Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Parametric Reduced Order Models Using Adaptive Sampling and Interpolation ⋆

Jeff Borggaard * **Kevin R. Pond** ** **Lizette Zietsman** *

* *Virginia Tech, Blacksburg, VA 24061 USA (e-mails: jborggaard@vt.edu, lzietsma@vt.edu).*
** *Air Force Institute of Technology, Wright Patterson AFB, OH 45433 USA (e-mail: kevin.pond@afit.edu)*

Abstract: Over the past decade, a number of approaches have been put forth to improve the accuracy of projection-based reduced order models over parameter ranges. These can be classified as either i.) building a global basis that is suitable for a large parameter set by applying sampling strategies, ii.) identifying parameter dependent coefficient functions in the reduced order model, or iii.) changing the basis as parameters change. We propose a strategy that combines sampling with basis interpolation. We apply sampling strategies that identify suitable parameter values from which associated basis functions are interpolated at any parameter value in a region. While our approach has practical limits to roughly a handful of parameters, it has the advantage of achieving a desired level of accuracy in parametric reduced-order models of relatively small size. We present this method using a proper orthogonal decomposition model of a nonlinear partial differential equation with variable coefficients and initial conditions.

1. INTRODUCTION

Reduced order models (ROMs) are essential tools to overcome computational hurdles in a number of application areas. These models promise either fast simulation or reduced problem sizes for applications such as weather forecasting, Cao et al. [2006], Daescu and Navon [2008], Fang et al. [2009], control of distributed parameter systems, Burns and King [1998], Ito and Ravindran [1998], Kunisch and Volkwein [1999], Tröltzsch and Volkwein [2009], PDE-constrained optimization, Arian et al. [2002], Bui-Thanh et al. [2004], Sachs and Volkwein [2010], and simulations of random and stochastic systems, Burkardt et al. [2007], van Wyk [2012]. In each of these examples, models are created from one or more simulations of a full-order system, at one or more parameter values, then used to represent systems at different values of these parameters. In most cases, the use of the reduced order model off-design (at parameter values not used to build the original model) leads to poor predictions and potential failure of the algorithm. In a few of the examples above, adaptivity is applied to guarantee model accuracy at new parameter values (e.g., Arian et al. [2002]) but these require iterating between on-line and off-line calculations.

To eliminate this shortcoming of reduced order models, a number of recent works have proposed strategies for creating models that are robust to parameter changes. These include applying sampling methods to build a basis from a sufficiently diverse collection of simulations, Grepl and Patera [2005], Burkardt et al. [2006], replacing coefficients in the reduced order models with functions fitted from data, Peterson [1989], Feng and Benner [2007], Benner et al. [2013], or adjusting the model basis by extrapolation, Hay et al. [2008, 2009, 2010] or interpolation, Amsallem and Farhat [2008], Haasdonk et al. [2008], Lassila et al. [2013]. In this study, we consider the combination of adaptive sampling and Lagrangian interpolation (with a strategy that generalizes to Hermite interpolation) of bases across the parameter region. This leads to low dimensional, parameter dependent models where only the Galerkin projection step needs to be performed on-line (though could be avoided by interpolating the models themselves).

The heart of this combined interpolation and sampling methodology is based on adaptively subdividing the d-dimensional parameter region into d-simplices based on an error indicator (see Pond [2010]). We note that a similar approach was studied in Haasdonk et al. [2011], however the present methodology requires no on-line simulations at the cost of significantly more off-line computational effort. This approach also has practical limits to the dimension of the parameter space where it can be applied, typically up to about seven through nine dimensions depending on the complexity of the underlying model.

In the remainder of this paper we test our combined interpolation and sampling strategy on a model problem. The next section presents the one dimensional Burgers equation with parameter dependent solutions along with a short description of the proper orthogonal decomposition/Galerkin projection approach to building reduced order models. This is followed by short sections that describe a basis interpolation strategy allowing for parameter dependent rank changes of basis functions as well as our sampling algorithm. The paper concludes with results of our numerical study and directions for further research.

⋆ The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

2. PROBLEM DESCRIPTION

Here we describe a parametrized solution to Burgers equation motivated by the study in Kunisch and Volkwein [1999]. The piecewise initial function is modified to allow for a parametrized description and along with varying the viscosity, we use this for testing our interpolation and sampling framework. Consider the $\mathbf{p} = (p_1, p_2, p_3)$-dependent, spatially *periodic* solution $u(x,t;\mathbf{p})$ to Burgers equation solving

$$u_t(x,t;\mathbf{p}) + u(x,t;\mathbf{p})u_x(x,t;\mathbf{p}) = p_1 u_{xx}(x,t;\mathbf{p}), \quad (1)$$

where x lies in the domain $\Omega = [0,1]$, the time $t \in (0,T)$ with $T = 10$, and the coefficient p_1 represents the viscosity parameter. For initial conditions in Ω, we consider

$$u(x,0;\mathbf{p}) = \begin{cases} p_2 \sin(2\pi x) + p_3 \sin(4\pi x), & 0 \le x \le 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

As we show in the numerical results, we obtain significantly different solutions as parameter values $p_1 \in [0.001, 0.01]$, $p_2 \in [0.2, 0.8]$, and $p_3 \in [0.2, 0.8]$ are changed. Parameter vectors will, later on, be indexed (e.g. $\mathbf{p}_1$, $\mathbf{p}_2$, etc.) to indicate the value of parameters at different vertices.

For a fixed parameter $\mathbf{p}$, a reduced order model is constructed using the proper orthogonal decomposition (POD), see e.g. Sirovich [1987], Aubry et al. [1988]. Therefore, assuming $u \in \mathcal{L}_2(0,T;\mathcal{L}_2(\Omega))$, a reduced order representation of the solution $u(x,t;\mathbf{p})$ is given by

$$u_r(x,t;\mathbf{p}) = \sum_{j=1}^{r} \phi_j(x;\mathbf{p}) a_j(t;\mathbf{p}) \quad (2)$$

where the basis functions $\{\phi_j\}$ are solutions to the Fredholm equation

$$\int_\Omega R^s(x,\bar{x};\mathbf{p})\phi(\bar{x};\mathbf{p})\, d\bar{x} = \lambda(\mathbf{p})\phi(x;\mathbf{p}), \quad (3)$$

with $R^s(x,\bar{x};\mathbf{p}) \equiv \frac{1}{T}\int_0^T u(x,t;\mathbf{p})u(\bar{x},t;\mathbf{p})dt$ known as the *spatial autocorrelation* kernel. Under this construction, if $a_j(t;\mathbf{p})$ satisfies $\int_\Omega u(x,t;\mathbf{p})\phi_j(x;\mathbf{p})dx$, the representation (2) is the optimal r-dimensional approximation to the solution $u(x,t;\mathbf{p})$. If we write $a_j(t;\mathbf{p})$ as $\sigma_j(\mathbf{p})\psi_j(t;\mathbf{p})$, where ψ_j is scaled so that $\frac{1}{T}\int_0^T \psi_j^2(t;\mathbf{p})dt = 1$, then $\lambda_j(\mathbf{p}) = \sigma_j^2(\mathbf{p})$ and (2) can be interpreted as a truncated singular value decomposition.

Note that in practice, the solution $u(x,t;\mathbf{p})$ is not known and we use the representation (2) and the basis functions $\{\phi_j\}$ to perform the Galerkin projection of (1) onto the POD basis functions. In this case, the vector of coefficients in (2), $\mathbf{a}(t;\mathbf{p})$ satisfy the nonlinear system of equations (Kunisch and Volkwein [1999])

$$[\dot{\mathbf{a}}(t;\mathbf{p})]_i = [\mathbf{A}(\mathbf{p})\mathbf{a}(t;\mathbf{p})]_i + \mathbf{a}^T(t;\mathbf{p})\mathbf{B}_i(\mathbf{p})\mathbf{a}(t;\mathbf{p}), \quad (4)$$

$i = 1, \ldots, r$, where for a given POD basis at parameter $\mathbf{p}$, the matrix $\mathbf{A}$ and indexed matrices (tensor) $\{\mathbf{B}_i\}_{i=1}^r$ can be efficiently precomputed from the POD basis functions and the notation $[\mathbf{w}]_i$ represents the ith component of the vector $\mathbf{w}$. The initial conditions for the system above can be found by projection $a_i(0;\mathbf{p}) = \int_\Omega \phi_i(x;\mathbf{p})u(x,0;\mathbf{p})dx$.

At a given sample point $\mathbf{p}$, the error between the finite element simulation to (1) and the POD model (2) with coefficients obtained by integrating equation (4) is easily controlled by increasing the size of the basis. The projection error is directly related to the truncated eigenvalues $\{\lambda_i\}_{i=r+1}^{\infty}$ of (3). For our study, we keep the size of the reduced order models fixed ($r = 6$) and control the error by determining a set of sample points that are vertices of d-simplices (here, tetrahedra since $d = 3$), computing a POD basis at each vertex, then interpolating these bases to other parameter values.

We are interested in controlling the error when we generate bases at the new points by interpolating stored bases that were generated at adaptively computed sample points. At any parameter $\mathbf{p}$ and using any rth order basis $\{\phi_j\}_{j=1}^r$, the error is defined as

$$e(\{\phi_j\}_{j=1}^r;\mathbf{p}) \equiv \left(\int_0^T \int_\Omega |u(x,t;\mathbf{p}) - u_r(x,t;\mathbf{p})|^2 dx dt \right)^{\frac{1}{2}} \quad (5)$$

where the coefficients of u_r are determined by solving (4) with matrices $\mathbf{A}$ and $\{\mathbf{B}_i\}_{i=1}^r$ calculated using the interpolated POD basis. Note that for estimating the error in our on-line calculations, we will be calling the function (5) with $\mathbf{p}$ and an r-dimensional basis that is computed by interpolating from the four vertices of the tetrahedron that contains $\mathbf{p}$. The details of the basis interpolation are provided below.

3. BASIS INTERPOLATION

As the following experiments show, one must exercise care when interpolating bases (eigenvectors) across parameters. In Fig. 1, we present the values of $\{\lambda_i\}_{i=1}^{20}$ for $p_2 = 0.5$, $p_3 = 0.2$, and p_1 ranging between values of 0.001 and 0.01. Likewise Fig. 2 presents the first 20 eigenvalues for

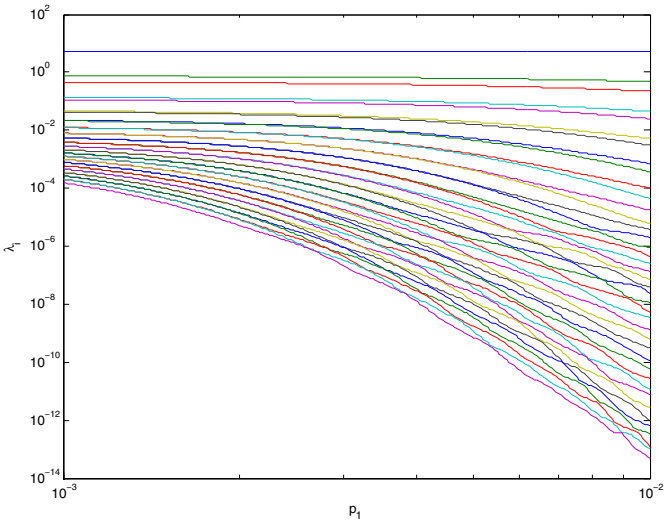

Fig. 1. Eigenvalues vs p_1 for ($p_2 = 0.5$, $p_3 = 0.2$)

$p_1 = 0.001$ and $p_3 = 0.2$. In Fig. 1 the observed larger magnitudes of eigenvalues for smaller values of p_1 indicates that the models are less accurate when the equation has a stronger convection term. The adaptive sampling strategy

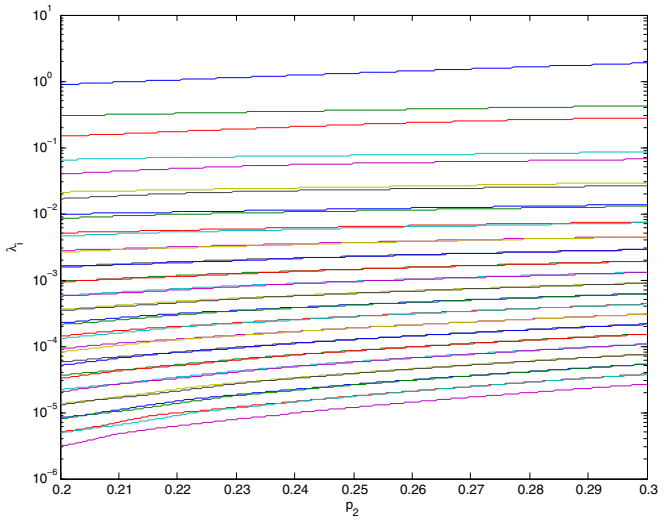

Fig. 2. Eigenvalues vs p_2 for ($p_1 = 0.001$, $p_3 = 0.2$)

described in Section 4 below naturally requires more points in this region since interpolated bases also happen to be less accurate for smaller values of p_1.

It can also be observed that higher order eigenvalues of (3) cross as the value of p_1 is increased from 10^{-3}. This crossing is also observed in Fig. 2 for fixed values of p_1 and p_3, but as p_2 is varied. Therefore the first step in the interpolation of basis elements is to ensure that the correct basis elements are identified at the different vertices of the tetrahedron containing $\mathbf{p}$. There are numerous approaches and many can be found in the survey paper, Benner et al. [2013]. For example, this may be handled using a *congruence transformation*, Lieu et al. [2005], Amsallem and Farhat [2011]. As an alternative, we simply compute a (signed) permutation matrix that accommodates the switching. Note that we also have access to more basis vectors at each vertex than we use to build the reduced-order model. The storage of these extra basis vectors allows us to attempt to accommodate switches in the eigenvector rank outside the set of the first r eigenpairs, $\{(\lambda_1(\mathbf{p}), \phi_1(\mathbf{p})), \ldots, (\lambda_r(\mathbf{p}), \phi_r(\mathbf{p}))\}$, such as when $\lambda_r(\mathbf{p})$ continuously changes to $\lambda_{r+1}(\tilde{\mathbf{p}})$ as $\mathbf{p}$ continuously changes to $\tilde{\mathbf{p}}$. Note that this approach would be challenging to study for $d = 1$ (comparing rank switches, touches, and repeated roots) and is only presented here for this $d = 3$ experiment as our first attempt at this heuristic.

Essentially, for any d-simplex, we identify $\mathbf{p}_1$ as the vertex closest to the desired parameter value. We then compute the representation of the POD bases at vertices $(\mathbf{p}_2, \mathbf{p}_3, \ldots, \mathbf{p}_{d+1})$ in the POD basis at vertex $\mathbf{p}_1$ and use these representations to produce appropriate permutation matrices to match up eigenpairs if rank switching occurs. We consider the basis interpolation between two parameter values. Here we present a strategy that preserves orthogonality at intermediate parameter values. To simplify the discussion, we consider the discretized version of the Fredholm problem (3). Thus columns of $\mathbf{\Phi}$ contain finite element coefficients corresponding to discretized basis functions where the jth column of $\mathbf{\Phi}$, written as ϕ_j, satisfies

$$\mathbf{R}\phi_j = \lambda_j \mathbf{M}\phi_j$$

as well as the discrete orthogonality conditions $\phi_i^T \mathbf{M}\phi_j = \delta_{ij}$. The matrix $\mathbf{R}$ is the discretized autocorrelation kernel and the matrix $\mathbf{M}$ is the finite element mass matrix. We index the bases represented by $\mathbf{\Phi}$ by their vertex numbers. Therefore, there is a basis $\mathbf{\Phi}_k$ defined for each $\mathbf{p}_k$. The interpolation of the basis to parameter $\mathbf{p}$ is performed by creating (signed) permutation matrices $\mathbf{P}_{1k}$ that represent sign changes in eigenvectors and the reranking of eigenvalues as $\mathbf{p}$ changes from $\mathbf{p}_1$ to $\mathbf{p}_k$. Essentially, we look to define $\mathbf{P}_{1k}$ as the solution to an orthogonal Procrustes problem

$$\min_{\mathbf{P}} \|\mathbf{\Phi}_1 - \mathbf{\Phi}_k \mathbf{P}\|_{2,\mathbf{M}}.$$

A straight-forward linear interpolation is defined as

$$\tilde{\mathbf{\Phi}} = r\mathbf{\Phi}_1 + s\mathbf{\Phi}_2\mathbf{P}_{12} + t\mathbf{\Phi}_3\mathbf{P}_{13} + (1-r-s-t)\mathbf{\Phi}_4\mathbf{P}_{14}.$$

For our experiments, the discretization level was such that $\tilde{\mathbf{\Phi}}$ nearly solved the Fredholm equation and the discrete orthogonality conditions. We applied the modified Gram-Schmidt algorithm with $\mathbf{M}$ weighted inner-products to find the $\mathbf{M}$-orthogonal interpolant $\mathbf{\Phi}$.

4. SAMPLING METHODOLOGY

The adaptive interpolation-based strategy known as the multidimensional adaptive quadrature routine over simplices (MAQS), Pond [2010], takes a given function [1] and provides a linear approximation over a given hypercube. This is accomplished by first subdividing the hypercube into simplices and then obtaining a linear approximation of the function, along with an estimate of the error [2] of the approximation, over each simplex. The algorithm creates a priority queue of the simplices, based on the size and estimated error associated with each simplex. The algorithm then proceeds to process (subdivide) the simplex associated with the highest priority. In this manner, we decrease the estimated error in each iteration (based on a predetermined heuristic). The algorithm subdivides a simplex by splitting it into 2^d sub-simplices and calculates new function estimates and error estimates for each simplex. This process is repeated until the required error tolerance or other stopping criteria (e.g. maximum number of function evaluations) is met.

The function estimate is based on a multivariate Lagrange interpolation rule in the spirit of Sauer and Xu [1994]. This allows function evaluations to be reused until the algorithm terminates. This stems from the fact that all one needs is a poised set of points and not a predetermined set of points, as required by Chebyshev methods, for our interpolation to exist. Additionally, each simplex can be processed independently, which allows for the parallel processing of multiple high priority simplices. Since we use interpolatory methods, restarting the calculation at a later time for greater precision requires no more additional function evaluations than if the desired tolerance were initially required. Changing the desired tolerance does not change how the algorithm adapts to the given function.

[1] These functions need to be well approximated by piecewise polynomial interpolants for MAQS to be useful.

[2] The error estimate is the absolute value of the difference between a linear and quadratic approximation of the function over a given simplex.

Also, using simplices allows for the algorithm to handle parameters that can range over arbitrary polyhedral domains, hence, approximated by tessellating the region. For more information on the methods used to derive the error estimation and proofs of convergence, please see Chapters 3 and 4 in Pond [2010]. The algorithm itself, is provided in the Appendix.

NUMERICAL RESULTS

For this study, we investigate the one dimensional viscous Burgers equation with three parameters. One parameter is the diffusion coefficient and the remaining parameters describe a range of initial conditions. Their values are allowed to vary in the three-dimensional cube defined by: $[0.001, 0.01] \times [0.2, 0.8] \times [0.2, 0.8]$. We run the algorithm until a total of 1,500 function evaluations are reached. Along the way to 1,500 we calculate the expected error and actual error given the current set of vertices in our scheme for the next set of vertices (the points needed in the next level of refinement.) This allows us to observe the relationship between what the algorithm expects (forecasts) the error to be and what the error actually is.

The ratio of the actual error to the expected error is shown in Fig. 3. This ratio fluctuates about one when the algorithm produces its first 1,000 vertices. The variations in this ratio beyond 1,000 vertices are due to the design of the algorithm. Essentially, the algorithm interrogates the function in places where it forecasts the error to be the greatest. However, for convergence to take place as the number of vertices increases, all simplices must be re-interrogated and further divided to make sure no hidden information remains. This is why the solid line is closer to one and sometimes above one. As the algorithm proceeds, the larger remaining simplices must be processed even if the expected error is small. However, this subdivision occasionally identifies some points not well represented by the existing polynomial, and is also required for the convergence theory. However the dotted line, being the average ratio of all simplices processed on a given iteration, shows that the number of cases that cause the ratio to be above one are relatively few.

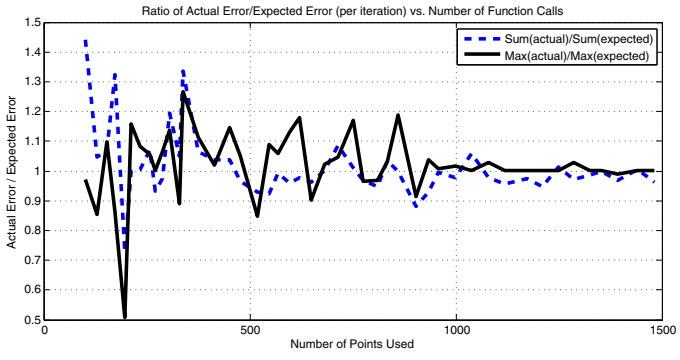

Fig. 3. Ratio of Actual Error/Expected Error (per iteration) vs. Number of Function Calls

Upon completion, the algorithm produced a set of vertices, error estimates at those vertices, as well as a POD basis that can be used to compute an associated ROM for that given vertex (parameter value $\mathbf{p}_k$). To test the

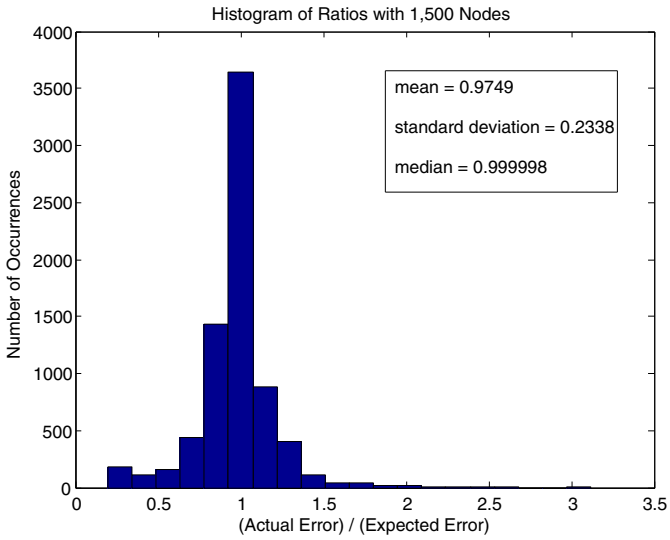

Fig. 4. Histogram of Ratios with 1,500 Vertices

accuracy of the resulting mesh (vertices and simplices) and to investigate the effectiveness of the approximation, we compared the expected errors to the actual errors (calculated by performing a simulation, computing a ROM, and measuring the actual ROM error) at the centroids of each simplex. Theoretically, this should be the place with the largest error based on our linear interpolation scheme. As observed in Fig. 4, the vast majority of the ratios of actual error to the expected error are right around one. This demonstrates again that the method under investigation produces a feasible error estimate.

5. ONGOING WORK

A number of natural extensions of this work are being carried out. First of all, we are considering modifications to the basis interpolation algorithm that promote better approximations to (3) and the orthogonality condition. We attempted to apply the Stiefel manifold interpolation strategy presented in Amsallem and Farhat [2008], Benner et al. [2013] and summarized here. The strategy is presented for interpolation of the POD bases at parameter $\hat{\mathbf{p}}$ from the POD bases computed at vertices of a tetrahedron in parameter space, $\{\mathbf{p}_i\}_{i=1}^4$. We denote these POD bases as $\{\mathbf{\Phi}_i\}_{i=1}^4$, respectively. Note that each POD basis is $\mathbf{M}$-orthogonal, i.e. $\mathbf{\Phi}_i^T \mathbf{M} \mathbf{\Phi}_i = \mathbf{I}$, and the interpolation strategy preserves this fact. Since $\mathbf{M}$ is a symmetric, positive definite matrix, we factor it as $\mathbf{M} = \mathbf{R}^T \mathbf{R}$. We set vertex $\mathbf{p}_1$ as a reference point and $\mathbf{T}_1 = \mathbf{0}$. For each $i = 2, \ldots, 4$, we compute

$$\mathbf{N}_i = \mathbf{R} \left(\mathbf{I} - \mathbf{\Phi}_1 \mathbf{\Phi}_1^T \mathbf{M} \right) \mathbf{\Phi}_i \left(\mathbf{\Phi}_1^T \mathbf{M} \mathbf{\Phi}_i \right)^{-1},$$

it's singular value decomposition $\mathbf{N}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$, and compute the projection to the tangent space

$$\mathbf{T}_i = \mathbf{U}_i \arctan\left(\mathbf{\Sigma}_i\right) \mathbf{V}_i^T.$$

We now apply our Lagrangian interpolation to the matrices $\{\mathbf{T}_i\}_{i=1}^4$,

$$\mathbf{T}(\hat{\mathbf{p}}) = \sum_{i=1}^{4} \ell_i(\hat{\mathbf{p}}) \mathbf{T}_i,$$

where ℓ_i is the Lagrange polynomial that is 1 at $\mathbf{p}_i$ and 0 at the remaining vertices. The singular value decomposition of $\mathbf{T}(\hat{\mathbf{p}})$ is computed as $\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^T$, then the interpolated POD basis at $\hat{\mathbf{p}}$ is computed as

$$\mathbf{\Phi}(\hat{\mathbf{p}}) = \mathbf{\Phi}_1\hat{\mathbf{V}}\cos(\hat{\mathbf{\Sigma}}) + \mathbf{R}^{-1}\hat{\mathbf{U}}\sin(\hat{\mathbf{\Sigma}}).$$

While the resulting basis $\mathbf{\Phi}(\hat{p})$ satisfied the $\mathbf{M}$-orthogonality condition, the basis interpolation procedure outlined in Section 3 was consistently more accurate. This was true even if we applied our transformation to incorporate the correct r basis functions first. We intent to continue looking into this issue.

While not an issue for this numerical study, we have encountered problems when interpolating over large values of Reynolds numbers in complex fluid flows (again, convection dominated). An approach that modifies the interpolation formula with an additional positive definite matrix, solving a Riccati equation, has shown promise in improving the orthogonality condition, but does not provide the same small residual of (3) as direct Lagrange interpolation.

This basis interpolation, and the interpolation-based sampling approach studied here, can be naturally extended to Hermite interpolation using the POD sensitivity analysis approaches developed in Hay et al. [2008, 2009, 2010]. Furthermore, the availability of derivatives of the eigenvalues with respect to parameters can also be used to assist in ranking the eigenvalues, especially when we are close to a crossing. The derivatives may provide information that helps determine the local parametric behavior of the problem.

Finally, since we do store additional basis vectors, the application of an *hp* strategy, such as described in Eftang et al. [2011], would be feasible. A suitable heuristic, to decide whether or not to use additional samples or more basis vectors would be based on the POD eigenvalues and the adaptive sampling error estimates we have in the MAQS algorithm.

ACKNOWLEDGEMENTS

This work was supported in part by the Air Force Office of Scientific Research under contract FA9550-12-1-0173.

REFERENCES

D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, July 2008.

D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33:2169–2198, 2011.

E. Arian, M. Fahl, and E. W. Sachs. Managing POD models by optimization methods. In *Proc. 41st IEEE Conference on Decision and Control*, pages 3300–3305, 2002. ThP04-6.

N. Aubry, P. Holmes, J. L. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192:115–173, 1988.

P. Benner, S. Gugercin, and K. Willcox. A survey of model reduction methods for parametric systems. Technical Report MPIMD/13-14, MPI Magdeburg, 2013.

T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42 (8):1505–1516, 2004.

J. Burkardt, M. Gunzburger, and H. C. Lee. POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1–3):337–355, 2006.

J. Burkardt, M. D. Gunzburger, and C. Webster. Reduced order modeling of some nonlinear stochastic partial differential equations. *International Journal of Numerical Analysis and Modeling*, 4(3–4):368–391, 2007.

J. A. Burns and B. B. King. A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems. *Journal of Vibration and Control*, 4:297–323, 1998.

Y. Cao, J. Zhy, Z. Lou, and I. M. Navon. Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition. *Computers and Mathematics with Applications*, 52:1373–1386, 2006.

D. N. Daescu and I. M. Navon. Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation. *International Journal for Numerical Methods in Fluids*, 2008.

J. L. Eftang, D. J. Knezevic, and A. T. Patera. An *hp* certified reduced basis method for paramtrized parabolic partial differential equations. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):395–422, 2011.

F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. E. Farrell, P. Allison, and A. J. H. Goddard. A POD reduced order 4D-Var adaptive mesh ocean modelling approach. *International Journal for Numerical Methods in Fluids*, 28:127–136, 2009.

L. Feng and P. Benner. A robust algorithm for parametric model order reduction. *Proceedings in Applied Mathematics and Mechanics*, 7 (1):1021501–1021502, December 2007. doi: http://dx.doi.org/10.1002/pamm.200700749.

M. A. Grepl and A. T. Patera. *A Posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced-basis method for evolution schemes with parameter-dependent explicit operators. *Electronic Transactions on Numerical Analysis*, 32:145–161, 2008.

B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):423–442, 2011.

A. Hay, J. Borggaard, and D. Pelletier. On the use of sensitivity analysis to improve reduced-order models. In *Proc. AIAA Flow Control Conference*, 2008. AIAA Paper 2008-4192.

A. Hay, J. Borggaard, and D. Pelletier. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of*

Fluid Mechanics, 629:41–72, 2009.

A. Hay, J. Borggaard, I. Akhtar, and D. Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of Computational Physics*, 229(4):1327–1352, 2010.

K. Ito and S. S. Ravindran. A reduced order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, 1998.

K. Kunisch and S. Volkwein. Control of Burgers' equation by a reduced order approach using proper orthogonal decomposition. *Journal on Optimization Theory and Applications*, 102:345–371, 1999.

T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Generalized reduced basis methods and n-width estimates for the approximation of the solution manifold of parametric PDEs. In *Analysis and Numerics of Partial Differential Equations*, pages 307–329. Springer, Milan, 2013.

T. Lieu, C. Farhat, and M. Lesionne. POD-based aeroelastic analysis of a complete F-16 configuration: ROM adaptation and demonstration. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2005. AIAA Paper 2005-2295.

J. N. Lyness and J. J. Kaganove. Comments on the nature of automatic quadrature routines. *ACM Transactions on Mathematical Software Transactions on Mathematical Software*, 2(1):65–81, 1976.

J. S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777–786, 1989.

K. R. Pond. *Multidimensional Adaptive Quadrature Over Simplices*. PhD thesis, Virginia Tech, 2010.

E. W. Sachs and S. Volkwein. POD-Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):194–208, 2010.

Thomas Sauer and Yuan Xu. On multivariate Lagrange interpolation. *Mathematics of Computation*, 64:1147–1170, 1994.

L. Sirovich. Turbulence and the dynamics of coherent structures, part III: Dynamics and scaling. *Quarterly of Applied Mathematics*, 45(3):583–590, 1987.

F. Tröltzsch and S. Volkwein. POD *a-posteriori* error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44:83–115, 2009.

H.-W. van Wyk. *A Variational Approach to Estimating Uncertain Parameters in Elliptic Systems*. PhD thesis, Virginia Tech, 2012.

Appendix A. THE MAQS ALGORITHM

Given a function $f \in C([0,1]^d)$, an error tolerance ϵ, and a maximum number of allowable function evaluations F_{max}, MAQS proceeds as follows.

(1) The domain, $\Omega = [0,1]^d$ is divided into $d!$ simplices by using only the corners of Ω as the vertices. (Ω can also be an arbitrary d-dimensional cube)

(2) The values $(Q_\Omega, E_\Omega, H_\Delta)$ are initialized to $(0, \infty, \infty)$. (Integral estimate, error estimate, and priority)

(3) The main *while* loop is initiated with the following criteria

$$\text{while}\,(E_\Omega = \Sigma_{\Delta\in\mathcal{T}_n} E_\Delta > \epsilon \;\text{ and }\; F_{evals} < F_{max})\,.$$

(a) For the current iteration, n, the simplices at the top of the priority queue or the simplices with the largest H_Δ value are processed.
- Here the midpoints of each edge of the simplex are evaluated to get a quadratic approximation to f over the simplex.
- This sampling also allows for a linear approximation to f over the 2^d sub-simplices.
- From the linear approximation, a linear quadrature estimate for each new simplex is returned as the Q_Δ value.
- From the integral of the absolute value of the difference of the linear interpolation over each new simplex with that of the quadratic interpolation over the parent simplex, the error estimate E_Δ is obtained for each new simplex.
- After the Q_Δ and E_Δ values are computed we assign each new simplex a priority value H_Δ based on the size and estimated error.
- These three values are stored for each simplex.

(b) After processing is complete, we calculate the new integral and error estimates over Ω and repeat the loop.

Several other input and output options appear in the literature. For a more complete list, see Lyness and Kaganove [1976].