

Suboptimal feedback control of flow separation by POD model reduction*

K. Kunisch and L. Xie

1 Abstract

The model reduction technique of proper orthogonal decomposition is applied to approximate the NS equations by a set of ODEs. Then feedback control is performed on this reduced order model by numerically solving the dynamic programming equation. In the numerical implementation, a multi-level method is applied to accelerate convergence of the fixed point iteration. Moreover to save computational time parallel computation is carried out, in which message passing interface is utilized to transfer data between the master computer and the slaves. The numerical results are presented for the distributed volume control of the incompressible viscous flow over a forward-facing step with an inlet on the upper channel. The feasibility of this method is also demonstrated by the computational results from a boundary control problem.

2 Introduction

Optimal design for fluid dynamical systems, which is a useful computational tool in aerodynamic drag reduction, turbulence delay and combustion control, etc., has been an interesting research area for many years. This research stands at the crossroads of optimization (to carry out optimal design) and computational fluid dynamics. Many computational approaches and theoretical investigations were carried out in the past two decades, such as [6, 8, 12, 21], to name a few.

*Supported by the Fonds zur Förderung der wissenschaftlichen Forschung under SFB 03 "Optimierung und Kontrolle".

Generally speaking it is unfeasible to carry out feedback design on infinite dimensional fluid dynamical systems by numerically solving the dynamic programming equation (DPE) or the Hamilton-Jacobi-Bellman (HJB) equation. The development of *proper orthogonal decomposition* (POD) techniques in recent years provides a promising approach to perform suboptimal feedback design on reduced order models. In this technique, a set of basis functions is constructed from pre-selected snapshots. Subsequently the fluid dynamic system is projected onto the space spanned by the basis functions, and the infinite-dimensional nonlinear partial differential equations (PDEs) of the fluid system are reduced to finite dimensional ordinary differential equations (ODEs). Then the optimal design is performed on this reduced system at reasonable computational cost. This suboptimal approach has been applied for open-loop design on the 1-D Burgers equation in [13] and on the 2-D NS equations with different geometrical shapes in [9, 19, 20]. POD based optimal control for distributed parameter systems was also investigated in [16] and [1, 2], for example. In [11] Ito and Schröter and in [13] Kunisch and Volkwein implemented an approach to carry out feedback control for distributed parameter systems on the basis of model reduction combined with solving the optimality systems for an array of open loop problems and interpolating the results thus obtained to define the closed loop synthesis. In [7] it was proved for the 2-D NS equations that the value function is the unique viscosity solution to the HJB equations. In [10] a suboptimal solution strategy for conservative systems (such as the NS equations) based on an ansatz for a solution to the HJB equation is proposed and in the authors' previous work [14, 15] solving the HJB equations for a POD-based reduced system of the Burgers equation is investigated.

The prohibitive computational cost of feedback design for nonlinear systems strongly suggests to profit from parallel computations. In this work several computers are exploited concurrently; furthermore, on different computers, a different number of MATLAB sessions is evoked, depending on the overall performance and the available computational power of each computer. The MATLAB based message passing interface is utilized to interchange data between MATLAB sessions on different computers. This parallel implementation reduces the computing time to an acceptable level.

The paper is arranged as follows. The distributed volume control problem and the numerical NS solver are briefly described in Section 3. Then the POD technique is applied in Section 4 to reduce the NS equations to a set of nonlinear ODEs, on which feedback design will be carried out. Numerical methods to solve the discretized DPE are discussed in Section 5. The computational results from the volume control are presented in Section 6. In these sections, the methodology is described for the distributed volume control problem. But it can also be applied to the boundary control problem, which is addressed in Section 7.

3 Distributed volume control problem

The dynamical system to be considered is incompressible viscous flow over a forward-facing step, with a uniform inlet on the upper bound of the channel. Let $\Omega \in R^2$

denote the flow domain with boundary Γ and let $y = (u, v)^T$ stand for the velocity. Then we have the NS equations in non-dimensional form,

$$\begin{aligned} y_t - \frac{1}{Re} \Delta y + (y \cdot \nabla) y + \nabla p &= f \\ \nabla \cdot y &= 0 \end{aligned} \quad (1)$$

in the domain $\Omega \times (0, \infty)$, where p denotes pressure and Re is the Reynold's number. Associated with the above governing equation we have the following boundary conditions

$$\begin{aligned} y &= y_i & \text{in } \Gamma_i \times [0, \infty) \\ pn - \frac{1}{Re} \frac{\partial y}{\partial n} &= 0 & \text{in } \Gamma_o \times [0, \infty) \\ y &= c(t) & \text{in } \Gamma_c \times [0, \infty) \\ y &= 0 & \text{in } \Gamma \setminus (\Gamma_i \cup \Gamma_o \cup \Gamma_c) \times [0, \infty), \end{aligned} \quad (2)$$

where Γ_i, Γ_o are the inflow and the outflow surfaces, respectively, Γ_c is the inlet and n denotes the unit normal to the boundary. The boundary condition on the outflow boundary is the stress-free condition. The initial condition is $y(0) = y_0$.

In this distributed volume control problem, the right hand side of the first component of the NS equations is set as a linear combination of a set of shape functions, $f = \Psi \nu(t)^T$, and the coefficients $\nu(t)$ are taken as the design variables. The cost functional to be minimized is defined as:

$$\mathcal{J} = \frac{1}{2} \left(\int_0^\infty \int_{\Omega_o} e^{-\mu t} \|y - y_{Stokes}\|^2 d\Omega dt + \int_0^\infty \beta e^{-\mu t} \|\nu\|^2 dt \right), \quad (3)$$

where the positive constants β and μ are the weight coefficient and the discount rate, respectively. The first term in (3) is of tracking type. Here y_{Stokes} denotes the solution to the Stokes equation associated to (1) with otherwise the same geometric and data settings. Since the Stokes solution is separation-free, a reduction of the recirculation regions should be observed for the successfully controlled NS solution. By Ω_o we denote the observation region. The set of admissible controls is

$$U_{ad} = \{\nu \in L^2(0, T) : \nu_a \leq \nu \leq \nu_b\}.$$

The weak form of the NS equations is given by

$$\begin{aligned} \langle y_t + (y \cdot \nabla) y, \psi \rangle_\Omega + \frac{1}{Re} \langle \nabla y, \nabla \psi \rangle_\Omega - \langle \nabla \cdot \psi, p \rangle_\Omega &= \langle f, \psi \rangle_\Omega \\ \langle \nabla \cdot y, q \rangle_\Omega &= 0 \end{aligned} \quad (4)$$

with boundary conditions

$$\begin{aligned} y &= y_i & \text{in } \Gamma_i \times [0, \infty), \\ y &= c(t) & \text{in } \Gamma_c \times [0, \infty), \\ y &= 0 & \text{in } \Gamma \setminus (\Gamma_i \cup \Gamma_o \cup \Gamma_c) \times [0, \infty), \end{aligned} \quad (5)$$

and initial condition

$$y(0) = y_0, \quad (6)$$

for all test function pairs $(\psi, q) \in V \times L^2(\Omega)$, where

$$V = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma \setminus \Gamma_o\} \quad (7)$$

and $\langle \cdot, \cdot \rangle_\Omega$ denotes the L_2 inner product over the flow domain Ω . This weak formulation will be utilized to obtain the reduced-order model for the dynamic system.

In the numerical NS solver, a finite difference method is applied to approximate the spatial differentiation over a non-uniform grid system, with an upwind scheme for the nonlinear convection term. A staggered grid system is applied, i.e., the horizontal velocities are located at the midpoints of the vertical cell edges, the vertical velocities are located at the midpoints of the horizontal cell edges and the pressure is defined at each cell center. At the inflow boundary, only the horizontal velocity is specified as a parabolic distribution with maximum equal to 1. The flow is initialized with the stationary Stokes flow. For time integration, we used the fractional-step- θ -scheme as applied in [22]. A macro time step from t^n to $t^{n+1} = t^n + \Delta t$ is split into three sub-steps, $t^n \rightarrow t^{n+\theta} \rightarrow t^{n+1-\theta} \rightarrow t^{n+1}$:

$$\begin{aligned} (I + \alpha \theta \Delta t D(y^{n+\theta}))y^{n+\theta} + \theta \Delta t \nabla p^{n+\theta} &= (I - \beta \theta \Delta t D(y^n))y^n + \theta \Delta t f^n \\ \nabla \cdot y^{n+\theta} &= 0, \\ (I + \beta \theta' \Delta t D(y^{n+1-\theta}))y^{n+1-\theta} + \theta' \Delta t \nabla p^{n+1-\theta} &= (I - \alpha \theta' \Delta t D(y^{n+\theta}))y^{n+\theta} + \theta' \Delta t f^{n+1-\theta} \\ \nabla \cdot y^{n+1-\theta} &= 0, \\ (I + \alpha \theta \Delta t D(y^{n+1}))y^{n+1} + \theta' \Delta t \nabla p^{n+1} &= (I - \beta \theta \Delta t D(y^{n+1-\theta}))y^{n+1-\theta} + \theta \Delta t f^{n+1} \\ \nabla \cdot y^{n+1} &= 0, \end{aligned} \quad (8)$$

where D denotes the diffusive and advective terms, $\theta = 1 - \frac{\sqrt{2}}{2}$, $\theta' = 1 - 2\theta$ and $\alpha = \frac{1-2\theta}{1-\theta}$, $\beta = 1 - \alpha$. This fractional step scheme combines advantages from the classical Crank-Nicolson method (2nd order accuracy) and the backward Euler method (strongly A-stable).

To avoid solving nonlinear equations and to ensure second order accuracy in time, the nonlinear convective term $y^{n+1} \cdot \nabla y^{n+1}$ in the left hand side of the first component of (8) is replaced by $(2y^n - y^{n-1}) \cdot \nabla y^{n+1}$, which is of second order accuracy.

4 POD-based reduced-order model

It is not yet possible to carry out feedback optimal control directly on the full NS equations. A suboptimal alternative is to utilize a reduced order model. In this section, the principles of proper orthogonal decomposition will be introduced to construct the reduced order model for incompressible NS equations. In Section 4.1 proper orthogonal decomposition will be briefly introduced. Then this principle is applied to the NS equation to get the reduced order model, which is presented in Section 4.2. Section 4.3 describes the method to obtain the shape functions for the distributed volume control.

4.1 Proper orthogonal decomposition

POD emerges from choosing a 'best' set of orthonormal basis functions from a given collection frequently referred to as the snapshots. It can be readily verified that this

problem is equivalent to an eigenvalue problem. Readers are referred to [1, 2, 4, 16], and the references given there, for more details. We will only briefly sketch the methodology of POD and recapitulate its properties.

Let the pre-selected collection of snapshots be denoted by

$$S = \{S_j, j = 1, 2, \dots, N | S_j \in H^1(\Omega)\}$$

and suppose that S has rank l , that is, $l = \dim\{\text{span}\{S\}\}$. If $\{\phi_i\}_{i=1}^l$ is an orthonormal basis for S , each element in S can be expressed as $S_j = \sum_{i=1}^l \langle S_j, \phi_i \rangle \phi_i$ for $j = 1, 2, \dots, N$. The orthonormal basis can be constructed in many ways. However, in POD we require the orthonormal basis to have the property of minimizing

$$\sum_{j=1}^N \|S_j - \sum_{i=1}^k \langle S_j, \phi_i \rangle \phi_i\|^2$$

for every $k \in \{1, 2, \dots, l\}$. Introduce the bounded linear operator $\mathcal{S} : R^n \mapsto L_2$

$$\mathcal{S}v = \sum_{j=1}^N v_j S_j.$$

Its adjoint $\mathcal{S}^* : L_2 \mapsto R^n$ is given by

$$\mathcal{S}^*z = (\langle z, S_1 \rangle, \dots, \langle z, S_N \rangle)^T.$$

The necessary optimality condition for this constrained minimization problem requires that basis functions satisfy

$$\mathcal{S}\mathcal{S}^*\phi = \lambda\phi.$$

Since $\mathcal{S}\mathcal{S}^*$ is non-negative and symmetric, its eigenvalues must be real and non-negative.

In numerical implements, we do not need to calculate the matrix $\mathcal{S}\mathcal{S}^*$; instead, the *singular value decomposition* (SVD) will be applied to \mathcal{S} such that $\mathcal{S} = U\Sigma V^T$, and the POD basis function is given by [13]

$$\phi_i = \frac{1}{\Sigma_i} \mathcal{S}v_i, 1 \leq i \leq l, \tag{9}$$

where Σ_i is the i^{th} singular value in Σ and v_i is the associated right eigenvector in V .

Let us reiterate that the POD-basis is the optimal orthonormal basis approximating in the mean all snapshots at arbitrary truncation level. In view of our application to the Navier-Stokes equations let us note that the basis elements inherit the boundary and incompressibility condition of the snapshots.

Our next task is to properly construct the snapshots from which the basis functions will be deduced. There are several requirements for the snapshots: firstly they must capture as much dynamics of the flow system as possible, since the basis

functions inherit the information already contained in the snapshots. Secondly, they should satisfy homogeneous boundary conditions, and thirdly the snapshots must be solenoidal for incompressible flow.

The immediate idea is to take the snapshots as the solutions of the NS equations at specified time instances. To satisfy the second requirement, a term accounting for the boundary conditions is subtracted from the instant solutions. With these considerations in mind, we can propose the snapshots

$$\tilde{S}_k = \{y(t^k) - \bar{y}\}, k = 1, 2, \dots, N,$$

where $y(t^k)$ is the NS solution at $t = t^k$ and the mean flow \bar{y} is given by

$$\bar{y} = \frac{1}{N} \sum_{k=1}^N y(t^k).$$

These snapshots satisfy homogeneous boundary conditions on the Dirichlet boundaries and the stress-free boundary condition is also satisfied by the snapshots on the outflow boundary. Furthermore to account for the effect of the nonuniform grid system, a diagonal matrix $\mathcal{M} \in R^{m \times m}$ is introduced, whose entries are the measures of the grid cells; hence the snapshot matrix now takes the following form

$$S = \{\mathcal{M}^{\frac{1}{2}} \tilde{S}_1, \mathcal{M}^{\frac{1}{2}} \tilde{S}_2, \dots, \mathcal{M}^{\frac{1}{2}} \tilde{S}_N\}.$$

Then the singular value decomposition is applied to the snapshot matrix using the MATLAB subroutine *svds* to find the largest $\ell < N$ eigenvalues and the corresponding eigenvectors,

$$S^\ell = P_\ell \Sigma_\ell Q_\ell^T, \quad (10)$$

where $P_\ell \in R^{m \times \ell}$, $\Sigma_\ell \in R^{\ell \times \ell}$ and $Q_\ell \in R^{N \times \ell}$. Actually S^ℓ is the best rank- ℓ approximation of the snapshot matrix S , that is we perform truncated SVD.

Let the column of Q_ℓ be $w_i, i = 1, 2, \dots, \ell$ and let the singular value in Σ_ℓ be denoted by $\sigma_i, i = 1, 2, \dots, \ell$. To determine the number of basis elements for the actual computations we choose $M \leq \ell$ such that $r = \Sigma_{k=1}^M \sigma_k^2 / \Sigma_{k=1}^\ell \sigma_k^2 > 99.5\%$. Then the basis functions can be expressed as

$$\phi_i = \frac{\tilde{S} w_i}{\sigma_i}, i = 1, 2, \dots, M.$$

The ration r is the percentage of the total energy captured in the first M POD basis functions, and therefore is an index to measure how close the M -dimensional subspace $\{\phi_1, \phi_2, \dots, \phi_M\}$ approximates the snapshot set. In this way the basis functions are orthonormal with respect to the L_2 norm. Furthermore, they are solenoidal and satisfy homogeneous boundary conditions on the Dirichlet boundaries and the stress-free boundary condition on the outflow boundary. These properties are inherited from the snapshots. The solution to NS equations is therefore be modelled by

$$\hat{y} = \bar{y} + \Sigma_{i=1}^M \alpha_i(t) \phi_i, \quad (11)$$

with $\alpha_i(t)$ to be determined from a Galerkin procedure.

4.2 Reduced order model

With the basis functions in hand, the reduced order model can be derived by restricting the modelled solution (11) to the weak form (4) of the NS equations. Furthermore, recalling that the basis functions are solenoidal and satisfy homogeneous boundary conditions on the Dirichlet boundaries, we can arrive at the ODE governing α

$$\dot{\alpha} = -\mathcal{A}\alpha - (\alpha^T H\alpha + P) + G, \quad (12)$$

where the components of matrices are determined as follows,

$$\begin{aligned} \mathcal{A}_{i,j} &= \frac{1}{Re} \langle \nabla \phi_j, \nabla \phi_i \rangle + \langle \phi_j \cdot \nabla \bar{y}, \phi_i \rangle + \langle \bar{y} \cdot \nabla \phi_j, \phi_i \rangle, \\ H_{i,j,k} &= \langle \phi_j \cdot \nabla \phi_k, \phi_i \rangle, \\ P_i &= \langle \bar{y} \cdot \nabla \bar{y}, \nabla \phi_i \rangle + \frac{1}{Re} \langle \nabla \bar{y}, \nabla \phi_i \rangle, \\ G_i &= \langle \Psi \nu^T, \phi_i \rangle. \end{aligned}$$

The boundary integration term on the outflow boundary is cancelled off since we chose stress-free outflow boundary conditions. The initial conditions for the reduced system can be derived from evaluating (11) at the initial time $t = 0$ by

$$y(0) = \bar{y} + \sum_{i=1}^M \alpha_i(0) \phi_i,$$

which can be put in the following form

$$\alpha_i(0) = \langle y_0 - \bar{y}, \phi_i \rangle, i = 1, 2, \dots, M. \quad (13)$$

Classical Galerkin approximations are typically obtained via sets of test functions that are not related to the dynamic system. Hence the resulting finite element model is a system with large number of unknowns. Contrary to that approach, the POD technique takes basis functions which reflect the system dynamics. In such a way, the reduced order model can be of much smaller scale. The stiffness matrix is full in this case, while for finite element or finite difference approximations it is typically sparse. This reduced order model is a nonlinear initial value problem, on which the feedback design will be carried out to minimize the cost functional,

$$\hat{\mathcal{J}} = \frac{1}{2} \int_0^T \int_{\Omega} e^{-\mu t} \|\hat{y} - y_{Stokes}\|^2 d\Omega dt + \frac{\beta}{2} \int_0^T e^{-\mu t} \|\nu\|^2 dt, \quad (14)$$

where \hat{y} is given by (11), and the α_i 's are the solutions to (12) with the initial condition (13).

4.3 Construction of shape functions

In the control problem, the distributed volume control is achieved by the linear combination of a set of shape functions, which are constructed in a similar way as to construct the basis functions for the states. First a set of snapshots is defined as

$$\mathcal{S} = y^0(t^k) - y_{Stokes}, \quad (15)$$

where $y^0(t^k)$ is the time dependent solution to the NS equations without any control, and y_{Stokes} is the Stokes solution, both these terms are under the action of the inlet. The same method as stated in Section 4.1 is then followed to find the basis functions of these snapshots; and the first two of them are taken as the shape functions. With the snapshot constructed in this way, the regions are filtered out for the separation bulbs. Therefore, this method will be very efficient, as can be seen from the numerical tests to be presented.

5 Feedback design for the reduced model

Feedback control can now be performed on the reduced ODE system. In this section, we first discretize the dynamic programming equation in time and spatial domain. Then a multi-level acceleration method and a parallel computation method are briefly described. Finally the method of retrieving the synthesis or the feedback law is introduced.

5.1 Discretization of DPE

For the optimal control problem

$$\begin{aligned} \min_{\nu \in U_{ad}} J &= \int_0^\infty e^{-\mu t} L(\alpha(t), \nu(t)) dt \\ \text{s.t. } \dot{\alpha} &= F(\alpha(t), \nu(t)), \quad \alpha(0) = \alpha_0 \in R^M, \end{aligned}$$

where $L : R^M \times R^K \mapsto R$ and $F : R^M \times R^K \mapsto R^M$, the value function can be defined as

$$\mathfrak{S}(\alpha_0) = \inf_{\nu \in U_{ad}} \int_0^\infty e^{-\mu t} L(\alpha, \nu) dt : R^M \mapsto R,$$

for any initial value α_0 . The principle of optimality by Bellman and a simply argument based on the semigroup property lead to the *dynamic programming equation* (or *dynamic programming principle*) [3]

$$\mathfrak{S}(\alpha_0) = \inf_{\nu \in U_{ad}} \left\{ \int_0^{\Delta T} e^{-\mu t} L(\alpha(t), \nu(t)) dt + \mathfrak{S}(\alpha(\Delta T)) e^{-\mu \Delta T} \right\} \quad (16)$$

for any $\Delta T > 0$ and any $\alpha_0 \in R^M$.

To derive the discrete dynamic programming equation, we assume that the control ν is constant in the time interval $[0, \Delta T]$. Thus we have to the first order accuracy

$$\begin{aligned} \mathfrak{S}(\alpha_0) = \inf_{\nu \in U_{ad}} \left\{ (L(\alpha(0), \nu) + e^{-\mu \Delta T} L(\alpha(\Delta T), \nu)) \frac{\Delta T}{2} \right. \\ \left. + \mathfrak{S}(\alpha(\Delta T)) e^{-\mu \Delta T} \right\}, \end{aligned} \quad (17)$$

where the trapezoid rule is applied for the integration of the cost functional on the interval $[0, \Delta T]$. Next the problem arises of calculating $\alpha(\Delta T)$ from the initial

condition $\alpha(0) = \alpha_0$. There are several candidates for this, such as the explicit Euler method, the modified Euler method (Heun method) and semi-implicit methods. In the numerical tests, an explicit scheme is applied based on a Runge-Kutta (4, 5) formula (the Dormand-Prince pair).

Then we introduce a bounded polyhedron $\Upsilon = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_M, b_M]$. Along each of its dimensions we distribute equidistant grid points to obtain a discretized polyhedron Υ_h . Let us denote these grid points by $\alpha_{0_j}, j = 1, 2, \dots, \bar{h}$. Rewriting the semi-discretized dynamic programming equation (17) at each grid point gives

$$\mathfrak{S}(\alpha_{0_j}) = \inf_{\nu \in U_{ad}} \left\{ (L(\alpha_{0_j}, \nu) + e^{-\mu\Delta T} L(\alpha_j(\Delta T), \nu)) \frac{\Delta T}{2} + \mathfrak{S}(\alpha_j(\Delta T)) e^{-\mu\Delta T} \right\} \quad (18)$$

where $\alpha_j(\Delta T)$ is the trajectory at $t = \Delta T$ from the initial position α_{0_j} .

The viability assumption is made for the polyhedron, i.e., for sufficiently small δt

$$x + \delta t F(x, u) \in \Upsilon \quad \forall y \in \Upsilon, \text{ and } \quad \forall u \in U_{ad}. \quad (19)$$

Since for our control problem, where no a priori constraint is imposed on the system state, the value function is continuous over the whole domain [3, Chapter 3], we can find a piecewise linear approximate solution to (18). For example the value function can be approximated at $\alpha_j(\Delta T)$ by the convex combination

$$\mathfrak{S}(\alpha_j(\Delta T)) = \sum_{i=1}^{\bar{h}} \Lambda_{ij} \mathfrak{S}(\alpha_{0_i}),$$

where the coefficients Λ_{ij} satisfy

$$0 \leq \Lambda_{ij} \leq 1 \quad \text{and} \quad \sum_{i=1}^{\bar{h}} \Lambda_{ij} = 1.$$

The construction of these coefficients is straightforward. In this way we arrive at the fully discretized dynamic programming equation

$$\mathfrak{S}(\alpha_{0_j}) = \inf_{\nu \in U_{ad}} \left\{ (L(\alpha_{0_j}, \nu) + e^{-\mu\Delta T} L(\alpha_j(\Delta T), \nu)) \frac{\Delta T}{2} + e^{-\mu\Delta T} \sum_{i=1}^{\bar{h}} \Lambda_{ij} \mathfrak{S}(\alpha_{0_i}) \right\} \quad (20)$$

for each $j = 1, 2, \dots, \bar{h}$. It can be proved [3, Appendix A] that the above equation is equivalent to the fixed point problem

$$\mathfrak{S}(\alpha_0) = \mathcal{T}(\mathfrak{S}(\alpha_0)) \quad (21)$$

with the map $\mathcal{T} : R^{\bar{h}} \mapsto R^{\bar{h}}$ defined by

$$\mathcal{T}(\alpha_{0_j}) = \inf_{\nu \in U_{ad}} \left\{ (L(\alpha_{0_j}, \nu) + e^{-\mu\Delta T} L(\alpha_j(\Delta T), \nu)) \frac{\Delta T}{2} + e^{-\mu\Delta T} \sum_{i=1}^{\bar{h}} \Lambda_{ij} \mathfrak{S}(\alpha_{0_i}) \right\}. \quad (22)$$

Close examination reveals that the map \mathcal{T} is a contraction map with respect to the maximum norm in R^h with contraction coefficient given by $e^{-\mu\Delta T}$. Consequently, the discretized equation (20) has a unique solution, which can be found by the common technique of fixed point iteration.

5.2 Numerical methods

The fixed point iteration begins at an initial guess to the value functions, which is zero in our tests. After the constrained minimization problem on the right hand side of (20) is solved, the minimum is set as the new guess to the value function. This iteration is repeated until the stopping criterion is reached. The convergence rate of the fixed point iteration depends on $\mu\Delta T$. When $\mu\Delta T$ approaches 0 the contraction coefficient $e^{-\mu\Delta T}$ tends to 1, and therefore only very slow convergence behavior will be observed. Falcone [3, Appendix A] suggested several techniques to speed up convergence for the fixed point iteration. But they are too expensive to carry out in our numerical tests.

However, in this study by using a multi-level method, a different method for speed-up is carried out. In this strategy the HJB equation is first solved for $\Delta T = 0.2$ until the relative residual is decreased to less than a prescribed tolerance. These results are taken as the initial guess to solve the HJB equations for $\Delta T = 0.05$.

In solving the dynamic programming equation, a constrained minimizing problem must be solved independently at each point of the polyhedron. For these tasks it is possible to perform the data parallel algorithm, also known as "fine-grained parallelism" [5]. In this parallelism, the same set of codes runs simultaneously on different pieces of data on various processors. The technique of MPI (*Message Passing Interface*) [17] is used in the current numerical tests. The mechanism used in MPI to distribute data (or information) is through explicit sending and receiving of data among the processors. Interested readers are referred to [18] for additional information. The Gauss-Seidel scheme can also be used to accelerate convergence, i.e., the newest available unknowns are utilized to calculate the remaining unknowns at each iteration. But this technique does not lend itself to parallel computations.

In practical implementations, the slaves are first started remotely from the master computer via host-based authentication. Then the MATLAB sessions are started on the slaves. (On each slave, more than one MATLAB session can be initialized.) For the parallel computations to be performed, we need to transfer the required data to each MATLAB session via MPI. On receiving data, each session on different slaves can execute computations concurrently. The portion of the computational work, to be performed on each session, can be adjusted according to the performance of the slaves. After all computations are finished on all slaves, the data will be collected from the MATLAB sessions.

5.3 Retrieving optimal control

After solving the discretized dynamic programming equations, we will have the value function $\mathfrak{V}(\alpha_{0j})$ and the corresponding control $\nu(\alpha_{0j})$ at each point of the polyhedron. The feedback control law or synthesis can be established through these

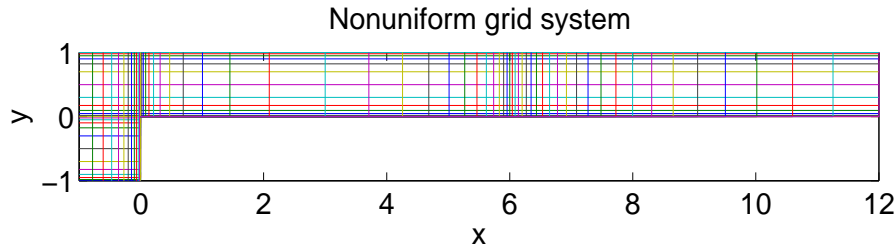


Figure 1. *Shape of the forward-facing step and the nonuniform grid system (shown in every other grid lines).*

data. Assuming the reduced order state at a time instance t^k is known as $\alpha(t^k)$, this state can be located in the polyhedron to find out the coefficients Λ_j for the convex combination,

$$\alpha(t^k) = \sum_{j=1}^{\bar{n}} \Lambda_j \alpha_{0j}.$$

The optimal control corresponding to the state at time t^k is then set as

$$\nu^*(t^k) = \sum_{j=1}^{\bar{n}} \Lambda_j \nu(\alpha_{0j}).$$

Under the assumption that the optimal control is piecewise constant, the NS equations is integrated in the time interval from t^k to t^{k+1} to find out the state of the flow at t^{k+1} , and hence the reduced order state $\alpha(t^{k+1})$ at that time instance. This procedure is repeated until reaching the final time. In this way the optimal trajectory and the associated piecewise constant optimal control are determined.

6 Numerical results from volume control

In the numerical experiments, a nonuniform 24×48 grid is chosen in the region $[-1, 0] \times [-1, 1]$ and a 96×24 grid for the region $[0, 12] \times [0, 1]$. The inlet is located at $(6, 1)$ with width 0.125 and its uniform velocity is $(-1, -1)^T$. Fig. 1 depicts the shape of the step and the nonuniform grid system used to solve the NS equations. The grid lines are concentrated toward the wall and the inlet for better capturing the separation bubbles. The Reynolds number is 1000 and the observation region Ω_o is $[-1, 12] \times [0, 1]$.

Fig. 2 shows the streamlines of the desired flow, which is the Stokes flow obtained with otherwise the same data setting. The streamlines for the uncontrolled flow at time instance $t = 10$ are shown in Fig. 3. There are separation bubbles around the step corner and behind the inlet flow. (The separation in the corner around the point $(0, -1)$ is not shown here or in the sequel since it is not of our

concerns from point of view of design target.) The design target is to reduced these two bubbles by tracking the desired Stokes flow. In the POD treatment, 6 basis functions are taken to capture more than 99.5% of the total energy. The distributed volume control is modelled by two shape functions, so that there are two design variables. The discount rate is $\mu = 2$. In the feedback design, there are 18480 grid points distributed in the six dimensional polyhedron. These data setting are summarized in the Tab. 1.

Six computers are utilized for the numerical experiments, on each of which a different number of MATLAB sessions are evoked. The finest time discretization is $\Delta T = 0.05$. The first computational case corresponds to $\beta = 0.1$. As shown in the first row of Tab. 2, the cost functional is 5.1374×10^{-2} under no control; and is decreased to 2.9264×10^{-2} with optimal control. The value function, corresponding to the specified initial state, is 2.8089×10^{-2} . The agreement between them is satisfactory. (Theoretically, the value function should be equal to the cost functional with the optimal control.) The streamlines are depicted in Fig. 4 for the optimally controlled flow at $t = 10$. Comparisons of Fig. 3 and 4 demonstrate that the separation bubbles are substantially attenuated in the sense that the separation lengths are significantly decreased.

We also solve the DPE for $\beta = 0.05$. The computational results are listed in the second row of Table 2. Again we observe good agreement between the value function and the cost functional with optimal control. The streamlines for this case are not shown here since they are similar to the streamlines for the case $\beta = 0.1$. As expected, the cost functional and the value function are smaller than the corresponding values for the case $\beta = 0.1$ where the controls are more expensive. Fig. 5 presents the optimal control for two cases, showing that all controls approach steady state as time evolves. It is worthwhile to note the difference between two cases. In the case $\beta = 0.05$, the second control hits the upper bound in some time interval. However, the bounds are not active throughout the whole time horizon in the case $\beta = 0.1$. This is consistent since we expect larger controls when the weighting coefficient β becomes smaller.

The main motivation for the construction of feedback design is its use in the context of noise to the system or to data. We performed a test in which uniform noise with values in the interval $[-0.5, 0.5]$ is added to the right hand side of the NS equations during the time interval $[0, 6]$. The optimal control subject to the noise can be easily retrieved from the feedback law which is already obtained from solving the DPE. The POD solution and the optimal control are shown in Fig. 6 in which the noise effect is visible. The controlled streamlines are indistinguishable from those in the noise-free case and are therefore not shown.

7 Feedback design for boundary control

In this section the proposed methodology is extended to a Dirichlet boundary control problem, in which a vertical blowing/suction actuator is located at the surface $([0, 3], 0)$ to attenuate the separation bubble (as shown in Fig. 7) in the flow over the forward-facing step of length 8. In this case no inlet is set on the upper channel

Reynolds number, Re	1000
Time horizon, T	10
No. of basis functions	6
No. of controls	2
Bounds on controls	$[-0.5, 0.5]$
Grid system	$12 \times 6 \times 6 \times 4 \times 3 \times 3$
Discount rate, μ	2
Observe region, Ω_o	$[-1, 12] \times [0, 1]$

Table 1. *Parameter Settings.*

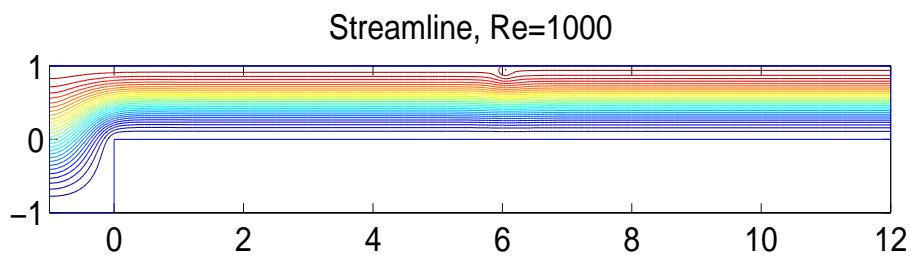


Figure 2. *Streamlines for the desired Stokes flow.*

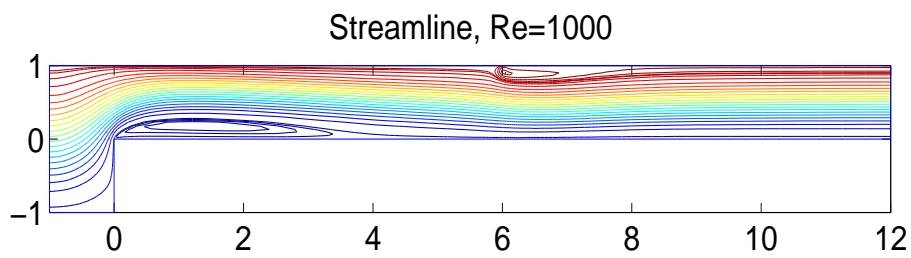


Figure 3. *Streamlines for the uncontrolled flow at $t = 10$.*

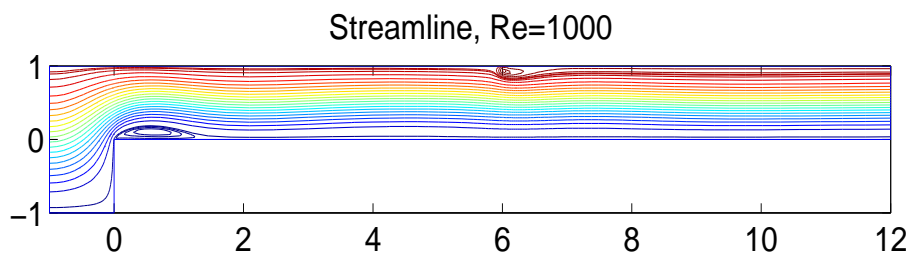


Figure 4. *Streamlines for the controlled flow at $t = 10$, $\beta = 0.1$.*

	cost w/o control	cost w. opt. control	value function
$\beta = 0.1$	5.1374e-2	2.9264e-2	2.8089e-2
$\beta = 0.05$	5.1374e-2	2.7547e-2	2.5762e-2

Table 2. Computational results.

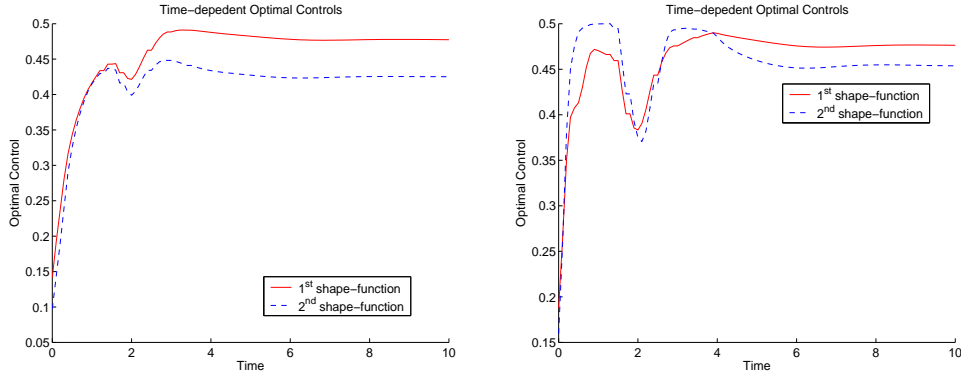


Figure 5. Optimal control for the distributed volume control problem, $\beta = 0.1$ (left) and $\beta = 0.05$ (right).

surface. Consequently we only concentrate the grid lines towards the wall surfaces. The actuator takes the following form,

$$v_c = \nu(t) \times \left(-\frac{0.05}{3}(x - 3)\right), \quad (23)$$

with $\nu(t)$ represents the time-depedent control. When ν is positive, the control is blowing, and otherwise it is suction. The cost functional to be considered in this design problem is

$$\mathcal{J} = \frac{1}{2} \left(\int_0^\infty \int_{\Omega_o} e^{-\mu t} \|u - u_{Stokes}\|^2 d\Omega dt + \beta \int_0^\infty e^{-\mu t} |\nu|^2 dt \right), \quad (24)$$

where u_{Stokes} is the u -component of Stokes solution, and the observation window is $\Omega_o = [0, 3] \times [0, 0.5]$. The reason for using the u -component in the cost functional is that the separation is directly related to the negative u -component velocity and the control is vertical blowing/suction. The dynamical system is the NS equations (1) where f is set to 0, together with boundary and initial conditions as in (2), except for the inject condition. Once snapshots are chosen, we proceed in an analogous way described in the previous sections for the volume control. This concerns the construction of the basis elements, the Galerkin process for obtaining the reduced order model, the approach of conducting feedback design and the method of retrieving the optimal control. We just describe in detail the snapshot construction. We first take

$$\hat{S} = \{y(t^k)|_{\nu(\cdot)=0}, \quad y(t^k)|_{\nu(\cdot)=2} - y_r\}, \quad k = 1, 2, \dots, N \quad (25)$$

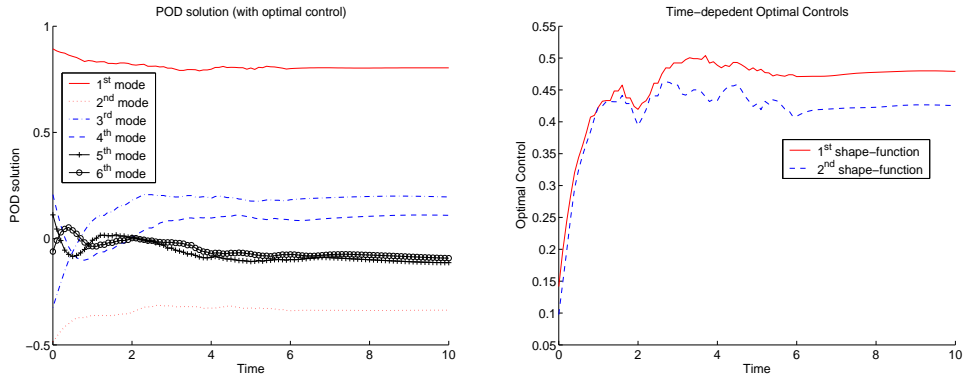


Figure 6. *POD solution and the optimal controls: with uniform noise in $[-0.5, 0.5]$ imposed on the right hand side during the time interval $[0, 6]$.*

where $y(t^k)_{\nu(\cdot)=0}$ and $y(t^k)_{\nu(\cdot)=2}$ denote the solution of the NS equation at time instance t^k without control and with constant control $\nu = 2$, respectively. The reference solution y_r is defined as

$$y_r = y_{Stokes}|_{\nu=2} - y_{Stokes}|_{\nu=0} \quad (26)$$

such that all the elements in the snapshot collection satisfy homogenous boundary condition on the Dirichlet boundaries. Finally the mean is subtracted from the snapshot set \hat{S} resulting in

$$\tilde{S} = \{\hat{S} - y_m\}, \quad (27)$$

with $y_m = \frac{1}{N} \sum_{k=1}^N \hat{S}_k$. After applying POD technique to obtain the basis functions ϕ_i , we can model the controlled NS solution as

$$\hat{y} = y_m + \sum_{i=1}^M \alpha_i \phi_i + \nu(t) y_r. \quad (28)$$

The reduced order model can now be obtained by a Galerkin process, namely, by restricting the modelled solution to the weak form (4). It is worthwhile to point out that the control now enters the modelled solution to account for the boundary conditions, differently from the reduced order model solution (11) in the volume control case. As a result, the reduced order ODEs have $M + 1$ components. In other words, given the same number of basis functions, the boundary control problem should deal with an ODE system with one more component than the distributed volume control problem. This is partly why the boundary control problem is more challenging.

In the design, the weighting coefficient is set to 0.001 and the discount rate is $\mu = 0.5$. The Reynolds number, the time horizon and the number of basis functions are set as the same values as in the previous design case. As shown in Tab. 3, the cost functional is decreased from 0.21674 to 0.17096. Meanwhile the value

	cost w/o control	cost w. opt. control	value function
$\beta = 0.001$	2.1674e-1	1.7096e-1	1.5920e-1

Table 3. Computational results for the boundary control problem.

function (0.15920) agrees well with the cost functional under the optimal control. The optimal control turns out to be blowing. Fig. 7 and 8 present the streamlines at $t = 10$ of the uncontrolled and the optimally controlled flow, respectively. The design objective is fulfilled in the sense that the separation bubble disappears in the controlled flow. The local flow pattern experiences large changes, which result in a relative small reduction in the cost functional.

We also tested a parabolic shape function for the spatial part of the actuator. Such a shape function is less effective in the sense that the reduction in the cost functional is smaller and that a small separation bulb still remains around the corner.

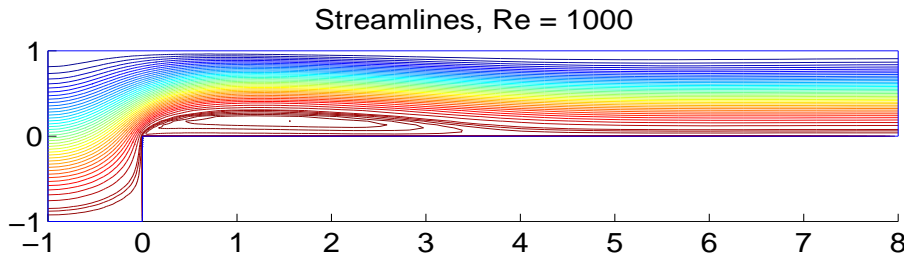


Figure 7. Streamlines for the uncontrolled flow at $t = 10$.

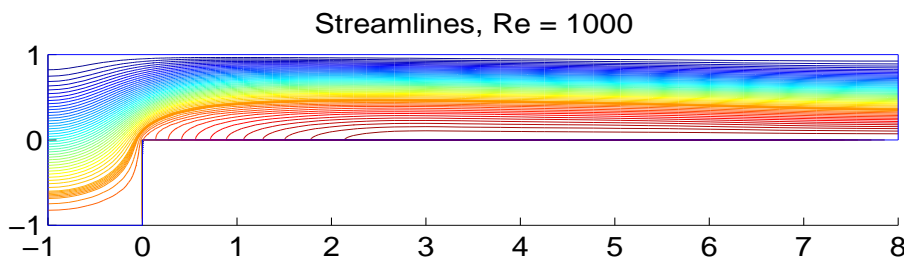


Figure 8. Streamlines for the flow with optimal boundary control, at $t = 10$.

Bibliography

- [1] J. A. ATWELL, *Proper orthogonal decomposition for reduced order control of partial differential equations*, PhD Thesis, Virginia Tech, Blacksburg, VA 24061, USA, 2000.
- [2] J. A. ATWELL AND B. B. KING, *Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations*, Math. Comput. Modelling, 33 (2001), pp. 1-19.
- [3] M. BARDI AND I. C.-DOLCETTA, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Systems & Control: Foundations & Applications, Birkhäuser, Boston, 1997.
- [4] M. FAHL, *Computation of POD basis functions for fluid flows with lanczos methods*, Math. Comput. Modelling, 34 (2001), pp. 91-107.
- [5] T. FURLANI, *Introduction to parallel computing at CCR*. http://www.ccr.buffalo.edu/documents/CCR-parallel_process_intro.pdf.
- [6] A. V. FURSIKOV, M. D. GUNZBURGER AND L. S. HOU, *Boundary value problems and optimal boundary control for the Navier-Stokes system: the two-dimensional case*, SIAM J. Control Optim., 36 (1998), pp. 852-894.
- [7] F. GOZZI, S. S. SRITHARAN AND A. ŚWIĘCH, *Viscosity solutions of dynamic-programming equations for the optimal control of the two-dimensional Navier-Stokes equations*, Arch. Rational Mech. Anal., 163 (2002), pp. 295-327.
- [8] R. M. HICKS AND P. A. HENNE, *Wing design by numerical optimization*, AIAA 77-1247, AIAA Aircraft System and Technology Conference, Seattle, Wash. Aug. 22-24, 1997.
- [9] M. HINZE AND K. KUNISCH, (2000). *Three control methods for time-dependent fluid flow*. Flow, Turbulence and Combustion, 65(2000), pp. 273-298.
- [10] K. ITO AND S. KANG, (1994). *A dissipative feedback control synthesis for systems arising in fluid dynamics*, SIAM J. Control Optim., 32 (1994), pp. 831-854.
- [11] K. ITO AND J. D. SCHRÖTER, *Reduced order feedback synthesis for viscous incompressible flows*, Math. Comput. Modelling, 33 (2001), pp. 173-192.

- [12] A. JAMESON, *Aerodynamic design via control theory*. J. Sci. Comput., 3 (1998).
- [13] K. KUNISCH AND S. VOLKWEIN, *Control of Burgers' equation by reduced order approach using proper orthogonal decomposition*, J. Optim. Theory Appl., 102 (1999), pp. 345-371.
- [14] K. Kunisch, S. Volkwein and L. Xie, *HJB-POD based feedback design for the optimal control of evolution problems*, SIAM J. Appl. Dynamic. Syst., to appear.
- [15] K. KUNISCH AND L. XIE, *POD-based feedback control of the Burgers equation by solving the evolutionary HJB equation*. Technical Report No. 287, Special Research Center F 003 *Optimization and Control*, Project area: *Continuous Optimization and Control*. University of Graz & Technical University of Graz, 2003. submitted to Math. Comput. Modelling.
- [16] H. V. LY AND H. T. TRAN, *Modelling and control of physical processes using proper orthogonal decomposition*, Math. Comput. Modelling, 33 (2001), pp. 223-236.
- [17] E. HEIBERG, *Matlab Parallization Toolkit*. see http://hem.passagen.se/einar_heiberg/
- [18] <http://mpi-forum.org/>
- [19] S. S. RAVINDRAN, *Reduced-order adaptive controllers for fluid flows using POD*, J. Sci. Comput., 15 (2002), pp. 457-478.
- [20] S. S. RAVINDRAN, *Control of flow separation over a forward-facing step by model reduction*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4599-4617.
- [21] edited by S. S. SRITHARAN, *Optimal control of viscous flow*, Siam Philadelphia, 1998.
- [22] S. TUREK, *A comparative study of some time-stepping techniques for the incompressible Navier-Stokes equations: from fully implicit nonlinear schemes to semi-implicit projection methods*, Int. J. Numer. Meth. Fluids, 22 (1996), pp. 987-1011.