Output Error Estimation for Projection-Based Reduced Models*

Gary Collins†, Krzysztof J. Fidkowski‡, and Carlos E. S. Cesnik§

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Computational modeling is a pillar of modern aerospace research and is increasingly becoming more important as computer technology and numerical methods grow more powerful and sophisticated. However, computational modeling remains expensive for many aerospace engineering problems, including high-fidelity solutions to highly-complex, three-dimensional unsteady vehicle simulations, large-scale aeroservoelastic control problems, and multidisciplinary design optimization. Model reduction methods have therefore garnered interest as an alternative means of preserving high fidelity at a much lower computational cost. Evaluating the accuracy of these models is of a high priority, and reliable adaptation of these reduced models is an on-going goal. This work presents a method for estimating the output error in projection-based reduced models by developing discrete steady adjoint solutions and applying an adjoint-weighted residual error estimation technique. Based on those estimates, this work introduces an adaptation technique which expands the projection basis of the reduced order model with basis vectors chosen based on the error estimate. Error estimation results ranging from scalar transport to compressible flow over a NLR 7301 airfoil are presented and adaptation results of the latter is demonstrated and analyzed.

I. Introduction

Advances in high-performance computing and numerical methods have enabled scientists and engineers to generate high-fidelity solutions to difficult computational fluids problems, such as high-degree of freedom unsteady simulations. Concurrently, the need for generating such high-fidelity solutions has increased with advances in computational design techniques, such as multidisciplinary design optimization (MDO). This need has outpaced the advances in CFD modeling, and the ability to generate solutions from high-fidelity models is a bottleneck in modern aircraft design and optimization.

Model reduction, also known as reduced-order modeling, has garnered increasing attention as a means for alleviating these computational issues. Two general types of model reduction techniques that have gained interest within the aerospace community are interpolation-based and projection-based techniques. Interpolation-based reduced-order modeling techniques attempt to create a surrogate model that directly maps inputs to outputs of a system. Commons ways of approaching this includes the use of Volterra series,[1,2] radial basis function neural network models,[3,4] Kriging surrogate models,[5,6] Gaussian processes,[7] and principal component analysis.[8] Advances in stochastic modeling and machine learning have increased the efficiency and accuracy with which interpolation-based reduced order models are able to predict aerodynamics. However, a drawback of these techniques is the replacement of the physics of the system with a black-box, often tunable, model.

On the other hand, projection-based techniques project the high-fidelity system to a lower-dimensional space, thereby retaining some of the physics of the original equations. This projection is typically done by using sampling techniques of the full-order model to find the most influential modes of the system and then projecting the full-order system to a reduced space spanned by these influential modes. Many projection-based techniques are based upon Proper Orthogonal Decomposition (POD), the use of which in aerodynamics has been common since the late 1990s.[9–12] These techniques include Trajectory Piecewise

*Manuscript to the 2019 AIAA Aviation CFD conference

†Ph.D. Candidate, Department of Aerospace Engineering, gggggggg@umich.edu

‡Associate Professor, Department of Aerospace Engineering, kfid@umich.edu, AIAA Senior Member

§Professor, Department of Aerospace Engineering, cesnik@umich.edu, AIAA Fellow

Linear models[13–15] (TPL), the Continuous and Discrete Empirical Interpolation Method[16–18] (EIM and DEIM), and Least-Squares Petrov-Galerkin methods[19,20] (GNAT and LSPG). The major drawbacks of projection-based methods are the losses in guaranteed stability and difficulties in quantifying the error and order of convergence of a reduced-order model.

Although reduced systems provide approximate solutions for a substantially lowered computational cost, little value can be placed in these solutions without an understanding and quantification of their error. Often, determining the true error of these systems would defeat the purpose of the model reduction as FOM solutions would need to be computed. Fortunately, methods for determining the bounds of the error and computing error estimates for projection-based reduced models have been investigated. With a first-order Taylor expansion of the residual, LeGresley and Alonso[21] estimated the error between a POD solution and a solution to the same problem solved with the union of the original POD basis and addition basis vectors. Applied to a transonic flow problem, this technique was used to determine spatial regions for domain decomposition. With a focus on control problems, Homescu, Petzold, and Serban[22] developed an error estimate based on a combination of adjoint solutions and the small sample statistical condition estimation method. Using this estimate, they were able to determine *regions of validity* for perturbations about initial conditions where the use of POD ROMs would be appropriate.

Another approach for estimating errors, presented by Meyer and Matthies,[23] uses the adjoint weighted-residual. Their technique was demonstrated on a rotor-blade problem where a forward-time primal POD solution and a backward-time dual problem POD solution were generated. Using the residuals from the forward-time problem and the adjoints from the backward-time problem, the error contribution of each basis vector used in the reduced model was determined. Meyer and Matthies then compared solutions generated from POD models with ordering based on singular value energy and from POD models with ordering based on error contribution from the earlier analysis, with variation of the reduced system size.

The work presented here is an extension of the work by Meyer and Matthies. The primary contribution of this paper is the verification of the adjoint-weighted residual error estimation, and the introduction of an adaptation technique based on the error estimates. The adapted basis presented by Meyer and Matthies is constructed by first solving the problem with a large number of basis vectors and then constructing a lower-rank POD basis from a ranking of the basis from the initial solve. Rather than filtering the POD basis from a fine POD ROM solve, a POD solution with a low number of basis vectors is first generated, and then the POD basis is expanded depending on the error estimates and the contribution of error from each basis vector in a higher-rank basis set. This allows for reduced systems to remain coarse where accurate solutions can still be generated and to be refined where inaccuracies in the output are present. Furthermore, this method for adaptation is performed iteratively, in order to account for the effects of basis vector coupling in nonlinear problems, and an analysis of the basis vectors activated during the adaptation is performed.

II. Projection-Based Model Reduction

II.A. System Formulation and Projection

A system of ordinary differential equations, generally arising from spatially-discretized partial differential equations, can be written as,

$$G : \begin{cases} \boldsymbol{M}\frac{d\boldsymbol{x}}{dt} + \boldsymbol{R}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \boldsymbol{0} \\ \boldsymbol{y}(t) = \boldsymbol{y}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \end{cases} . \tag{1}$$

Generally, most fluid dynamics problems can be cast in this setting, with $\boldsymbol{x} \in \mathbb{R}^N$ a vector of flow state variables, $\boldsymbol{u} \in \mathbb{R}^p$ a vector of inputs (such as geometric parameters or boundary conditions), $\boldsymbol{y} \in \boldsymbol{R}^q$ a vector of outputs, and t the time. In the above system, $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is the (sparse) mass matrix, and $\boldsymbol{R} \in \mathbb{R}^N$ is the discretized spatial residual.

II.B. Proper Orthogonal Decomposition

Model reduction by means of projection reduces the degrees of freedom of the system by representing the state vector as a linear combination of basis vectors,

$$\boldsymbol{x} = \boldsymbol{V}\hat{\boldsymbol{x}}, \tag{2}$$

where $\boldsymbol{V} : \hat{\boldsymbol{x}} \rightarrow \boldsymbol{x} \in \mathbb{R}^{N \times n}$ is a matrix whose columns contain the basis vectors, and $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^n$ with $n \ll N$ is the reduced state. Substituting (2) into (1), and dropping the time dependence notation for clarity, yields

$$\begin{cases} \boldsymbol{M} \frac{d}{dt} (\boldsymbol{V}\hat{\boldsymbol{x}}) + \boldsymbol{R}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}, t) = \boldsymbol{0} \\ \boldsymbol{y} = \boldsymbol{y}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}, t) \end{cases}.$$

The first equation above can be multiplied on the left with a left projection matrix, $\boldsymbol{W}^T$, where $\boldsymbol{W}^T\boldsymbol{V} = \boldsymbol{I}$, to create the reduced system,

$$\begin{cases} \hat{\boldsymbol{M}} \frac{d}{dt} (\hat{\boldsymbol{x}}) + \boldsymbol{W}^T\boldsymbol{R}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}, t) = \boldsymbol{0} \\ \boldsymbol{y} = \boldsymbol{y}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}, t) \end{cases}, \tag{3}$$

where $\hat{\boldsymbol{M}} \equiv \boldsymbol{W}^T\boldsymbol{M}\boldsymbol{V}$. Note, it is assumed that both $\boldsymbol{V}$ and $\boldsymbol{W}$ are time invariant. (3) represents the reduced system, where the trajectory of $\hat{\boldsymbol{x}}$ is solved rather than $\boldsymbol{x}$, reducing the degrees of freedom from N to n.

For steady problems, the time derivative term and time dependence are not present, and so the full system reads

$$\begin{cases} \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0} \\ \boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{u}) \end{cases}, \tag{4}$$

and the reduced formulation is

$$\begin{cases} \hat{\boldsymbol{R}}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}) = \boldsymbol{0} \\ \boldsymbol{y} = \boldsymbol{y}(\boldsymbol{V}\hat{\boldsymbol{x}}, \boldsymbol{u}) \end{cases}, \tag{5}$$

with

$$\hat{\boldsymbol{R}} = \boldsymbol{W}^T\boldsymbol{R}.$$

There are several aspects of the above system that categorize the model reduction method in use. Many methods employ either a Galerkin projection framework ($\boldsymbol{W} = \boldsymbol{V}$), while others use a Petrov-Galerkin based framework ($\boldsymbol{W} \neq \boldsymbol{V}$). In this work the Galerkin-based framework is used.

A common method for deriving the set of basis vectors $\boldsymbol{V}$ is through proper orthogonal decomposition (POD), also known as principal component analysis, combined with the method of snapshots, introduced by Sirovich.[24] This method involves collecting a series of full-order solutions (called "snapshots"), performing a singular value decomposition of the set of snapshots, and truncating the resulting left singular vector set to create the state basis vector set. For this discussion denote the set of solution snapshots by

$$\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_K] \in \mathbb{R}^{N \times K},$$

where $\boldsymbol{s}_i \in \mathbb{R}^N$ is a solution for the full-order system for a given set of conditions and K is the number of snapshots collected.

For many discretizations, a singular value decomposition of the snapshot matrix suffices for computing the state basis vector set. However, for modal finite-element discretizations, such as the DG method used in this work, the discrete singular value decomposition will depend on the finite-element basis functions used in the discretization. To eliminate this dependence, a continuous spatial inner product is used, and the singular vectors and values are computed from an eigendecomposition of the corresponding normal matrix. This normal matrix takes the form,

$$\boldsymbol{K}_{\text{norm}} = \boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{M}\boldsymbol{S}, \tag{6}$$

where $\mathbf{M}$ is the mass matrix. The eigenvectors of this system represent the singular vectors of the snapshot set, and the corresponding eigenvalues are the squares of the corresponding singular values ($\sigma_i = \sqrt{\lambda_i(\boldsymbol{K}_{\text{norm}})}$).

These singular vectors have two important properties: they span the snapshot space, and each basis vector is associated with a singular value. The first property implies that if the snapshot set is a good representation of the solution space of interest, then so is the basis set. Obtaining a "good" representation of the solution space thus typically requires that the snapshots are obtained from effectively sampling the parameters and inputs of interest of the system. The second property effectively ranks each singular vector in order of importance to the recreation of the snapshot set. This means that basis vectors associated with relatively small singular values can be removed without very large losses in solution space approximation.

The left singular vectors are truncated, and this smaller set of basis vectors is used as the projection basis, $\boldsymbol{V}$. The criterion for which basis vectors to remove is usually the cumulative sum of the singular values compared to the total sum of singular values, where, for example,

$$\frac{\Sigma_{i=1}^{n}\sigma_i}{\Sigma_{i=1}^{k}\sigma_i} \geq 99.9\% \tag{7}$$

is a typical threshold. Although this criterion measures the accuracy with which the state space can be represented, it does not measure how well the outputs of the system, $\boldsymbol{y}$, can be resolved, and often these are of greater interest. This paper aims to presents ways to estimate the error in output predictions of POD-ROMs by applying adjoint-based techniques in a reduced framework and to adapt based on these estimations.

III. Discrete Steady Adjoints for Reduced Order Models

III.A. Fully Discrete Adjoints and Error Estimation

The adjoint is the sensitivity of an output of the system to perturbations in the residual of the system. Let $y \in \boldsymbol{y}$ be a scalar output of (4), and define by $\boldsymbol{\psi} \in \mathbb{R}^N$ the associated adjoint. Given an infinitesimal perturbation in the residual of (4), the change in y is

$$\delta y = \boldsymbol{\psi}^T \delta \boldsymbol{R}, \tag{8}$$

The infinitesimal residual change elicits a state change, $\delta \boldsymbol{x}$, which must satisfy the linearization of (4),

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\delta \boldsymbol{x} + \delta \boldsymbol{R} = \boldsymbol{0}. \tag{9}$$

Linearizing the output with respect to the state and substituting (8) and (9) yields,

$$\begin{aligned} \delta y &= \frac{\partial y}{\partial \boldsymbol{x}}\delta \boldsymbol{x} \\ \frac{\partial y}{\partial \boldsymbol{x}} &= -\boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}. \end{aligned} \tag{10}$$

Transposing and rearranging yields the *discrete adjoint equation*,

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\right]^T \boldsymbol{\psi} + \left[\frac{\partial y}{\partial \boldsymbol{x}}\right]^T = \boldsymbol{0}. \tag{11}$$

Adjoint-weighted residual error estimation is now a commonly used technique for quantifying the output error in CFD solutions. Such estimates can serve as output corrections or drive mesh adaptation.[25] Presently, the adjoint-weighted residual error estimate technique is briefly presented in order to discuss its application in a reduced setting. The adjoint-weighted residual method estimates the error between the output computed on a fine-space and that computed on a coarse-space, using state computation only on the coarse-space. Typically, coarse and fine-spaces result from differences in mesh refinement or approximation order. Fine-space variables are distinguished from coarse-space variables via h and H, respectively.

Given a coarse-space solution, $\boldsymbol{x}_H$, the injection of the coarse-space solution into the fine-space is defined by, $\boldsymbol{x}_h^H = \boldsymbol{I}_h^H \boldsymbol{x}_H$, where $\boldsymbol{I}_h^H$ is a prolongation matrix that interpolates the state variables from the coarse-space to the fine-space. Also define the state error $\delta \boldsymbol{x}_h \equiv \boldsymbol{x}_h^H - \boldsymbol{x}_h$, where $\boldsymbol{x}_h$ is the fine-space solution. The output error between the coarse and fine-space solutions can then be estimated on the fine-space as

$$y_H(\boldsymbol{x}_H) - y_h(\boldsymbol{x}_h) \approx \frac{\partial y_h}{\partial \boldsymbol{x}_h}\delta \boldsymbol{x}_h \approx -\boldsymbol{\psi}_h^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\delta \boldsymbol{x} \approx -\boldsymbol{\psi}_h^T \left(\boldsymbol{R}_h(\boldsymbol{x}_h^H) - \boldsymbol{R}_h(\boldsymbol{x}_h)\right) = -\boldsymbol{\psi}_h^T \boldsymbol{R}_h(\boldsymbol{x}_h^H) \tag{12}$$

A similar adjoint equation and adjoint-weighted residual formula are presented for a reduced setting in the next section.

III.B. Fully Discrete Adjoints and Error Estimation within a Projected Framework

The reduced-model analog of (11) is based on (5) and reads

$$\left[\frac{\partial \hat{\boldsymbol{R}}}{\partial \hat{\boldsymbol{x}}}\right]^T \hat{\boldsymbol{\psi}} + \left[\frac{\partial y}{\partial \hat{\boldsymbol{x}}}\right]^T = \boldsymbol{0}. \tag{13}$$

Assuming a Galerkin projection and expanding the linearizations with respect to $\hat{\boldsymbol{x}}$ via the chain rule yields

$$\underbrace{\left[\boldsymbol{V}^{\boldsymbol{T}}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\boldsymbol{V}\right]^T}_{n\times n} \underbrace{\hat{\boldsymbol{\psi}}}_{n\times 1} + \underbrace{\left[\frac{\partial y}{\partial \boldsymbol{x}}\boldsymbol{V}\right]^T}_{n\times 1} = \boldsymbol{0}. \tag{14}$$

As mentioned earlier, the typical criterion for truncating the state basis set is based on singular value, (7). This criterion, however, does not indicate how well the particular reduced system will be able to predict the outputs of the system. In this work presents an adjoint-weighted residual error estimate that could be used to provide such a metric. This technique was first introduced by Meyer and Matthies[23] to evaluate basis vectors used for reducing a rotor-blade model.

First, let the coarse-space and the fine-space reduced solutions be labeled with $\hat{\boldsymbol{x}}_H$ and $\hat{\boldsymbol{x}}_h$, and the coarse and fine-space state basis vector sets with $\boldsymbol{V}_H$ and $\boldsymbol{V}_h$ which differ in the number of basis vectors retained. Note, the coarse and fine-space solutions are related to their full-order expansions by $\boldsymbol{x}_H = \boldsymbol{V}_H\hat{\boldsymbol{x}}_H$ and $\boldsymbol{x}_h = \boldsymbol{V}_h\hat{\boldsymbol{x}}_h$. If these two sets of vectors arise from the same snapshot set, it can been seen that $\boldsymbol{V}_H \subset \boldsymbol{V}_h$, which means that the state injection operator $\boldsymbol{I}_h^H$ is lossless. The derivation of the output error estimate remains the same as in (12), and the result is

$$y_H(\hat{\boldsymbol{x}}_H) - y_h(\hat{\boldsymbol{x}}_h) \approx -\hat{\boldsymbol{\psi}}_h^T \hat{\boldsymbol{R}}_h(\hat{\boldsymbol{x}}_h^H) \tag{15}$$

where $\hat{\boldsymbol{x}}_h^H = \boldsymbol{I}_h^H\hat{\boldsymbol{x}}_H$. The fine-space adjoint can be computed exactly by solving (13) on $\boldsymbol{V}_h$ by linearizing about the injected coarse-space primal solution. The next section introduces a novel adaptation technique based on the error estimate presented above.

III.C. POD Adaptation using Adjoint-Weighted Error Estimations

The basic idea of the proposed adaptation method is to use the error estimation in (15) as an adaptation indicator for a coarse solution, and the inner product in (15) as a metric for deciding which basis vectors to add to the coarse ROM when adapting. The general procedure is to generate a POD solution on a coarse number of basis vectors (chosen in order of singular value); predict the error between the coarse basis and a fine set of basis vectors; then, if the predicted error is greater than a chosen tolerance, create a new POD basis that is the union of the coarse POD basis and the basis vectors with the largest inner product components in (15), and then resolve the problem on this new basis. The reasoning for this type of adaptation is that the reduced system can remain coarse in locations where outputs are sufficiently accurate and can be refined where they are not. Additionally, the refinement comes in the form of expansion and not filtering. This follows that the coarse basis is formed to sufficient span the solution space by truncating based on singular value energy. Removal of these basis vectors may harm this property, and thus expansion allows for more fluid dynamics to be introduced to the ROM with little effect on the state representation.

The inner product components in (15) describe the error contribution of the individual basis vectors in the fine-space basis. When the coarse-space basis is a subset of the fine-space, the error contribution of the shared basis vectors is zero. This is due to the coarse-space solution being found via a residual minimization

process,

$$\begin{aligned}
&\hat{\boldsymbol{x}}_H = \arg\min_{\hat{\mathbf{z}}} ||\boldsymbol{V}_H \mathbf{R}(\hat{\mathbf{z}})||_{\mathbf{2}}, \\
\Rightarrow \quad & y_H(\hat{\boldsymbol{x}}_H) - y_h(\hat{\boldsymbol{x}}_h) \approx -\hat{\boldsymbol{\psi}}_h^T \hat{\boldsymbol{R}}_h(\hat{\boldsymbol{x}}_h^H) = -\begin{bmatrix} \hat{\boldsymbol{\psi}}_H^T & \hat{\boldsymbol{\psi}}_{h\notin H}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_H^T \boldsymbol{R}(\hat{\boldsymbol{x}}_H) \\ \boldsymbol{V}_{h\notin H}^T \boldsymbol{R}(\hat{\boldsymbol{x}}_H) \end{bmatrix}, \\
& = -\begin{bmatrix} \hat{\boldsymbol{\psi}}_H^T & \hat{\boldsymbol{\psi}}_{h\notin H}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{V}_{h\notin H}^T \boldsymbol{R}(\hat{\boldsymbol{x}}_H) \end{bmatrix} = \hat{\boldsymbol{\psi}}_{h\notin H}^T \boldsymbol{V}_{h\notin H}^T \boldsymbol{R}(\hat{\boldsymbol{x}}_H)
\end{aligned}$$

where $h \notin H$ are the components of the fine-space basis and adjoint that are not in the coarse-space.

If the estimated error of the coarse-space solution is higher than a desired error threshold, the error contribution of the non-shared basis vectors can be used to determine which of them should be added to the coarse-space basis. This allows each solution process to start with a very low-rank ROM and then only increase the fineness of the solution where the error predictions are high, reducing the overall costs of the online process. Algorithm 1 presents the adjoint-driven POD-ROM adaption process.

Algorithm 1 Simple Adjoint-Driven POD Adaptation

1: Given a coarse-space basis, $\boldsymbol{V}_H$, and a fine-space basis, $\boldsymbol{V}_h$, such that $\boldsymbol{V}_H \subset \boldsymbol{V}_h$.
2: Solve $\hat{\boldsymbol{x}}_H = \arg\min_{\hat{\mathbf{z}}} ||\boldsymbol{V}_H \mathbf{R}(\hat{\mathbf{z}})||_{\mathbf{2}}$
3: Compute the error estimate, $\hat{\boldsymbol{\psi}}_{h\notin H} \boldsymbol{V}_{h\notin H}^T \boldsymbol{R}(\hat{\boldsymbol{x}}_H) = \sum_{i=n_H+1}^{n_h} e_i = E$, from (15) between the coarse and fine-spaces.
4: **if** $E >$ tolerance **then**
5: Sort e_i from largest to smallest, i.e. find $j \in \boldsymbol{J}$ where $e_{j_1} \geq e_{j_2} \geq \cdots \geq e_{j_{n_h}}$.
6: Choose c by $\sum_{i=1}^{c} e_{j_c}$, or *apriori*.
7: Construct $\boldsymbol{V}_{H+c} = \boldsymbol{V}_H \cup \boldsymbol{V}_{j_1,\ldots,j_c}$
8: Solve $\hat{\boldsymbol{x}}_{H+c} = \arg\min_{\hat{\mathbf{z}}} ||\boldsymbol{V}_{H+c} \mathbf{R}(\hat{\mathbf{z}})||_{\mathbf{2}}$
9: **end if**

This process can also be done progressively, where the adaption process is done with multiple small iterative steps, as shown in Algorithm 2. This is attractive for nonlinear problems because due to the inherent coupling between basis vectors, the addition of a basis vector to the working basis set can affect the relative error contribution of the remaining basis vectors not yet added. However, this will inherently increase the computational cost of the solution process, as the residual minimization process may need to be undertaken more times than in the simple adaptation show in Algorithm 1.

Algorithm 2 Progressive Adjoint-Driven POD Adaptation

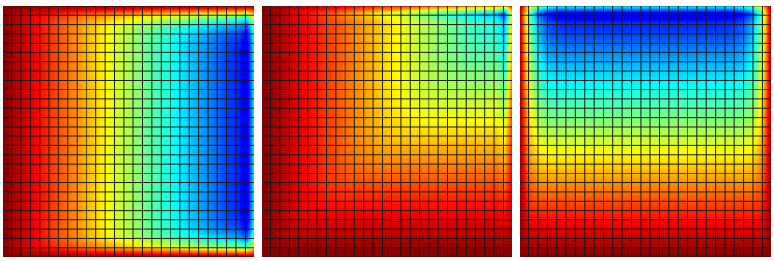
1: Given a coarse-space basis, $\boldsymbol{V}_H$, and a fine-space basis, , $\boldsymbol{V}_h$, such that $\boldsymbol{V}_H \subset \boldsymbol{V}_h$.
2: Set working basis to coarse-space basis $\boldsymbol{V}_W = \boldsymbol{V}_H$, and $W = n_H$.
3: Choose c *apriori*.
4: **while** $E >$ tolerance **do**
5: Solve $\hat{\boldsymbol{x}}_W = \arg\min_{\hat{\mathbf{z}}} ||\boldsymbol{V}_W \mathbf{R}(\hat{\mathbf{z}})||_{\mathbf{2}}$
6: Compute the error estimate, $\hat{\boldsymbol{\psi}}_{h\notin W} \boldsymbol{V}_{h\notin W}^T \boldsymbol{R}(\hat{\boldsymbol{x}}_W) = \sum_{i=W+1}^{n_h} e_i = E$, from (15) between the coarse and fine-spaces.
7: Sort e_i from largest to smallest, i.e. find $j \in \boldsymbol{J}$ where $e_{j_1} \geq e_{j_2} \geq \cdots \geq e_{j_{n_h}}$.
8: Construct $\boldsymbol{V}_W = \boldsymbol{V}_W \cup \boldsymbol{V}_{j_1,\ldots,j_c}$
9: Set $W = W + c$
10: **end while**

Later sections of the paper use these two algorithms to adapt a nonlinear POD model, which shows the benefits of using the progressive adaptation model over the simple adaptation model.

IV. Error Estimation on a Scalar POD Model

A scalar model serves as a perfect problem to verify error estimation technique. This is because, due to the linear formulation, the error estimation in (15) will be exact. A simple scalar advection-diffusion system

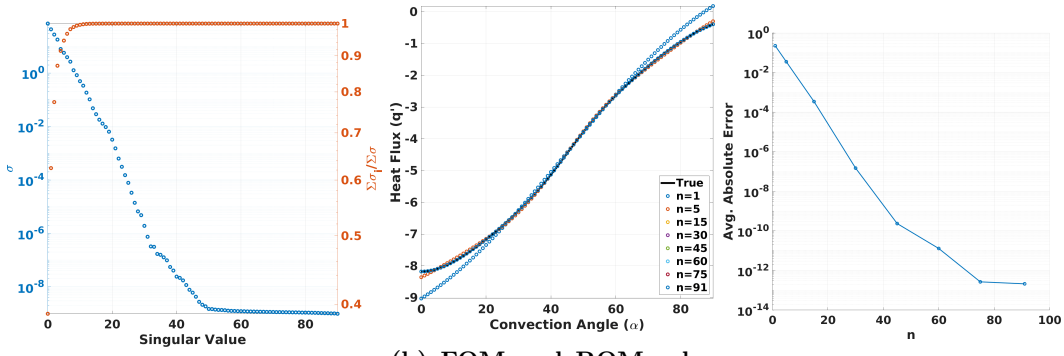
within a square on a uniform grid will serve as the linear test case. The mesh for this case is constructed with 27^2 quadrilateral elements with bi-linear solution interpolation, resulting in 2916 degrees of freedom. State snapshots are obtained by varying the flow direction, α, in the range $[0°, 90°]$ with a velocity magnitude of 1. The boundary conditions are all homogeneous Dirichlet (zero), and a constant unit source is added to the equation. The diffusivity is 0.01, for a domain-based Peclet number of 100. The target output is heat flux through the right boundary.



(a) $\alpha = 0°$ **(b)** $\alpha = 45°$ **(c)** $\alpha = 90°$

Figure 1: Solutions to scalar test problem.

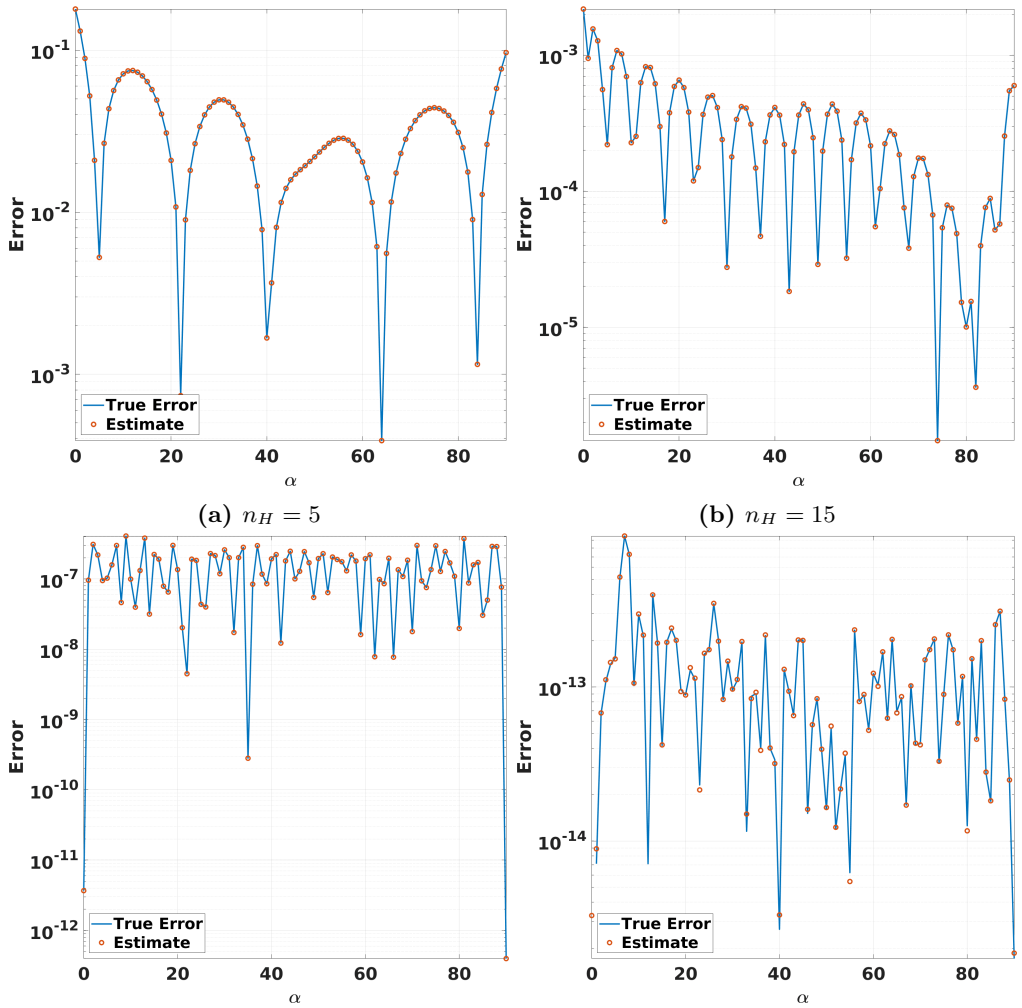
Solutions to this system were generated with `xflow` – an in-house, high-order CFD solver.[26] 91 snapshots were taken with even sampling of the internal flow angle. 99.9% of the total singular value energy is contained in the first 15 singular values. Figure 2 shows a plot of the singular values, the heat flux of the FOM, as well as the POD-ROM reconstruction, and their errors.



(a) Singular Values **(b) FOM and ROM solutions** **(c) Errors**

Figure 2: Scalar FOM and ROM results.

Using the reduced-model error estimation technique developed in Section III.B, the between the outputs of several coarse reduced-model solutions and the 91-basis vector reduced-model were computed. These error estimates were exact, as shown in Figure 3. This demonstrates the validity of the error estimation for linear problems. Nonlinear problems, however, are of much more interest and are more commonly found in modern engineering applications. For nonlinear problems (15) will not be exact, but it will still be useful for analysis and adaptation. This is demonstrated in the following section.



(a) $n_H = 5$ (b) $n_H = 15$

(c) $n_H = 30$ (d) $n_H = 75$

Figure 3: Error estimation of output between coarse-space reduced models with n_H number of basis vectors and a fine-space reduced model with 91 basis vectors.

V. Nonlinear Example

V.A. Error Estimation of POD solutions

The previous section demonstrated the ROM error estimation technique for a linear problem with benign physics. The following example tests the error estimates on a more complicated fluid model, turbulent flow, which is highly nonlinear.

Using `xflow`, snapshots of an NLR 7301 airfoil at a constant 2° angle of attack with a varying Mach number were collected, sampled at 51 uniformly distributed points in $\text{Ma} \in [0.30, 0.80]$. This range was chosen so that both subsonic and transonic flow regimes would be present in the snapshot set. The Reynolds number for all snapshots was $\text{Re} = 2.11\text{M}$, and the snapshots were Reynolds-Averaged Navier-Stokes solutions generated with a Spalart-Allmaras turbulence model. To normalized the system, the static pressure at the boundaries was kept constant, as was the chord length. Variation of the Mach number was achieved by variation of the inflow speed. A FOM solution of the system can be seen in Figure 4.

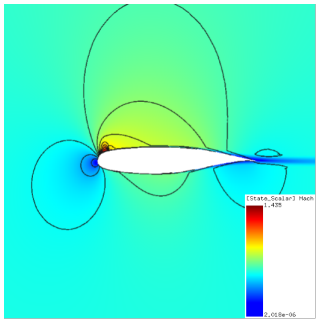
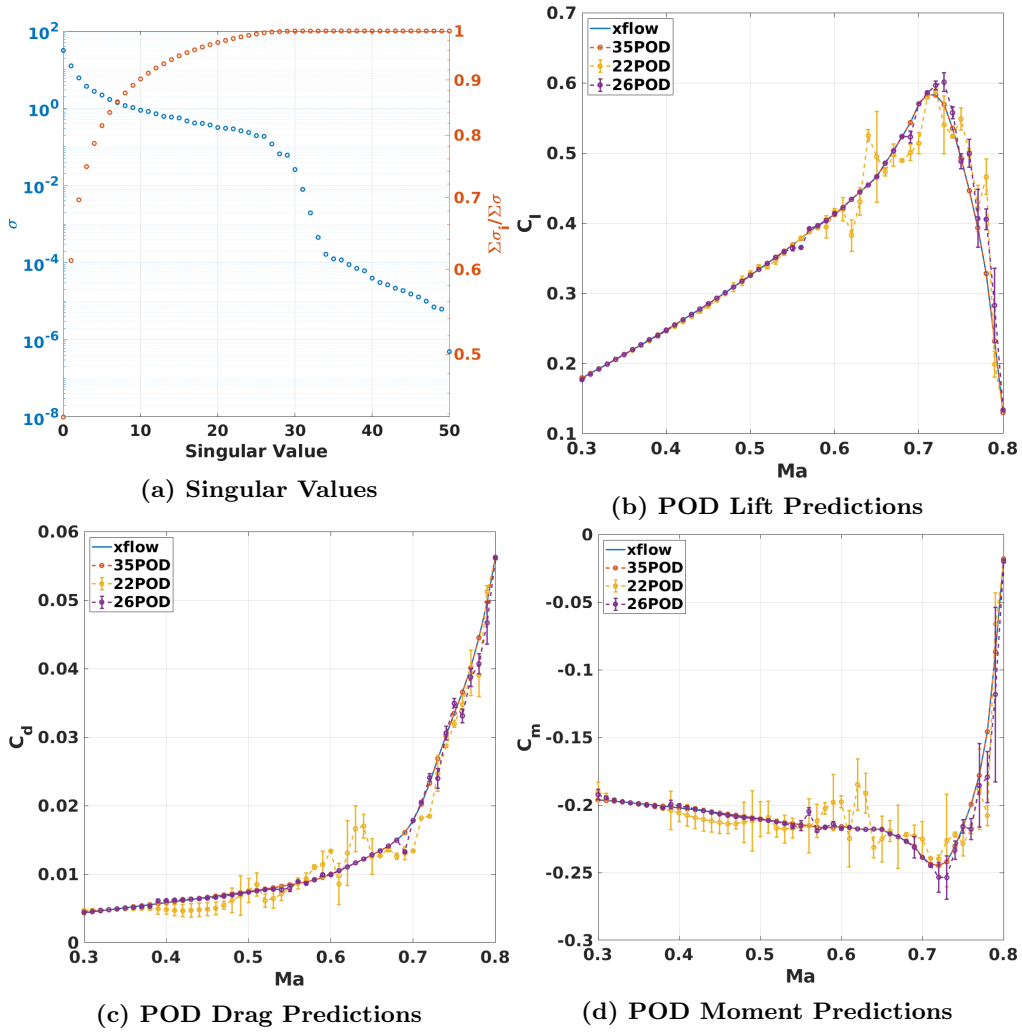


Figure 4: `xflow` Mach contours solution for $\text{Ma} = 0.59$ flow around an NLR 7301 airfoil

The normalized cumulative sum of singular values reached 98% by the 22^{nd} singular value (22POD), 99% by the 26^{th} singular value (26POD), and 99.999%

by the 35$^{\text{th}}$ singular value (35POD). Using these subsets of basis vectors, POD solutions were generated on the Mach values contained in the snapshot set. The lift, drag, and pitch-up, nose-centered moment of these solutions were determined in post-processing. Then, using the adjoint-based error estimation technique, errors were predicted between the 22POD/26POD models and the 35POD model. As seen in Figure 5, the outputs of the 35POD model align with the `xflow` solutions. Between the two coarser models, the 26POD output predictions are more accurate. Additionally, its error estimates are generally smaller than the error estimates of the 22POD model. Although, these error predictions are not exact, it is clear that the error predictions are of the same order as the true error and are generally higher in locations of higher amounts of error. This allows us to run adaptation as described in Section III.C.



(a) Singular Values

(b) POD Lift Predictions

(c) POD Drag Predictions

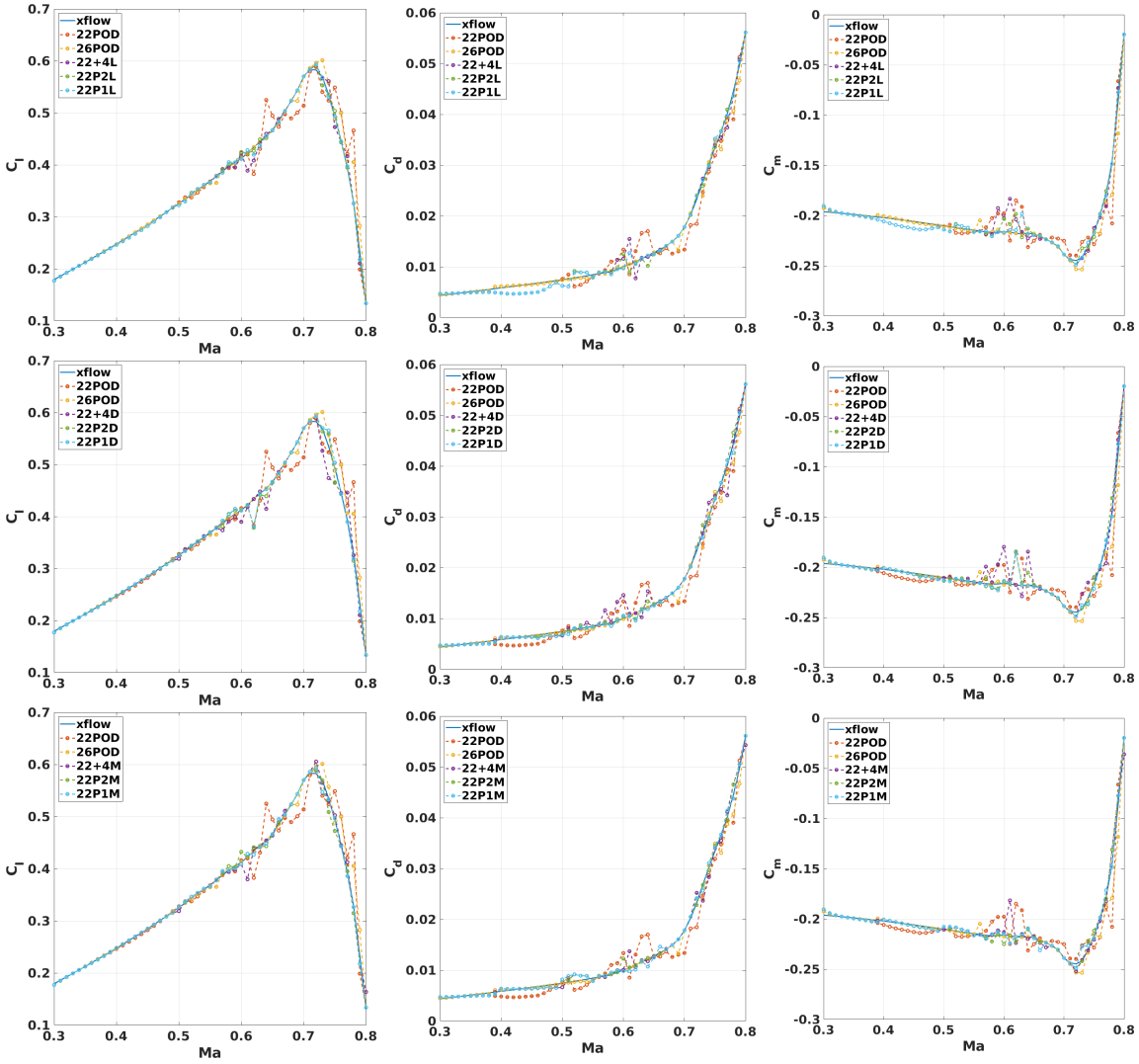
(d) POD Moment Predictions

Figure 5: Singular values of the NLR 7301 example and POD predictions for lift and drag using 22, 26, and 35 state basis vectors.

V.B. POD Adjoint-based Adaptation Example

To test the adaptation scheme, the 22POD basis solutions were adapted to have up to 26 basis vectors, using the pool of 35POD basis vectors as the fine-space. Lift, drag, and moment coefficient errors were targeted for adaptation, and for each of the loads targeted, three adapted solutions were generated: a simple adaptation with 4 basis vectors added (22+4L, 22+4L, 22+4M), a progressive adaptation with 2 basis vectors added per iteration (22P2L, 22P2D, 22P2M), and a progressive adaptation with 1 basis vector added per iteration (22P1L, 22P1D, 22P1M). The tolerances for adaptation were 10^{-3} (lift), 10^{-4} (drag), and 10^{-3} (moment) and were chosen with respect to the order of magnitude of their values in the FOM. Solution snapshots were generated in a Mach number sweep, using the solution at the previous Mach number value as the initial

guess for the next Mach number value.



(a) Lift solutions **(b) Drag solutions** **(c) Moment solutions**

Figure 6: Loads of POD solutions with and without adaptation.

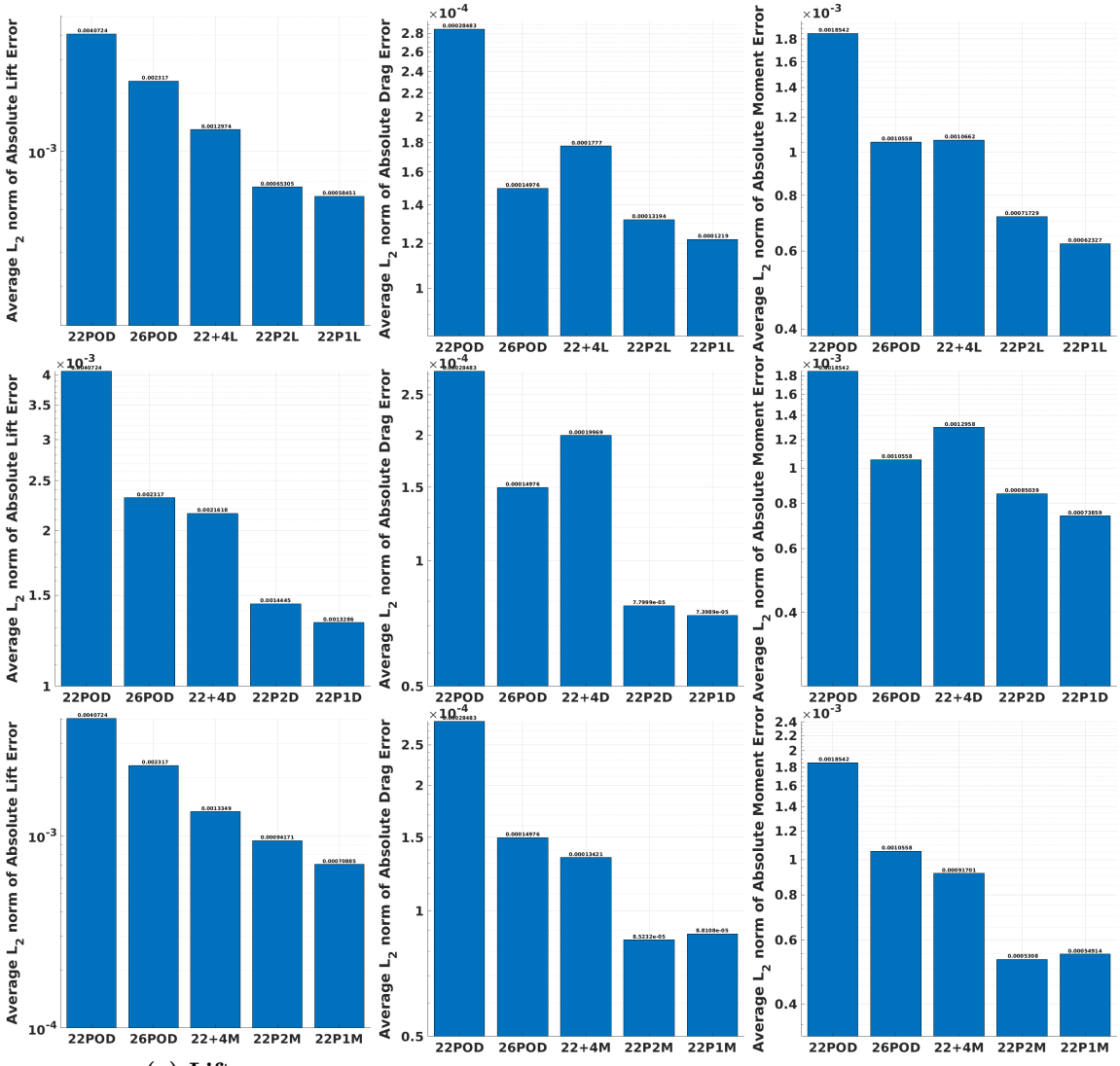
From Figure 7, it can be seen that all of the adapted solutions improve the error of their targeted output from the solution of the 22POD model. The average system size for each adapted solution lies between 23 and 24 basis vectors, with each finer progressive adaptation needing fewer basis vectors. This demonstrates a strength of the adaptive mechanism as the adapted solutions are either on par or exceeded the 26POD solution's accuracy but at a lower cost. Additionally, the accuracy generally improves with the fineness of the progressive adaptation. This supports the earlier assumption that due to nonlinear coupling when a basis vector is added to the system, the contributions of the other basis vectors to the accuracy of the output are affected. Although the progressively adapted solutions use fewer basis vectors than the simple adapted solution, their computation requires more nonlinear ROM solves.

The lift and drag outputs are orthogonal, while the moment output is derived from a linear combination of the two. Likewise, moment-targeted adaptation seems to be more robust and generally tends to adapt in locations where drag or lift have large errors. This can be seen in Figure 8, which displays the activated basis vectors for the adapted solutions. The lift-targeted solutions did not begin to adapt until about 0.59 Mach. This is because the 22POD model is fairly accurate in its lift predictions until this point. Moment and drag, however, both detect the error at $\mathrm{Ma} = 0.4$, and consequently adapt at that point. This corrects the path of the drag and moment predictions for the moment and drag targeted adapted solutions in the region $\mathrm{Ma} \in [0.4, 0.5]$, while the lift targeted adapted solution has the same errors as the 22POD solution

for drag and moment.

Interestingly, the drag and moment targeted solutions activate the same exact basis vectors for the Ma = 0.4 adaptation and are nearly the same for Ma = 0.51 and Ma = 0.52. An explanation for this is that moment is a linear combination of lift and drag. In the lower Mach regions, the value of lift and its contribution to the value and error estimation of moment is low, allowing for the influence of drag to trigger the moment-targeted adaptation. As the Mach number increases, lift begins to increase in value, the influence of drag on the moment decreases, and errors in drag might not be able to contribute to the error in moment above the adaptation tolerance, especially since moment is dominated by lift.

In addition to improving the targeted outputs, all of the adapted solutions show improvements to the predictions of the other integrated outputs. This indicates the possibility that the same characteristics that are relevant to one of the integrated outputs are relevant to the other integrated outputs and are contained in basis vectors that are not necessarily the next largest rank in singular value. This is important, as it demonstrates that the use of singular value as a metric for basis selection, while giving the most optimal basis for state representation, does not give the optimal basis for output prediction.



(a) Lift (b) Drag (c) Moment

Figure 7: Average Total Absolute Error in loads.

The level of consistency with which the adaptation methods select additional basis vectors can be determined by measuring the similarity of basis activation between all of the adapted runs. This can reveal if there is a common set of basis vectors which are used for adaptation and can support the idea that singular values do not choose the most optimal basis vectors for output prediction.

The cosine similarity is used to properly assess the level of coherency between the activated basis vectors for the lift, drag, and moment targeted runs. For each of the adapted solutions, the sum of activations for each basis vector is computed. For example, for the 22P1D solutions, the total number of activations for each basis vector from basis vector 23 to 35 can be expressed as,

$$C_{22\text{P1D}} = \begin{bmatrix} 5 & 2 & 8 & 2 & 5 & 0 & 6 & 2 & 5 & 9 & 8 & 8 & 4 \end{bmatrix},$$

which coincides with the histogram in Figure 9.

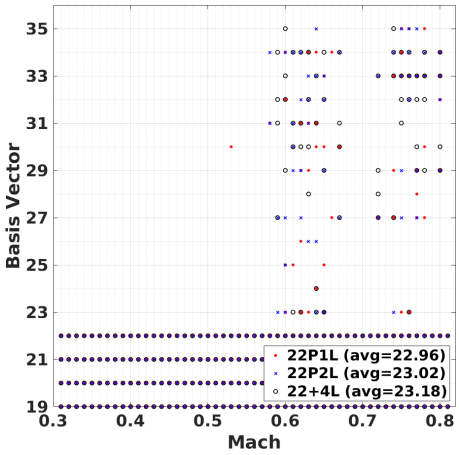
The cosine similarity measures the level of orthogonality between two vectors, $\boldsymbol{a}$ and $\boldsymbol{b}$, of equal length. A common use of cosine similarity is to examine the similarity of two pieces of texts by comparing frequency of specific terms in each.[27] The benefits of cosine similarity comes from the normalization with the magnitudes of the two vectors, preventing the length of the texts from biasing the results. Similarly here, the number of adaptations does not affect the similarity measurement. The cosine similarity is defined by,

$$\mathbf{sim}(\boldsymbol{a}, \boldsymbol{b}) = \cos\theta_{\boldsymbol{a},\boldsymbol{b}} = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}||\ ||\boldsymbol{b}||}, \tag{16}$$

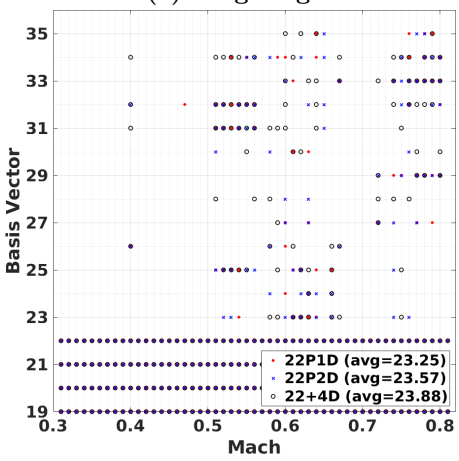
and its value varies between 0 (completely orthogonal/uncorrelated) and 1 (exactly similar). The closer the cosine similarity is to 1, the more similar the two vectors are.

Table 2 presents a matrix of the cosine similarities between the lift, drag, and moment adapted runs. The subscripts reflect the activation count vector with $C_{22\text{totL}}$, $C_{22\text{totD}}$, and $C_{22\text{totM}}$ being the activation count vector for all of the lift, drag, and moment runs, respectively. The diagonal blocks of this table compare runs of similar adaptation targets, while the off-diagonal blocks compare runs that target different outputs.

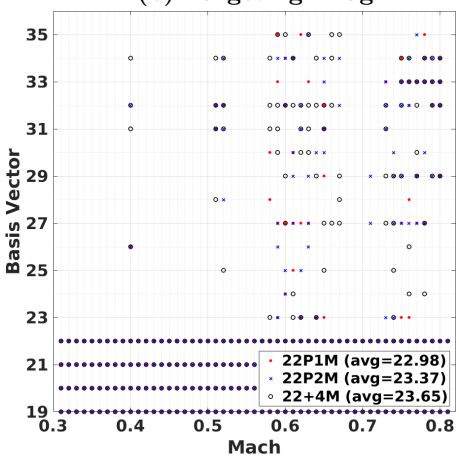
Comparing the totals, the moment and lift adapted outputs have the highest similarity (0.9504), followed by the similarity between the moment and drag (0.9361), with the similarity between the drag and lift falling last (0.9101). These similarity rankings between all of the total runs comport with earlier observations. Since moment is composed of both lift and drag, it can be expected that basis vectors that are relevant to either lift or drag would be relevant to moment. Additionally, the contribution from lift dominates the moment value, naturally leading it to have a higher similarity with moment than drag. Additionally, drag and lift are orthogonal to each other, which explains their lower similarity; however, the similarity between all of the runs remains very high.



(a) Targeting Lift



(b) Targeting Drag



(c) Targeting Moment

Figure 8: Activated basis vectors for adapted solutions

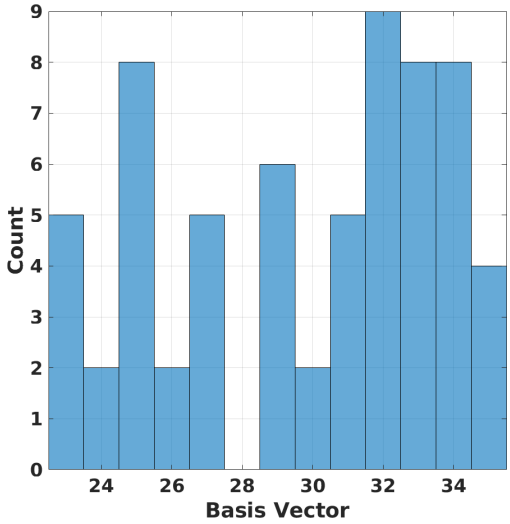


Figure 9: Histogram of 22P1D activations.

Table 1: Modes of total activation vectors.

Targeted Output	Most activated basis (descending)							
Lift	34	33	29	23	27	32	31	30
Drag	32	34	33	25	31	23	29	27
Moment	32	34	27	33	29	23	31	30

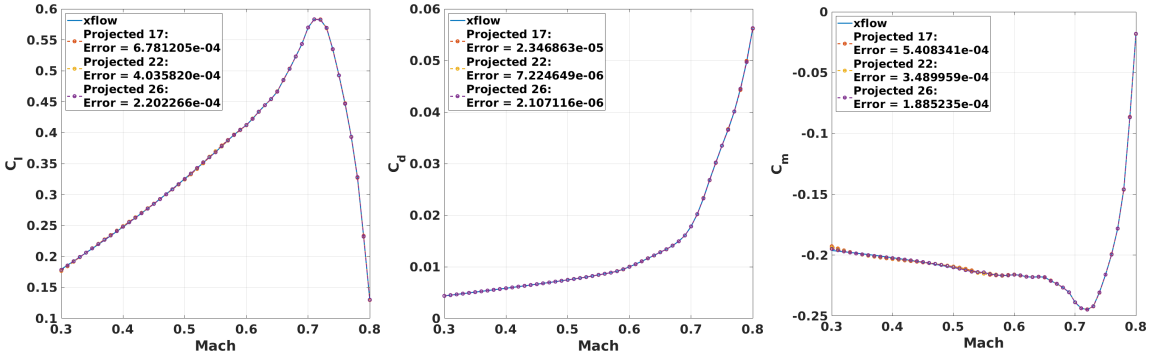
From (Figure 1), it can be seen that basis vectors 33 and 34 both appear in the first 4 most commonly added basis vectors; basis vector 32 appears as the most commonly activated basis for both drag and moment; and basis vectors 29, 27, and 23 appear to be somewhat important for all three runs as well. This supports the earlier assertion that singular value ranking does not choose the most optimal basis for output prediction as many of the commonly chosen basis vectors are outside of the range of the 24^{th} to 26^{th} basis vectors, with the exception of the 23^{rd} which appears as only the 4^{th} most commonly chosen basis vector for lift adaptation.

Table 2: Cosine similarity between adapted runs targeting drag and moment. Lift to Drag, Lift to Moment, Drag to Moment.

	C_{22totL}	C_{22P1L}	C_{22P2L}	C_{22+4L}	C_{22totD}	C_{22P1D}	C_{22P2D}	C_{22+4D}	C_{22totM}	C_{22P1M}	C_{22P2M}	C_{22+4M}
C_{22totL}	1.000	0.9658	0.9724	0.9736	0.9101	0.9117	0.9009	0.8762	0.9504	0.9278	0.9564	0.8921
C_{22P1L}	0.9658	1.000	0.9182	0.9085	0.8788	0.8865	0.8775	0.8359	0.8822	0.8880	0.8939	0.8066
C_{22P2L}	0.9724	0.9182	1.000	0.9144	0.8687	0.8977	0.8650	0.8138	0.9331	0.9370	0.9383	0.8605
C_{22+4L}	0.9736	0.9085	0.9144	1.000	0.9006	0.8733	0.8813	0.8813	0.9442	0.8785	0.9462	0.9159
C_{22totD}	0.9101	0.8788	0.8687	0.9006	1.000	0.9789	0.9883	0.9792	0.9361	0.8657	0.9146	0.9306
C_{22P1D}	0.9117	0.8865	0.8977	0.8733	0.9789	1.000	0.9742	0.9251	0.9354	0.8738	0.9211	0.9185
C_{22P2D}	0.9009	0.8775	0.8650	0.8813	0.9883	0.9742	1.000	0.9443	0.9379	0.8847	0.9154	0.9226
C_{22+4D}	0.8762	0.8359	0.8138	0.8813	0.9792	0.9251	0.9443	1.000	0.8936	0.8066	0.8690	0.9038
C_{22totM}	0.9504	0.8822	0.9331	0.9442	0.9361	0.9354	0.9379	0.8936	1.000	0.9384	0.9910	0.9743
C_{22P1M}	0.9278	0.8880	0.9370	0.8785	0.8657	0.8738	0.8847	0.8066	0.9384	1.000	0.9241	0.8462
C_{22P2M}	0.9564	0.8939	0.9383	0.9462	0.9146	0.9211	0.9154	0.8690	0.9910	0.9241	1.000	0.9544
C_{22+4M}	0.8921	0.8066	0.8605	0.9159	0.9306	0.9185	0.9226	0.9038	0.9743	0.8462	0.9544	1.000

A final point worth noting is how the adaptively added basis vectors do not contribute to the state solution significantly but enable the ROM to push the coarse basis vectors towards the desired solution by contributing to the test space of the problem. At 95% of the singular value energy, a 17POD model

already spans the solution space well. This can be seen in Figure 10, which shows the projections of the FOM solutions into a 17POD, 22POD, and 26POD coarse-spaces. The errors of the outputs of the 22POD projection are orders of magnitude smaller than the results from any of the ROM solutions. This shows that improving a ROM once the POD basis contains a sufficient amount of singular value energy is most effectively done by improving the test space of the POD-ROM as additional basis vectors do not significantly contribute to the trial space. Singular value energy gives the metric of a basis vector's contribution to the trial space but not to the test space and thus does not optimally choose the basis vectors to add to the test space. Higer fidelity was achieved from the adapted ROMs when choosing basis vectors to add to 22POD that were not necessarily the next largest in terms of singular value.



(a) Lift **(b) Drag** **(c) Moment**

Figure 10: FOM solutions projected into coarse POD spaces.

VI. Conclusions

This paper demonstrated an adjoint-based error estimation techniques for projection-based reduced order models and developed adaptation methods based on those techniques. These error estimates were verified as exact on a linear scalar problem. Following the scalar example, the error estimates were demonstrated on a nonlinear problem consisting of turbulent, compressible flow around an airfoil. Although these error estimates were not exact, they were nevertheless relatively accurate. Following this study, a POD adaptation scheme driven by these error estimations was demonstrated on the nonlinear problem. The overall results demonstrated that with fewer basis vectors, a higher output prediction accuracy could be achieved when basis vectors are chosen progressively with adaptation. Additionally, there was a high similarity between the basis vectors activated during the lift, moment, and drag adaptation. These basis vectors tended not to be in the set of the basis vectors with the next highest singular value energies. This showed that the choice of basis vectors based on singular value energy does not produce the most optimal basis for output prediction. Singular value only measures the contribution of a basis vector to the trial space and not the test space of a ROM; when the value of normalized cumulative sum is already very high, contribution to the test space become more important, and thus error estimation based basis expansion is able to create a more optimal basis than singular value expansion for output prediction.

Acknowledgments

The material is based upon work supported by Airbus in the frame of the Airbus/Michigan Center for Aero-Servo-Elasticity of Very Flexible Aircraft. Special thanks to Joël Brezillon and Hans Bleecke for their expert advice and review of this paper.

References

[1]Cheng, C., Peng, Z., Zhang, W., and Meng, G., "Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review," *Mechanical Systems and Signal Processing*, Vol. 87, 2017, pp. 340–364.

[2]Balajewicz, M. and Dowell, E., "Reduced-order modeling of flutter and limit-cycle oscillations using the sparse Volterra series," *Journal of Aircraft*, Vol. 49, No. 6, 2012, pp. 1803–1812.

[3]Zhang, W., Wang, B., Ye, Z., and Quan, J., "Efficient method for limit cycle flutter analysis based on nonlinear aerody-

namic reduced-order models," *AIAA journal*, Vol. 50, No. 5, 2012, pp. 1019–1028.

[4]Winter, M. and Breitsamter, C., *Reduced-order modeling of unsteady aerodynamic loads using radial basis function neural networks*, Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2014.

[5]Glaz, B., Liu, L., and Friedmann, P. P., "Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework," *AIAA journal*, Vol. 48, No. 10, 2010, pp. 2418–2429.

[6]Da Ronch, A., Ghoreyshi, M., and Badcock, K., "On the generation of flight dynamics aerodynamic tables by computational fluid dynamics," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 597–620.

[7]Kou, J. and Zhang, W., "Multi-kernel neural networks for nonlinear unsteady aerodynamic reduced-order modeling," *Aerospace Science and Technology*, Vol. 67, 2017, pp. 309–326.

[8]Klock, R. and Cesnik, C. E., "Rapid Simulation of a Hypersonic Vehicle Through Singular Value Decomposition," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 0178.

[9]Alonso, J. J., LeGresley, P. A., Legresley, P. A., and Alonso, J., "Airfoil design optimization using reduced order models based on proper orthogonal decomposition," 2000.

[10]Dowell, E., Hall, K., Thomas, J., Florea, R., Epureanu, B., and Heeg, J., "Reduced order models in unsteady aerodynamics," 1999.

[11]Willcox, K. and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 40, No. 11, Nov 2002, pp. 23232330.

[12]Rowley, C. W., "Model reduction for fluids, using balanced proper orthogonal decomposition," *Modeling And Computations In Dynamical Systems: In Commemoration of the 100th Anniversary of the Birth of John von Neumann*, World Scientific, 2006, pp. 301–317.

[13]Rewienski, M. and White, J., "Improving trajectory piecewise-linear approach to nonlinear model order reduction for micromachined devices using an aggregated projection basis," *Proceedings of the 5th International Conference on Modeling and Simulation of Microsystems*, 2002, pp. 128–131.

[14]Gratton, D. and Willcox, K., *Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics*, Ph.D. thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2004.

[15]Yao, W. and Liou, M.-S., "A nonlinear modeling approach using weighted piecewise series and its applications to predict unsteady flows," *Journal of Computational Physics*, Vol. 318, 2016, pp. 58–84.

[16]Galbally, D., *Nonlinear model reduction for uncertainty quantification in large-scale inverse problems: application to nonlinear convection-diffusion-reaction equation*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

[17]Chaturantabut, S., *Dimension reduction for unsteady nonlinear partial differential equations via empirical interpolation methods*, Ph.D. thesis, Rice University, 2009.

[18]Xu, J. and Duraisamy, K., "Reduced-Order Modeling of Model Rocket Combustors," *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, 2017, p. 4918.

[19]Carlberg, K., Cortial, J., Amsallem, D., Zahr, M., and Farhat, C., "The GNAT nonlinear model reduction method and its application to fluid dynamics problems," *6th AIAA Theoretical Fluid Mechanics Conference*, Vol. 2730, 2011, pp. 2011–3112.

[20]Carlberg, K., Barone, M., and Antil, H., "Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction," *Journal of Computational Physics*, Vol. 330, 2017, pp. 693–734.

[21]LeGresley, P. and Alonso, J., "Dynamic domain decomposition and error correction for reduced order models," *41st Aerospace Sciences Meeting and Exhibit*, 2003, p. 250.

[22]Homescu, C., Petzold, L. R., and Serban, R., "Error estimation for reduced-order models of dynamical systems," *SIAM Journal on Numerical Analysis*, Vol. 43, No. 4, 2005, pp. 1693–1714.

[23]Meyer, M. and Matthies, H. G., "Efficient model reduction in non-linear dynamics using the Karhunen-Loeve expansion and dual-weighted-residual methods," *Computational Mechanics*, Vol. 31, No. 1-2, 2003, pp. 179–191.

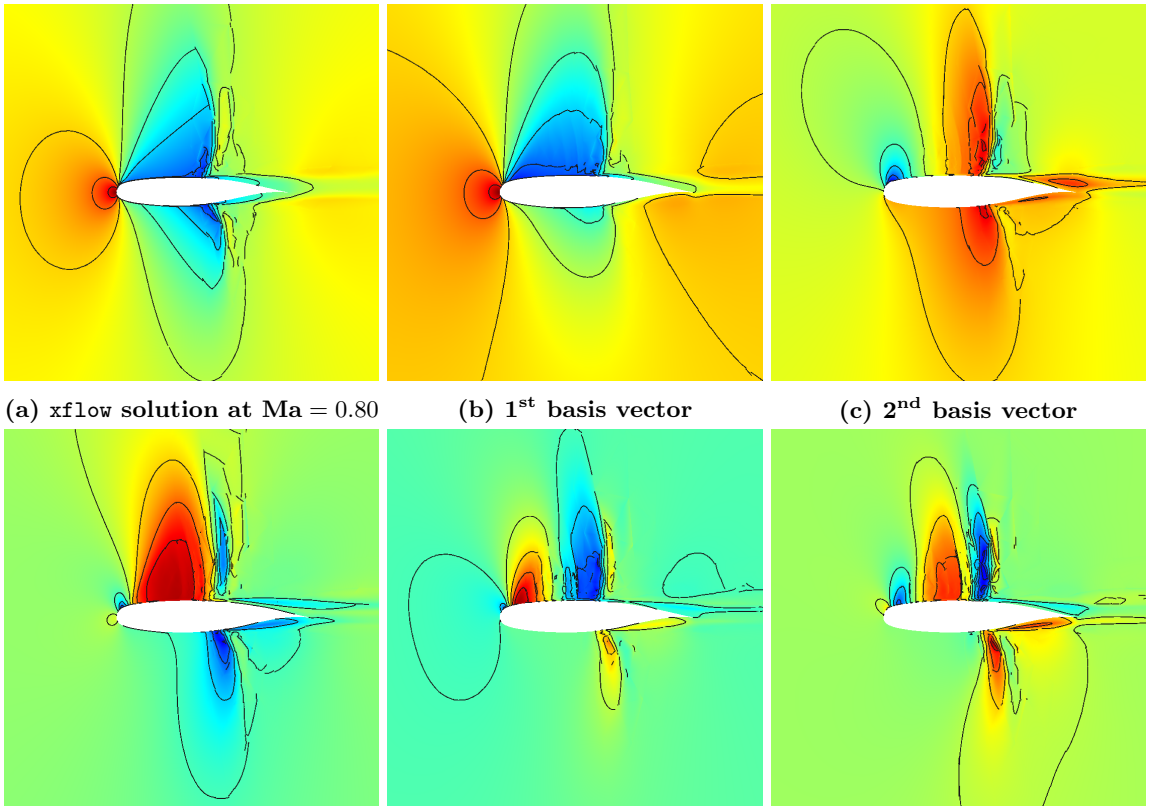
[24]Sirovich, L., "Method of snapshots," *Quarterly of applied mathematical*, Vol. 45, No. 3, 1987, pp. 561–571.

[25]Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[26]Fidkowski, K. J. and Roe, P. L., "An entropy adjoint approach to mesh refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.

[27]Ricardo, B.-Y. et al., *Modern information retrieval*, Pearson Education India, 1999.

A. `xflow` Solution and First 5 POD Basis Vectors

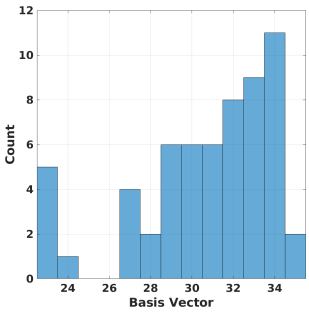


(a) `xflow` solution at Ma $= 0.80$ (b) 1^{st} basis vector (c) 2^{nd} basis vector

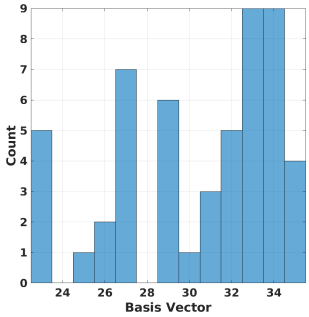
(d) 3^{rd} basis vector (e) 4^{th} basis vector (f) 5^{th} basis vector

Figure 11: Density contours for FOM at Ma $= 0.80$ and first 5 state basis vectors.

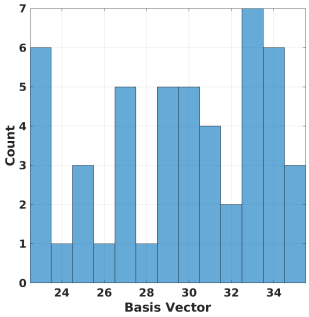
B. Histograms of Activated Basis Vectors



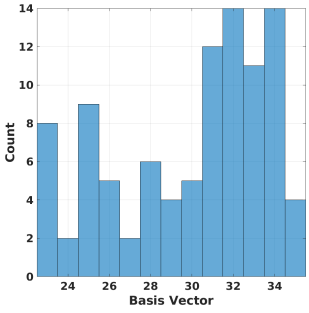
(a) 22+4L Adapted Basis Histogram



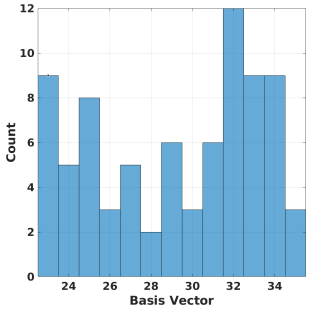
(b) 22P2L Adapted Basis Histogram



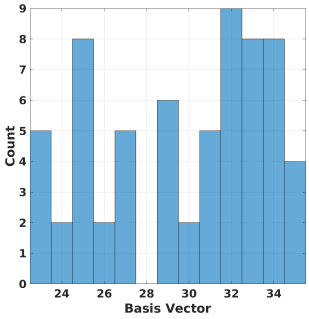
(c) 22P1L Adapted Basis Histogram



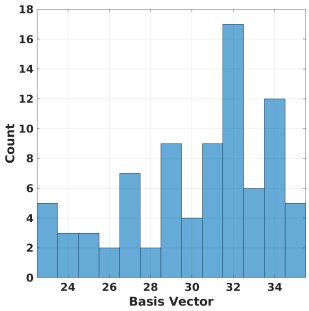
(d) 22+4D Adapted Basis Histogram



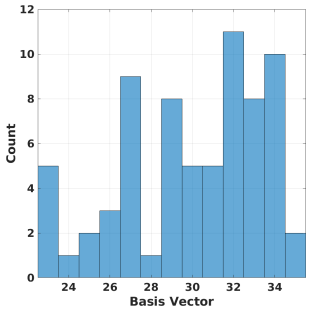
(e) 22P2D Adapted Basis Histogram



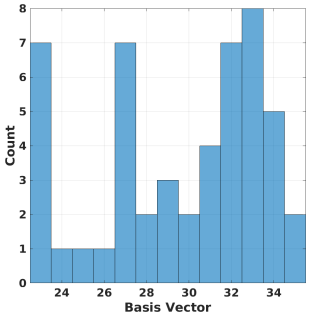
(f) 22P1D Adapted Basis Histogram



(g) 22+4M Adapted Basis Histogram



(h) 22P2M Adapted Basis Histogram



(i) 22P1M Adapted Basis Histogram

Figure 12: Activated basis vectors for adapted runs.