computed optimal control. Both position and control as unknown simultaneously; Starting values: $u \equiv 0$, $a = 0.7$. Final values: $\|u\| = 0.24$, $a = 0.36$.

$$
\begin{aligned}
f(x,t) &= \begin{cases} x, & (x,t) \in (0,0.7) \times (0,T); \\ 7(1-x)/3, & (x,t) \in [0.7,1) \times (0,T), \end{cases} \\
y(0) &= 0 \qquad\qquad\qquad\quad x \in (0,1), \\
y_T &= 5(e^x - 1)x(1-x) \quad x \in (0,1).
\end{aligned}
$$

The results are given in Figures 9 and 10.

# 3 Boundary Control of a Linear Advection-Diffusion Problem

## 3.1 Generalities

Many physical phenomena can be viewed as a combination of a diffusion and an advection phenomenon. In the last section, pointwise control of a nonlinear advection-diffusion problem in one space dimension was considered. In this section, we consider a *boundary control problem* in two space dimension, but we will restrict ourselves to a linear state equation.

Let $\Omega \in \mathbb{R}^2$ be an open, bounded and connected region with sufficiently smooth boundary and let $0 < T < +\infty$. We assume that the boundary is partitioned into two

disjoint parts, $\Gamma_c$ and $\Gamma_1$. The state equation considered in this section is

$$\begin{aligned} \frac{\partial y}{\partial t} - \alpha \Delta y + \mathbf{V} \cdot \nabla y &= f \quad \text{in } \Omega \times (0,T), \\ y &= v \quad \text{on } \Gamma_c \times (0,T), \\ ay + \alpha \frac{\partial y}{\partial n} &= ag \quad \text{on } \Gamma_1 \times (0,T), \\ y(0) &= y_0, \end{aligned} \tag{3.1}$$

where $\mathbf{V} : \Omega \times (0,T) \to \mathbb{R}^2$ such that $\nabla \cdot \mathbf{V}(t) = 0$, $\alpha > 0$ is a constant, $a : \Gamma_1 \to \mathbb{R}$, and $\partial/\partial n$ denotes the outward normal derivative. A possible physical interpretation of the system (3.1) is given below.

Consider a flow of an incompressible Newtonian fluid that is *convection driven*, that is, the temperature effects on the mechanical properties of the flow can be neglected. This allows a decoupling of the relevant energy and flow equations, so that the velocity field of the fluid, $\mathbf{V}$, can be assumed to be known independently of the temperature field, $y$. Assume also that the heat flow is governed by Fourier's law,

$$\mathbf{q} = -\kappa \nabla y,$$

with a constant *thermoconductivity*, $\kappa > 0$. The constant $\alpha$ in the state equation is the *thermal diffusivity*, $\alpha = \kappa/\rho c$, where $\rho$ is the density of the fluid (assumed to be constant), and $c$ the specific heat per unit mass. The term $f$ represents the heat generated through viscous forces and will have the form

$$f = 2\beta \left( \nabla \mathbf{V} + (\nabla \mathbf{V})^T \right) \cdot \left( \nabla \mathbf{V} + (\nabla \mathbf{V})^T \right), \tag{3.2}$$

where $\beta = \mu/\rho c \geq 0$, and where $\mu$ is the *dynamic viscosity coefficient*. Under these assumptions, the temperature field of the fluid will satisfy the differential equation (3.1). Regarding the boundary conditions on $\Gamma_1$, we assume that the heat transfer on $\Gamma_1$ is governed by *Newton's law of external heat transfer*, that is,

$$\mathbf{n} \cdot \mathbf{q} = C(y - y_s),$$

where $\mathbf{n}$ is the unit outward normal of $\Gamma_1$, $y_s$ is the temperature of the surroundings, and $C$ is a constant (the *heat transfer number*). This, together with Fourier's law, yields a boundary condition like the one on $\Gamma_1$.

## 3.2 The Control Problem

Let us consider the following problem: Given a particular temperature distribution, $y_T$, in the domain $\Omega$, we want the temperature at time $t = T$ to approximate $y_T$, that is, we want $y(T) \approx y_T$. We consider the case when our means of control is the temperature as a function of space and time restricted to a part, $\Gamma_c$, of the boundary. As before, we will add the restriction that the a control should be of minimal cost in a norm sense. This *Dirichlet boundary control* problem may not be the most natural in the interpretation of equation (3.1) that is given in Section 3.1. Controlling the flux at the boundary may be physically more reasonable. However, Dirichlet boundary control is the most

natural in similar, but more complicated cases like the incompressible Navier-Stokes equations. Since this form of control gives rise to some regularity complications (cf. below), it might be fruitful to first consider this relatively simple case.

To find a mathematical formulation of the problem above, let $y_T$ be a target function defined in $\Omega$. Consider a cost function of the general form

$$J(v) = \frac{1}{2}\|v\|_{\mathscr{U}}^2 + \frac{k}{2}\|y(T) - y_T\|_K^2, \tag{3.3}$$

where $k > 0$ is a *penalty parameter* which, as in the cost function (2.4), balances the need for a cheap cost and a close approximation of the target function. The control problem considered here is

$$\begin{aligned} &\textit{Find } u \in \mathscr{U} \textit{ such that} \\ &J(u) \leq J(v), \qquad \forall v \in \mathscr{U}. \end{aligned} \tag{3.4}$$

To make this precise, the space of admissible controls, $\mathscr{U}$, and the space in which the error will be measured, $K$, has to be specified. There is a certain freedom in the choice of those spaces, but as a minimal requirement, we need to choose spaces so that the functional $J$ is differentiable with respect to the control. This is the case if $\mathscr{U}$ and $K$ are chosen so that, for each $v \in \mathscr{U}$, the function $t \mapsto y(t)$ is continuous in $K$. To avoid having to evaluate derivatives on $\Gamma_c$, a natural choice for the control space is $\mathscr{U} = L^2((0,T) \times \Gamma_c)$. Unfortunately, by standard regularity results (see e.g. [32]), the function $t \mapsto y(t)$ will not be continuous in $K = L^2(\Omega)$ for this choice of control space. However, by defining

$$V_0 = \{z | z \in H^1(\Omega) : z|_{\Gamma_c} = 0\}, \tag{3.5}$$

we have[6], $y \in \mathscr{C}^0([0,T]; V_0'(\Omega))$, where $V_0'$ is the dual space of $V_0$. (How to evaluate the dual norm is shown below.)

Choosing $\mathscr{U} = L^2((0,T) \times \Gamma_c)$ and $K = V_0'$, we obtain the cost function

$$J(v) = \frac{1}{2}\int_0^T \int_{\Gamma_c} v^2 \, d\Gamma \, dt + \frac{k}{2}\int_\Omega |\nabla\phi|^2 \, dx, \tag{3.6}$$

where, for $y_T \in V_0'$, $\phi$ is the solution to the following Poisson's equation:

$$\begin{aligned} -\Delta\phi &= y(T) - y_T \quad &&\textit{in } \Omega, \\ \phi &= 0 &&\textit{on } \Gamma_c, \\ \frac{\partial\phi}{\partial n} &= 0 &&\textit{on } \Gamma_1. \end{aligned} \tag{3.7}$$

The mapping $(y(T) - y_T) \to \phi$ may be viewed as a *smoothing*; the $V_0'$-norm will be less sensitive to highly oscillatory components of the error than if, for instance, a $L^2$-norm would have been applicable and used instead. This smoothing makes it possible to assume very little smoothness of the control and still get a well-defined expression of the residual $y(T) - y_T$.

[6] Assuming that $\mathbf{V} \in L^\infty(\Omega \times (0,T))^2$, $a \in L^\infty(\Gamma_1)$, and (e.g.) $f \in L^2(\Omega)$, $g \in L^2(\Gamma_1)$.

**Remark 3.7:** An important issue in this context is the *controllability* of the system (3.1). Under suitable conditions we have *approximate controllability* (see [20] for the precise assumptions and a proof), that is, for all $y_T \in K$, there are controls in $\mathscr{U}$ making $\|y(T) - y_T\|_K$ arbitrarily small. Another intresenting question is whether we have approximate controllability in case we constrain the control to be constant or having a fixed distribution on $\Gamma_c$, so that the control is a function of time only. This appears to be an open problem.

With the above choices of $\mathscr{U}$ and $K$, the problem (3.4) has a unique solution. However, for a given value of the penalty parameter $k$, we have no knowledge *a priori* about the size of $\|y(T) - y_T\|_K$. This can be expected to depend strongly on the character of the flow, $\mathbf{V}$, the location and size of $\Gamma_c$, and on the balance between advection and diffusion, which is determined by the size of the parameter $\alpha$. To ensure that the advection actually assists in controlling the system, we will always assume that $\Gamma_c$ is at an inflow region of the boundary, that is, $\mathbf{n} \cdot \mathbf{V} < 0$ on $\Gamma_c$. Another parameter of interest, especially for highly advective flows, is the final time $T$, since the advection represents a finite speed of propagation in $\Omega$ for a control applied at $\Gamma_c$.

A similar calculation as in § 2.3, shows that the derivative of $J$ at $v$ in the direction $w \in \mathscr{U}$ is

$$\int_0^T \int_{\Gamma_c} J'(v) w \, d\Gamma \, dt = \int_0^T \int_{\Gamma_c} \left( v - \alpha \frac{\partial p}{\partial n} \right) w \, d\Gamma \, dt, \tag{3.8}$$

where $p$ is the solution to the following adjoint equation:

$$\begin{aligned} -\frac{\partial p}{\partial t} - \alpha \Delta p - \mathbf{V} \cdot \nabla p = 0 \qquad & \text{in } \Omega \times (T,0), \\ p = 0 \qquad & \text{on } \Gamma_c \times (T,0), \\ \alpha \frac{\partial p}{\partial n} + ap + \mathbf{n} \cdot \mathbf{V} p = 0 \qquad & \text{on } \Gamma_1 \times (T,0), \\ p(T) = k\phi. & \end{aligned} \tag{3.9}$$

## 3.3 The Semi-Discrete Control Problem

The state equation (3.1) is discretized in time using a similar scheme as in Section 2.4 for the Burgers equation. We divide the time interval $(0,T)$ into $N$ subintervals of equal length $\Delta t = T/N$ and, with obvious notation, approximate (3.1) by

$$y^0 = y_0; \tag{3.10}_1$$

$$\begin{aligned} &\text{for } n = 1, \ldots, N, \text{ solve} \\ &\quad \frac{y^n - y^{n-1}}{\Delta t} - \alpha \Delta y^n + \mathbf{V}^n \cdot \nabla y^{n-1} = f^n \quad \text{in } \Omega, \\ &\quad y^n = v^n \qquad \text{on } \Gamma_c, \\ &\quad a y^n + \alpha \frac{\partial y^n}{\partial n} = a g^n \qquad \text{on } \Gamma_1. \end{aligned} \tag{3.10}_2$$

Thus, the diffusion term is treated implicitly while the convection term is treated explicitly; this yields a constant, symmetric, positive definite system of equations to solve at each time step. To approximate $\mathscr{U}$ and $J$, we choose

$$\mathscr{U}^{\Delta t} = \left(L^2(\Gamma_c)\right)^N, \tag{3.11}$$

and

$$J^{\Delta t}(v) = \frac{\Delta t}{2} \sum_{n=1}^{N} \int_{\Gamma_c} (v^n)^2 \, d\Gamma + \frac{k}{2} \int_{\Omega} |\nabla \phi|^2 \, dx, \tag{3.12}$$

where $\phi$ is defined as before (equation (3.7)).

By a perturbation technique, as in Section 2.4, we obtain from (3.10) and (3.12) the semi-discrete adjoint equation

$$p^{N+1} = k\phi; \tag{3.13}_1$$

$$\begin{cases} -\dfrac{p^{N+1} - p^N}{\Delta t} - \alpha \Delta p^N = 0 & \text{in } \Omega, \\ p^N = 0 & \text{on } \Gamma_c, \\ ap^N + \alpha \dfrac{\partial p^N}{\partial n} = 0 & \text{on } \Gamma_1; \end{cases} \tag{3.13}_2$$

$$\text{for } n = N-1, \ldots, 1$$

$$\begin{cases} -\dfrac{p^{n+1} - p^n}{\Delta t} - \alpha \Delta p^n - \mathbf{V}^{n+1} \cdot \nabla p^{n+1} = 0 & \text{in } \Omega, \\ p^n = 0 & \text{on } \Gamma_c, \\ ap^n + \alpha \dfrac{\partial p^n}{\partial n} + \mathbf{n} \cdot \mathbf{V}^{n+1} p^{n+1} = 0 & \text{on } \Gamma_1, \end{cases} \tag{3.13}_3$$

and the following expression for the derivative of $J^{\Delta t}$:

$$\Delta t \sum_{n=1}^{N} \int_{\Gamma_c} \frac{\partial J^{\Delta t}}{\partial v^n} w_n \, d\Gamma = \Delta t \sum_{n=1}^{N} \int_{\Gamma_c} (v^n - \alpha \frac{\partial p^n}{\partial n}) w^n \, d\Gamma. \tag{3.14}$$

### 3.4 Space Discretization

We assume that $\Omega$ is a polygonal domain in $\mathbb{R}^2$. Let $\mathcal{T}_h = \{K_i\}_{i=1}^M$ be a triangulation of $\Omega$ with $M$ elements, $K_i$, and let $h = \max_{K_i \in \mathcal{T}_h} \operatorname{diam}(K_i)$. Let the space $V_h$ be the space of continuous, piecewise linear functions on $\Omega$, that is

$$V_h = \left\{ z | z \in \mathscr{C}^0(\overline{\Omega}) : z|_{K_i} \in P_1, \, \forall K_i \in \mathcal{T}_h \right\},$$

where $P_1$ is the space of polynomials of degree $\leq 1$. The space $V_0$ as defined in (3.5) is approximated by

$$V_{0h} = \{ z | z \in V_h : z|_{\Gamma_c} = 0 \}.$$

The space of traces of $V_h$ on $\Gamma_c$ is

$$\gamma V_h = \left\{ \mu | \mu \in \mathscr{C}^0(\Gamma_c), \text{ where } \mu = z|_{\Gamma_c} \text{ for some } z \in V_h \right\}.$$

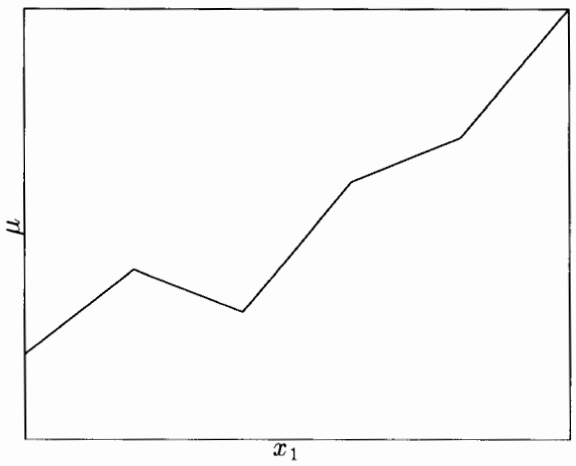


Figure 11: A function $\mu \in \gamma V_h$ when $\Gamma_c$ is a piece of the boundary $x_2 = \text{Const}$.

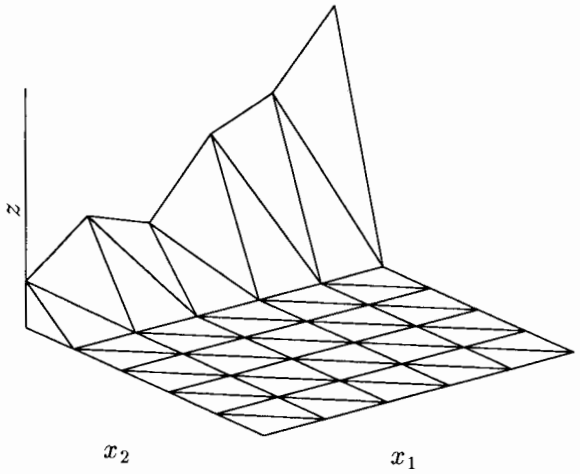


Figure 12: The extension $z \in \mathcal{M}_h$ corresponding to the function $\mu \in \gamma V_h$ visualized in Figure 11.

We take as the space of discrete admissible controls the Cartesian product of $N$ such spaces, that is,

$$\mathcal{U}_h^{\Delta t} = \gamma V_h \times \ldots \times \gamma V_h,$$

where $\dim(\mathcal{U}_h) = N \dim(\gamma V_h)$. Thus, for any $\mathbf{v} \in \mathcal{U}_h^{\Delta t}$, we have $\mathbf{v} = \{v_h^n\}_{n=1}^N$ and $v_h^n \in \gamma V_h$.

The following, unique decomposition of $V_h$ will be used:

$$V_h = \mathcal{M}_h \oplus V_{0h},$$

where

$$\mathcal{M}_h = \left\{ z | z \in V_h : \overline{K_i} \cap \Gamma_c = \varnothing \Rightarrow z|_{K_i} = 0, \forall K_i \in \mathcal{T}_h \right\}.$$

The space $\mathcal{M}_h$ can be viewed as an extension of $\gamma V_h$ into $V_h$; for every function $\mu \in \gamma V_h$, we associate a unique function $z \in V_h$ such that $z$ will be zero at every node except at the nodes of $\Gamma_c$ where it will be equal to $\mu$. Figure 11 shows an example of a function $\mu \in \gamma V_h$ when $\Gamma_c$ is a piece of the line $x_2 = \text{Const}$. Figure 12 shows the corresponding extension $z \in \mathcal{M}_h$.

We take as the fully discrete state equation

$$y_h^0 = y_{0h}, \text{ with } y_{0h} \in V_h \text{ an approximation of } y_0; \tag{3.15}_1$$

*for* $n = 1, \ldots, N$, $y_h^n$ *is obtained from* $y_h^{n-1}$ *through the solution of the discrete elliptic problem*

$$\begin{cases} y_h^n \in V_h \text{ such that } y_h^n|_{\Gamma_c} = v_h^n \text{ and, } \forall z \in V_{0h}, \\ \displaystyle\int_\Omega \frac{y_h^n - y_h^{n-1}}{\Delta t} z \, dx + \alpha \int_\Omega \nabla y_h^n \cdot \nabla z \, dx + \int_\Omega \mathbf{V}^n \cdot \nabla y_h^{n-1} z \, dx \\ \qquad + \displaystyle\int_{\Gamma_1} a y_h^n z \, d\Gamma = \int_\Omega f^n z \, dx + \int_{\Gamma_1} a g^n z \, d\Gamma, \end{cases} \tag{3.15}_2$$

and as our discrete cost function

$$J_h^{\Delta t} = \frac{\Delta t}{2} \sum_{n=1}^{N} \int_{\Gamma_c} (v_h^n)^2 \, d\Gamma + \frac{k}{2} \int_{\Omega} |\nabla \phi_h|^2 \, dx, \tag{3.16}$$

where

$$\begin{cases} \phi_h \in V_{0h} \text{ such that, } \forall z \in V_{0h}, \\ \displaystyle\int_{\Omega} \nabla \phi_h \cdot \nabla z \, dx = \int_{\Omega} (y_h^N - y_T) z \, dx. \end{cases} \tag{3.17}$$

**Remark 3.8:** Note that in (3.6) and (3.12), we prescribed a target function $y_T$ in the dual space $V_0'$ whereas in (3.17), we assume that the target is smooth enough to be represented as a function in $L^2(\Omega)$.

The fully discrete analogue to the problem 3.4 is

$$\begin{aligned} &\text{Find } u_h^{\Delta t} \in \mathscr{U}_h^{\Delta t} \text{ such that} \\ &J_h^{\Delta t}(u_h^{\Delta t}) \leq J_h^{\Delta t}(v), \qquad \forall v \in \mathscr{U}_h^{\Delta t}. \end{aligned} \tag{3.18}$$

The state equation (3.15) and the cost function (3.16) defines the fully discrete adjoint equation

$$p_h^{N+1} = k \phi_h; \tag{3.19}_1$$

$$\begin{cases} p_h^N \in V_{0h} \text{ such that, } \forall z \in V_{0h}, \\ \displaystyle\int_{\Omega} \frac{p_h^N - p_h^{N+1}}{\Delta t} z \, dx + \alpha \int_{\Omega} \nabla p_h^N \cdot \nabla z \, dx + \int_{\Gamma_1} a p_h^N z \, d\Gamma = 0; \end{cases} \tag{3.19}_2$$

*for $n = N-1, \ldots, 1$, $p_h^n$ is obtained from $p_h^{n+1}$ through the solution of the discrete elliptic problem*

$$\begin{cases} p_h^n \in V_{0h} \text{ such that, } \forall z \in V_{0h}, \\ \displaystyle\int_{\Omega} \frac{p_h^n - p_h^{n+1}}{\Delta t} z \, dx + \alpha \int_{\Omega} \nabla p_h^n \cdot \nabla z \, dx + \int_{\Omega} p_h^{n+1} \mathbf{V}^{n+1} \cdot \nabla z \, dx \\ \qquad + \displaystyle\int_{\Gamma_1} a p_h^n z \, d\Gamma = 0, \end{cases} \tag{3.19}_2$$

and the following expression for the derivative of $J_h^{\Delta t}$:

$$\Delta t \sum_{n=1}^{N} \int_{\Gamma_c} \frac{\partial J_h^{\Delta t}}{\partial v^n} w_n \, d\Gamma = \Delta t \sum_{n=1}^{N} \int_{\Gamma_c} (v^n - \lambda_h^n) w^n \, d\Gamma, \tag{3.20}$$

where, $\{\lambda_h^n\}_{n=1}^N$ are the solutions to

$$\begin{aligned} &\lambda_h^n \in \gamma V_h \text{ such that, } \forall z \in \mathcal{M}_h, \\ &\int_{\Gamma_c} \lambda_h^n z \, d\Gamma = \\ &\quad = \begin{cases} \displaystyle\int_{\Omega} \frac{p_h^n - p_h^{n+1}}{\Delta t} z \, dx + \alpha \int_{\Omega} \nabla p_h^n \cdot \nabla z \, dx, & \text{if } n = N \\ \displaystyle\int_{\Omega} \left( \frac{p_h^n - p_h^{n+1}}{\Delta t} - \mathbf{V}^{n+1} \cdot \nabla p_h^{n+1} \right) z \, dx + \alpha \int_{\Omega} \nabla p_h^n \cdot \nabla z \, dx, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.21}$$

Comparing (3.14) and (3.20), we note that the quantity $\lambda_h^n$ is an approximation to $\alpha \partial p^n / \partial n$. To see why the approximation should be of this form, we return to the semi-discrete case and consider the space

$$V_1 = \{z | z \in H^1(\Omega), z|_{\Gamma_1} = 0\}.$$

By the Green's formula, we have

$$\int_{\Gamma_c} \alpha \frac{\partial p^n}{\partial n} z \, d\Gamma = \alpha \int_\Omega \Delta p^n z \, dx + \alpha \int_\Omega \nabla p^n \cdot \nabla z \, dx, \quad \forall z \in V_1.$$

Reducing the term containing the Laplacian with the help of the adjoint equation (3.13) yields, $\forall z \in V_1$,

$$\int_{\Gamma_c} \alpha \frac{\partial p^n}{\partial n} z \, d\Gamma = \begin{cases} \displaystyle\int_\Omega \frac{p^n - p^{n+1}}{\Delta t} z \, dx + \alpha \int_\Omega \nabla p^n \cdot \nabla z \, dx, & \text{if } n = N, \\ \displaystyle\int_\Omega \left( \frac{p^n - p^{n+1}}{\Delta t} - \mathbf{V}^{n+1} \cdot \nabla p^{n+1} \right) z \, dx + \alpha \int_\Omega \nabla p^n \cdot \nabla z \, dx, & \text{otherwise.} \end{cases} \tag{3.22}$$

In the fully discrete case, we have that $\nabla p_h^n$ is discontinuous at each element boundary, so it is questionable what an expression like $\partial p_h^n / \partial n$ would mean. However, the right-hand side of (3.22) is well-defined even in the fully discrete case. Using this observation, it is straightforward (but somewhat tedious) to derive the discrete gradient (3.20) where $\lambda_h^n$ is the discrete analogue of $\alpha \partial p^n / \partial n$; see [4] for an example of a calculation in a similar situation.

## 3.5 Implementation and Results

In this section, we consider a number of test problems with $\Omega = (0,1) \times (0,1)$. Let the independent variable be denoted $x = \{x_1, x_2\}$. The edge $x_2 = 0$ was chosen to be the part of the boundary where the control is applied, $\Gamma_c$ , and $\Gamma_1$ is the rest of the boundary. We used a standard, quasi-uniform triangulation of $\Omega$ (Figure 13).

All the integrals are computed using the trapezoidal rule. This means that the fully discretized state and adjoint equation will have precisely the same structure as if a standard finite difference scheme were used for the space discretizations with the Laplacian discretized by the *five-point formula* (cf. [26, p. 31], for instance). The remaining linear systems were solved by using the FISHPACK library routine **depx4**, based on the work by Swarztrauber [49]. This solves the system by a *generalized cyclic reduction algorithm*, which is a fast direct solver.

The term $\int_\Omega (p_h^n - p_h^{n+1}) z \, dx$ in (3.21) vanishes when the trapezoidal rule is used (recall that $p_h = 0$ on $\Gamma_c$ and that $z|_{K_i} = 0$ whenever $\overline{K_i} \cap \Gamma_c = \varnothing$). For this triangulation, the term $\int_\Omega \nabla p_h^n \cdot \nabla z \, dx$ reduces to a finite difference approximation of $\partial p^n / \partial n$. However, the term involving $\mathbf{V}$ also contributes for $n < N$, so, altogether, $\lambda_h^n$ will *not* reduce to a straight-forward difference approximation of $\partial p^n / \partial n$.

Regarding the implementation of the conjugate gradient algorithm to solve the minimization problem (3.18), it is basically the same as for the Burgers equation (see Section 2.5.2) with one major difference; there is no need for an inexact line search

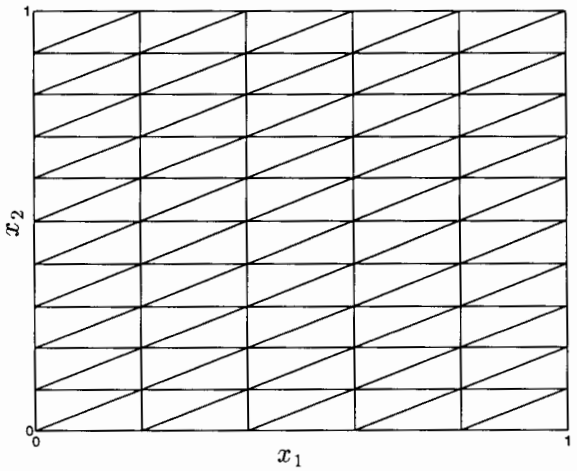

Figure 13: The triangulation of $\Omega$.

since due to the linearity of the state equation, the optimal step length can be explicitly computed at each iteration step at no extra cost in terms of evaluations of the state equation.

As a first test problem, we consider the following target function:

$$y_T = \frac{1}{a_0^2 + (x_1 - a_1)^2 + (x_2 - a_2)^2},$$

the constant advecting field $\mathbf{V} = \{0, 1\}$, and the following parameter values:

$$\begin{aligned} &T = 1.4, && k = 10^4, && h_1 = h_2 = 1/20, \\ &\Delta t = 2 \times 10^{-2}, && \alpha = 10^{-2}, && \epsilon = 10^{-6}, \\ &a_0 = \frac{1}{10}, && a_1 = a_2 = \frac{1}{2}. \end{aligned} \tag{3.23}$$

As a convergence criterion for the conjugate gradient algorithm we used (2.33). At convergence, we got

$$\frac{\|y_h^N - y_T\|}{\|y_T\|} = \begin{cases} 0.0018 & (V_0'\text{-norm}), \\ 0.083 & (L^2\text{-norm}); \end{cases} \qquad \|u_h\|_{L^2} = 7.4.$$

Figures 14 and 15 show cross sections of the target state and the computed final state at $x_2 = 1/2$ and $x_1 = 1/2$ respectively. Figure 16 shows the computed optimal control as a function of time and space. The number of iterations was about 250 for this problem, as for all converging test problems reported in this section.

A few runs were performed with different values of the parameter $\alpha$. When the system becomes highly dissipative ($\alpha \sim 1$), it also becomes quite hard to control. The controllability increased with decreasing values of $\alpha$ down to $\alpha \sim 10^{-2}$. Convergence problems started to occur for even smaller values. The reason might be that the simple discretization scheme used here is too unsophisticated for a strongly advection-dominated equation.

This encouraging result makes it tempting to try a few harder problems. As a second problem, consider the pyramid-shaped target function visualized in Figure 17.

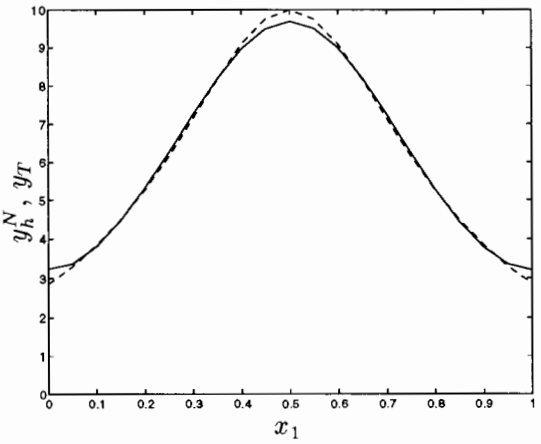


Figure 14: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$, (solid) at $x_2 = 1/2$.

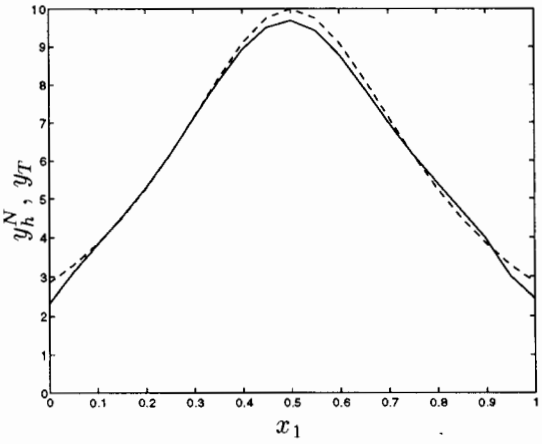


Figure 15: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$ (solid) at $x_1 = 1/2$.

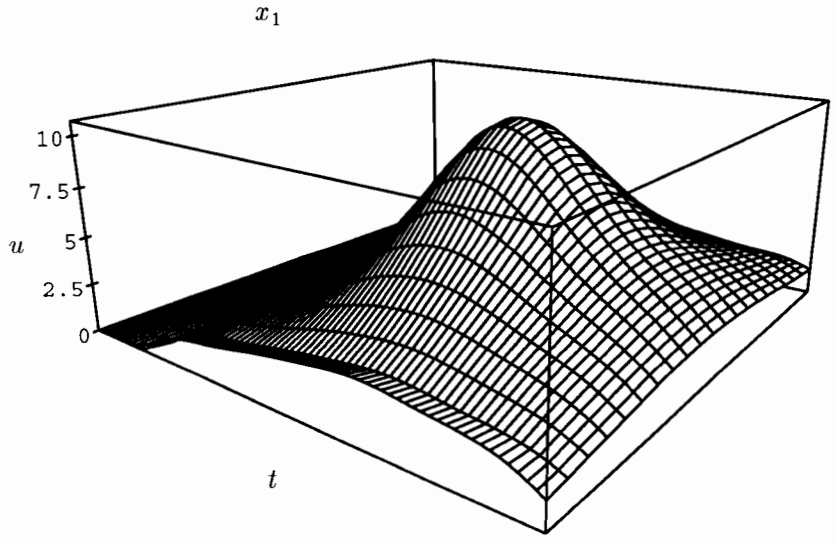


Figure 16: The computed optimal control associated with the final state given in Figures 14 and 15.

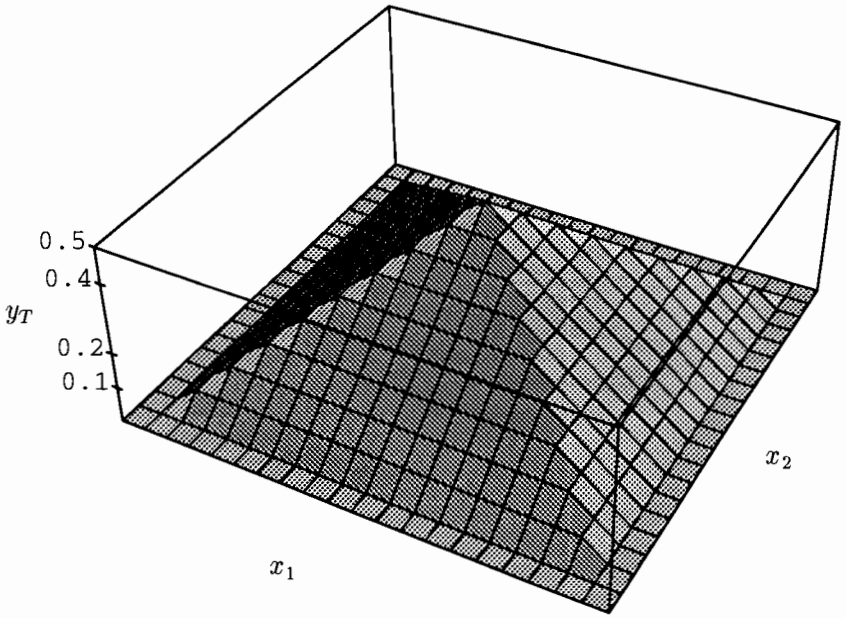


Figure 17: The pyramid-shaped target function.

We obtained the following numerical results using the same parameter values (3.23) as above:

$$\frac{\|y_h^N - y_T\|}{\|y_T\|} = \begin{cases} 0.0066 & (V_0'\text{-norm}), \\ 0.067 & (L^2\text{-norm}); \end{cases} \qquad \|u_h\|_{L^2} = 0.19.$$

Figure 18 shows the final state $y_h^N$ and Figures 19 and 20 show the cross sections at $x_2 = 1/2$ and $x_1 = 1/2$ respectively.

A series of targets converging towards a discontinuous function was selected as a third test problem. Here, we present the results of two of the runs. Let $r_1$, $r_2$ be two parameters such that $0 < r_1 \leq r_2 < 1/2$, and let

$$r = \left[ \left( x_1 - \frac{1}{2} \right)^2 + \left( x_2 - \frac{1}{2} \right)^2 \right]^{1/2}.$$

The target function is then defined by

$$y_T = \begin{cases} 1 & \text{if } 0 \leq r < r_1, \\ \dfrac{r - r_2}{r_1 - r_2} & \text{if } r_1 \leq r < r_2, \\ 0 & \text{if } r_2 \leq r < r_1. \end{cases} \tag{3.24}$$

This target approaches a discontinuous target—a circular cylinder with radius $r$ centered at $\{1/2, 1/2\}$—as $r_2 \to r_1$. The actual target function used was a piecewise linear approximation of the function given above. The target for $r_1 = 0.2$, $r_2 = 0.45$ is visualized in Figure 21. Using this target and the parameter values (3.23), we obtained the

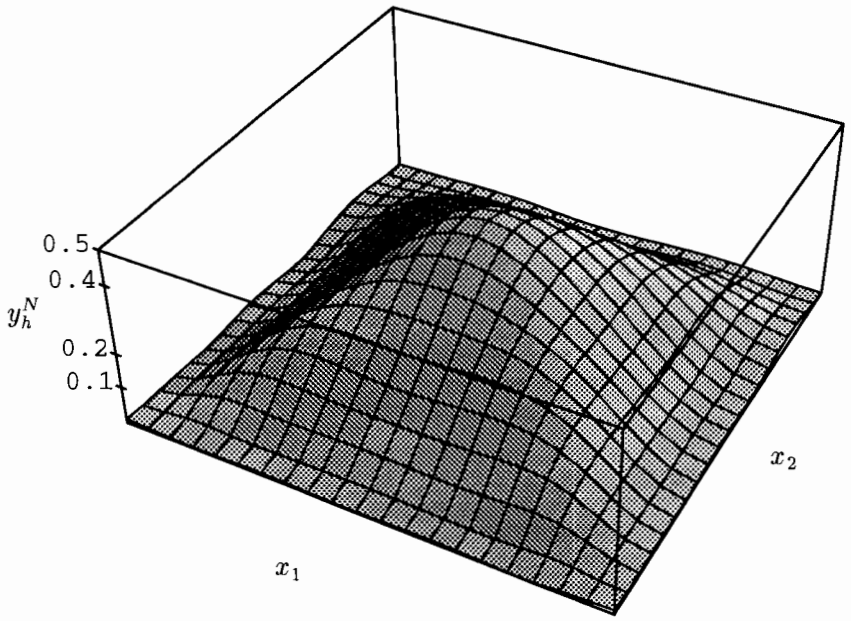


Figure 18: The final state corresponding to the target state in Figure 17.

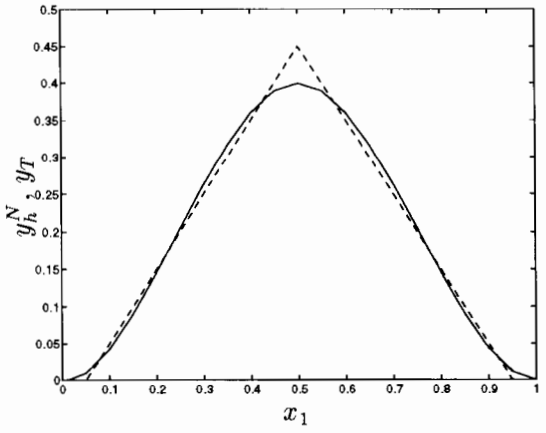


Figure 19: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$, (solid) at $x_2 = 1/2$.

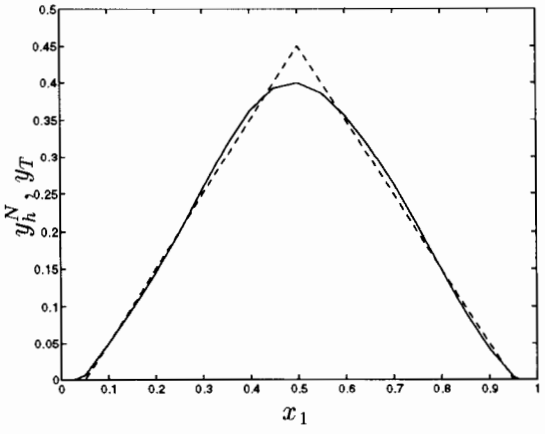


Figure 20: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$ (solid) at $x_1 = 1/2$.

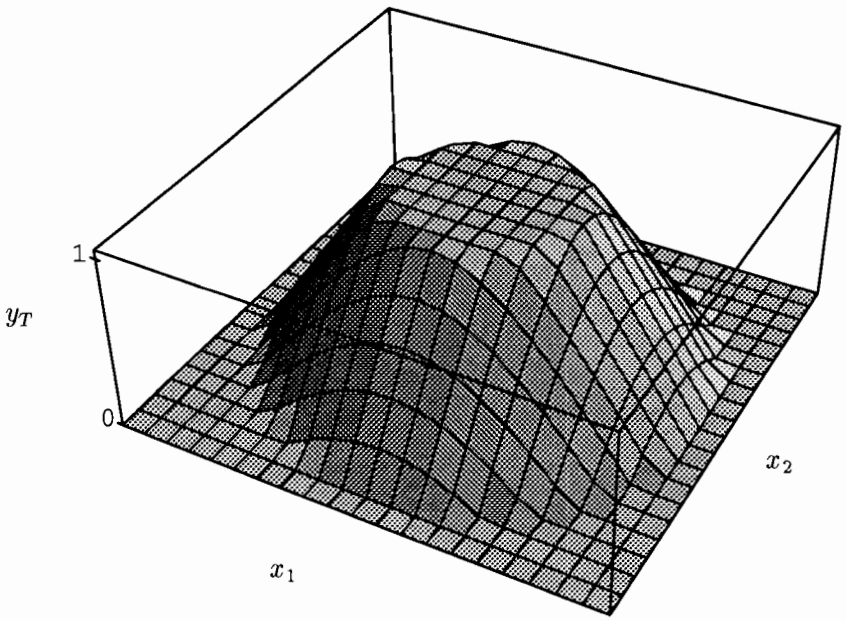


Figure 21: A target state.

following results

$$\frac{\|y_h^N - y_T\|}{\|y_T\|} = \begin{cases} 0.0050 & (V_0'\text{-norm}), \\ 0.047 & (L^2\text{-norm}); \end{cases} \qquad \|u_h\|_{L^2} = 0.58.$$

Figure 22 shows the computed state at $t = T$ and Figures 23 and 24 show the cross sections at $x_2 = 1/2$ and $x_1 = 1/2$ respectively.

The steeper the slope in the target function, the more badly conditioned the problem becomes, until finally the algorithm fails to converge. A "limit case" is shown in Figures 25–28, showing the target function and the computed final state. Figure 29 shows the control as a function of time. In this case, we used the following parameter values:

$$\begin{aligned} &r_1 = 0.25, \quad r_2 = 0.3, \quad T = 1.4, \quad k = 10^4, \\ &h_1 = h_2 = 1/40, \quad \Delta t = 10^{-2}, \quad \alpha = 10^{-2}. \end{aligned} \tag{3.25}$$

The iterations were stopped after 400 iterations. At that point, the norm of the gradient had been reduced by a factor of $\sim 2 \times 10^3$, instead of $10^6$ as required for full convergence. The result were

$$\frac{\|y_h^N - y_T\|}{\|y_T\|} = \begin{cases} 0.026 & (V_0'\text{-norm}), \\ 0.25 & (L^2\text{-norm}); \end{cases} \qquad \|u_h\|_{L^2} = 0.60.$$

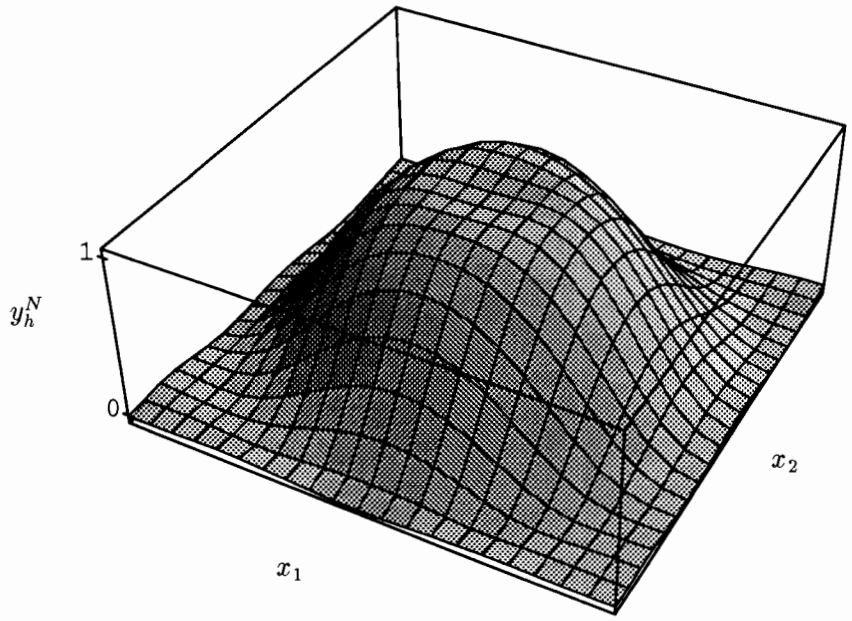


Figure 22: The final state corresponding to the target state in Figure 21.

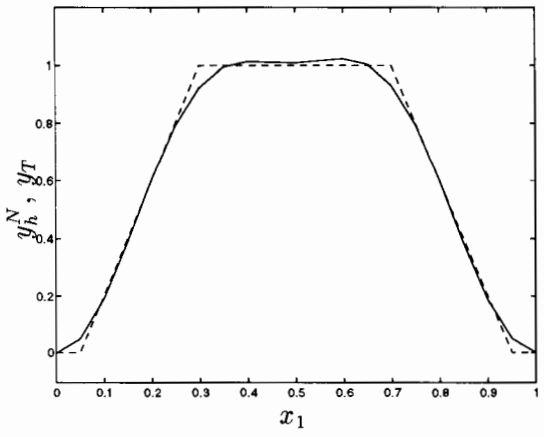


Figure 23: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$, (solid) at $x_2 = 1/2$.

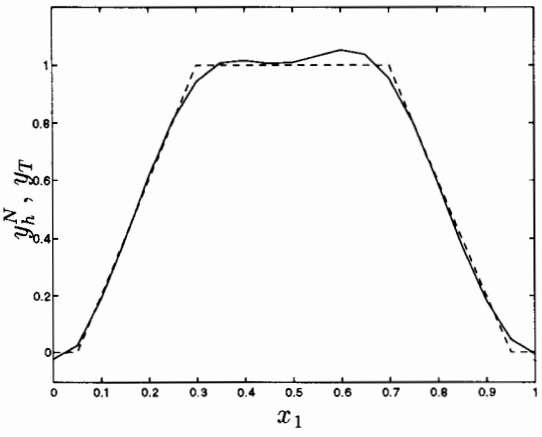


Figure 24: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$ (solid) at $x_1 = 1/2$.

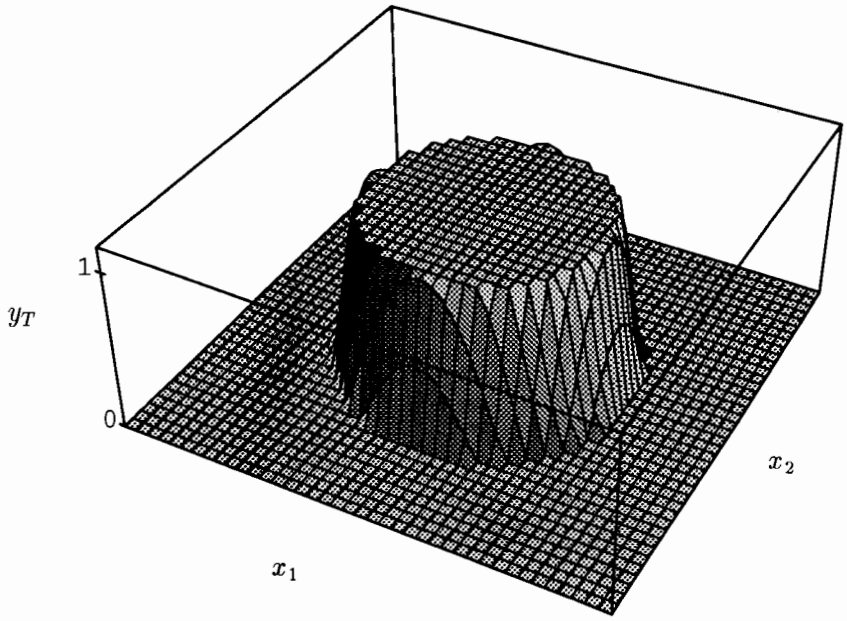


Figure 25: A target state.

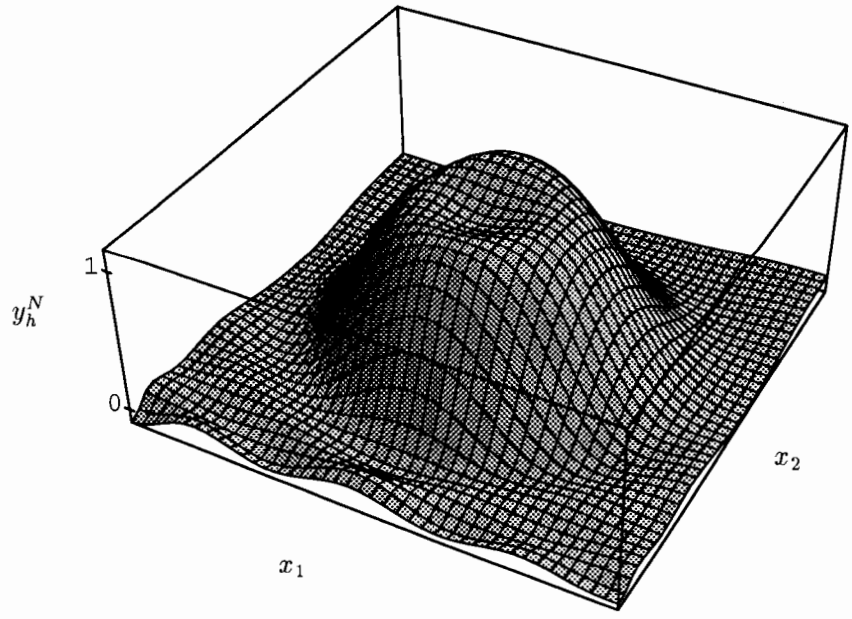


Figure 26: The final state corresponding to the target state in Figure 25.

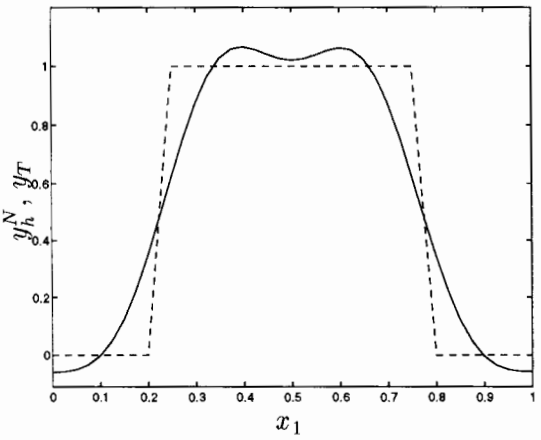


Figure 27: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$, (solid) at $x_2 = 1/2$.

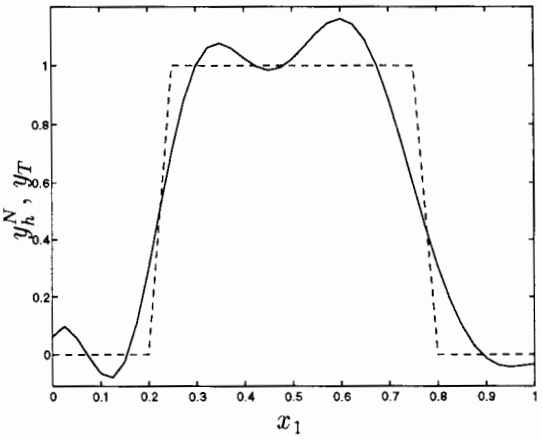


Figure 28: Cross section of the target state, $y_T$, (dashed) and the final state, $y_h^N$ (solid) at $x_1 = 1/2$.

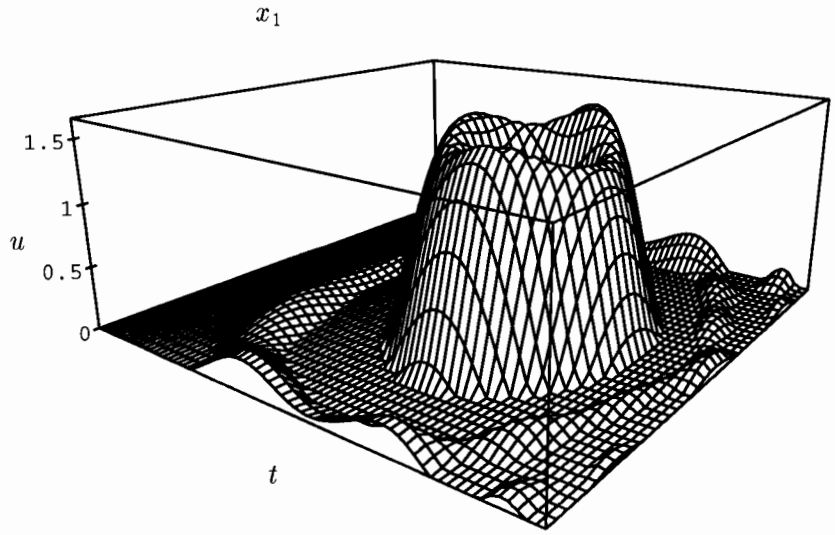


Figure 29: The computed optimal control corresponding to the final state given in Figure 25.

# 4 Forcing Control of Flow Governed by the Unsteady Stokes Equations

## 4.1 Generalities and Problem Formulation

In this section, we consider systems governed by the *Unsteady Stokes equations*,

$$\begin{aligned} \frac{\partial \mathbf{y}}{\partial t} - \nu \Delta \mathbf{y} + \nabla \pi &= \mathbf{f} \quad && \text{in } \Omega, \\ \nabla \cdot \mathbf{y} &= \mathbf{0} \quad && \text{in } \Omega, \\ \mathbf{y} &= \mathbf{g}_0 \quad && \text{on } \Gamma_0, \\ \nu \frac{\partial \mathbf{y}}{\partial n} - \mathbf{n}\pi &= \mathbf{g}_1 \quad && \text{on } \Gamma_1, \\ \mathbf{y}(0) &= \mathbf{y}_0. \end{aligned} \tag{4.1}$$

Here, $\mathbf{y}(\mathbf{x}, t) \in \mathbb{R}^d$ models the velocity field and $\pi(\mathbf{x}, t) \in \mathbb{R}$ the pressure field of a fluid which fills the open and connected region $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3 is the space dimension), and $\nu$ is a positive constant. These equations are obtained by linearization of the incompressible Navier-Stokes equations about a zero flow. At a part of the boundary, $\Gamma_0$, we have a prescribed velocity field $\mathbf{g}_0$. The boundary condition at the rest of the boundary, $\Gamma_1$, is not particularly "physical", but it can be used to implement downstream boundary conditions for flow in unbounded regions. Considering the control of a system like (4.1) is an important step towards the control of the full Navier-Stokes equations.

**Remark 4.9:** The proper approximation for flow at very low Reynolds numbers—when the viscous forces are considerably greater than the inertia forces—is the *Stokes equations*. These differ from (4.1) in that there is no time-derivative term $\partial \mathbf{y}/\partial t$.

The control problem is of the same type as before: we want the velocity field $\mathbf{y}$ at time $t = T$ to approximate a given target state, $\mathbf{y}_T$. In this case, the system is controlled through the forcing term which will be nonzero only in the open, nonempty subdomain $\omega \subset \Omega$,

$$\mathbf{f}(\mathbf{x}, t) = \begin{cases} \mathbf{v}(\mathbf{x}, t) & \text{if } \mathbf{x} \in \omega, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

We define the space of admissible control to be $\mathscr{U} = L^2(\omega \times (0,T))^d$, and we consider the cost function

$$J(v) = \frac{1}{2} \int_0^T \int_\omega |\mathbf{v}|^2 \, dx \, dt + \frac{k}{2} \int_\Omega |\mathbf{y}(T) - \mathbf{y}_T|^2 \, dx, \tag{4.2}$$

where $k > 0$ is a penalty parameter as in (2.4) and (3.3), and where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^d$. The control problem is to find the function $\mathbf{u}$ which minimizes $J$ among all $\mathbf{v} \in \mathscr{U}$.

With the above choice of $\mathscr{U}$, the function $t \mapsto \mathbf{y}(t)$ is continuous[7] in $L^2(\Omega)$, so the cost function (4.2) is well defined and the control problem will have a unique solution.

[7] For this, we also need to assume that the domain has a sufficiently smooth boundary, and that the inhomogeneous terms in (4.1) are in suitable function classes.

**Remark 4.10:** Another quite interesting problem is the *Dirichlet boundary control of* (4.1). As for the advection-diffusion problem in Section 3, if we define the admissible controls in an $L^2$ space on the boundary, $t \mapsto \mathbf{y}(t)$ will *not* be continuous in $L^2(\Omega)$. Thus, in this case we need to strengthen the regularity requirements on the control or, as we did for the advection-diffusion problem, use a weaker norm for the error $\mathbf{y}(T) - \mathbf{y}_T$.

Some quite powerful controllability results hold for this system [20]. Letting $V$ be the space of reachable final states, that is,

$$V = \{\mathbf{y}(T) | \mathbf{y} \textit{ is a solution of (4.1) for some } \mathbf{v} \in \mathscr{U}\},$$

and $H$ the space

$$H = \{\mathbf{z} | \mathbf{z} \in L^2(\Omega)^d : \nabla \cdot \mathbf{z} = 0\},$$

we have $\overline{V} = H$ in $L^2(\Omega)^d$. So every "reasonable" target function can be approximated arbitrarily well in a least-squares sense. In fact, for this result to hold, it is enough to control only two of the components of $\mathbf{v}$ in three-space and only one component in two-space.

**Remark 4.11:** An interesting problem, still under investigation, is the control with respect to *domain variations*; from the work of I. Diaz, E. Zuazua and J.L. Lions, there are indications that the approximate controllability property holds with respect to "small" domain perturbations.

Thus, in the case $d = 2$, we can assume that the control is given in the form $\mathbf{v} = \{v, 0\}$. A similar calculation as in previous sections yields the gradient $J'(v) = v - p_1|_\omega$, where $\mathbf{p} = \{p_1, p_2\}$ is the solution to the adjoint equation

$$\begin{aligned}
-\frac{\partial \mathbf{p}}{\partial t} - \nu \Delta \mathbf{p} + \nabla \sigma &= \mathbf{0} && \text{in } \Omega, \\
\nabla \cdot \mathbf{p} &= \mathbf{0} && \text{in } \Omega, \\
\mathbf{p} &= \mathbf{0} && \text{on } \Gamma_0, \\
\nu \frac{\partial \mathbf{p}}{\partial n} - \mathbf{n}\sigma &= \mathbf{0} && \text{on } \Gamma_1, \\
\mathbf{p}(T) &= k(\mathbf{y}_T - \mathbf{y}(T)).
\end{aligned} \tag{4.3}$$

## 4.2 Discretization and Implementation

We use the following simple implicit scheme for the time discretization ($\Delta t = T/N$):

$$\begin{aligned}
&\mathbf{y}^0 = \mathbf{y}_0, \\
&\text{for } n = 1, \ldots, N, \text{ solve} \\
&\quad \frac{\mathbf{y}^n - \mathbf{y}^{n-1}}{\Delta t} - \nu \Delta \mathbf{y}^n + \nabla \pi^n = \mathbf{f}^n \quad \textit{in } \Omega, \\
&\quad \nabla \cdot \mathbf{y}^n = \mathbf{0} \quad \textit{in } \Omega, \\
&\quad \mathbf{y}^n = \mathbf{g}_0^n \quad \textit{on } \Gamma_0, \\
&\quad \nu \frac{\partial \mathbf{y}^n}{\partial n} - \mathbf{n}\pi^n = \mathbf{g}_1^n \quad \textit{on } \Gamma_1,
\end{aligned} \tag{4.4}$$

where, for $n = 1, \ldots, N$,

$$\mathbf{f}^n = \begin{cases} \{v^n, 0\} & \text{in } \omega, \\ \mathbf{0} & \text{in } \Omega \setminus \omega. \end{cases}$$

We approximate the space of admissible controls $\mathscr{U}$ with

$$\mathscr{U}^{\Delta t} = L^2(\omega)^N,$$

and the cost function $J$ with

$$J^{\Delta t}(v) = \frac{\Delta t}{2} \sum_{n=1}^{N} \int_\omega (v^n)^2 \, dx + \frac{k}{2} \int_\Omega (\mathbf{y}(T) - \mathbf{y}_T)^2 \, dx.$$

For the numerical experiments below, we use a method that only needs solutions of the state equation. However, for completeness we state the expression for the gradient of $J^{\Delta t}$ and the adjoint equation; for $v, w \in \mathscr{U}^{\Delta t}$, we have, with obvious notation,

$$< \nabla J^{\Delta t}(v), w >= \Delta t \sum_{n=1}^{N} \int_\omega (v^n - p_1^n) w^n \, dx,$$

where $\{\mathbf{p}^n\}_{n=1}^N$, $\mathbf{p}^n = \{p_1^n, p_2^n\}$ is the solution to

$$\begin{aligned} &\mathbf{p}^{N+1} = k(\mathbf{y}_T - \mathbf{y}^N), \\ &\textit{for } n = N, \ldots, 1, \textit{ solve} \\ &\quad -\frac{\mathbf{p}^{n+1} - \mathbf{p}^n}{\Delta t} - \nu \Delta \mathbf{p}^n + \nabla \sigma^n = \mathbf{0} \quad \textit{in } \Omega, \\ &\quad \nabla \cdot \mathbf{p}^n = \mathbf{0} \quad \textit{in } \Omega, \\ &\quad \mathbf{p}^n = \mathbf{0} \quad \textit{on } \Gamma_0, \\ &\quad \nu \frac{\partial \mathbf{p}^n}{\partial n} - \mathbf{n}\sigma^n = \mathbf{0} \quad \textit{on } \Gamma_1. \end{aligned} \tag{4.5}$$

This discretization leads to a sequence of stationary, Stokes-like problems for both equation (4.4) and (4.5). The fully discretized versions of these problems are solved by a preconditioned conjugate gradient algorithm ([10], [18]).

As before, we use continuous, piecewise linear functions for the space approximations. To avoid instabilities resulting in spurious oscillations in the solution, we define the pressure on a grid twice as coarse as the grid associated with the velocity. In this way we satisfy a certain compatibility condition between the velocity and pressure approximations, the *Babuska-Brezzi* or *inf-sup* condition. We refer to the literature for the details ([17], [18], [26], [40]).

An important property of the unsteady Stokes equations (4.1) (as well as their discrete analogue) is their *time invariance*; the coefficients in the equations do not depend on time, this in contrast to the other state equations considered in this article (equations (2.2) and (3.1)). The time invariance yields a particular structure to the control problem which makes it feasible, if the problem is not too large, to employ a *direct method*, based on one of the classical methods developed for the *linear least-squares problem*. Appendix B shows that the present control problem can be formulated, using

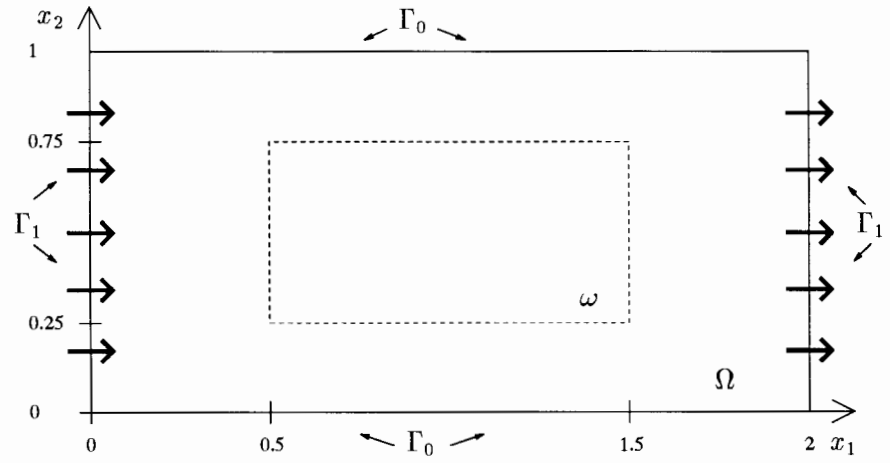


Figure 30: The domain used for the test problem. The arrows indicate flow in and out of the region.

the standard linear-algebra notation, as a version of the linear least-squares problem. Appendix B also briefly discusses some of the classical computational methods for this problem and when they can be efficiently invoked to solve this kind of control problem. For the experiments reported below, we used a direct method based on the *Singular Value Decomposition* of the *forward map* (cf. Appendix B).

### 4.3 Numerical Results

As a test problem, we chose the damping of a *plane Poiseuille flow*. The domain (Figure 30) is the rectangle $\Omega = (0,2) \times (0,1)$ with $\Gamma_0$ being the edges $x_2 = 0$ and $x_2 = 1$, and $\Gamma_1$ the edges $x_1 = 0$ and $x_1 = 2$. The subdomain in which the control is applied is $\omega = (1/2, 3/2) \times (1/4, 3/4)$. The boundary conditions are homogeneous, that is, $\mathbf{g}_0 = \mathbf{g}_1 = 0$ (cf. equation (4.1)), and the initial condition is the parabolic velocity profile $\mathbf{y}_0 = \{4x_2(1 - x_2), 0\}$. Solving equation (4.1) under these conditions with no forcing $\mathbf{f}$ yields a solution exponentially decaying in time with a rate determined by the size of the parameter $\nu$. The control problem is to find a control, acting only on the subdomain $\omega$, such that the velocity field in full domain $\Omega$ is dampened as much as possible at a given final time $T$. To obtain this, we simply choose the target $\mathbf{y}_T = \mathbf{0}$. We choose to keep the control constant in the subdomain $\omega$ and to update the control each second time step. The size of the control problem, that is, the dimension of the discrete space of admissible controls, is then $\dim(\mathcal{U}_h^{\Delta t}) = N/2$, where $N$ is the number of time steps. The Lapack routine `dgesvd` [15] was used to perform the SVD. The computer program was written in Fortran 77 and all runs were performed on a Sun Sparc 10.

We used the following parameter values:

$$T = 1.0, \quad \nu = \frac{1}{20}, \quad k = 20,\ 10^2,\ 10^3,\ 10^4,$$

and the following discretization:

$$h_1 = h_2 = 1/16, \quad \Delta t = 0.02.$$

Table 1: Damping coefficients and the cost of the control for different values of the penalty parameter.

| $k$ | $\frac{\|\mathbf{y}_v(T)\|_{L^2(\Omega)}}{\|\mathbf{y}(T)\|_{L^2(\Omega)}}$ | $\|v\|_{L^2(\omega\times(0,T))}$ |
|---|---|---|
| 20 | 0.10 | 1.1 |
| 100 | 0.050 | 1.4 |
| 1,000 | 0.018 | 1.8 |
| 10,000 | 0.0062 | 2.2 |

The size of the forward map $A$, as defined in Section B.1, is 1122 by 25 with this discretization. Due to the time-invariant nature of the state equation and the fact that the control has no spatial dependence in this case, the state equation needs to be solved only once to compute the map $A$ (cf. Appendix B). Note that the adjoint equation does not need to be computed at all. In fact, once $A$ is computed, the state equation is not needed any more either. From that point of view, the direct approach is optimal in this case. The limitation is the size of the problems that can be treated. Both the storage and the the time needed to compute (in this case) the Singular Value Decomposition will be serious concerns for larger problems.

Figure 31 shows how the introduction of the control affects the temporal development of the the norm of the state, $\|\mathbf{y}(t)\|_{L^2(\Omega)}$. The curve given by asterisks shows how the norm decays without any control, and the rest of the curves show the behavior for different values of the penalty parameter $k$. Figure 32 shows the controls as a function of time. Table 1 gives some numerical values on the efficiency of the damping in terms of the the ratio between the value $\|\mathbf{y}_v(T)\|_{L^2(\Omega)}$ (the norm of the final state with the optimal control applied) and the value $\|\mathbf{y}(T)\|_{L^2(\Omega)}$ (without any control) for the different values of $k$. Figures 33 and 34 show the energy distribution in the final state for two different values of $k$.

The fact that the control problem is so easily solved for different values of the parameter $k$ can be used to analyze the least-squares problem in a way proposed by Lawson and Hanson [31]. The control problem was solved for 15 values of the regularization parameter ranging from $k = 1/10$ to about $k = 1/2 \times 10^{11}$. (Recall that this only involves repeated solutions of the diagonal system (B.10) with $\epsilon = 1/k$) The norm of the control and the norm of the residual, $\|\mathbf{y}(T) - \mathbf{y}_T\|$, were computed for each value of $k$. The norm of the control versus the norm of the residual for each value of $k$, is plotted in Figure 35. Note that this curve constitutes the boundary for all possible coordinate combinations because of the minimum property (B.7); for any control vector $v$, the pair $\{\|v\|, \|\mathbf{y}(T; v) - \mathbf{y}_T\|\}$ lies on or above the curve. From the figure we see that the norm of the residual decreases quite dramatically when $k$ is increased up to about $10^5$ after which the curve starts to flatten out. For $k$ larger than about $10^8$, the norm of the residual stays practically constant. Thus, if the objective is "maximal damping to a reasonable cost", $k \sim 10^8$ seems to be a good choice.

Using $k = 10^8$, we get a damping coefficient (in the sense of Table 1) of $7 \times 10^{-4}$,

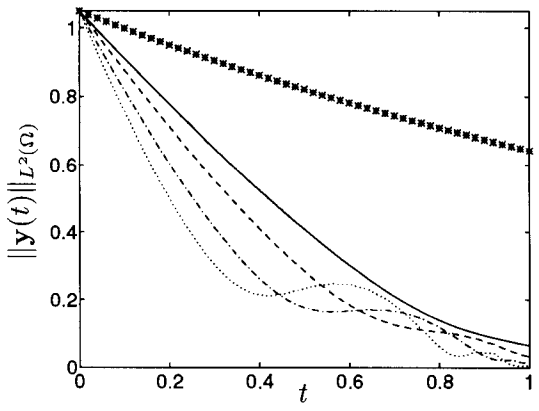


Figure 31: The spatial norm of the state, $\|\mathbf{y}(t)\|_{L^2(\Omega)}$, as a function of time without the control (asterisks) and with the computed optimal control for $k = 20$ (solid), 100 (dashes), 1,000 (dashdots), and 10,000 (dots).

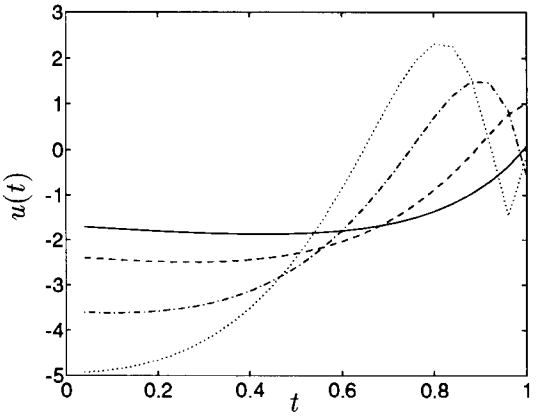


Figure 32: The control as a function of time for $k = 20$ (solid), 100 (dashes), 1,000 (dash-dots), and 10,000 (dots).

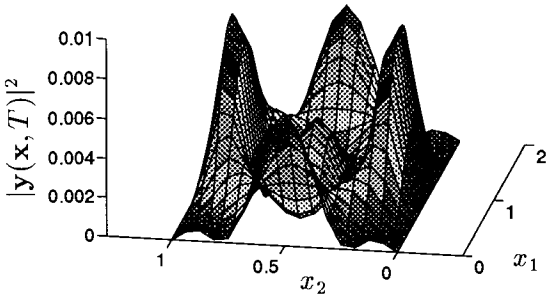


Figure 33: The energy distribution of the final state, $|\mathbf{y}(\mathbf{x},T)|^2$, for the case $k = 20$.

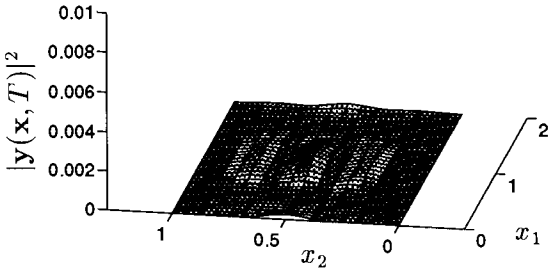


Figure 34: The energy distribution of the final state, $|\mathbf{y}(\mathbf{x},T)|^2$, for the case $k = 1,000$.

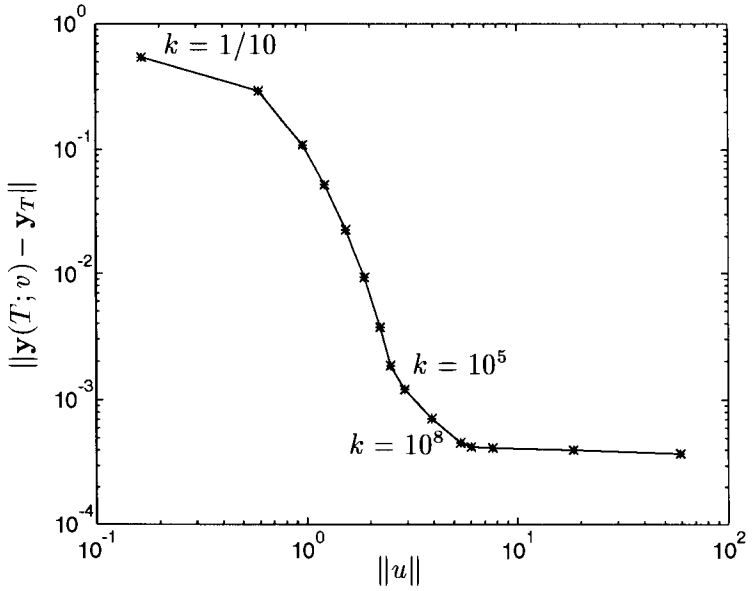


Figure 35: The norm of the control versus the norm of the residual for values of the regularization parameter ranging from $k = 1/10$ to $k = 1/2 \times 10^{11}$.

and the norm of the the control becomes $\|u\| = 5.9$. Figure 31 shows the norm of the state as a function of time without any forcing term (asterisks) and with the computed optimal control. Figure 32 shows the control as a function of time. Thus, in this case we get a very efficient damping of the flow to the prize of less regular behavior of the control and the state.

# 5 Conclusions

In this paper, we have discussed numerical solutions of controllability problems to some fairly simple flow models. For these models, the numerical methods that are described in this paper provide efficient solution algorithms. One of our main objectives is to apply these computational methods to the flow control of more complicated systems like, for example, the systems modelled by the Navier-Stokes equations for viscous, incompressible flow.

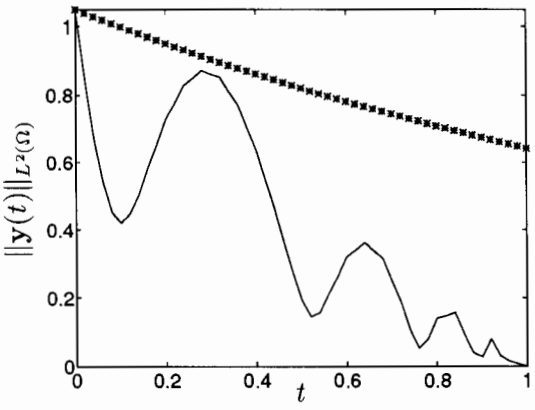


Figure 36: The spatial norm of the state, $\|\mathbf{y}(t)\|_{L^2(\Omega)}$, as a function of time without the control (asterisks) and with the computed optimal control for $k = 10^8$ (solid).

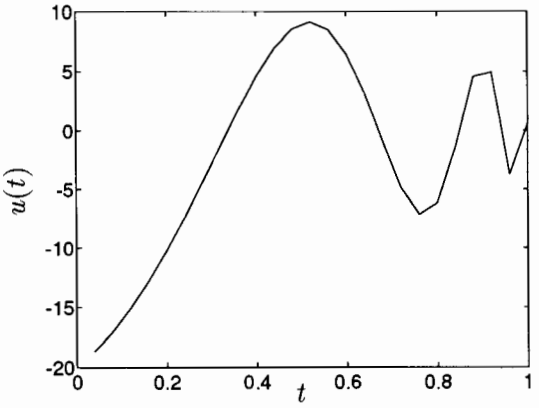


Figure 37: The control as a function of time for $k = 10^8$.

# Appendix A

## Reducing the Storage Requirements in Gradient Computations

When computing the gradient of the cost function (2.29), we first solve the state equation (2.30) and then the adjoint equation, (2.31). Finally, we use the information from the solution of the adjoint equation to compute the gradient (2.32). As we see from (2.31), a non-linear state equation yields an adjoint equation with coefficients that contain the solution of the state equation. For large, multi-dimensional problem it will be very memory demanding to store all the $N$ values of the state variable simultaneously. However, there is a strategy to reduce the storage requirements at the expense of an extra solution of the state equation. Below, we demonstrate the strategy for the gradient given in Section 2.5.1.

We assume that the number of time steps $N$ is factored as

$$N = PQ, \tag{A.1}$$

where $P$ and $Q$ are positive integers. Briefly, the strategy can be described as follows. The interval $(0, N\Delta t)$ is partitioned into $P$ slices, each consisting of $Q$ time steps. First, we solve the state equation up to time step $n = N - Q$, storing only the $P$ values of the states on the boundary between the slices. Then, for each slice, starting with the last one, we solve the state equation for the $Q - 1$ "internal" time steps and store corresponding states, and solve the adjoint equation in that same slice using the information about the states that is just computed.

In algorithmic form, the computation of the gradient becomes

*Solve the state equation (2.30) for* $n = 0, \ldots, N - Q$, *storing only the sampled set* $S = \{y_h^{lQ}\}_{l=0}^{P-1}$ $(A.2)_1$

*for* $l = P - 1, P - 2, \ldots, 0$,

*with* $y_h^{lQ} \in S$ *as initial condition,*
*for* $n = lQ + 1, lQ + 2, \ldots, (l+1)Q - 1$, $(A.2)_2$
*compute* $y_h^n$ *from* $y_h^{n-1}$ *by solving (2.30)*$_2$;

*set* $T \leftarrow \{y_h^n\}_{n=lQ+1}^{(l+1)Q-1}$; $(A.2)_3$

*if* $l = P - 1$,
*compute* $y_h^N$ *from* $y_h^{N-1} \in T$ *by solving (2.30)*$_2$; $(A.2)_4$
*set* $p_s \leftarrow p_h^{N+1}$, *where* $p_h^{N+1}$ *is the solution to (2.31)*$_1$;

*with* $p_s$ *as initial condition,*
*for* $n = (l+1)Q, (l+1)Q - 1, \ldots, lQ$,
*compute* $p_h^n$ *from* $p_h^{n+1}$ *by solving (2.31)*$_3$ *(or (2.31)*$_2$*)* $(A.2)_5$
*using the information in* $T$,
*and add the contribution from* $p_h^n$ *to the gradient;*

*set* $p_s \leftarrow p_h^{lQ}$; $(A.2)_6$

Since $S$ and $T$ contain $P$ and $Q-1$ vectors respectively, the storage requirement is $P+Q-1$ state vectors. Thus, to minimize storage, we should minimize $P+Q-1$ among all factorizations (A.1). If $N^{1/2}$ is an integer, the minimum is attained for $P=Q=N^{1/2}$, giving a required storage of $2N^{1/2}-1$ vectors instead of $N$, which will be a substantial saving for large problems.

The price for this reduction in storage is the extra computations of the states in step $(A.2)_1$. However, note that the state equation needs to be solved only up to $n=N-Q$ at that step, and that the state equation is solved only for the "internal" steps at step $(A.2)_2$. The work accumulated at steps $(A.2)_1$ and $(A.2)_2$ corresponds to $2-1/P-1/Q$ solutions of the state equation, and the work accumulated at step $(A.2)_5$ corresponds to 1 solution of the adjoint equation.

The "best" choice of $P$ and $Q$ depends on the size of the problem and on how costly it is to compute the state equation:

- If the problem is small enough, it might be feasible to store all the states. This corresponds to the choice $P=1$, $Q=N$ above. Then, $N$ state vectors need to be stored and the state equation is solved once.
- For medium-large problems, we can choose $P=2, Q=N/2$. In this case, $N/2+1$ state vectors need to be stored and the state equation needs to be solved close to one and a half times.
- For large problems, we choose $P=Q$ (or approximately so). Then, $2N^{1/2}-1$ state vectors need to be stored and the state equation needs to be solved (almost) two times.

**Remark A.1:** The above approach can be further generalized. Let us consider the factorization

$$N=PQR,$$

where $P$, $Q$, and $R$ are positive integers. The interval $(0, N\Delta t)$ is still partitioned into $P$ slices, but now each such slice is in its turn subdivided into $Q$ slices, each consisting of $R$ time steps. A similar algorithm as (A.2) for this situation needs a storage of $P+Q+R-2$ state vectors, and the state equation needs to be solved $3-1/P-1/Q-1/R$ times. Thus if $P=Q=R$, the storage needed is roughly $3N^{1/3}$ and the state equation needs to be solved about 3 times.

Driving this factoring technique to an extreme, and assuming than

$$N=2^M,$$

for a positive integer $M$, we get a storage of $O(\log N)$. In this case, the corresponding algorithm would need a storage of $M+1$ state vectors and the state equation needs to be solved $M/2$ times.

**Remark A.2:** Dr. W.W. Symes at Rice University pointed out to us that a similar memory-saving device har been introduced by A. Griewank ([22]) in the context of Automatic Differentiation. A minor modification of Grieswank's algorithm has been implemented by J. Blanch in a code for linearized inversion for 2D viscoacoustic media ([50]).

# Appendix B

# The Control Problem as an Algebraic Least-Squares Problem

## B.1 Equivalent Formulations of the Control Problem

For linear state equations, the control problems that are treated in this article are closely related to the *linear least-squares problem*. Formulating the problem in this way makes it natural to try some of the classical computational methods that have been developed for this class of problems.

To see how the control problem can be reformulated as a least-squares problem using the standard linear-algebra-type notation, let us consider a fully discretized control problem of the type treated in this article. Below, for notational convenience, we drop the sub- and superscripts $h$ and $\Delta t$, and use subscripts for the time levels.

Assume that we have a (finite-dimensional) space of admissible controls $\mathscr{U}$. Given an element $v \in \mathscr{U}$, we solve the fully discretized state equation and obtain the discrete states $\{y_n\}_{n=1}^N$ where $y_n$'s are in some finite-dimensional space $F$. We define $\tilde{A} : \mathscr{U} \to F$ as the map $v \mapsto y_N$. Wanting to approximate $y_N$ with a given function $y_T \in F$, we consider the least-squares problem

$$\min_{v \in \mathscr{U}} \|\tilde{A}(v) - y_T\|^2, \tag{B.1}$$

where $\|\cdot\|$ denotes a vector norm in $F$ The map $\tilde{A}$ is affine if the state equation is linear *which we will assume from now on*. Thus we have $\tilde{A}(v) = \tilde{A}(0) + Av$, where $A : \mathscr{U} \to F$, the *forward map*, is linear. The action of $A$ corresponds to solving a homogeneous version of the state equation for an given control $v$ and observing the final state $y_N$. Letting $z = y_T - A(0)$, we see that (B.1) also can be written

$$\min_{v \in \mathscr{U}} \|Av - z\|^2, \tag{B.2}$$

which is clearly a linear least-squares problem. If the minimization is done in the Euclidean (or two-) norm, then $u$ is a solution to (B.2) if and only if it satisfies *the normal equations*

$$A^T A u = A^T z, \tag{B.3}$$

where $A^T$ is the transpose of $A$.

**Remark B.3:** Here we consider only minimization in the 2-norm. Everything discussed in this Appendix can easily be generalized to the case when the norm is given by

$$\|z\|_M = (z^T M z)^{1/2},$$

for a symmetric, positive definite matrix $M$.

The normal equations (B.3) always has a solution, and the solution is unique if $A$ has at least as many rows as columns and if the columns of $A$ are linearly independent. To stabilize the solution of the normal equations (B.3) in the case of an underdetermined

system (more columns than rows in $A$), or in the case of linearly dependent or nearly linearly dependent columns in $A$, one may introduce a positive parameter $\epsilon$ and solve the *regularized normal equations*

$$(\epsilon I + A^T A)u_\epsilon = A^T z. \tag{B.4}$$

In fact, solving the *regularized* normal equations is equivalent to solving the *penalized* control problem considered in this article,

$$\begin{aligned} &\textit{Find } u_k \in \mathscr{U} \textit{ such that} \\ &J(u_k) \leq J(v), \quad \forall v \in \mathscr{U}. \end{aligned} \tag{B.5}$$

where

$$J(v) = \frac{1}{2}\|v\|^2 + \frac{k}{2}\|Av - z\|^2.$$

The unique solution to this problem is $u_k = u$, where $u$ is the solution to (B.4) with $\epsilon = 1/k$.

**Remark B.4:** Equation (B.4) is called a *Tikhonov regularization* of the, in general, *ill-posed* problem $Au = z$ ([30], [52]). When equation (B.4) is a subproblem arising from a non-linear least-squares problem, this regularization procedure is known as the *Levenberg-Marquardt* method ([37], [14]).

**Remark B.5:** Equation (B.4) is defined in the space of the control variables, $\mathscr{U}$. Introducing

$$f = \frac{1}{\epsilon}(z - Au_\epsilon),$$

we see that $f$ and $u_\epsilon$ satisfy

$$\begin{aligned} (\epsilon I + AA^T)f &= z, \\ u_\epsilon &= A^T f. \end{aligned} \tag{B.6}$$

This change of variables yields an equation, equivalent to (B.4), but in the space of the state variables $F$. This formulation is especially advantageous if $\dim(F) \ll \dim(\mathscr{U})$. Applying the *Reverse Hilbert Uniqueness Method* to controllability problems associated with evolution equations yields in a natural way the formulation (B.6) ([33], [19], [34]).

**Remark B.6:** There are several ways to motivate the introduction of the regularization parameter $\epsilon$ (or, equivalently, the penalization parameter $k$). First, as mentioned above, it guarantees a unique solution in the case of a rank-deficient forward map $A$. Secondly, as discussed in the main portion of the text, it can be viewed as a parameter that balances the need for a cheap control, $\|v\|$, and a small residual, $\|Av - z\|$. A third interpretation is to view $\epsilon$ as multiplier that is connected to a particular constrained minimization problem, as shown below.

Since the $u_\epsilon$ that solves equation (B.4) also minimizes the cost function (B.5) for $k = 1/\epsilon$, we have for all $v \in \mathscr{U}$,

$$\frac{\epsilon}{2}\|u_\epsilon\|^2 + \frac{1}{2}\|Au_\epsilon - z\|^2 \leq \frac{\epsilon}{2}\|v\|^2 + \frac{1}{2}\|Av - z\|^2.$$

In particular, for all $v$ with $\|v\| \leq \|u_\epsilon\|$, we have

$$\|Au_\epsilon - z\| \leq \|Av - z\|.$$

That is, the solution $u_\epsilon$ of equation (B.4) solves the constrained minimization problem

$$\begin{aligned} &\min_{v \in \mathscr{U}} \|Av - z\|^2 \\ &\textit{under the constraint } \|v\| \leq \delta. \end{aligned} \tag{B.7}$$

with $\delta = \|u_\epsilon\|$

Conversely, specifying a *trust region* $\delta \geq 0$, the solution of problem (B.7) satisfies equation (B.4) for some $\epsilon \geq 0$. It follows from the *Kuhn-Tucker necessary conditions* for constrained minimization [35] that if $u$ solves the problem (B.7), there exists an $\epsilon \geq 0$ so that $\{u, \epsilon\}$ is a stationary point of

$$\mathcal{L}(v, \lambda) = \frac{1}{2}\|Av - z\|^2 + \frac{\lambda}{2}(\|v\|^2 - \delta).$$

Setting $\nabla_u \mathcal{L}$ equal to zero yields equation (B.4) and setting $(\partial/\partial\lambda)\mathcal{L} = 0$ yields $\|u\| = \delta$. Thus, if $u$ solves the problem (B.7) for a given $\delta \geq 0$, there exists an $\epsilon \geq 0$ such that $u = u_\epsilon$, where $u_\epsilon$ solves equation (B.6) and $\|u\| = \delta$.

## B.2 Direct Solution Methods

We observed in the previous section that for linear state equations, the minimization problem of interest, (B.5), is equivalent to solving the regularized normal equations (B.4) or (B.6). The advantage of using the conjugate gradient algorithm to solve this problem is that we do not need an explicit matrix representation for $A$; we only need the action of $A$ (or $\tilde{A}$) and $A^T$, computed through the solution of the state and the adjoint equation respectively. By a *direct* method to solve the control problem, we mean a method that uses a matrix representation of $A$, $A^TA$, or $AA^T$ and solves the normal equations (B.4) or (B.6) directly. It turns out that this can be a quite efficient and stable approach when the size of the problem is not too large.

Below we give a short review of three classical, direct methods to solve the regularized least-squares problem. There is a vast literature concerning the numerical aspects of the linear least-squares problem to which we refer the reader for further details and clarifications ([5], [31], [21, Ch. 5], [48, Ch. 5] e.g.). We discuss methods for the problem in the formulation (B.4) only; methods for the problem in the formulation (B.6) will be similar.

Since the matrix on the left-hand side of equation (B.4) is symmetric and positive definite, the system can be solved by a Cholesky-factorization of $\epsilon I + A^TA$. However, explicitly forming the product $A^TA$ might lead to numerical instabilities that can be avoided by using *Householder factorizations* of an augmented matrix $A_L$ instead. For this, we consider the least-squares problem

$$\min_{v \in \mathscr{U}} \|A_L u - z_L\|^2 \tag{B.8}$$

where

$$A_L = \begin{pmatrix} A \\ \epsilon^{1/2} I \end{pmatrix}, \quad z_L = \begin{pmatrix} z \\ 0 \end{pmatrix}.$$

The normal equations associated with (B.8) are $A_L^T A_L u = A_L^T z_L$ which by the definition of $A_L$ and $z_L$ is the same as equation (B.4). Consider the Householder (or $QR$) factorization of $A_L$,

$$A_L = QR = (Q_1, Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix},$$

where $Q$ is orthogonal and $R_1$ is upper triangular. For all $v \in \mathscr{U}$,

$$\begin{aligned} \|A_L v - z_L\|^2 &= \|Q^T(A_L v - z_L)\|^2 = \|Rv - Q^T z_L\|^2 \\ &= \left\| \begin{pmatrix} R_1 \\ 0 \end{pmatrix} v - \begin{pmatrix} Q_1^T z_L \\ Q_2^T z_L \end{pmatrix} \right\|^2 \\ &= \|R_1 v - Q_1^T z_L\|^2 + \|Q_2^T z_L\|^2 \geq \|Q_2^T z_L\|^2, \end{aligned}$$

which means that minimum is attained for $v = u$, where $u$ is the solution to $R_1 u = Q_1^T z_L$.

Another solution technique where $A^T A$ need not to be formed is to use the *singular value decomposition* (SVD)

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T, \tag{B.9}$$

where, if $A$ is an $m$-by-$n$ matrix, $U$ is an orthogonal $m$-by-$m$ matrix, $V$ an orthogonal $n$-by-$n$ matrix, and $\Sigma$ an $n$-by-$n$ diagonal matrix which contains the square root of the eigenvalues of $A^T A$. Such a decomposition exists for each rectangular matrix $A$. Substituting (B.9) in (B.4) yields

$$(\epsilon I + \Sigma^2) V^T u = (\Sigma \; 0) U^T z.$$

Thus, if $U_n$ is the first $n$ columns of $U$, solving equation (B.4) is equivalent to solving

$$(\epsilon I + \Sigma^2) w = \Sigma U_n^T z \tag{B.10}$$

and setting $u = Vw$.

Note that solving equation (B.10) is trivial since the system is diagonal. This is a clear advantage with this method. Once the matrix $A$ has been computed and decomposed, it is very easy and fast to solve for different $\epsilon$. Compared to the SVD, both the Cholesky and the QR factorizations have smaller operation counts for a single application, but these factorizations have to be redone for each value of $\epsilon$. Another advantage with the SVD is that it contains explicit information about the conditioning of the problems in terms of the singular values $\Sigma$.

Regardless of which of these direct methods that is used, note also that it is easy to change the target since all information about the target function is contained in the vector $z$. This is in sharp contrast to when the conjugate gradient algorithm is used, when all the the work has to be redone for each target.

## B.3 Computing a Matrix Representation for the Forward Map

The drawback with the approach sketched above is that the matrix $A$ has to be explicitly constructed which can be both computationally expensive and memory demanding. The matrix can be computed, column by column, by applying unit vectors from $\mathcal{U}$ to the state equation and computing the corresponding final states. In general, this will be a very costly calculation.

However, for problems with a certain structure, the matrix representation for the forward map is quite cheap to compute. Recall that the forward map $A$, as defined in Section B.1, is the composition of applying the control $v \in \mathcal{U}$, solving the *homogeneous* part of the state equation, and observing the final state $y_N$. Let us partition an element $v \in \mathcal{U}$ in the following way

$$v = \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix},$$

where the control at time level $n$, $v_n$, is in some finite-dimensional space $\mathcal{U}_\Delta$, whose dimension is the *spatial degrees of freedom* for the control. Given a control $v \in \mathcal{U}$, we consider the solution of an abstract state equation

$$\begin{aligned} &y_0 = 0; \\ &\text{for } n = 1, \ldots, N \\ &\qquad y_n = E_n(y_{n-1} + Bv_n), \end{aligned} \tag{B.11}$$

where $E_n \in \mathcal{L}(F)$ denotes the linear "solution procedure" at each time step, and $B \in \mathcal{L}(\mathcal{U}_\Delta, F)$ the linear application of the control through a boundary condition or a forcing term. The homogeneous part[8] of the state equation (3.15) is of the type (B.11). Likewise, the homogeneous part of the fully discrete analogue of (4.4) is also of the form (B.11). In the latter case we have something stronger, namely *time invariance*; the "solution procedure" $E_n$ does not change with $n$. In such a case, when $E_n = E$, we can conclude by induction that the final state is the discrete convolution

$$y_N = E^N B v_1 + E^{N-1} B v_2 + \cdots + EBv_N = Av,$$

where

$$A = (E^N B, E^{N-1} B, \ldots, EB). \tag{B.12}$$

Thus, for time invariant state equations, the forward map has the block *Krylov structure* evident in (B.12), and to compute $E^n B$, we can simply apply $E$ to $E^{n-1}B$. Making use of this recursive structure, the total work involved in computing $A$ corresponds to solving the state equation $\dim(\mathcal{U}_\Delta)$ times (recall that the column dimension of $B$ is $\dim(\mathcal{U}_\Delta)$). As a comparison, when the state equation is time-variant, corresponding calculation corresponds to solving the state equation about $N \dim(\mathcal{U}_\Delta)/2$ times.

To summarize: Explicitly computing a matrix representation for the forward map will be computationally cheap if the state equation is time invariant and if the spatial degrees of freedom for the control—the parameter $\dim(\mathcal{U}_\Delta)$—is small.

[8] That is, (3.15) with $y_{0h} = 0$, $f^n = 0$, and $g^n = 0$ for all $n$.

## Acknowledgements

We would like to acknowledge the helpful comments and suggestions from the following individuals: J.O. Blanch, C. Dalton, E.J. Dean, T. Kearsley, J.L. Lions, D.C. Sorensen, and W.W. Symes.

The support from the following institutions is also acknowledged; CERFACS, Rice University, and the University of Houston. We also benefited from the support of the Texas Board of Higher Education (Grants ARP 003652091, ATP 003652146).

# References


[1] F. Abergel and R. Temam. "On some control problems in fluid mechanics". *Theoretical and Computational Fluid Dynamics*, 1:303–326, 1990.

[2] M. Al-Baali. "Descent Property and Global Convergence of the Fletcher-Reeves Method with Inexact Line Search". *IMA Journal of Numerical Analysis*, 5:121–124, 1985.

[3] G.V. Alekseyev and V.V. Malikin. "Numerical Analysis of Optimal Boundary Control Problems for the Stationary Navier-Stokes Equations". *Computational Fluid Dynamics Journal*, 3(1):1–26, 1994.

[4] M. Berggren. "Control and Simulation of Advection-Diffusion Problems". Master's Thesis, Mechanical Engineering, University of Houston, 1992.

[5] Å. Björk. "Solution of Equations in $\mathbb{R}^N$ (Part 1)". In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis, Volume 1.* North-Holland, Amsterdam, 1990.

[6] R.B. Buchanan and R.L. Bras. "A study of real time adaptive closed-loop reservoir control algorithm". In *Optimal Allocation of Water Resources*, pages 317–326, 1982. IAHS Publication no. 135.

[7] J.A. Burns and S. Kang. "A stabilization problem for Burgers' equation with unbounded control and observation". In *Estimation and control of distributed parameter systems: proceedings of an International Conference on Control and estimation of Distributed Parameter Systems, Vorau, July 8–14, 1990.* International Series of Numerical Mathematics vol. 100, Birkhauser Verlag, Basel, 1991.

[8] J.A. Burns and H. Marrekchi. "Optimal fixed-finite-dimensional compensator for Burgers' Equation with unbounded input/output operators". Technical Report 93-19, ICASE, 1993.

[9] D.M. Buschnell and J.N. Hefner. "Viscous Drag Reduction in Boundary Layers". American Institute of Aeronautics and Astronautics, Washington DC, 1990.

[10] J. Cahouet and J.P. Chabard. "Some Fast 3D Solvers for the Generalized Stokes Problem". *International Journal of Numerical Methods in Fluids*, 8:869–895, 1988.

[11] H. Choi, R. Temam, P. Moin, and J. Kim. "Feedback control for unsteady flow and its application to the stochastic Burgers equation". *Journal of Fluid Mechanics*, 253:509–543, 1993.

[12] R. Dautray and J.L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology*, volume 5: Evolution Problems I. Springer, Berlin, 1992.

[13] E.J. Dean and P. Gubernatis. "Pointwise Control of Burgers' Equation—A Numerical Approach". *Computers and Mathematics with Applications*, 22(7):93–100, 1991.

[14] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Non-linear Equations.* Prentice-Hall, Englewood Cliffs, N.J., 1983.

[15] E. Anderson *et al. Lapack Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[16] R. Fletcher and C.M. Reeves. "Function minimization by conjugate gradients". *Computer Journal*, 7:149–154, 1964.

[17] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer, Berlin, 1986.

[18] R. Glowinski. "Finite Element Methods for the Numerical Solution of Incompressible Viscous Flow. Introduction to the Control of the Navier-Stokes Equations". In *Lectures in Applied Mathematics*, volume 28, pages 219–301. American Mathematical Society, Providence, RI, 1991.

[19] R. Glowinski and J.L. Lions. "Exact and approximate controllability for distributed parameter systems (Part I)". *Acta Numerica*, pages 269–378, 1994.

[20] R. Glowinski and J.L. Lions. "Exact and approximate controllability for distributed parameter systems (Part II)". *Acta Numerica*, 1995.

[21] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, 1989.

[22] A. Griewank. "Achieving Logarithmic Growth of Temporal and Spatial Complexity in Reverse Automatic Differentiation". *Optimization Methods and Software*, 1:35–54, 1992.

[23] M.D. Gunzburger, L.S. Hou, and T.P. Svobodny. "Numerical Approximation of an Optimal Control Problem Associated with the Navier-Stokes Equations". *Applied Mathematics Letters*, 2(1):29–31, 1989.

[24] M.D. Gunzburger, L.S. Hou, and T.P. Svobodny. "Analysis and finite element approximation of optimal control problems for the stationary Navier-Stokes equations with distributed and Neumann controls". *Mathematics of Computation*, 57:123–151, 1991.

[25] M.D. Gunzburger, L.S. Hou, and T.P. Svobodny. "Boundary velocity control of incompressible flow with an application to viscous drag reduction". *SIAM Journal on Control and Optimization*, 30:167–181, 1992.

[26] C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, Cambridge, 1987.

[27] M. Kawahara and Y. Shimada. "Optimal control with constraints on the control value to operate water gate of dam". In *Proc. Finite Elements in Fluids, New Trends and Applications*, pages 1272–1280, Barcelona, 1993.

[28] M. Kawahara and Y. Shimada. "Gradient Method of Optimal Control Applied to the Operation of a Dam Water Gate". *International Journal for Numerical Methods in Fluids*, 19:463–477, 1994.

[29] T. Kawasaki and M. Kawahara. "A flood control of dam reservoir by conjugate gradient method and finite element method". In T.J. Chung and G.R. Karr, editors, *Finite Element Analysis in Fluids*, pages 629–634. University of Alabama in Huntsville Press, Huntsville AL, 1989.

[30] R. Kress. *Linear Integral Equations.* Springer-Verlag, Berlin, 1989.

[31] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[32] J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications.* Springer, Berlin and New York, 1972.

[33] J.L. Lions. *Controlabilité Exacte, Perturbation et Stabilisation des Systèmes Distribués.* Masson, Paris, 1988.

[34] J.L. Lions. "Exact Controllability, Stabilization and Perturbations for Distributed Systems". *SIAM Review*, 30(1):1–68, March 1988.

[35] D.G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading Massachusetts, 1989.

[36] K. McManus, T. Poinsot, and S. Candel. "Review of active control of combustion instabilities". *Progress in Energy and Combustion Science*, 19:1–29, 1993.

[37] J.J. Moré. "The Levenberg–Marquardt algorithm: implementation and theory". In G.A. Watson, editor, *Numerical Analysis*, pages 105–116. Lecture Notes in Math. 630, Springer-Verlag, Berlin, 1977.

[38] J. Muskatirovic and R. Kapor. "Analysis of the control of floods caused by the failure of a cascade system of dams". In *Proc. 2nd Int. Conf. on the Hydraulics of Floods and Flood Control*, pages 49–61, 1985.

[39] M. Papageorgio. "Optimal multireservoir network control by the discrete maximum principle". *Water Resources Research*, 21:1824–1830, 1985.

[40] O. Pironneau. *Finite Element Methods for Fluids.* J. Wiley, Chichester, 1989.

[41] E. Polak. *Computational Methods in Optimization.* Academic Press, New York, 1971.

[42] R.H. Sellin and T. Moses. *Drag Reduction in Fluid Flows.* Ellis Horwood, Chichester, 1989.

[43] Y. Shimada, M. Kawahara, and T. Umetsu. "Adaptive control methods to operate the water gate in dam as the practical problem". In *Proc. Forth Int. Conf. on Computing in Civil and Building Engineering*, page 324, Tokyo, 1991.

[44] Y. Shimada, M. Kawahara, and T. Umetsu. "Adaptive control to operate water gate of dam". In *Proc. Asian Pacific Conf. on Computational Mechanics*, pages 1749–1754, Hong Kong, 1991.

[45] B.K. Shivamoggi. *Theoretical fluid dynamics.* Martinus Nijhoff Publishers, Dordrecht, 1985.

[46] S.S. Sritharan. "An optimal control problem for exterior hydrodynamics". In G. Chen, E.B. Lee, W. Littman, and L. Markus, editors, *Distributed Parameter Control Systems: New Trends and Applications*, pages 385–417. Marcel Dekker, New York, 1991.

[47] S.S. Sritharan. "Dynamic programming of Navier-Stokes equations". *Syst. Control Lett.*, 16:299–307, 1991.

[48] G.W. Stewart. *Introduction to Matrix Computations.* Academic Press, San Diego, 1973.

[49] P. Swarztrauber. "A Direct Method for the Solution of Separable Elliptic Equations". *SIAM Journal of Numerical Analysis*, 11(6):1136–1150, 1974.

[50] W.W. Symes and J.O. Blanch, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892. Private Communications, November 1994.

[51] Y. Tanaka and R. Kawahara. "Control of flood using finite element method and optimal control theory". In T.J. Chung and G.R. Karr, editors, *Finite Element Analysis in Fluids*, pages 617–622. University of Alabama in Huntsville Press, Huntsville AL, 1989.

[52] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-Posed Problems.* V.H. Winston and Sons, Washington, D.C., 1977.

[53] T. Umetsu, Y. Tanaka, and M. Kawahara. "Optimal control of flood using finite element method". *Proceedings of JSCE*, 9(1):11s–23s, 1992.