Goal-Oriented, Model-Constrained Optimization for Reduction of Large-Scale Systems

T. Bui-Thanh [a], K. Willcox [a,*] O. Ghattas [c],

B. van Bloemen Waanders [b],

[a] *Massachusetts Institute of Technology, Cambridge, MA 02139*

[b] *Sandia National Laboratories, Albuquerque, NM 87185* [1]

[c] *University of Texas at Austin, Austin, TX 78712*

Abstract

Optimization-oriented reduced-order models should target a particular output functional, span an applicable range of dynamic and parametric inputs, and respect the underlying governing equations of the system. To achieve this goal, we present an approach for determining a projection basis that uses a goal-oriented, model-constrained optimization framework. The mathematical framework permits consideration of general dynamical systems with general parametric variations and is applicable to both linear and nonlinear systems. Results for a simple linear model problem of the two-dimensional heat equation demonstrate the ability of the goal-oriented approach to target a particular output functional of interest. Application of the methodology to a more challenging example of a subsonic blade row governed by the unsteady Euler flow equations shows a significant advantage of the new method over the proper orthogonal decomposition.

Key words: Model reduction, optimization, partial differential equations

1991 MSC: 49N99, 65D99, 76R50

1 Introduction

Model reduction entails the systematic generation of cost-efficient representations of large-scale systems that result, for example, from discretization of partial differential equations (PDEs). The task of determining these representations may be posed as an optimization problem: determine the reduced model that provides the optimal representation (with respect to some quantity of interest) of the large-scale system behavior. For very large systems, determination of the best reduced model via direct optimization has not been pursued, due to challenges in solving the resulting optimization problem. Instead, several reduction methods have been developed that trade off optimality for tractability, and these have been applied in many different settings with considerable success, including controls, fluid dynamics, structural dynamics, and circuit design. However, a number of open issues remain with these methods, including the reliability of reduction techniques, guarantees associated with the quality of the reduced models, and the generation of reduced mod-

* Corresponding author.
Email addresses: tanbui@mit.edu (T. Bui-Thanh), kwillcox@mit.edu (K. Willcox), omar@ices.utexas.edu (O. Ghattas), bartv@sandia.gov (B. van Bloemen Waanders).

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

els that are suitable for optimal design, optimal control and inverse problem applications.

Recent advances in scalable algorithms for large-scale optimization of systems governed by PDEs have permitted solution of problems with millions of state and optimization variables [1,2]. The problem of determining a reduced model can be cast in a similar model-constrained optimization framework. In particular, we consider a *goal-oriented* formulation in which the reduced model is chosen to optimally represent a particular output functional. Whereas other large-scale reduction methods, such as the proper orthogonal decomposition (POD), are purely data-driven and do not consider the underlying equations, our *model-constrained* optimization approach enforces the reduced-order governing equations as constraints. This improves on a data-driven approach by bringing additional knowledge of the reduced-order governing equations into the construction of the basis.

Most large-scale model reduction frameworks are based on a projection approach, which can be described in general terms as follows. Consider the general linear, time-invariant (LTI) dynamical system

$$M\dot{u} + Ku = f, \tag{1}$$
$$g = Cu, \tag{2}$$

with initial condition

$$u(0) = u_0, \tag{3}$$

where $u(t) \in \mathbb{R}^N$ is the system state, $\dot{u}(t)$ is the derivative of $u(t)$ with respect to time, and the vector u_0 contains the specified initial state. In general, we are interested in systems of the form (1) that result from spatial discretiza-

tion of PDEs. In this case, the dimension of the system, N, is very large and the matrices $M \in \mathbb{R}^{N\times N}$ and $K \in \mathbb{R}^{N\times N}$ result from the chosen spatial discretization method. The vector $f(t) \in \mathbb{R}^N$ defines the input to the system and the matrix $C \in \mathbb{R}^{Q\times N}$ defines the Q outputs of interest, which are contained in the output vector $g(t)$.

A reduced-order model of (1)–(3) can be derived by assuming that the state $u(t)$ is represented as a linear combination of m basis vectors,

$$\hat{u} = \Phi\alpha, \tag{4}$$

where $\hat{u}(t)$ is the reduced model approximation of the state $u(t)$ and $m \ll N$. The projection matrix $\Phi \in \mathbb{R}^{N\times m}$ contains as columns the basis vectors ϕ_i, i.e., $\Phi = [\phi_1\ \phi_2\ \cdots\ \phi_m]$, and the vector $\alpha(t) \in \mathbb{R}^m$ contains the corresponding modal amplitudes. This yields the reduced-order model with state $\alpha(t)$ and output $\hat{g}(t)$

$$\hat{M}\dot{\alpha} + \hat{K}\alpha = \hat{f}, \tag{5}$$
$$\hat{g} = \hat{C}\alpha, \tag{6}$$
$$\hat{M}\alpha_0 = \Phi^T M u_0, \tag{7}$$

where $\hat{M} = \Phi^T M\Phi$, $\hat{K} = \Phi^T K\Phi$, $\hat{f} = \Phi^T f$, $\hat{C} = C\Phi$, and $\alpha_0 = \alpha(0)$.

Projection-based model reduction techniques seek to find a basis Φ so that the reduced system (5)–(7) provides an accurate representation of the large-scale system (1)–(3) over the desired range of inputs. An optimal reduced model can be defined as one that minimizes the H-infinity norm of the difference between the reduced and original system transfer functions; however, no polynomial-time algorithm is known to achieve this goal. Algorithms such as optimal Hankel model reduction [3–5] and balanced truncation [6] have been used

widely throughout the controls community to generate suboptimal reduced models with strong guarantees of quality. These algorithms can be carried out in polynomial time; however, the computational requirements make them impractical for application to large systems such as those arising from the discretization of PDEs, for which system orders typically exceed 10^4.

While considerable effort has been applied in recent years towards development of algorithms that extend balanced truncation to large-scale LTI systems [7–9], efficient algorithms for very large systems remain a challenge. In addition, application of balanced truncation methods to systems that are linear time-varying or have parametric variation has been limited to small systems [10,11]. The proper orthogonal decomposition (POD) [12–14] has emerged as a popular alternative for reduction of very large dynamical systems; however, it lacks the quality guarantees of methods such as balanced truncation.

Optimal design, optimal control and inverse problem applications present additional challenges for model reduction methods. In such cases—where the physical system must be simulated repeatedly—the availability of reduced models can greatly facilitate solution of the optimization problem, particularly for real-time and/or large-scale applications. To be useful for optimization purposes, the reduced model must provide an accurate representation of the high-fidelity model over a wide range of parameters. In particular, discretization produces high-dimensional input spaces when the input parameters represent continuous fields (such as initial conditions, boundary conditions, distributed source terms, and heterogeneous material fields). Model reduction for high-dimensional input spaces remains a challenging problem. Approaches developed for dynamical systems, such as POD and Krylov-based methods, have been applied in an optimization context [15–17]; however, the number

of parameters in the optimization application was small. In recent work for steady-state problems, methods are presented for constructing reduced models that are of guaranteed quality over a range of inputs via the use of error estimates and adaptivity [18].

In this paper, we formulate the task of determining a projection basis as a goal-oriented, model-constrained optimization problem. The mathematical framework permits consideration of general dynamical systems with general parametric variations and is applicable to both linear and nonlinear systems. We propose an efficient solution strategy that borrows concepts from the POD and employs recent methods for optimization of systems governed by PDEs to make the approach tractable for large-scale problems. We begin with a description of the general dynamical system framework with parametric variations. This is followed by a description of the goal-oriented basis optimization formulation and the proposed model reduction methodology. The approach is then demonstrated with two examples. The first is a simple linear model problem that considers the unsteady two-dimensional heat equation with parametrically varying boundary control inputs. The second is a more complicated example that considers the two-dimensional linearized Euler equations governing the unsteady motion of a subsonic blade row. Finally, we present conclusions and directions for future research.

2 Dynamical System Framework

The standard LTI system framework is defined by (1)–(3). In this section, we present the more general case that includes parametric variation in the system. An overview of the existing POD method of snapshots, a commonly

used approach to define the reduced basis, is described.

2.1 Parametric input variations

We consider a finite set of instantiations of the governing equations (1)–(3) that could arise from variations in the coefficient matrices M and K, the input f, or the initial state u_0. For example, where (1)–(3) represent a spatially discretized PDE, these variations stem from changes in the domain shape, boundary conditions, coefficients, initial conditions, or distributed sources of the underlying PDEs. The general dynamical system for S different instances is thus written

$$M^k \dot{u}^k + K^k u^k = f^k, \quad k = 1, \ldots, S, \tag{8}$$
$$u^k(0) = u_0^k \quad k = 1, \ldots, S, \tag{9}$$
$$g^k = C^k u^k, \quad k = 1, \ldots, S, \tag{10}$$

where the superscript k denotes the kth instance of the system, with corresponding state $u^k(t)$ and output $g^k(t)$.

Using the projection framework described in the previous section, a reduced-order model of (8)–(10) is obtained as

$$\hat{M}^k \dot{\alpha}^k + \hat{K}^k \alpha^k = \hat{f}^k, \quad k = 1, \ldots, S, \tag{11}$$
$$\hat{g}^k = \hat{C}^k \alpha^k, \quad k = 1, \ldots, S, \tag{12}$$
$$\hat{M}^k \alpha_0^k = \Phi^T M^k u_0^k, \quad k = 1, \ldots, S, \tag{13}$$

where $\hat{M}^k = \Phi^T M^k \Phi$, $\hat{K}^k = \Phi^T K^k \Phi$, $\hat{f}^k = \Phi^T f^k$, and $\hat{C}^k = C^k \Phi$.

2.2 Proper orthogonal decomposition

POD is a widely used approach to determine the reduced basis Φ. POD can be applied efficiently to large systems using the method of snapshots [12] as follows. Consider the collection of "snapshots", $u^k(t_j),\ j = 1,\ldots,T,\ k = 1,\ldots,S$, where $u^k(t_j) \in \mathbb{R}^N$ is the solution of the governing equations (8) at time t_j for parameter instance k. T time instants are considered for each parameter instance, yielding a total of ST snapshots. We define the snapshot matrix $U \in \mathbb{R}^{N\times ST}$ as

$$U = \left[u^1(t_1)\ u^1(t_2)\ \cdots\ u^1(t_T)\ u^2(t_1)\ \cdots\ \cdots\ u^S(t_T)\right], \tag{14}$$

and we will refer to the ith column of U as the ith snapshot, denoted by U_i.

The POD basis vectors are chosen to be the orthonormal set that solves the optimization problem [19]

$$\phi = \arg\max_{\varphi} \frac{\langle |(u,\varphi)|^2 \rangle}{(\varphi,\varphi)}, \tag{15}$$

where (u,φ) denotes the scalar product of the vector φ with the field $u(t)$ evaluated over the domain, and $\langle\ \rangle$ represents a time-averaging operation. In the case of the discrete snapshots contained in U, (15) is maximized when the m basis vectors are chosen to be the first m left singular vectors of U. For a fixed basis size, the POD basis therefore minimizes the error between the original snapshots and their representation in the reduced space defined by

$$E = \sum_{k=1}^{S}\sum_{j=1}^{T} \left[u^k(t_j) - \tilde{u}^k(t_j)\right]^T \left[u^k(t_j) - \tilde{u}^k(t_j)\right], \tag{16}$$

where $\tilde{u}^k(t_j) = \Phi\Phi^T u^k(t_j)$. This error is equal to the sum of the squares of the singular values corresponding to those singular vectors not included in the basis,

$$E = \sum_{i=m+1}^{ST} \sigma_i^2, \tag{17}$$

where σ_i is the i^{th} singular value of U.

The POD is an optimal basis in the sense that it minimizes the data reconstruction error given by (16); however, it is important to note that this optimality applies only to the representation of a known state solution $u^k(t_j)$ in the reduced space, i.e. $\tilde{u}$ is computed as $\tilde{u}^k(t_j) = \Phi\Phi^T u^k(t_j)$, not by solution of the reduced model ($\tilde{u} \neq \hat{u}$). Therefore, the error expression does not apply to the resulting POD reduced-order model (5). In particular, the error expression yields no rigorous information regarding the accuracy of the solution of the reduced model and thus whether $\hat{u}$ is a good approximation of u. Moreover, the POD basis does not account for the system outputs, although methods to augment the standard approach have been proposed that use adjoint information [20,21]. In addition, because no information regarding the governing equations is included in the POD process, the POD basis does not properly reflect the fact that the snapshots $u^k(t_j)$ are associated with different parametric instances of the system.

In the following section we present an alternative method to determine the reduced-space basis. This method seeks to minimize an error similar in form to (16). However, we will improve upon the POD, first, by minimizing the error in the outputs (as opposed to states) and, second, by imposing additional constraints that $\hat{u}^k(t)$ should result from satisfying the reduced-order

governing equations for each parameter instance k.

3 Optimized Reduced-Order Basis

3.1 Constrained optimization formulation for projection basis

We pose the problem of selecting the basis Φ as a goal-oriented optimization problem that seeks to minimize the difference between the full-space and reduced-order output solution over a selected set of inputs and the interval $(0, t_f)$, subject to satisfying the underlying governing equations. The problem of determining the optimal basis, $\Phi \in \mathbb{R}^{N\times m}$, can be written as

$$\min_{\Phi,\alpha} \mathcal{G} = \frac{1}{2}\sum_{k=1}^{S}\int_0^{t_f}\left(g^k - \hat{g}^k\right)^T\left(g^k - \hat{g}^k\right)\,dt + \frac{\beta}{2}\sum_{j=1}^{m}\left(1-\phi_j^T\phi_j\right)^2 + \frac{\beta}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{m}\left(\phi_i^T\phi_j\right)^2, \tag{18}$$

subject to

$$\Phi^T M^k \Phi\dot{\alpha}^k + \Phi^T K^k \Phi\alpha^k = \Phi^T f^k,\ k=1,\ldots,S, \tag{19}$$

$$\Phi^T M^k \Phi\alpha_0^k = \Phi^T M^k u_0^k,\ k=1,\ldots,S, \tag{20}$$

$$\hat{g}^k = C^k\Phi\alpha^k,\ k=1,\ldots,S. \tag{21}$$

In the case of a linear relationship between outputs and state as in (10), the objective function can be written

$$\mathcal{G} = \frac{1}{2}\sum_{k=1}^{S}\int_0^{t_f}\left(u^k - \hat{u}^k\right)^T H^k\left(u^k - \hat{u}^k\right)\,dt + \frac{\beta}{2}\sum_{j=1}^{m}\left(1-\phi_j^T\phi_j\right)^2 + \frac{\beta}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{m}\left(\phi_i^T\phi_j\right)^2, \tag{22}$$

where $H^k = C^{kT}C^k$ can be interpreted as a weighting matrix that defines the states relevant to the specified output. While the first term in the objective function (22) has similarities with that minimized by the POD, given by (16), there are two important distinctions to note. First, the goal-oriented nature of the formulation (22) focuses on reduction of the error for a particular output functional rather than for the general state vector. Second, through the constraints (19)–(21), the optimization approach requires satisfaction of the reduced-order governing equations to determine $\hat{u}$. The error minimized by the optimization approach is thus tied rigorously to the reduced-order model, whereas the POD is based purely on snapshot data. In both cases, however, the definition of the error is limited to a discrete set of observations.

The second and third terms in (22) are regularization terms that penalize the deviation of the basis vectors from an orthonormal set, with β as a regularization parameter. This regularization acts only in the null space of the projected Hessian matrix of the first term of (22). Therefore, the reduced output approximation, $\hat{g}$, is unaffected by the regularization term, yet the conditioning of the optimization problem is improved. Note, however, that there remains a null space of the projected Hessian matrix that admits arbitrary rotations of the basis vectors; the optimization method chosen to solve (18)–(21) should therefore be tolerant of singular projected Hessian matrices. It is also important to note that the optimization problem (18)–(21) is nonlinear and nonconvex; hence, there is no guarantee that a purely local optimization method will converge to the global optimum. As a result, generating the initial guess is very important; strategies to address this issue will be discussed in the next section.

3.2 Optimality conditions and the reduced gradient

The optimality conditions for the system (18)–(21) can be derived by defining the Lagrangian functional

$$\begin{aligned}
\mathcal{L}\left(\Phi, \alpha^k, \lambda^k, \mu^k\right) &= \frac{1}{2}\sum_{k=1}^{S}\int_0^{t_f}\left(u^k - \Phi\alpha^k\right)^T H^k\left(u^k - \Phi\alpha^k\right)\, dt \\
&+ \frac{\beta}{2}\sum_{j=1}^{m}\left(1 - \phi_j^T\phi_j\right)^2 + \frac{\beta}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{m}\left(\phi_i^T\phi_j\right)^2 \\
&+ \sum_{k=1}^{S}\int_0^{t_f}\lambda^{kT}\left(\Phi^T M^k\Phi\dot{\alpha}^k + \Phi^T K^k\Phi\alpha^k - \Phi^T f^k\right)\, dt \\
&+ \sum_{k=1}^{S}\mu^{kT}\left(\Phi^T M^k\Phi\alpha_0^k - \Phi^T M^k u_0^k\right),
\end{aligned} \tag{23}$$

where $\lambda^k = \lambda^k(t) \in \mathbb{R}^m$ and $\mu^k \in \mathbb{R}^m$ are Lagrange multipliers (also known as adjoint state variables) that respectively enforce the state ODE system and initial conditions for the kth sample. The optimality system can be derived by taking variations of the Lagrangian with respect to the adjoint, state, and basis vector variables.

Setting the first variation of the Lagrangian with respect to λ^k to zero and arguing that the variation of λ^k is arbitrary in $(0, t_f)$, and setting the derivative of the Lagrangian with respect to μ^k to zero, simply recovers the state equation and initial conditions (19)–(20).

Setting the first variation of the Lagrangian with respect to the α^k to zero, and arguing that the variation of α^k is arbitrary in $(0, t_f)$, at $t = 0$, and at $t = t_f$, yields the adjoint equation, final condition and definition of μ

$$-\Phi^T M^k\Phi\dot{\lambda}^k + \Phi^T K^{kT}\Phi\lambda^k = \Phi^T H^k\left(u^k - \Phi\alpha^k\right),\ k = 1, \ldots, S, \tag{24}$$

$$\lambda^k(t_f) = 0,\ k = 1, \ldots, S, \tag{25}$$

$$\mu^k = \lambda^k(0),\ k = 1, \ldots, S. \tag{26}$$

Note that, without loss of generality, M is assumed to be a symmetric matrix.

Taking the derivative of the Lagrangian functional with respect to the basis vector variables Φ yields the following matrix equation,

$$\begin{aligned}
\delta\mathcal{L}_\Phi &= \sum_{k=1}^{S} \int_0^{t_f} H^k \left(\Phi\alpha^k - u^k\right) \alpha^{kT}\ dt + 2\beta\Phi\left(\Phi^T\Phi - I\right) \\
&+ \sum_{k=1}^{S} \int_0^{t_f} \left[M^k\Phi\left(\lambda^k\dot{\alpha}^{kT} + \dot{\alpha}^k\lambda^{kT}\right) + K^{kT}\Phi\lambda^k\alpha^{kT} + K^k\Phi\alpha^k\lambda^{kT} - f^k\lambda^{kT}\right]\ dt \\
&+ \sum_{k=1}^{S} M^k\Phi\mu^k\alpha_0^T + \sum_{k=1}^{S} M^k\left(\Phi\alpha_0^k - u_0^k\right)\mu^{kT} = 0.
\end{aligned} \tag{27}$$

The combined system (19)–(20), (24)–(26), and (27) represents the first-order Karush-Kuhn-Tucker optimality conditions for the optimization problem (18)–(21).

3.3 Solution of the optimization problem

To solve the constrained optimization problem (18)–(21), we choose to solve an equivalent unconstrained optimization problem in the Φ variables by eliminating the state variables α^k and state equations (19). That is, we replace $\min_{\phi,\alpha} \mathcal{G}(\alpha, \phi)$ with $\min_\phi \tilde{\mathcal{G}}(\alpha(\phi), \phi)$, where the dependence of α on ϕ is implicit through the state equations (19)–(20).

We solve this unconstrained optimization problem by a trust-region inexact-Newton conjugate-gradient method. That is, we use the conjugate gradient (CG) method to solve the linear system of equations arising at each Newton step and globalize by a trust region scheme (see, for example, [22]). We

terminate CG when any of the three following conditions is satisfied: (1) a negative curvature direction is encountered; (2) the norm of the residual of the Newton system is brought down to a sufficiently small value relative to the norm of the gradient; or (3) the Newton step iterate exits the trust region. This method combines the rapid locally-quadratic convergence rate properties of Newton's method, the effectiveness of trust region globalization for treating ill-conditioned problems, and the Eisenstat-Walker idea of preventing oversolving [23].

The gradient of the unconstrained function $\hat{\mathcal{G}}$ with respect to ϕ, as required by Newton's method, can be computed efficiently by an adjoint method. The gradient is given by $\delta\mathcal{L}_\Phi$ when the α^k satisfy the state equations and (λ^k, μ^k) satisfy the adjoint equations. The procedure to compute the gradient can therefore be summarized as follows. First, solve the state equations (19)–(20) to determine $\alpha^k(t)$. Second, solve the adjoint equations (24)–(26) to determine $\lambda^k(t)$ and μ^k. Finally, use the computed α^k, λ^k, and μ^k in (27) to determine the gradient. The Hessian-vector product as required by CG is computed on-the-fly; because it is a directional derivative of the gradient its computation similarly involves solution of state-like and adjoint-like equations. Therefore, the optimization algorithm requires solution of a pair of state and adjoint systems at each CG iteration. Note that the state and adjoint equation each consist of S uncoupled ODE systems, each corresponding to one instance of the parameter. For more details on Newton-Krylov methods for solution of simulation-constrained optimization problems and the associated computational cost, see [24].

3.4 Basis computation

The formulation defined by equations (18)–(21) provides a mathematical definition of the desired optimal basis; however, in practice this optimization problem may not be tractable for large-scale problems. First, we may not be able to afford storage of the entire time history for the full model, which leads us to adopt a snapshot-based approach. As in the POD, accurate numerical approximation of the time integrals in (18) can be replaced by summation over a more coarsely sampled subset of time instants. Our method therefore requires *a priori* computation of a set of high-fidelity solutions over a pre-determined set of time instants and input parameter values.

Second, even with this simplification, the number of optimization variables is equal to mN — the desired number of basis functions multiplied by the length of each basis vector — where for many applications $N \geq O(10^6)$. Therefore, it will be assumed that each basis vector can be represented as a linear combination of snapshots,

$$\phi_j = \sum_{i=1}^{ST} \gamma_i^j U_i \quad j = 1, \ldots, m, \tag{28}$$

where the coefficients γ_i^j are the variables in the modified optimization problem. This approximation reduces the number of optimization variables from mN to mST; for large-scale applications, typically $ST \ll N$. As a consequence, neither the gradient computation nor the optimization step computation (which dominate the cost of an optimization iteration) scale with the full system size N. Approximating the basis vectors as a linear combination of snapshots is motivated by the singular value decomposition (SVD) theory underlying the POD, for which the relation (28) is exact (this is equivalent to

solving the inner versus the outer SVD problem).

Equation (28) can be written in matrix form as

$$\Phi = U\Gamma, \tag{29}$$

where γ_i^j is the ijth element of $\Gamma \in \mathbb{R}^{ST\times m}$. Gradients of the objective function with respect to Γ are related simply to gradients with respect to Φ by

$$\frac{\partial \mathcal{L}}{\partial \Gamma} = U^T \frac{\partial \mathcal{L}}{\partial \Phi}. \tag{30}$$

The modified optimization formulation offers no guarantees of convexity and the choice of initial guess for the basis is thus very important. In this paper, we present two possible strategies. The first is to use the POD basis as an initial starting point. Since a snapshot set is required anyway, the additional cost of computing the POD basis is small. A second strategy is to employ continuation on the basis dimension. In this approach, the initial guess for the case of m basis vectors is chosen to be the solution of the optimization problem for $m-1$ basis vectors plus an arbitrary mth vector. This iterative procedure can be initialized at any value $m \geq 1$ with the POD basis vectors as an initial guess on the first iteration.

4 Results

Results are presented for two examples. The first example is a simple heat conduction model problem of moderate dimension that permits detailed assessment of the optimized basis methodology. The second example is a large-scale CFD problem that clearly demonstrates the advantages of the new method

over the POD.

4.1 Heat Conduction Example

Results are presented for a simple model problem that considers the two-dimensional time-dependent heat equation with boundary temperature inputs. The initial-boundary value problem is given by

$$\frac{\partial \bar{u}}{\partial t} - \kappa \nabla^2 \bar{u} = 0 \quad \text{in } \Omega, \tag{31}$$

$$\bar{u} = \bar{u}_c \quad \text{on } \Gamma_c, \tag{32}$$

$$\bar{u} = 0 \quad \text{on } \Gamma_D, \tag{33}$$

$$\frac{\partial \bar{u}}{\partial n} = 0 \quad \text{on } \Gamma_N, \tag{34}$$

$$\bar{u} = \bar{u}_0 \quad \text{in } \Omega \text{ for } t = 0, \tag{35}$$

where $\bar{u}(x, y, t)$ is the temperature field defined on the domain Ω, κ is the thermal diffusivity, $\bar{u}_c(x, y)$ is the boundary control function (which is assumed to be constant in time) applied on the boundary Γ_c, Γ_D and Γ_N are Dirichlet and Neumann boundaries, respectively, and $\bar{u}_0(x, y)$ is the given initial temperature field. The output of interest is the temperature over a specified sub-region of the domain.

Spatial discretization is by linear triangular finite elements, yielding a dynamical system of the form (8)–(10), where $u^k(t)$ represents the spatially discretized temperature field corresponding to forcing input f^k, and $g^k(t)$ contains those elements of u^k that lie within the specified region of interest. Figure 1 shows the problem domain Ω and corresponding mesh that was used. Results are presented for a discretization containing $N = 480$ temperature unknowns. The specified initial condition is $u = 0$ at $t = 0$, and time integration is by

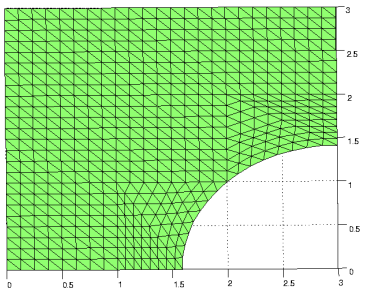

Fig. 1. Problem domain and boundary conditions for heat conduction example: Neumann on right side, Dirichlet on all other boundaries.

implicit Euler with a constant time step over the time interval $(0, t_f)$. Note that the adjoint equation is marched backward in time. The boundary control is applied on $\Gamma_c = \{(0, y) : 0 \leq y \leq 3\}$, i.e., Dirichlet control on the left boundary of the domain. A Neumann boundary condition is specified on $\Gamma_N = \{(3, y) : 1.5 \leq y \leq 3\}$, and a homogeneous Dirichlet condition is imposed on the remaining part of the boundary, Γ_D.

Snapshots were generated by solving the system under different boundary forcing conditions. The forcing was generated by applying a temperature distribution along the boundary Γ_c. For the results presented here, the forcing functions considered were parameterized by sinusoidal distributions with varying spatial frequency. Snapshots were generated over $S = 5$ instances of the control parameter forcing with $T = 20$ time instants for each parameter instance. Using the optimization formulation (18)–(21), we seek the m basis functions that minimize the error defined by (18) while satisfying the reduced-order state equations for each control instance. The basis functions are assumed to be a linear combination of available snapshots; hence there are $mST = 100m$ basis function variables in the optimization problem. The state and adjoint equations each consist of $S = 5$ uncoupled ODE systems of

dimension m.

4.1.1 Optimized basis performance

For the first set of results, the output of interest is defined to be the temperature over a strip of the domain in the region $0.5 < x < 1.0$, $0.5 < y < 2.5$, yielding an output vector of size $Q = 47$. Figure 2 shows values of the resulting objective function (22), i.e. the error in the outputs, for bases ranging in size from $m = 1$ to $m = 10$. Figure 2 also shows the values of (22) for the POD bases over this range of m. It can be seen clearly that the optimized basis outperforms the POD in all cases, particularly when m is small. For larger reduced models, $m > 10$, the POD resolves the temperature field very accurately, since the first 10 POD eigenvalues contain over 98% of the snapshot energy, that is $(\sum_i^{10} \lambda_i)/(\sum_i^{100} \lambda_i) > 0.98$. In these cases, the optimized basis does not provide an improvement over the POD.

In order to provide a quantitative metric by which to judge the performance of the optimized basis, balanced truncation was applied to this problem. The problem was converted to standard LTI form by considering each parametric forcing function as an independent input. Figure 2 plots values of (22) for truncated balanced models of size $m = 1$ through $m = 10$. It can be seen that in most cases the optimized basis provides a substantial improvement over POD when both are compared to the results of balanced truncation. It is also important to note that balanced truncation uses both a left and a right projection basis, and thus has twice as many degrees of freedom as the goal-oriented optimized basis.

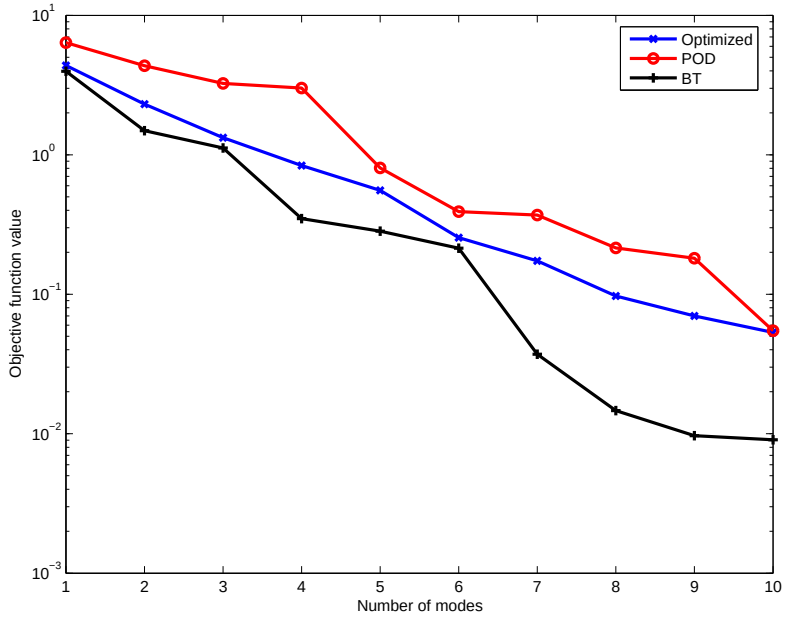

Fig. 2. The output error between reduced and full-order models (22) versus number of modes for the goal-oriented optimized basis, the POD basis, and balanced truncation applied to the heat conduction example.

4.1.2 Comparison with POD

A significant advantage of the goal-oriented approach is that the basis can be optimized with respect to a particular output functional, whereas the POD seeks to minimize the reconstruction error over all states. Several different output definitions were considered in order to gain insight into the optimized basis results.

If the output considered is to minimize the error of state prediction over the entire domain, that is, $H^k = 1$ in (22), then the goal-oriented approach seeks to minimize the same error as the POD. However, it is important to note again the difference in the representation of the term $\tilde{u}^j$, which for POD is computed directly from the known solution u^j, i.e. $\tilde{u}^j = \Phi\Phi^T u^j$. In this sense the POD is a purely data-based method that does not account for the

underlying governing equations. In contrast, our method computes $\hat{u}^j$ in (22) by requiring the solution to satisfy the governing equations in the reduced-order space.

Results for this case are shown in the first row of Table 1. Using the POD basis as an initial guess, the optimizer makes little improvement in the objective function. As shown in Table 1, the reduction in the error is just 1%. For different values of S, T and m, the POD basis is found to be almost optimal with respect to state reconstruction error for this example. Due to the symmetry properties of the system (M and K are symmetric matrices), any congruent basis transformation, such as the POD, is guaranteed to preserve the stability of the system. Thus we expect that the POD should perform well on this heat conduction example. As the results show, the additional error from solution of the governing equations in the reduced space is not significant in this case. As the next example will show, in more complicated problems the optimized basis can provide an advantage over the POD even for full state reconstruction, particularly in the case of non-symmetric systems for which the POD basis can routinely produce unstable reduced-order models.

Table 1 shows the results for other outputs corresponding to various specified output regions (and thus different weightings H in the objective function (22)). Note that the POD basis is computed in the standard way and thus is insensitive to the choice of output functional. The values in the column $\mathcal{G}_{pod}$ represent the standard POD basis evaluated using the criterion defined by (22) for each different instance of H (i.e. the metric $\mathcal{G}_{pod}$ is case-dependent). It can be seen that by targeting an output functional, the goal-oriented basis can yield substantial improvements in errors over the POD basis. It should be emphasized that our method does not simply "ignore" states that lie outside of

Table 1

Comparison of optimization results for the heat conduction example. The objective function given by (22) is evaluated for the optimized basis ($\mathcal{G}_{opt}$) and the POD basis ($\mathcal{G}_{pod}$).

Minimize prediction error over	S	T	m	$\mathcal{G}_{opt}$	$\mathcal{G}_{pod}$
All states	5	20	5	28.9829	29.2762
$x = 0.625, y = 0.625$	5	20	5	2.9038e-3	0.01066
$0.5 < x < 1, 0.5 < y < 1$	5	20	5	0.01282	0.1932
$0.5 < x < 1, 0.5 < y < 2.5$	5	20	5	0.5555	0.8062

the region of interest, since $\hat{u}^j$ is computed by solving the reduced-order equations over the entire domain. Therefore the basis must represent *all* states – but the optimization formulation allows the basis energy to be focused appropriately to achieve the desired objective. One might draw conceptual parallels between this approach and goal-oriented *a posteriori* error estimates to drive grid adaptivity.

Figure 3 shows the output errors in the case of an output functional defined over the region $0.5 < x < 1$, $0.5 < y < 1$. Each plot in the figure corresponds to one of the grid points that lie within the region of interest (for clarity, just four of the nine points are shown). The first $T = 20$ snapshots correspond to the first instance of control forcing, the second $T = 20$ correspond to the second instance, and so on. The figure shows that for almost every snapshot in the ensemble, the optimized basis results in a more accurate prediction of the temperature at the point of interest. In many cases, the error is reduced by almost an order of magnitude.

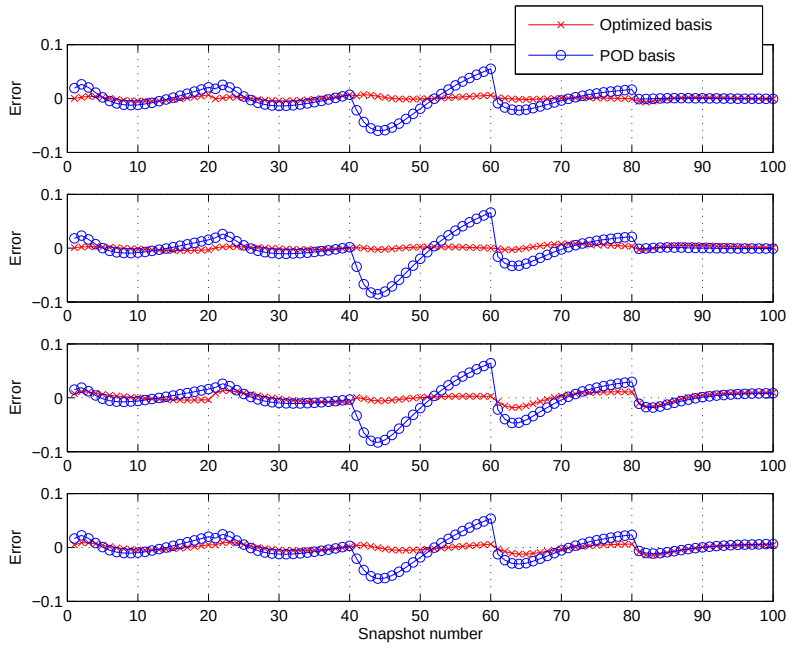

Fig. 3. Error in temperature prediction for each snapshot using POD and optimized basis. The optimized basis was selected to minimize the error over the region $0.5 < x < 1$, $0.5 < y < 1$. Errors are shown for four of the nine points contained within this region.

The reduced output errors shown in Figure 3 come at a cost. Figure 4 shows the norm of the errors computed over the entire domain for each snapshot. In order to reduce the errors at the specified points, the optimized basis yields less accurate predictions for other states. However, it is again important to note that this trade-off in accuracy is done in a systematic way using both the governing equations and the defined output functional. According to the optimization result, the larger errors observed in other areas of the domain are compatible with the task of reducing the error in the region of interest.

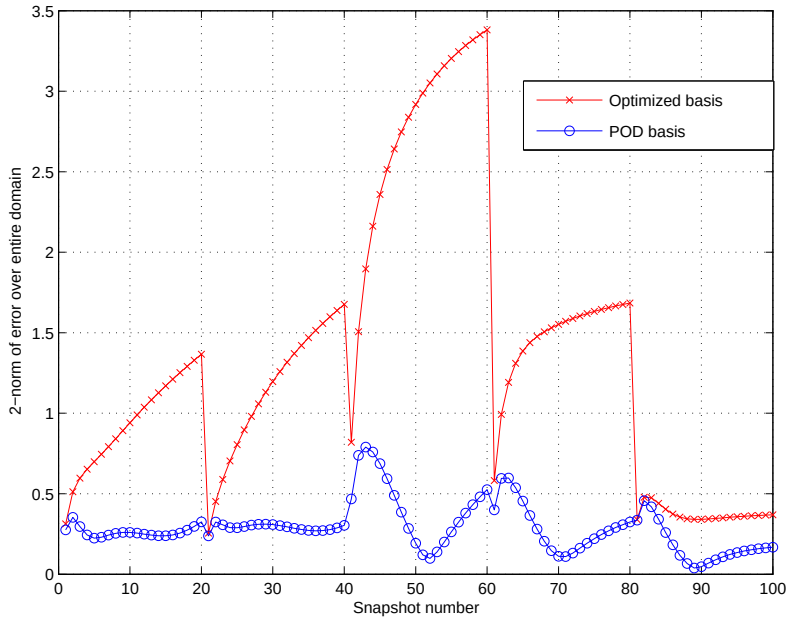

Fig. 4. Norm of the error in temperature prediction over the entire domain for each snapshot using POD and optimized basis. The optimized basis was selected so as to minimize the error over the region $0.5 < x < 1$, $0.5 < y < 1$.

4.2 Subsonic Rotor Blade Example

The second example considers forced response of a subsonic rotor blade that moves in unsteady rigid motion. The flow is modeled using the two-dimensional Euler equations written at the blade mid-section. In this case the governing PDEs are given by

$$\frac{\partial w}{\partial t} + \nabla \cdot \mathcal{F}(w) = 0, \tag{36}$$

where $w(x, y, t)$ is the conservative state vector,

$$w = (\rho,\ \rho u,\ \rho v,\ \rho E)^T, \tag{37}$$

and $\mathcal{F} = (F^x, F^y)$ is the inviscid Euler flux,

$$F^x = \left(\rho u,\ \rho u^2 + P,\ \rho uv,\ \rho uH\right)^T,$$
$$F^y = \left(\rho v,\ \rho uv,\ \rho v^2 + P,\ \rho vH\right)^T. \tag{38}$$

In the above equations, ρ is the density, u and v are respectively the $x-$ and $y-$component of velocity, E is the total energy, P is the pressure, and $H = E + P/\rho$ is the total enthalpy. The equation of state is the ideal gas law

$$P = (\gamma - 1)\left[\rho E - \frac{1}{2}\rho\left(u^2 + v^2\right)\right], \tag{39}$$

where γ is the ratio of specific heats.

The geometry of the blade is shown in Figure 5 along with the unstructured grid for a single blade passage, which contains 4028 triangular elements. The Euler equations (36) are discretized in space with a discontinuous Galerkin (DG) method, as described in [25]. To solve the forced response problem of interest here, the steady-state solution is first obtained by solving the discretized nonlinear system of equations. For the case considered here, the incoming steady-state flow has a Mach number of $M = 0.113$ and a flow angle of $\beta = 59°$. Flow tangency boundary conditions are applied on the blade surfaces. Since the rotor is cyclically symmetric, the steady flow in each blade passage is the same and the steady-state solution can be computed on a computational domain that describes just a single blade passage. Periodic boundary conditions are applied on the upper and lower boundaries of the grid to represent the effects of neighboring blade passages.

A linearized model is derived for unsteady flow computations by assuming that the unsteady flow is a small deviation from steady state. The details of the unsteady DG method are given in Bui-Thanh et al. [26]. Linearization of the unsteady Euler equations about the steady state yields a linear time-invariant

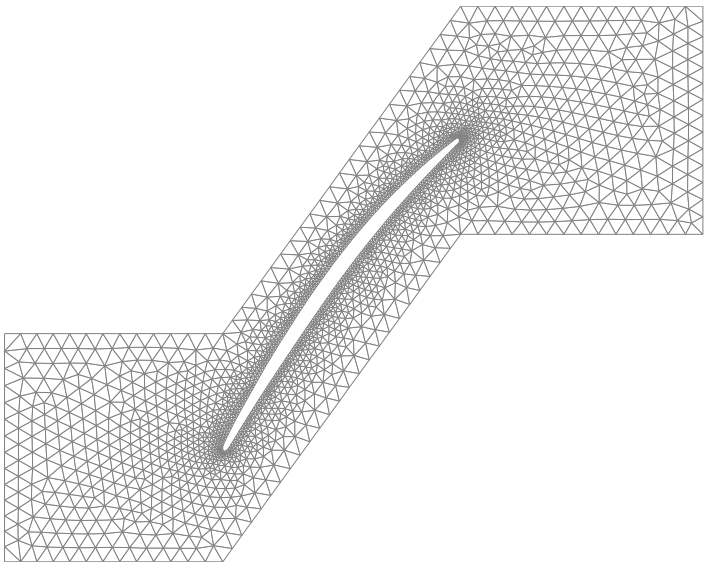

Fig. 5. Geometry and CFD mesh for a single blade passage.

system of the form (1)–(3), where the state vector, $u(t)$, contains the unknown perturbation flow quantities (density, Cartesian momentum components and energy). For the DG formulation, the states are the coefficients corresponding to each nodal finite element shape function. Using linear elements, there are 12 degrees of freedom per element, giving a total state-space size of $N = 48336$ per blade passage. For the problem considered here, the forcing input, $f(t)$, describes the unsteady motion of the blade, which in this case is assumed to be rigid plunging motion (vertical motion with no rotation). The output of interest, $g(t)$, is the unsteady lift force generated on the blade. The initial perturbation flow is given by $u_0 = 0$.

Using the linearized Euler equations reduces the complexity and computational time of unsteady calculations considerably. In particular, in the linearized setting, unsteady computations can be done on a single blade passage in the frequency domain, using symmetry arguments and complex periodicity conditions to account for the effects of neighboring blade rows. However,

the model for a single blade passage has dimension $N = 48336$, and solution of the linearized equations can be too costly for many applications, such as aeroelastic analyses, which require coupling of a fluid dynamic and structural model, and analysis of the effects of blade mistuning, which is the case when all blades in the rotor are not identical. When mistuning exists, symmetry can no longer be exploited in the frequency domain and unsteady computations must be carried out on the full rotor, which for this example has 56 blades. The goal is therefore to create a reduced-order model that accurately represents the dynamic relationship between blade motion and lift force.

Snapshots were taken by computing the response of the blade to a pulse input in plunging motion. For this input, the blade vertical position as a function of time is given by

$$h(t) = \bar{h} e^{-g(t-t_0)^2}, \tag{40}$$

where the parameters $\bar{h} = 1$, $g = 0.02$, and $t_0 = 40$ were chosen based on the range of motions that are expected in practice, and all quantities are non-dimensionalized with the blade chord as a reference length and the inlet speed of sound as a reference velocity. The unsteady simulation was performed with a timestep of $\Delta t = 0.1$ from $t = 0$ to $t_f = 200$. A set of POD basis vectors was computed from this collection of 2000 snapshots. POD reduced-order models were then obtained by projecting the linearized Euler equations onto the subspace spanned by the POD basis vectors, for various basis sizes.

Although the POD is very commonly used for fluid dynamic applications such as this one, an important limitation is highlighted by the results shown in Table 2. The table shows the value of the cost functional defined by (18) for

each of the POD-based reduced-order models (note that in this case there is no variation of parameters, i.e. $S = 1$). Using this set of snapshots to compute the POD basis and then applying the projection as in (4) and (5) results in unstable POD-based reduced models for bases of size 1 through 10. That is, for $m \leq 10$, one or more of the generalized eigenvalues of the system (5) has negative real part. As a result, the magnitudes of the state vectors $\hat{u}$ computed with these unstable models grow in time, and thus the magnitude of first term of the cost functional in (22) approaches infinity. Even though the POD basis is optimal in the sense that it provides the most efficient representation of the given snapshot set, the results in Table 2 emphasize that this optimality is not related to the quality of the resulting reduced-order model. For this example, the POD basis provides satisfactory models if a larger number of states is used (for example, the error with $m = 11$ states is considered to be acceptable). Modification of the snapshot set might yield better POD-based reduced models. A set of POD-based reduced models was also created using a step input to generate the snapshots. In this case, the POD-based reduced models were again found to be unstable for most choices of the basis size. Improvements to the POD-based models could possibly be achieved by further ad-hoc modification to the snapshot set.

The goal-oriented, model-constrained optimization reduction methodology was applied to this example for a similar range of basis sizes evaluated with the POD. In each case, a continuation in the parameter m was used to initialize the optimization. The set of 2000 snapshots was first reduced to a set of 19 vectors using SVD. Then, as described by (28), we seek the optimal basis that is a linear combination of these vectors. This compression greatly reduces the size of the optimization problem without a significant loss of information, since

Table 2

The objective function given by (22) for POD-based reduced-order models generated using a pulse plunge displacement input for the blade example.

Number of POD basis vectors	$\mathcal{G}_{pod}$
1	Unstable
2	Unstable
3	Unstable
4	Unstable
5	Unstable
6	Unstable
7	Unstable
8	Unstable
9	Unstable
10	Unstable
11	3.78e-09
12	2.16e-10
13	6.08e-11
14	6.58e-12

in this case 19 vectors are sufficient to represent the information contained in the snapshot set. Although we chose to reduce the number of degrees of freedom in the optimization problem, the objective function (18) is defined over

the full set of 2000 solutions used to generate the snapshots. For this example, the *a priori* reduction of the snapshot set works well; however, there is no guarantee that the smaller set will be sufficiently rich to yield an accurate reduced model. In particular, for nonlinear systems, discarding information may be problematic if states that appear unimportant via the SVD analysis are needed to obtain an accurate reduced basis. Defining the error term in (18) over all solutions helps to identify such cases, as the optimizer will be unable to reduce the objective function to a sufficiently low value with the available degrees of freedom.

Figure 6 shows the values of the cost functional (18) versus the number of modes, using the optimized basis vectors. The plot does not show a comparison with the POD-based models, since those models are all unstable for this range of basis functions and the corresponding cost functional values are extremely large. It can be seen that, by attempting to reduce the difference between full-order and model-constrained reduced-order outputs, the optimization approach not only yields a stable reduced model, but also provides very accurate response over the specified range of behavior. The reduced-order output matches the full-order output very accurately with a low number of states. For example, comparing the results in Figure 6 with those in Table 2, it can be seen that the accuracy of the 7th-order optimized reduced model is comparable to that obtained with an 11th-order POD model.

Balanced truncation was also applied to this example; however, we were not able to obtain satisfactory results. The large dimension of the system precludes use of a balancing method that employs dense matrix algebra, such as that in the SLICOT library [27]. Approximate balancing can be achieved for problems of this size using sparse iterative methods such as the low rank square

root method [28] implemented in the Lyapack toolbox [29]. However, application to this example of the Lyapack approximate balanced truncation method with the default settings resulted, as for the POD, in unstable reduced-order models. The Lyapack users' manual suggests experimentation with different parameter combinations until the reduced model is stable [29]. Using this ad-hoc approach, we were able to obtain stable approximate balanced truncation models, but the quality of the models was extremely poor.

Figure 7 shows simulation results for a pulse with $\bar{h} = 1$, $g = 0.02$, and $t_0 = 40$ (i.e. the same parameters used to generate the reduced model). With just $m = 3$ states in the reduced model obtained using the optimized basis, there is a small discrepancy between the full-order and reduced-order outputs. With $m = 8$ reduced states, the results are indistinguishable. Figure 8 shows the eigenvalues of the unstable POD reduced model with eight basis vectors and its stabilized counterpart computed using the optimized basis. It can be seen that the spectra of the models differ widely.

5 Conclusions

The goal-oriented, model-constrained optimization approach presented here provides a general framework for construction of reduced models, and is particularly applicable to optimal design, optimal control and inverse problems. The optimization approach provides significant advantages over the POD by targeting the projection basis to output functionals of interest, by providing a framework in which to treat multiple parameter instances, and by incorporating the reduced-order governing equations as constraints in the basis derivation.

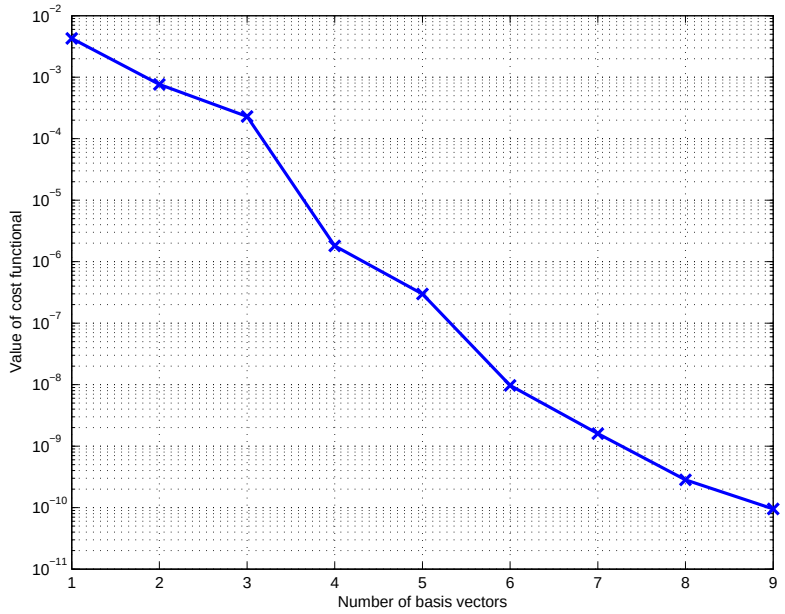

Fig. 6. Cost functional values versus number of modes using the optimized basis vectors for the blade example.

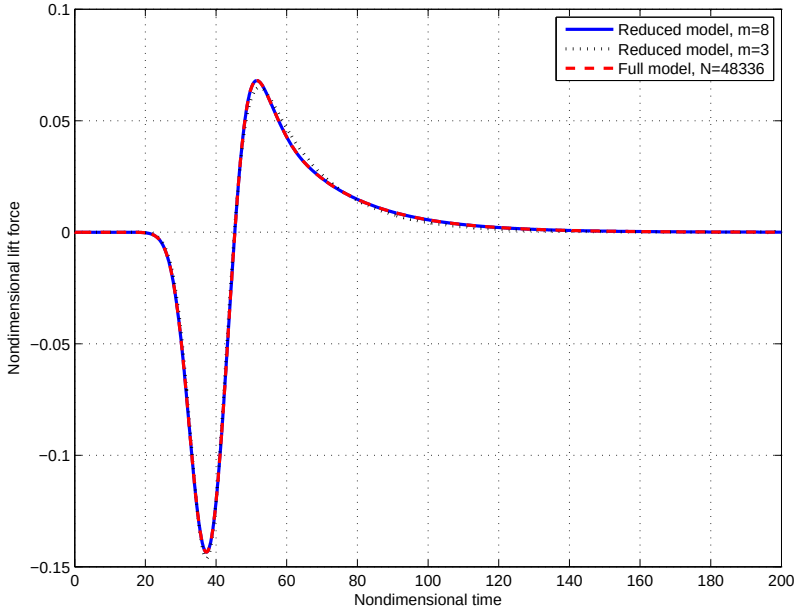

Fig. 7. Simulation results for a pulse input in blade plunge displacement using reduced models of size $m = 3$ and $m = 8$ compared with full-order CFD results.

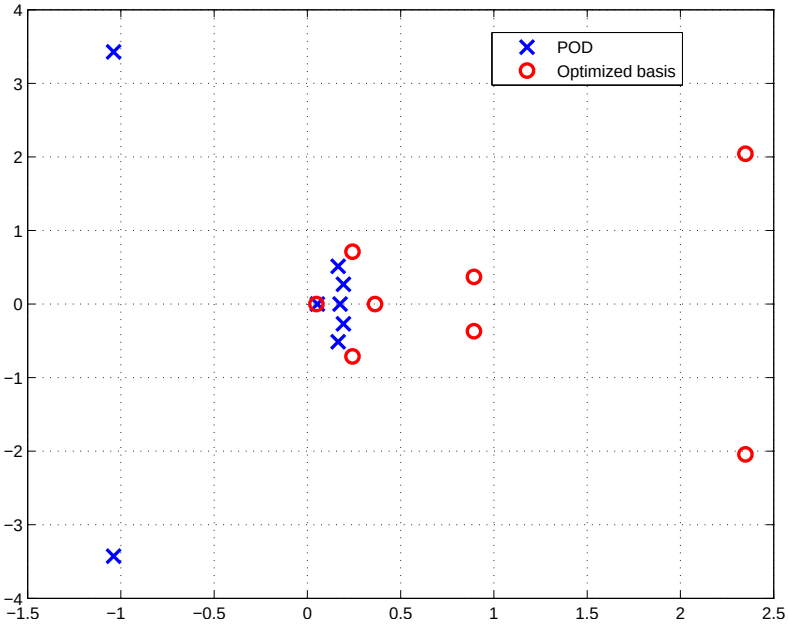

Fig. 8. Eigenvalues of 8th-order reduced models using POD and optimized basis for the blade example.

Selection of an appropriate snapshot set remains an open issue that is not addressed by the model-constrained optimization approach presented in this paper. Like the POD, the quality of the optimized basis depends on the richness of information contained within the available snapshots. For large-scale problems, increasing the size of the snapshot set may increase the computational demands of determining the optimized basis, since the dimension of the optimization variables scales linearly with the number of snapshots. However, when an adjoint-based matrix-free Newton-conjugate gradient method is used to solve the optimization problem (as in this paper) the number of iterations, and thus the cost of solving the optimization problem, scales linearly with the number of dominant modes, as opposed to the number of available snapshots. Even in the case when the number of dominant modes is large, solution of the optimization problem remains tractable, since each conjugate-gradient iteration requires forward and adjoint solves of just the reduced model, and not

the full-scale model.

6 Acknowledgements

The MIT authors gratefully acknowledge support from the Singapore-MIT Alliance, and Universal Technology Corporation under contract number 04-S530-0022-07-C1, technical contract monitor Dr. Cross of AFRL. This work was partially supported by the Computer Science Research Institute at Sandia National Laboratories, and the National Science Foundation under DDDAS grants CNS-0540372 and CNS-0540186, program manager Dr. Darema. The authors also gratefully acknowledge the help of Dr. Bader to create the heat conduction finite element model.

References

[1] V. Akçelik, G. Biros, O. Ghattas, Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation, in: Proceedings of SC2002, 2002.

[2] V. Akçcelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. Kim, J. Lopez, D. O'Hallaron, T. Tu, J. Urbanic, Terascale forward and inverse earthquake modeling, in: Proceedings of SC2003, 2003.

[3] V. Adamjan, D. Arov, M. Krein, Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem, Math. USSR Sbornik 15 (1971) 31–73.

[4] M. Bettayeb, L. Silverman, M. Safonov, Optimal approximation of continuous-time systems, in: Proceedings of the 19th IEEE Conference on Decision and Control, Volume 1, 1980.

[5] S.-Y. Kung, D. Lin, Optimal Hankel-norm model reductions: Multivariable systems, IEEE Transactions on Automatic Control AC-26 (1) (1981) 832–52.

[6] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, IEEE Transactions on Automatic Control AC-26 (1) (1981) 17–31.

[7] D. Sorensen, A. Antoulas, The Sylvester equation and approximate balanced reduction, Linear Algebra and its Applications 351–352 (2002) 671–700.

[8] J. Li, J. White, Low rank solution of Lyapunov equations, SIAM Journal on Matrix Analysis and Applications 24 (1) (2002) 260–280.

[9] S. Gugercin, A. Antoulas, A survey of model reduction by balanced truncation and some new results, International Journal of Control 77 (2004) 748–766.

[10] G. Wood, P. Goddard, K. Glover, Approximation of linear parameter-varying systems, Proceedings of the 35th IEEE Conference on Decision and Control 1 (1996) 406–411.

[11] S. Shokoohi, L. Silverman, P. van Dooren, Linear time-variable systems: Balancing and model reduction, IEEE Transactions on Automatic Control AC-28 (1983) 810–822.

[12] L. Sirovich, Turbulence and the dynamics of coherent structures. Part 1: Coherent structures, Quarterly of Applied Mathematics 45 (3) (1987) 561–571.

[13] P. Holmes, J. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, UK, 1996.

[14] A. Deane, I. Kevrekidis, G. Karniadakis, S. Orszag, Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders, Phys. Fluids 3 (10) (1991) 2337–2354.

[15] L. Daniel, O. Siong, L. Chay, K. Lee, J. White, Multiparameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models, Transactions on Computer Aided Design of Integrated Circuits 23 (5) (2004) 678–693.

[16] M. Hinze, S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control, in: P. Benner, V. Mehrmann, D. Sorensen (Eds.), Dimension Reduction of Large-Scale Systems, Lecture Notes in Computational and Applied Mathematics, 2005, pp. 261–306.

[17] K. Kunisch, S. Volkwein, Control of Burgers' equation by reduced order approach using proper orthogonal decomposition, Journal of Optimization Theory and Applications 102 (1999) 345–371.

[18] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. Patera, G. Turinici, Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods, Journal of Fluids Engineering 124 (2002) 70–80.

[19] G. Berkooz, P. Holmes, J. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annual Review of Fluid Mechanics 25 (1993) 539–575.

[20] S. Lall, J. Marsden, S. Glavaski, A subspace approach to balanced truncation for model reduction of nonlinear control systems, International Journal on Robust and Nonlinear Control 12 (5) (2002) 519–535.

[21] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA Journal 40 (11) (2002) 2323–30.

[22] J. Nocedal, S. Wright, Numerical Optimization, Springer, New York, 1999.

[23] S. Eisenstat, H. Walker, Choosing the forcing terms in an inexact Newton method, SIAM Journal on Scientific Computing 17 (1996) 16–32.

[24] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, B. van Bloemen Waanders, Parallel algorithms for PDE-constrained optimization, in: M. Heroux, P. Raghaven, H. Simon (Eds.), Frontiers of Parallel Computing, SIAM, 2006.

[25] D. Darmofal, R. Haimes, Towards the next generation in computational fluid dynamics, AIAA Paper 2005-0087, presented at 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January (2005).

[26] T. Bui-Thanh, K. Willcox, Model reduction for large-scale CFD applications using the balanced proper orthogonal decomposition, AIAA Paper 2005-4617, presented at 16th AIAA Computational Fluid Dynamics Conference, Toronto, Canada, June (2005).

[27] A. Varga, Model reduction software in the SLICOT library, in: B. Datta (Ed.), Applied and Computational Control, Signals, and Circuits, Vol. 2, Kluwer Academic Publishers, Boston, 2001, pp. 239–282.

[28] T. Penzl, Algorithms for model reduction of large dynamical systems, Linear Algebra and its Applications 415 (2–3) (2006) 322–343.

[29] T. Penzl, LYAPACK. A MATLAB Toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problems, and Linear Quadratic Optimal Control Problems. Users' Guide (Version 1.0) (1999).