INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2011; **86**:155–181
Published online 28 October 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.3050

Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations

Kevin Carlberg[1,*,†], Charbel Bou-Mosleh[4] and Charbel Farhat[1,2,3]

[1]*Department of Aeronautics and Astronautics, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A.*
[2]*Department of Mechanical Engineering, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A.*
[3]*Institute for Computational and Mathematical Engineering, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A.*
[4]*Department of Mechanical Engineering, Notre Dame University, P.O. Box 72 Zouk Mikael, Louaize, Lebanon*

SUMMARY

A Petrov–Galerkin projection method is proposed for reducing the dimension of a discrete non-linear static or dynamic computational model in view of enabling its processing in real time. The right reduced-order basis is chosen to be invariant and is constructed using the Proper Orthogonal Decomposition method. The left reduced-order basis is selected to minimize the two-norm of the residual arising at each Newton iteration. Thus, this basis is iteration-dependent, enables capturing of non-linearities, and leads to the globally convergent Gauss–Newton method. To avoid the significant computational cost of assembling the reduced-order operators, the residual and action of the Jacobian on the right reduced-order basis are each approximated by the product of an invariant, large-scale matrix, and an iteration-dependent, smaller one. The invariant matrix is computed using a data compression procedure that meets proposed consistency requirements. The iteration-dependent matrix is computed to enable the least-squares reconstruction of some entries of the approximated quantities. The results obtained for the solution of a turbulent flow problem and several non-linear structural dynamics problems highlight the merit of the proposed consistency requirements. They also demonstrate the potential of this method to significantly reduce the computational cost associated with high-dimensional non-linear models while retaining their accuracy. Copyright © 2010 John Wiley & Sons, Ltd.

Received 11 May 2010; Revised 16 August 2010; Accepted 18 August 2010

KEY WORDS: non-linear model reduction; compressive approximation; discrete non-linear systems; gappy data; Petrov–Galerkin projection; proper orthogonal decomposition

1. INTRODUCTION

During the past two decades, giant strides have been made in many aspects of computational science and engineering. Higher fidelity mathematical models, better approximation methods, and faster solution algorithms have been developed for many applications. Computing speed barriers have also been shattered by hardware manufacturers. As a result, the potential of physics-based numerical simulations for providing a deeper understanding of complex engineering systems, enhancing their reliability, and, most importantly, reducing design-cycle time, is recognized today in almost every field of engineering. However, in many engineering applications, such numerical

*Correspondence to: Kevin Carlberg, Department of Aeronautics and Astronautics, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A.
†E-mail: carlberg@stanford.edu

simulations are so computationally intensive that they either cannot be performed as often as needed or are more often performed in special circumstances than routinely. For this reason, the recognized potential of physics-based numerical simulations on time-critical operations, such as design, design optimization, and active control has yet to materialize. Time-critical applications call instead for model reduction methods that replace large-scale computational models by smaller approximations of these models that are nevertheless capable of faithfully reproducing their essential features at a fraction of their computational cost.

The following key observation lies at the core of model reduction: although the state of a large-scale system is in general represented by an element of a large-dimensional vector space, it actually resides on a lower dimensional manifold induced by time evolution and input parameter variation. In order to approximately capture an associated subspace of interest, snapshots of the solution are first computed *offline* for a carefully chosen set of input parameters, then are compressed into a reduced-order basis by a procedure, such as the Proper Orthogonal Decomposition (POD) method [1, 2]. Next, the large-scale computational model of interest is projected onto the subspace spanned by this reduced-order basis. Finally, real-time numerical predictions are performed *online* by seeking approximate solutions to the problems of interest in the lower dimensional subspace spanned by the aforementioned reduced-order basis.

The computational cost of the projection-based model reduction technique outlined above scales with the large dimension of the original computational model. For this reason, this approach to model reduction is efficient primarily for those problems where the sought-after reduced operators need to be constructed only once. These include linear time-invariant dynamical systems with fixed parameter values [3], linear static systems governed by affinely parameterized elliptic partial differential equations (PDEs) [4], and systems with at most quadratic non-linearities [5–7]. Within these contexts, the projection-based model reduction technique has been successfully applied to problems in aerodynamics [8–12], aeroelasticity [13–17], control [18], structural dynamics [19–22], and non-destructive evaluation and parameter estimation [23], to name only a few.

Unfortunately, standard projection techniques are not generally effective for non-linear model reduction for two main reasons: they do not necessarily capture the non-linearities of the problem of interest, and their application to the reduction at each Newton iteration of the linearized computational model is CPU intensive because, as mentioned above, the associated computational cost scales with the dimension of the large-scale computational model. To overcome these roadblocks, several approaches have been recently proposed. Most of them rely on empirical techniques for capturing the non-linearities, and on evaluating the non-linear terms at a subset of the points where they are defined to reduce the computational cost.

The 'empirical interpolation' method developed for linear elliptic and coercive static problems with non-affine parameter dependence, and for non-linear elliptic and parabolic coercive problems [24], reduces the computational cost associated with the evaluation of non-linear functions and their Jacobians by interpolating their values at a few selected spatial locations using an empirically derived basis. A variant of this method using 'best [interpolation] points' and a POD basis can be found in [25]. Whereas both of these methods operate at the continuous level, related approaches have also been proposed at the semi-discrete level—that is, at the level of the ordinary differential equation (ODE) obtained after discretizing the PDE of interest in space. These include the model reduction approach developed in [26], which reconstructs the non-linear function in the least-squares sense using the same basis adopted for representing the solution, and semi-discrete analogs to the empirical and best points interpolation methods developed in [27, 28] for parameterized non-linear static problems, and in [27] for non-linear dynamic problems.

Alternatively, this paper presents a model reduction method for problems in non-linear dynamics that differs from the approaches outlined above in several ways. Unlike the best points interpolation method presented in [25], which operates at the continuous level, and that proposed in [27], which operates at the semi-discrete level, the proposed non-linear model reduction method operates at the fully discrete level—that is, at the level of the algebraic system obtained after discretizing the target PDE in both space and time—independently of the type of the non-linear problem

DOI: 10.1002/nme

of interest and its discretization. Furthermore, whereas the method presented in [27] relies on a Galerkin projection technique that is not in general optimal, the proposed method is based on a Petrov–Galerkin projection where the right reduced-order basis is constructed using a one-time snapshot collection procedure, and the left reduced-order basis is chosen to minimize in a least-squares sense the residual of the linearized system of equations arising at each Newton iteration. To avoid the significant computational cost associated with the projection of the updated Jacobian and non-linear residual at each Newton iteration, a compressive tensor approximation procedure is considered. However, instead of interpolating the non-linear residual and action of the Jacobian on the right reduced-order basis, these are approximated in the proposed method by an optimal least-squares approach [29] that is shown numerically to offer higher accuracy in many cases. Overall, the non-linear model reduction method proposed in this paper is designed around consistency requirements that ensure that its approximations do not introduce any new error in a certain limit, and around optimality requirements that guarantee convergence by monotonically decreasing an error measure as the approximation space is expanded. These requirements dictate the construction of separate reduced-order bases for the non-linear functions and their Jacobians that furthermore are different from those used for approximating the sought-after solution; this further distinguishes the proposed method from the works published in [24, 25, 27, 28, 30]. It is noted here that while the non-linear reduction method proposed herein is presented in the context of static and implicit dynamic computations, its extension to explicit transient computations is straightforward.

The remainder of this paper is organized as follows. Section 2 formulates the problem of interest and outlines the rationale behind the approach chosen here for developing an efficient non-linear model reduction method. Section 3 develops optimal projectors and a consistent snapshot collection procedure for reducing the order of the non-linear problem and representing the state of the non-linear system of interest. It also illustrates these with the solution of a turbulent flow problem. Section 4 introduces a hierarchy of snapshot collection procedures for the fast reconstruction of non-linear function bases that satisfy certain consistency conditions. Section 5 discusses in detail some aspects of the overall non-linear model reduction method and presents a novel greedy algorithm for index selection. Section 6 illustrates the proposed non-linear reduction method with the solution of a set of geometrically non-linear structural dynamics problems, highlights the merit of the consistency requirements adopted for its design, and demonstrates its computational efficiency. Finally, Section 7 concludes the paper.

2. PROBLEM FORMULATION

2.1. Parameterized non-linear systems

Consider first a parameterized non-linear static computational model described by the large-scale system of discrete equations

$$R(y;\mu)=0 \tag{1}$$

and the outputs

$$\begin{aligned} z &= G(y,\mu) \\ &= L(\mu). \end{aligned} \tag{2}$$

Here, $y\in\mathbb{R}^N$ is the state implicitly defined by Equation (1), $\mu\in\mathscr{D}\subset\mathbb{R}^d$ is the vector of input *configuration parameters* (such as shape parameters, material properties, boundary conditions, etc.), $\mathscr{D}$ is the input parameter domain, and $z\in\mathbb{R}^p$ represents the outputs that are of primary interest to the analyst. The function $R:\mathbb{R}^N\times\mathscr{D}\to\mathbb{R}^N$ is non-linear in at least its first argument, $G:\mathbb{R}^N\times\mathscr{D}\to\mathbb{R}^p$ and $L:\mathscr{D}\to\mathbb{R}^p$. The second of Equation (2) follows from the dependence of y on μ.

DOI: 10.1002/nme

Next, consider an ODE written in state–space form that results from the semi-discretization of a PDE governing a dynamic problem

$$\begin{aligned} E\frac{\mathrm{d}v}{\mathrm{d}t}(t) &= F(v(t), t; \mu) \\ v(0) &= v_0, \end{aligned} \tag{3}$$

with outputs of interest

$$z = H(v(t), \mu). \tag{4}$$

Here, $t \in \mathbb{R}^+$ denotes time, $v(t) \in \mathbb{R}^{N^\star}$ and $z \in \mathbb{R}^p$ are the state and outputs, respectively, $F : \mathbb{R}^{N^\star} \times \mathbb{R}^+ \times \mathscr{D} \to \mathbb{R}^{N^\star}$ is non-linear in at least its first argument, $E \in \mathbb{R}^{N^\star \times N^\star}$ and $H : \mathbb{R}^{N^\star} \times \mathscr{D} \to \mathbb{R}^p$.

When Equation (3) is solved by an implicit time-integrator, a sequence of non-linear problems arises. Assuming for simplicity a uniform time-step Δt, each of these non-linear problems can be written as

$$R(y^{n+1}; y^n, \ldots, y^0; t^n; \mu) = 0, \tag{5}$$

with outputs

$$\begin{aligned} z &= G(y^0, \ldots, y^n, \ldots, y^{n_\mathrm{t}}, \mu) \\ &= L(\mu). \end{aligned} \tag{6}$$

Here, the superscript n designates the value of a variable at the time-instance $n\Delta t$, $y^n \in \mathbb{R}^N$ with $0 \leqslant n \leqslant n_\mathrm{t}$ are intermediate variables, $N \leqslant N^\star$[‡] and $z \in \mathbb{R}^p$. The operator $R : \mathbb{R}^N \times \cdots \times \mathbb{R}^N \times \mathbb{R}^+ \times \mathscr{D} \to \mathbb{R}^N$ is non-linear in at least its first argument and $G : \mathbb{R}^N \times \cdots \times \mathbb{R}^N \times \mathscr{D} \to \mathbb{R}^p$. The second of Equation (6) follows from the dependence of y^n on μ. The intermediate variables y^n are implicitly defined by Equation (5) given t^n, μ, and y^j, $0 \leqslant j \leqslant n-1$, and are used to explicitly update the corresponding full state v^n. For simplicity, the intermediate variables will hereafter be referred to as the state.

2.2. *Offline–online predictions*

Consider the following objective: given inputs $\mu^\star \in \mathscr{D}$, compute approximations to the outputs $\tilde{L}(\mu^\star) \approx L(\mu^\star)$ in a sufficiently small amount of time. When the number of dofs N of the high-dimensional model is sufficiently large, solving Equation (1) or Equation (5) with $\mu = \mu^\star$ and computing the corresponding outputs becomes a major obstacle for the stated objective. Instead, the following two-stage 'offline-online' strategy can be employed. In the offline stage, Equations (1)–(2) or Equations (5)–(6) are solved for $n_s \geqslant 1$ 'training' vectors of input configuration parameters $\mathscr{D}_s = \{\mu^j\}_{j=1}^{n_s} \subset \mathscr{D}$, and results (the data) are acquired. Then, these data are used to construct one or more surrogate models for the original parameterized systems that are capable, with or without adaptation, of reproducing in real-time the behavior of the high-dimensional model at arbitrary points in $\mathscr{D}$. In the online stage, such surrogate model(s) is (are) exploited for real-time prediction.

Two surrogate modeling strategies have been primarily adopted in this context: data fitting and model reduction. Data fitting methods, such as response surfaces, Kriging models, and other supervised learning approaches, bypass the computation of the state entirely and instead directly model the input–output map $z = L(\mu)$ as

$$\tilde{z} = \tilde{L}(\mu), \tag{7}$$

where $\tilde{z} \in \mathbb{R}^p$ are the predicted outputs and $\tilde{L} : \mathscr{D} \to \mathbb{R}^p$.[§] However, since the prediction $\tilde{L}(\mu)$ bypasses the state equations, it is 'blind' to the underlying physics of the problem. This can lead

[‡] N is typically the number of degrees of freedom (dofs) of the system of interest and $N^\star$, which is the dimension of the state space, is often a multiple of N.

[§] Statistical quantities carrying information about the uncertainty associated with the prediction $\tilde{L}(\cdot)$ are also often modeled.

DOI: 10.1002/nme

to significant errors when the problem's physics are strongly input-dependent. Furthermore, data fits tend to be impractical for large-dimensional input spaces (large values of d) [31].

On the other hand, model reduction seeks to achieve real-time prediction by *approximately* solving the state equations for the online configuration $\mu^\star$ and then computing the resulting outputs. Since the state equations contain information about the physics at the input of interest $\mu^\star$, this approach can be expected to generate predictions that are more robust than data fits with respect to input space dimension and rapidly changing physics. This approach is also more amenable to 'what-if' experimentation, as arbitrary excitations and boundary conditions can be applied to the approximated system.

2.3. Single vs multiple and sampled vs unsampled configurations

The comparison of Equations (1)–(2) with Equations (5)–(6) shows that for the purpose of data acquisition in view of model reduction, static problems and dynamic problems discretized by an implicit time-integration scheme have the same structure and therefore can be addressed by the same computational framework. Namely, given a set of sample input configurations $\mathscr{D}_s$, the sampled problems can be simply treated as a general set of n_p systems of non-linear equations of the form

$$R^i(y^i)=0, \quad i=1,\ldots,n_\mathrm{p}, \tag{8}$$

with $n_\mathrm{p}=n_s$ in the static case, and $n_\mathrm{p}=n_s n_\mathrm{t}$ in the dynamic case. The associated outputs of interest can then be written as

$$\mathsf{z}=\mathsf{G}(y^1,\ldots,y^{n_\mathrm{p}}), \tag{9}$$

where $\mathsf{z}\in\mathbb{R}^{pn_s}$ and $\mathsf{G}:\mathbb{R}^N\times\cdots\times\mathbb{R}^N\to\mathbb{R}^{pn_s}$ may contain outputs computed for multiple configurations.

Hence, potential configuration changes are embedded within the formulation (8)–(9), which is equally applicable to static and dynamic problems. However, there is a significant difference between the static and dynamic cases. In the static case, the systems of non-linear equations (8) are usually associated with different input configuration parameters (including different loading and/or boundary conditions). In the dynamic case, they are not necessarily associated with different problem configurations: they can be associated simply with different time-instances, or a combination of different configurations and time-instances. Consequently, generating a sufficient amount of data for model reduction often requires considering multiple configurations in the static case, whereas a single configuration often suffices for this purpose in the dynamic case.

In this setting, the non-linear model reduction method proposed in this paper is developed and assessed for dynamic problems using a single set of input configuration parameters ($n_s=1$) for data acquisition. The validity or performance for unsampled configurations of the non-linear reduced-order model (ROM) obtained by the proposed approach is beyond the scope of this paper for three reasons: (1) to be able to compute accurate, real-time solutions for unsampled configurations ($\mu^\star\notin\mathscr{D}_s$), a model reduction method should be shown first to be able to compute real-time and accurate solutions for configurations that have already been sampled ($\mu^\star\in\mathscr{D}_s$), (2) there are several competing approaches for addressing the problem of unsampled configurations, which range from global approximation approaches based on data acquisition for multiple configurations, to interpolation of reduced-order information on suitable manifolds (for example, see [17, 22]), and (3) among these approaches, the most suitable one for adapting the proposed non-linear model reduction method to unsampled configurations can be expected to be problem- and parameter-dependent; therefore, its identification will be the subject of future work.

In summary, the remainder of this paper focuses on the systems of non-linear equations (8) associated with different time-instances of a high-fidelity (or high-dimensional) dynamic computational model built at a single configuration, with Equation (9) defining the corresponding outputs of interest. The objective of the non-linear model reduction method presented in this paper is to compute real-time (online), yet accurate solutions of these systems of equations after they have been sampled offline.

DOI: 10.1002/nme

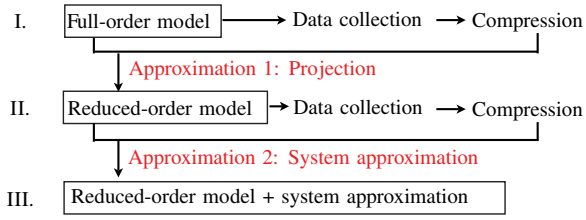

Figure 1. Model hierarchy.

2.4. *Consistency-driven approach for non-linear model reduction*

Model reduction of non-linear systems is often executed in a somewhat *ad hoc* manner, where approximations are constructed using intuition and past experience without much reference to properties that a 'good' approximation should satisfy. To avoid this pitfall, this work adopts a strategy that enables approximations to be carefully constructed to meet the desired conditions. In the proposed approach, if a given model is deemed too computationally expensive for real-time evaluation, an additional approximation is introduced, resulting in another less accurate but more economical model. This results in a hierarchy of models characterized by tradeoffs between accuracy and computational efficiency. The approximations, which are introduced consecutively, are constructed to generate minimal error with respect to the previous model by satisfying *optimality* and *consistency* properties that are defined more precisely below.

As shown in Figure 1, the model hierarchy employed in this work consists of three computational models: an original model and two increasingly 'lighter' approximated versions. Each approximated model is generated by acquiring data during the evaluation of the more accurate model for sample inputs, then compressing the data, and finally introducing the approximation that exploits the compressed data.

The high-dimensional model will be referred to as Model I and is taken to be the 'truth'. When evaluating this model is too computationally intensive for real-time prediction, a projection approximation (Approximation 1) is introduced to reduce the dimensionality of the state equations. This leads to the ROM or Model II. If this ROM is still too CPU intensive for online computations, a system approximation (Approximation 2) is introduced to reduce the computational complexity of its processing. The result of the application of this system approximation to Model II can be interpreted as a computational model and therefore will be referred to as Model III in the remainder of this paper.

As previously stated, the approximations should introduce minimal error with respect to the previous model in the hierarchy. To this end, Approximations 1 and 2 will be constructed to be: (1) consistent and (2) optimal in the sense defined below.

Consistent approximation: An approximation is said here to be consistent if, when implemented without data compression, it introduces no additional error in the solution of the same problem for which data were acquired.

Optimal approximation: An approximation is said here to be optimal if it leads to approximated quantities that minimize some error measure with respect to the previous model in the hierarchy.

Here, consistency implies that the approximation is exact for the same problem for which data were acquired in the limit of no truncation. On the other hand, optimality ensures that some measure of the error monotonically decreases as the approximation spaces expand, leading to guaranteed *a priori* convergence. Lack of optimality can lead to the undesirable property of deterioration in solution quality as the approximation spaces expand, as has been observed in [32], for example. It will be shown in this paper that ensuring the above properties formalizes the construction of approximations.

DOI: 10.1002/nme

3. PETROV–GALERKIN PROJECTION

To simplify the presentation, this section focuses on one representative instance of problems (8) written as

$$R(y)=0, \tag{10}$$

where the state $y\in\mathbb{R}^N$ is implicitly defined by Equation (10) and the mapping $R:\mathbb{R}^N\to\mathbb{R}^N$, $w\to R(w)$ is non-linear. Furthermore, Equation (10) is considered in the following sections to be the system of full state equations of a representative non-linear high-dimensional model—that is, Model I.

However, it should not escape the reader that when in the remainder of this paper reference is made to snapshots that are collected, these snapshots are computed for several instances of Equation (10) that correspond in this work to different time-instances.

3.1. Dimension reduction via optimal projection

To reduce the dimension of Equation (10), a projection process is employed. Specifically, an approximate solution $\tilde{y}$ is sought in the affine search subspace $y^{(0)}+\mathscr{Y}$ of dimension $n_y\ll N$

$$\tilde{y}=y^{(0)}+\Phi_y y_r, \tag{11}$$

where $y^{(0)}\in\mathbb{R}^N$ is an initial guess for the solution, $\Phi_y\in\mathbb{R}^{N\times n_y}$ is a basis for the subspace $\mathscr{Y}\subset\mathbb{R}^N$, and $y_r\in\mathbb{R}^{n_y}$ are generalized coordinates for the state.

Substituting Equation (11) into Equation (10) yields an overdetermined system of N non-linear equations in n_y unknowns. Therefore, n_y constraints are introduced by enforcing the orthogonality of the non-linear residual to a left subspace $\mathscr{L}\subset\mathbb{R}^N$. This results in the reduced system of n_y state equations with n_y unknowns

$$\Psi^{\mathrm{T}}R(y^{(0)}+\Phi_y y_r)=0, \tag{12}$$

where $\Psi\in\mathbb{R}^{N\times n_y}$ provides a basis for $\mathscr{L}$ and the superscript 'T' denotes the transpose operation.

Applying Newton's method to solve the reduced non-linear problem (12) results in the following iterations: for $k=1,\ldots,K$, solve

$$\Psi^{\mathrm{T}}J^{(k)}\Phi_y p^{(k)}=-\Psi^{\mathrm{T}}R^{(k)} \tag{13}$$

$$y_r^{(k+1)}=y_r^{(k)}+\alpha^{(k)}p^{(k)}, \tag{14}$$

where K is determined by the satisfaction of a convergence criterion, $y_r^{(0)}=0$, and $R^{(k)}\equiv R(y^{(0)}+\Phi_y y_r^{(k)})$ and $J^{(k)}\equiv(\partial R/\partial w)(y^{(0)}+\Phi_y y_r^{(k)})$ are the non-linear residual and Jacobian at iteration k, respectively. The step length $\alpha^{(k)}$ is computed by executing a line search in the direction $p^{(k)}$ or is set to the canonical step length of unity. Equations (13)–(14) will be referred to as the reduced Newton iterations.

3.2. Consistent search subspace and POD

POD [1] is a popular data compression method that is often used in model reduction to compute the subspace $\mathscr{Y}$ for the approximate solution. Its widespread use is due to its optimal compression property: it minimizes the sum of squared distances to vector-valued 'snapshots' that constitute the data of interest. For this reason, POD is used in this work for all data compressions (see Figure 1), including the computation of $\mathscr{Y}$.

Formally, the POD subspace of dimension $n\leqslant n_x$ is defined as

$$\mathscr{P}(n,X)\equiv\arg\min_{\mathscr{A}\in\mathscr{G}(n,N)}\sum_{i=1}^{n_x}\|(I-P_{\mathscr{A}})x^i\|_2^2, \tag{15}$$

DOI: 10.1002/nme

where n_x denotes the number of pre-computed snapshots $x^i \in \mathbb{R}^N$, $1 \leqslant i \leqslant n_x$—usually $n_x \ll N$, $X \equiv [x^1 \ \dots \ x^{n_x}]$ denotes the snapshot matrix, $\mathcal{G}(\alpha, N)$ is the set of α-dimensional linear subspaces of $\mathbb{R}^N$ (the Grassmann manifold [17, 33]), $\|\cdot\|_2$ is the Euclidean norm, and $P_{\mathscr{A}}$ is the orthogonal projector onto the subspace $\mathscr{A}$ and is defined as

$$P_{\mathscr{A}} u \in \mathscr{A}, \quad ((I - P_{\mathscr{A}})u, v) = 0 \quad \forall v \in \mathscr{A}, \tag{16}$$

where $(\cdot, \cdot)$ is the Euclidean inner product. The POD basis $\Phi(n, X) \equiv [\phi^1 \ \dots \ \phi^n] \in \mathbb{R}^{N \times n}$ for the subspace $\mathscr{P}(n, X)$ can be computed using the singular value decomposition (SVD)[¶] as described in Algorithm 1.

Algorithm 1 POD basis computation.

Input: Snapshot matrix $X \in \mathbb{R}^{N \times n_x}$
Output: $\Phi(n, X) \in \mathbb{R}^{N \times n}$
1: Compute the thin SVD: $X = U \Sigma V^{\mathrm{T}}$
2: Choose the basis dimension $n \in \{1, 2, \dots, n_x\}$
3: Compress the data by truncating the basis $\Phi(n, X) = [u^1 \ \dots \ u^n]$, where $U \equiv [u^1 \ \dots \ u^{n_x}]$

Note that the POD basis vectors are optimally ordered in the following sense:

$$\operatorname{span}\{\phi^i\}_{i=1}^n = \mathscr{P}(n, X), \quad 1 \leqslant n \leqslant n_x, \tag{17}$$

so that $\mathscr{P}(n, X) \subset \mathscr{P}(n+1, X)$ for $1 \leqslant n \leqslant n_x - 1$. Also, since $\Phi(n_x, X)$ spans the range of X, the POD subspace spans the entire set of snapshots when the POD basis is not truncated.

The singular values $\Sigma \equiv \operatorname{diag}(\sigma_1, \dots, \sigma_{n_x})$ computed in Step 1 can be used to compute the cost function

$$\sum_{i=1}^{n_x} \|(I - P_{\mathscr{P}(n,X)})x^i\|_2^2 = \sum_{k=n+1}^{n_x} \sigma_k^2 \tag{18}$$

which is minimized by POD. This function is often interpreted as the 'energy' of the snapshots omitted by the POD basis. The dimension of the POD basis n is thus often chosen in Step 2 of Algorithm 1 so that the relative energy retained by the basis is greater than or equal to some threshold $\upsilon \in [0, 1]$—that is, $n \leftarrow n_{\mathscr{E}}(\Sigma, \upsilon)$, where

$$\begin{aligned} n_{\mathscr{E}}(\Sigma, \upsilon) &\equiv \min_{n \in \mathscr{V}(\Sigma, \upsilon)} n, \\ \mathscr{V}(\Sigma, \upsilon) &\equiv \left\{ n \in \{1, 2, \dots, n_x\} \,\middle|\, \sum_{i=1}^{n} \sigma_i^2 \Big/ \sum_{j=1}^{n_x} \sigma_j^2 \geqslant \upsilon \right\}. \end{aligned} \tag{19}$$

Proposition 1
If $(y - y^{(0)})$ is one of the collected snapshots used to compute the POD basis Φ_y and $y^{(0)}$ is sufficiently close to y, the projection approximation outlined in Section 3.1 is consistent.

Indeed, in this case, when $n_y = n_x$—that is, when Φ_y is not truncated—Φ_y will contain the solution $(y - y^{(0)})$ of Equation (10) (Model I), and under the usual assumptions, Newton's method applied to Equation (12) (Model II) will converge to this solution.

Because of Proposition 1 above, it is chosen here to compute Φ_y by the POD method using snapshots $(y - y^{(0)})$ obtained by solving Equation (10) for different time-instances. Note that other consistent snapshot collection procedures exist for non-linear dynamical systems; this constitutes an interesting subject for future investigation.

[¶]When other norms are used to compute the POD basis, the symmetric eigenvalue decomposition is often a more computationally efficient choice.

DOI: 10.1002/nme

3.3. Optimal left subspaces

To satisfy the optimality requirement of the projection approximation, the left subspace $\mathscr{L}$ is selected here to minimize in some norm Θ the difference between the solution based on reduced state equations (Model II), and that of the full state equations (Model I). As Newton's method is employed to solve the reduced state equations, the left subspace is chosen here to minimize at each iteration the error in the search direction—that is, to obtain a search vector $\Phi_y p^{(k)}$ with $p^{(k)}$ verifying

$$p^{(k)} = \arg \min_{p \in \mathbb{R}^{n_y}} \| \Phi_y p - {J^{(k)}}^{-1} R^{(k)} \|_\Theta, \tag{20}$$

where Θ is a symmetric positive definite (SPD) matrix and $\|x\|_\Theta \equiv \sqrt{x^{\mathrm{T}} \Theta x}$. Note that when (20) is satisfied, the error measure $\| \Phi_y p^{(k)} - {J^{(k)}}^{-1} R^{(k)} \|_\Theta$ decreases monotonically as vectors are added to the POD basis. This occurs because $\mathrm{span}\{\phi_y^i\}_{i=1}^{n_y} \subset \mathrm{span}\{\phi_y^i\}_{i=1}^{n_y+1}$; thus, for a given Newton system with $J^{(k)}$ and $R^{(k)}$ held fixed, the minimization in (20) is done over a superset of the previous space as the POD basis expands. This causes the error measure in (20) to decrease monotonically.

3.3.1. Galerkin projection. When the Jacobians are SPD matrices, the Galerkin projection defined by $\mathscr{L} = \mathscr{Y}$ leads to search directions $p^{(k)}$ that are optimal in the sense defined in (20) with $\Theta = J^{(k)}$ (for example, see [34], p. 135 for a proof). In this case, the reduced Newton iterations (13)–(14) become

$$\Phi_y^{\mathrm{T}} J^{(k)} \Phi_y p^{(k)} = -\Phi_y^{\mathrm{T}} R^{(k)} \tag{21}$$

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} p^{(k)}. \tag{22}$$

Unfortunately, the Jacobians of a non-linear problem are not in general SPD matrices, and therefore the Galerkin projection is not always optimal in the sense of Equation (20). For this reason, a different projection is proposed next for general non-linear problems.

3.3.2. Least-squares Petrov–Galerkin projection. Petrov–Galerkin projection with $\mathscr{L} = J^{(k)} \mathscr{Y}$ produces the reduced Newton iterations

$$\Phi_y^{\mathrm{T}} J^{(k)T} J^{(k)} \Phi_y p^{(k)} = -\Phi_y^T J^{(k)T} R^{(k)}, \tag{23}$$

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} p^{(k)}. \tag{24}$$

In this case, the computed search directions $p^{(k)}$ are optimal in the sense of (20) for $\Theta = J^{(k)T} J^{(k)}$. This is because the projection scheme underlying Equations (23)–(24) is the normal equation form of the least-squares problem

$$p^{(k)} = \arg \min_{a \in \mathbb{R}^{n_y}} \| J^{(k)} \Phi_y a + R^{(k)} \|_2, \tag{25}$$

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} p^{(k)}, \tag{26}$$

and

$$\| J^{(k)} \Phi_y a + R^{(k)} \|_2 = \| \Phi_y a - {J^{(k)}}^{-1} R^{(k)} \|_{J^{(k)T} J^{(k)}} \tag{27}$$

Note that the chosen left subspace $\mathscr{L} = J^{(k)} \mathscr{Y}$ changes from one Newton iteration to another. Therefore, it can capture in principle the non-linear effects but is practical primarily when the non-linear model reduction is performed at the fully discrete level.

The numerical conditioning of the above least-squares problem can be improved by using the thin QR [35] factorization $J^{(k)} \Phi_y = Q_{J^{(k)} \Phi_y} R_{J^{(k)} \Phi_y}$, where $Q_{J^{(k)} \Phi_y} \in \mathbb{R}^{N \times n_y}$ and $R_{J^{(k)} \Phi_y} \in \mathbb{R}^{n_y \times n_y}$. This transforms Equations (23)–(24) into their equivalent form

$$R_{J^{(k)} \Phi_y} p^{(k)} = -Q_{J^{(k)} \Phi_y}^{\mathrm{T}} R^{(k)}, \tag{28}$$

DOI: 10.1002/nme

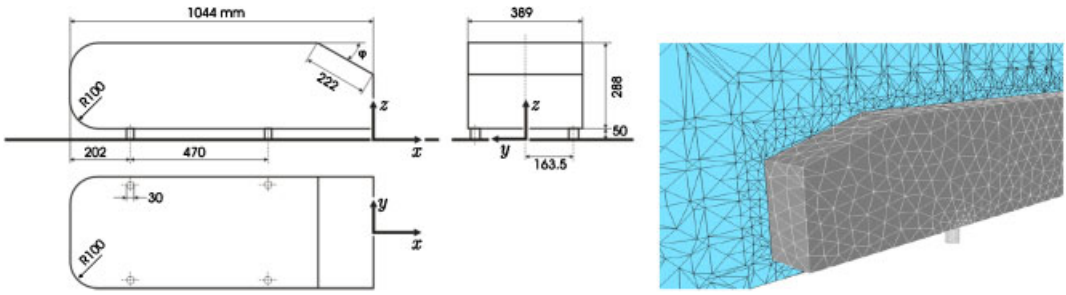

Figure 2. Ahmed body geometry (from [40]) and corresponding CFD mesh for $\varphi = 20^\circ$.

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} p^{(k)}. \tag{29}$$

When the scalars $\alpha^{(k)}$, $1 \leqslant k \leqslant K$ satisfy the Wolfe conditions (see [36], Equations (3.6)), Equations (28)–(29) constitute the globally convergent Gauss–Newton method [36] for solving the non-linear least-squares problem

$$\underset{\bar{y} \in y^{(0)} + \mathscr{Y}}{\text{minimize}} \, \| R(\bar{y}) \|_2 \tag{30}$$

that arises from choosing the search subspace[‖]. This highlights further the optimality of the choice $\mathscr{L} = J^{(k)}\mathscr{Y}$ as it generates Model II solutions that minimize the Model I non-linear residual over the entire search subspace.

Because $\mathscr{L} = J^{(k)}\mathscr{Y}$ leads at each Newton iteration to the least-squares problem (25), this projection method will be referred to as the least-squares Petrov–Galerkin (LSPG) projection. This type of projection was also recently used in [38] but in the context of linear model reduction.

3.4. Numerical comparison of left subspaces

To demonstrate the practical significance of employing in the practice of employing the LSPG left subspace $\mathscr{L} = J^{(k)}\mathscr{Y}$ as opposed to the Galerkin left subspace $\mathscr{L} = \mathscr{Y}$ for non-linear problems with non-SPD Jacobians, consider the problem of computing the unsteady airflow past the so-called Ahmed body shown in Figure 2. This problem, which is often used as a benchmark for turbulent flows by the automotive industry, was initially investigated by Ahmed *et al.* [39] in the 1980s. Here, the slant angle is set to $\varphi = 20^\circ$ and the free-stream velocity to $V_\infty = 60\,\text{m/s}$, which corresponds to a Reynolds number of $Re = 4.29 \times 10^6$. A relatively coarse mesh is chosen for the numerical solution of this problem since the purpose here is only to compare the performance of two different non-linear model reduction methods. This mesh is depicted in Figure 2. It consists of 146 517 nodes and 837 894 tetrahedra and results in $N = 879\,102$ dofs. Several Navier–Stokes computations are performed using the DES turbulence model and Reichardt's wall law. First, the turbulent flow is computed using the high-dimensional model (Model I) for a time-interval of 0.25 s. During this computation, n_t snapshots $x^i = y_{\text{I}}^n$, $1 \leqslant i \leqslant n_t$, are collected. Using these snapshots, a POD basis of dimension $n_y = 400 \ll N$ is constructed. Then, the reduced system of state equations (Model II)—whose dimension is only 0.046% of the dimension of the system of equations arising from the high-dimensional model—are solved using both the Galerkin (21)–(22) and LSPG (23)–(24) projection methods and a fixed unit step length. In each case, convergence of the iterative Newton process is declared when $\| R(y) \|_2 \leqslant 10^{-3} \| R(y^{(0)}) \|_2$, and the reduced Newton equations are solved using the LU factorization method. As for this problem the Jacobians are not SPD, the LSPG projection method is expected to generate more accurate results than the Galerkin projection method. Since the Jacobians are not SPD for this problem.

[‖]For SPD Jacobians, Galerkin projection search directions satisfy $p^{(k)} = \arg\min_{a \in \mathbb{R}^{n_y}} \| J^{(k)} \Phi_y a + R^{(k)} \|_{J^{(k)-1}}$ at each iteration (Lemma 2.1.1 of [37]). For $\alpha^{(k)}$ satisfying the Wolfe conditions and $[\partial^2 R / (\partial w_i \partial w_j)]$ negligible, Galerkin projection is the globally convergent Gauss–Newton method for minimizing $\| (\partial R / \partial w)^{-1/2} (\bar{y}) R(\bar{y}) \|_2$ for $\bar{y} \in y^{(0)} + \mathscr{Y}$.

DOI: 10.1002/nme

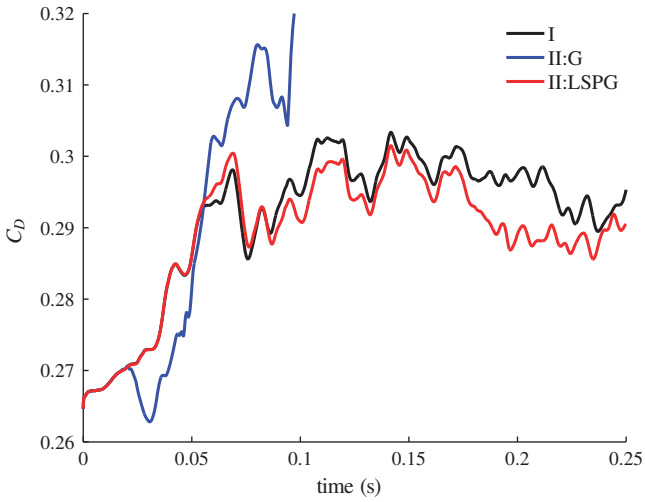

Figure 3. Predicted time histories of the drag coefficient using the high-fidelity Model I (I) and Model II based on the Galerkin (II:G) and LSPG (II:LSPG) projections.

Figure 3 reports the obtained time-histories of the drag coefficient C_D for Model I and both variants of the reduced-order Model II. The reader can observe that the non-linear ROM based on the Galerkin projection quickly ceases to be accurate and eventually becomes numerically unstable. On the other hand, the non-linear ROM based on the LSPG projection is shown to reproduce the results of the higher dimensional computational model with a relative root mean square error

$$\sqrt{\sum_{n=1}^{n_t}(C_{D,\mathrm{I}}^n - C_{D,\mathrm{II}}^n)^2} \Big/ \sqrt{\sum_{n=1}^{n_t}(C_{D,\mathrm{I}}^n)^2} = 1.31 \times 10^{-2} \tag{31}$$

that is, accurately. These observations confirm the stated expectations.

3.5. *Computational complexity*

As demonstrated in the previous section for the computation of the turbulent flow past the Ahmed body, the reduced equations (12) can be of much smaller dimension $n_y \ll N$ than the full state equations (10) and still deliver accurate results. However, they remain computationally expensive to form.

For example, consider the reduced Newton iterations (13)–(14) for Model II. To perform these iterations, the product $\Psi^{\mathrm{T}} J^{(k)} \Phi_y$ must be repeatedly computed. This product alone requires $2N\omega n_y + 2Nn_y^2$ floating-point operations (flops), where ω denotes the average number of non-zero entries in a row of $J^{(k)}$. Hence, this computational cost scales with the dimension of the large-scale system of state equations, N. For this reason, a non-linear model reduction method based on a projection technique rarely delivers the expected cost savings, unless it is equipped with a further approximation. This issue is addressed next in Section 4, which presents a novel system approximation method that decreases the computational cost associated with forming the reduced Newton equations (13)–(14) and leads to Model III in Figure 1.

4. SYSTEM APPROXIMATION VIA COMPRESSIVE TENSOR APPROXIMATIONS

4.1. *Approximation of the reduced state equations*

The large computational cost associated with the reduced Newton iterations (25) can be mitigated by approximating Equation (25). In the spirit of related previous works [24, 25, 27, 28, 30], such

DOI: 10.1002/nme

a system approximation is carried out here by constructing surrogates $\tilde{R}^{(k)}$ and $\widetilde{J^{(k)}\Phi_y}$ for the non-linear residual $R^{(k)}$ and column-reduced Jacobian $J^{(k)}\Phi_y$, respectively, using a data fitting approach that employs their values at a few dofs (or indices) and a set of pre-determined shape functions. However, because of the consistency requirement introduced in Section 2.4 and further developed for this specific case in Section 4.2, the surrogates $\tilde{R}^{(k)}$ and $\widetilde{J^{(k)}\Phi_y}$ proposed in this paper are constrained to lie in the ranges of *different* POD bases $\Phi_R \equiv [\phi_R^1 \ \dots \ \phi_R^{n_R}] \in \mathbb{R}^{N\times n_R}$ and $\Phi_J \equiv [\phi_J^1 \ \dots \ \phi_J^{n_J}] \in \mathbb{R}^{N\times n_J}$, respectively. Furthermore, $\widetilde{J^{(k)}\Phi_y}$ is treated as a collection of columns, hence its columns are computed independently. This strategy differs from those advocated in [24, 25, 27, 28, 30].

Hence, the system approximation performed here can be written as the tensor approximations

$$\tilde{R}^{(k)} = \Phi_R R_r^{(k)}, \tag{32}$$

$$\widetilde{J^{(k)}\Phi_y} = \Phi_J J_r^{(k)}, \tag{33}$$

where $R_r^{(k)} \in \mathbb{R}^{n_R}$ and $J_r^{(k)} \in \mathbb{R}^{n_J\times n_y}$ are generalized coordinates for the non-linear residual and column-reduced Jacobian, respectively. The approach for computing $R_r^{(k)}$ and $J_r^{(k)}$ is discussed in Section 4.2. Here, it is emphasized that the approximations (32)–(33) correspond to Approximation 2 in Figure 1 and are applied to the reduced Newton iterations (25) associated with the LSPG projection method. As a result, these reduced equations are transformed into their approximate form

$$\tilde{p}^{(k)} = \arg\min_{a\in\mathbb{R}^{n_y}} \| \widetilde{J^{(k)}\Phi_y} a + \tilde{R}^{(k)} \|_2, \tag{34}$$

$$y_r^{(k+1)} = y_r^{(k)} + \alpha^{(k)} \tilde{p}^{(k)}, \tag{35}$$

which defines the non-linear ROM with system approximation described as Model III in Figure 1. In other words, the global problem solved using Model III is the non-linear least squares problem (30), and the algorithm employed for this purpose is the Gauss–Newton algorithm with the additional system approximation (32)–(33).

Most importantly, it is pointed out here that the search directions computed for this Model III are optimal in the sense of (34)—that is, the system approximation (32)–(33) preserves the least-squares optimality property established for the LSPG projection in the case of Model II. That is, as vectors are added to Φ_y while maintaining $R^{(k)}$, $J^{(k)}$, Φ_R, and Φ_J fixed, the error measure $\| \widetilde{J^{(k)}\Phi_y} a + \tilde{R}^{(k)} \|_2$ monotonically decreases. This occurs because as columns are added to Φ_y, columns are also added to the matrix $\widetilde{J^{(k)}\Phi_y}$ without modification of its current columns, since the columns of $\widetilde{J^{(k)}\Phi_y}$ are computed independently. These additional columns cause the minimization (34) to be performed over a superset of the previous space, thereby driving the aforementioned error to decrease monotonically.

It is also noted here that even when the non-linear problem of interest has SPD Jacobians that warrant the application of the Galerkin projection method described in Section 3.1 for non-linear model reduction, the matrices $\Phi_y^{\mathrm{T}} \widetilde{J^{(k)}\Phi_y}$ are not guaranteed to be SPD. Therefore, in the presence of a system approximation, the Galerkin projection method is not optimal in any sense. For this reason, only the LSPG projection is recommended here and is pursued in the remainder of this paper.

4.2. Gappy data reconstruction

Gappy data reconstruction is a technique for reconstructing vector (or tensor)-valued data that has 'gaps', i.e. unknown—or more appropriately in the context of this work, 'uncomputed'—entries. Therefore, it can also be viewed as a data compression procedure. From a mathematical viewpoint, this technique is equivalent to a least-squares regression in one discrete variable using empirical basis functions. It was introduced in [29] for the purpose of image reconstruction. It has also been used for steady [41, 42] and unsteady [43, 44] flow field reconstruction, inverse design [42],

DOI: 10.1002/nme

design variable mapping for multi-fidelity optimization [45], and to some extent, for non-linear computations [26, 46]. In this paper, a specific form of gappy data reconstruction is proposed for computing the generalized coordinates $R_r^{(k)}$ and $J_r^{(k)}$ of Equations (32)–(33). For this reason, and to keep this paper as self-contained as possible, gappy data reconstruction is first briefly reviewed.

Any vector $g \equiv [g_1 \ \dots \ g_N]^{\mathrm{T}} \in \mathbb{R}^N$ can be optimally reconstructed in the least-squares sense using a basis $\Phi_g \equiv [\phi_g^1 \ \dots \ \phi_g^{n_g}] \in \mathbb{R}^{N \times n_g}$ and $n_{\mathrm{i}} \leqslant N$ sampled entries g_{i_k}, $1 \leqslant k \leqslant n_{\mathrm{i}}$. Let $\mathscr{I} \equiv \{\mathrm{i}_1, \mathrm{i}_2, \dots, \mathrm{i}_{n_{\mathrm{i}}}\} \subset \{1, \dots, N\}$ denote the set of n_{i} sample indices for which g is evaluated. The gappy data reconstruction method computes an approximation $\tilde{g}$ to g of the form

$$\tilde{g} = \Phi_g g_r \tag{36}$$

with

$$g_r = \arg \min_{a \in \mathbb{R}^{n_g}} \|\hat{\Phi}_g a - \hat{g}\|_2, \tag{37}$$

where for a given $\mathscr{I}$, the restriction operator $\hat{\cdot}$ is defined on any $h \in \mathbb{R}^{N \times p}$ by

$$\hat{h} \equiv \begin{bmatrix} h_{\mathrm{i}_1}^1 & \cdots & h_{\mathrm{i}_1}^p \\ \vdots & \ddots & \vdots \\ h_{\mathrm{i}_{n_{\mathrm{i}}}}^1 & \cdots & h_{\mathrm{i}_{n_{\mathrm{i}}}}^p \end{bmatrix} = P^{\mathrm{T}} h, \tag{38}$$

where $P = [e_{\mathrm{i}_1} \ \cdots \ e_{\mathrm{i}_{n_{\mathrm{i}}}}]$ are selected columns of the identity matrix. For problem (37) to have a unique solution, the *alternant matrix* $\hat{\Phi}_g$ must have full column rank, and therefore $n_{\mathrm{i}} \geqslant n_g$ is required. When the number of sampled entries is equal to the number of basis functions—that is, $n_{\mathrm{i}} = n_g$—$\hat{\Phi}_g$ is a square matrix and $g_r = \hat{\Phi}_g^{-1} \hat{g}$. In this case, $\tilde{g}$ interpolates g at the sample indices ($\tilde{g}_i = g_i \forall i \in \mathscr{I}$) [27, 28]. For $n_{\mathrm{i}} > n_g$, the resulting least-squares problem can be solved by performing a thin QR factorization $\hat{\Phi}_g = Q_{\hat{\Phi}_g} R_{\hat{\Phi}_g}$ and solving

$$R_{\hat{\Phi}_g} g_r = Q_{\hat{\Phi}_g}^{\mathrm{T}} \hat{g}. \tag{39}$$

By design, the gappy data reconstruction method is optimal in the sense that the error measure $\|\hat{\tilde{g}} - \hat{g}\|_2$ monotonically decreases as the number of basis functions n_g is increased.

In the specific case of the Gappy POD method [29], the basis Φ_g is chosen to be the POD basis computed using snapshots of g for different inputs, time-instances, and Newton iterations. In this work, the Gappy POD method is employed to reconstruct the residual and, independently, each column of the column-reduced Jacobian. The POD bases $\Phi_R \equiv [\phi_R^1 \ \dots \ \phi_R^{n_R}] \in \mathbb{R}^{N \times n_R}$ and $\Phi_J \equiv [\phi_J^1 \ \dots \ \phi_J^{n_J}] \in \mathbb{R}^{N \times n_J}$ chosen for gappy data reconstruction in this work are specified next.

4.3. Consistency-driven construction of the bases Φ_R and Φ_J

This section develops sufficient conditions on the snapshots to be used for computing the POD bases Φ_R and Φ_J introduced in Equations (32) and (33), respectively, in order to meet the consistency requirement for the system approximation (Approximation 2 in Figure 1). It will be shown that this requirement, which ensures that the resulting Model III produces the same solution as Model II in the limit of no truncation of the POD bases, calls for using different snapshots for the bases Φ_R and Φ_J that are pre-computed during the evaluation of Model II. This constitutes a significant departure from the strategy that currently dominates the literature [24, 25, 27, 28, 30], where $\Phi_J = \Phi_R$ and the underlying snapshots are collected during the evaluation of Model I.

In the following, $p^{(k)}$ denotes the Model II search direction satisfying Equation (25) for given $R^{(k)}$, $J^{(k)}$, and Φ_y.

DOI: 10.1002/nme

Lemma 1
Let $\bar{R}^{(k)} \in \mathbb{R}^N$ and $\overline{J^{(k)}\Phi_y} \in \mathbb{R}^{N \times n_y}$ denote generic approximations of $R^{(k)}$ and $J^{(k)}\Phi_y$, respectively. Assume that $\overline{J^{(k)}\Phi_y}$ has full column rank. If:

1. the non-linear residual is exactly approximated

$$\bar{R}^{(k)} = R^{(k)}, \tag{40}$$

2. *the action of the approximated column-reduced Jacobian on the* Model II *search direction* $p^{(k)}$ is exactly approximated

$$\overline{J^{(k)}\Phi_y}\, p^{(k)} = J^{(k)}\Phi_y\, p^{(k)}, \tag{41}$$

3. and the range of the approximated column-reduced Jacobian is the same as the range of the true column-reduced Jacobian

$$\text{range}(\overline{J^{(k)}\Phi_y}) = \text{range}(J^{(k)}\Phi_y), \tag{42}$$

then the solution $\bar{p}^{(k)}$ of the approximated reduced Newton equations

$$\bar{p}^{(k)} = \arg \min_{a \in \mathbb{R}^{n_y}} \| \overline{J^{(k)}\Phi_y} a + \bar{R}^{(k)} \|_2 \tag{43}$$

is equal to the Model II search direction $p^{(k)}$.

Lemma 1 above is proved in Appendix A.

The next lemma provides sufficient conditions on the bases Φ_R and Φ_J for satisfying conditions 1–3 of Lemma 1 in the case where gappy data reconstruction is employed for the tensor approximations.

Lemma 2
Assume that the approximations $\bar{R}^{(k)}$ and $\overline{J^{(k)}\Phi_y}$ are chosen as $\bar{R}^{(k)} = \tilde{R}^{(k)}$ and $\overline{J^{(k)}\Phi_y} = \widetilde{J^{(k)}\Phi_y}$ as in (32) and (33), respectively, and that the restricted matrices $\hat{\Phi}_R$ and $\hat{\Phi}_J$ have full column rank. Then:

I. If

$$R^{(k)} \in \text{range}(\Phi_R), \tag{44}$$

condition 1 of Lemma 1 holds.

II. If

$$J^{(k)}\Phi_y\, p^{(k)} \in \text{range}(\Phi_J), \tag{45}$$

condition 2 of Lemma 1 holds.

III. If each column of $J^{(k)}\Phi_y$ is in the range of Φ_J—that is,

$$J^{(k)}\phi_y^j \in \text{range}(\Phi_J), \quad 1 \leqslant j \leqslant n_y, \tag{46}$$

condition 3 of Lemma 1 holds.

Lemma 2 above is proved in Appendix B. It provides sufficient conditions on Φ_R and Φ_J to obtain an error-free system approximation (32)–(33). Proposition 2 below provides sufficient conditions on the underlying snapshots $X_R \equiv [x_R^1 \ \dots \ x_R^{n_{x_R}}]$ and $X_J \equiv [x_J^1 \ \dots \ x_J^{n_{x_J}}]$ for ensuring that the non-truncated POD bases Φ_R and Φ_J satisfy conditions I–III of Lemma 2.

Proposition 2
Consistency of the system approximation. If:

DOI: 10.1002/nme

i. the residual $R^{(k)}$ is one of the snapshots used for computing Φ_R—that is,

$$R^{(k)} \in \{x_R^i\}_{i=1}^{n_{x_R}}, \tag{47}$$

ii. the action of the column-reduced Jacobian on the Model II search direction $p^{(k)}$ is one of the snapshots used for computing Φ_J—that is,

$$J^{(k)}\Phi_y p^{(k)} \in \{x_J^i\}_{i=1}^{n_{x_J}}, \tag{48}$$

iii. and each column of $J^{(k)}\Phi_y^j$ is a snapshot used for computing Φ_J—that is,

$$J^{(k)}\phi_y^j \in \{x_J^i\}_{i=1}^{n_{x_J}}, \quad 1 \leqslant j \leqslant n_y, \tag{49}$$

then the system approximation (32)–(33) is consistent.

Proposition 2 above is proved in Appendix C. It provides conditions on the snapshots collected for Φ_R and Φ_J that ensure consistency. That is, when the quantities specified in Proposition 2 are collected as snapshots for all Newton iterations during the evaluation of Model II for a given problem, Model III generates the same solution as Model II for the same problem. This result provides a rational starting point for the development of snapshot collection procedures for the system approximation (32)–(33). More specifically, it suggests that a snapshot collection method would benefit from satisfying all three identified conditions, or only a subset of them if computational feasibility dictates it. It also suggests that accuracy could be improved by using different bases for the residual and column-reduced Jacobian.

Guided by this result, several snapshot collection procedures for constructing the POD bases Φ_R and Φ_J in view of performing the tensor approximations $\tilde{R}^{(k)}$ and $\widetilde{J^{(k)}\Phi_y}$ are presented in Table I. These procedures correspond to various tradeoffs between consistency and computational feasibility. A (k) superscript indicates that snapshots are saved at each Newton iteration. Subscripts I and II specify the model for which the snapshots are collected. All methods satisfy the condition of Proposition 1, which ensures consistency of the projection approximation (Approximation 1).

Method 3, which is consistent because it satisfies all three conditions of Proposition 2, is infeasible for most problems as it requires saving n_y+1 vectors at each Newton step during the evaluation of Model II. Method 2, which is not consistent but satisfies consistency conditions i and ii of Proposition 1, is computationally feasible as it requires saving only two vectors per Newton step. Method 1 is marginally more economical than Method 2 as it requires saving only one vector per Newton iteration; yet, it does not satisfy condition ii. For this reason, Method 1 is not generally preferred over Method 2. Method 0, which is similar to the approach commonly adopted in the literature [24, 25, 27, 28, 30], satisfies none of these sufficient conditions, but requires only one simulation (Model I) to be executed offline. For the above reasons, Method 2 is recommended in this work.

Table I. Snapshot collection procedures for Model III of Figure 1.

Procedure identifier	0	1	2	3
Snapshots for y	$y_{\mathrm{I}}-y_{\mathrm{I}}^{(0)}$	$y_{\mathrm{I}}-y_{\mathrm{I}}^{(0)}$	$y_{\mathrm{I}}-y_{\mathrm{I}}^{(0)}$	$y_{\mathrm{I}}-y_{\mathrm{I}}^{(0)}$
Snapshots for $R^{(k)}$	$R_{\mathrm{I}}^{(k)}$	$R_{\mathrm{II}}^{(k)}$	$R_{\mathrm{II}}^{(k)}$	$R_{\mathrm{II}}^{(k)}$
Snapshots for $J^{(k)}\Phi_y$	$R_{\mathrm{I}}^{(k)}$	$R_{\mathrm{II}}^{(k)}$	$[J^{(k)}\Phi_y p^{(k)}]_{\mathrm{II}}$	$[J^{(k)}\Phi_y]_{\mathrm{II}}$
# Simulations	1	2	2	2
# Snapshots per Newton iteration	1	1	2	n_y+1
Proposition 2 conditions satisfied	None	i	i, ii	i, ii, iii

DOI: 10.1002/nme

5. COMPUTATIONAL PROCEDURES

5.1. Offline procedure

In the offline stage of an offline-online computational strategy designed for real-time prediction, data is acquired to construct the surrogate model to be used in the online stage. The computational steps executed during the offline stage of the proposed method can be summarized as follows:

1. Acquire data by collecting snapshots X_y, X_R, and X_J, and compute the POD bases Φ_y, Φ_R, and Φ_J. Table I provides possible snapshot collection methods, and Algorithm 1 contains the procedure for computing the POD bases using these snapshots. To ensure uniqueness of the reconstructed gappy quantities, the conditions $n_R \geqslant n_y$ and $n_J \geqslant n_y$ are required.
2. Determine the set of n_i indices $\mathscr{I} \equiv \{\mathsf{i}_1, \mathsf{i}_2, \ldots, \mathsf{i}_{n_\mathrm{i}}\}$ for reconstructing the residual and column-reduced Jacobian. A greedy method to do so is presented in Section 5.4. To ensure uniqueness in the least-squares gappy reconstruction, the conditions $n_\mathrm{i} \geqslant n_R$ and $n_\mathrm{i} \geqslant n_J$ are required.
3. Compute the matrices required for online computation using Algorithm 2.

Algorithm 2 Offline computation of online matrices.

Input: Φ_R, Φ_J, $\mathscr{I}$
Output: A, B

1: Build restricted matrices $\hat{\Phi}_R \in \mathbb{R}^{n_\mathrm{i} \times n_R}$ and $\hat{\Phi}_J \in \mathbb{R}^{n_\mathrm{i} \times n_J}$ using indices $\mathscr{I}$ and the restriction operator defined in Equation (38)
2: Compute thin QR factorizations $\hat{\Phi}_R = Q_{\hat{\Phi}_R} R_{\hat{\Phi}_R}$ and $\hat{\Phi}_J = Q_{\hat{\Phi}_J} J_{\hat{\Phi}_J}$
3: Solve $2n_\mathrm{i}$ linear systems for pseudoinverses $D = R_{\hat{\Phi}_R}^{-1} Q_{\hat{\Phi}_R}^{\mathrm{T}} \in \mathbb{R}^{n_R \times n_\mathrm{i}}$ and $E = R_{\hat{\Phi}_J}^{-1} Q_{\hat{\Phi}_J}^{\mathrm{T}} \in \mathbb{R}^{n_J \times n_\mathrm{i}}$
4: Compute $C = \Phi_J E \in \mathbb{R}^{N \times n_\mathrm{i}}$ and its thin QR factorization $C = Q_C R_C$
5: Set $A \leftarrow R_C$ with $A \in \mathbb{R}^{n_\mathrm{i} \times n_\mathrm{i}}$
6: Compute $B = Q_C^{\mathrm{T}} \Phi_R D \in \mathbb{R}^{n_\mathrm{i} \times n_\mathrm{i}}$

Algorithm 2 can be derived by considering the approximated reduced Newton equations (34)

$$\begin{aligned}
\tilde{p}^{(k)} &= \arg\min_{a \in \mathbb{R}^{n_y}} \| \widehat{J^{(k)} \Phi_y} a + \tilde{R}^{(k)} \|_2 \\
&= \arg\min_{a \in \mathbb{R}^{n_y}} \| \Phi_J J_r^{(k)} a + \Phi_R R_r^{(k)} \|_2 \\
&= \arg\min_{a \in \mathbb{R}^{n_y}} \| \Phi_J R_{\hat{\Phi}_J}^{-1} Q_{\hat{\Phi}_J}^{\mathrm{T}} \widehat{J^{(k)} \Phi_y} a + \Phi_R R_{\hat{\Phi}_R}^{-1} Q_{\hat{\Phi}_R}^{\mathrm{T}} \hat{R}^{(k)} \|_2 .
\end{aligned} \tag{50}$$

Equation (50) uses the gappy approximation (36)–(37) to compute $J_r^{(k)}$ and $R_r^{(k)}$. Setting $C = \Phi_J R_{\hat{\Phi}_J}^{-1} Q_{\hat{\Phi}_J}^{\mathrm{T}} \in \mathbb{R}^{N \times n_\mathrm{i}}$ and computing the thin QR factorization $C = Q_C R_C$ transforms Equation (50) into

$$\tilde{p}^{(k)} = \arg\min_{a \in \mathbb{R}^{n_y}} \| A \widehat{J^{(k)} \Phi_y} a + B \hat{R}^{(k)} \|_2, \tag{51}$$

where $A = R_C \in \mathbb{R}^{n_\mathrm{i} \times n_\mathrm{i}}$ and $B = Q_C^{\mathrm{T}} \Phi_R R_{\hat{\Phi}_R}^{-1} Q_{\hat{\Phi}_R}^{\mathrm{T}} \in \mathbb{R}^{n_\mathrm{i} \times n_\mathrm{i}}$.

The next section describes the online stage, where the objective is to solve the approximated reduced Newton equations (51).

5.2. Online procedure

As indicated by Equation (51), the online work at each Newton iteration consists simply of computing the restricted quantities $\widehat{J^{(k)} \Phi_y}$ and $\hat{R}^{(k)}$ and solving Equation (51) using, for example, Algorithm 3. The partial update of the state in Step 1 is explained in the following section.

DOI: 10.1002/nme

Algorithm 3 Online computation of an approximated search direction (solution of Equation (51)).

Input: A, B, $\check{y}^{(0)}$, $y_r^{(k)}$
Output: $\tilde{p}^{(k)}$

1: Partially update the state $\check{y}^{(k)} = \check{y}^{(0)} + \check{\Phi}_y y_r^{(k)}$, where the operator $\check{\cdot}$ is defined by (54)
2: Compute $\hat{R}^{(k)} \in \mathbb{R}^{n_\mathrm{i}}$ and $\widehat{J^{(k)}\Phi_y} \in \mathbb{R}^{n_\mathrm{i} \times n_y}$ by evaluating $R_i^{(k)}$ and $\left[J^{(k)}\Phi_y\right]_i$ $\forall i \in \mathscr{I}$
3: Compute $\bar{B} = B\hat{R}^{(k)}$ and $\bar{A} = A\widehat{J^{(k)}\Phi_y}$
4: Compute the thin QR factorization $R_{\bar{A}} = Q_{\bar{A}}^{\mathrm{T}} \bar{A}$
5: Compute $Q_{\bar{A}}^{\mathrm{T}} \bar{B}$
6: Solve $\tilde{p}^{(k)} = -R_{\bar{A}}^{-1} Q_{\bar{A}}^{\mathrm{T}} \bar{B}$

Table II reports the computational complexity of the online computations associated with Algorithm 3. In this table, ω and γ denote the average number of floating point operations needed for evaluating one entry of the residual and one row of the Jacobian, respectively, and therefore are application-dependent. ξ denotes the average number of non-zeros per row of the Jacobian matrix, and n_j denotes the number of state vector entries required for the partial update of the state. The results reported in Table II show that the computational cost of the online component of processing Model III is independent of the dimension N of the large-scale system of state equations.

Algorithm 4 describes the online solution, using the non-linear ROM Model III, of the n_t problems of interest of the form given in (5), for a configuration defined by $\mu^\star$.

Algorithm 4 Online solution of non-linear problems (5)–(6) using Model III.

Define the online configuration $\mu^\star$ and the initial condition $\check{y}^0$
for $n = 0, \ldots, n_\mathrm{t} - 1$ **do**
  $k \leftarrow 0$
  $\check{y}^{(0)} \leftarrow \check{y}^n$, $y_r^{(0)} \leftarrow 0$
  while $\|Q_{\bar{A}}^{\mathrm{T}} \bar{B}\|_2 > \varepsilon$ **do**
    Solve Equation (51) for $\tilde{p}^{(k)}$ using Algorithm 3
    $y_r^{(k+1)} \leftarrow y_r^{(k)} + \alpha^{(k)} \tilde{p}^{(k)}$ where $\alpha^{(k)}$ is computed by a line search
    $k \leftarrow k + 1$
  end while
  $\check{y}^{n+1} \leftarrow \check{y}^{(k)}$
end for
Compute the outputs $G(\acute{y}^0, \ldots, \acute{y}^{n_\mathrm{t}}, \mu^\star)$, where the operator $\acute{\cdot}$ is defined by (56)

5.3. Partial computation of the state

As the residual and column-reduced Jacobian are evaluated at only a subset of indices $\mathscr{I} \subset \{1, \ldots, N\}$, in most cases it is unnecessary to compute the entire state $y \in \mathbb{R}^N$. This is particularly

Table II. Computational complexity of Algorithm 3 for performing the online component of the evaluation of Model III.

Step of Algorithm 3	Approximate operation count
1	$2n_\mathrm{j}n_y$
2	$\omega n_\mathrm{i} + \gamma n_\mathrm{i} + 2\xi n_\mathrm{i} n_y$
3	$2n_\mathrm{i}^2(n_y + 1)$
4	$2n_\mathrm{i}n_y^2$
5	$4n_\mathrm{i}n_y - 2n_y^2$
6	n_y^2
Total	$\sim 2n_\mathrm{j}n_y + 2\xi n_\mathrm{i}n_y + 2n_\mathrm{i}n_y^2 + 2n_\mathrm{i}^2 n_y$

DOI: 10.1002/nme

true if the Jacobian $J^{(k)}$ is sparse. Previous work [27] has also pointed out this opportunity for efficiency.

The set of n_{j} state vector indices required to compute the restricted residual and column-reduced Jacobian for a given problem $R(y)=0$ is given by

$$\mathscr{J}(\mathscr{I}) \equiv \{\mathrm{j}_1, \mathrm{j}_2, \ldots, \mathrm{j}_{n_{\mathrm{j}}}\} \tag{52}$$

$$\equiv \{j \in \{1, \ldots, N\} \mid \exists k \ \exists i \in \mathscr{I} \quad \text{with } J_{ij}^{(k)} \neq 0\}. \tag{53}$$

Here, $J_{ij}^{(k)}$ is the (i, j) element of $J^{(k)}$ and $[\Phi_y]_j$ is the jth row of Φ_y. As the topology is assumed constant for all problems, $\mathscr{J}$ is unchanged between problems.

The mask operator $\check{\cdot}: \mathbb{R}^N \to \mathbb{R}^{n_{\mathrm{j}}}$ is thus defined on a vector $h \in \mathbb{R}^N$ given $\mathscr{J}$ as

$$\check{h} \equiv [h_{\mathrm{j}_1} \ \ldots \ h_{\mathrm{j}_{n_{\mathrm{j}}}}]^{\mathrm{T}}. \tag{54}$$

When evaluating functions $f: \mathbb{R}^N \to \mathbb{R}^q$ with masked arguments, an alternative function $\underline{f}: \mathbb{R}^{n_{\mathrm{j}}} \to \mathbb{R}^q$ is used as follows:

$$\underline{f}(\check{h}) \equiv f(\acute{h}), \tag{55}$$

where

$$\acute{h}_j \equiv \begin{cases} 0, & j \notin \mathscr{J}, \\ h_j, & j \in \mathscr{J}. \end{cases} \tag{56}$$

Practically, this can be implemented using the compressed sparse row format [27].

5.4. *Greedy method for sample index selection*

Several methods have been proposed in the literature for selecting sample points or indices that define the restriction operator $\hat{\cdot}$ (see Equation (38)). In particular, approaches have been developed to select points that minimize the error in the reconstructed snapshots [27, 30], the difference between the reconstructed snapshots and their optimal projections [25, 28], and the condition number of the normal-equations matrix used for reconstruction [26, 44].

Here, the proposed method for sample index selection alternates between minimizing (1) the error in the gappy reconstruction of the POD basis vectors Φ_R and Φ_J and (2) the number of entries n_{j} of a state vector required to evaluate the ROM online. This method essentially modifies the approach of [27, 30] to take into account the effect of selecting sample indices on the computational complexity of the online component of the ROM evaluation. For this purpose, a suboptimal but economical greedy minimization algorithm is chosen. Furthermore, the minimization procedure is designed to operate on the optimally ordered POD bases rather than the snapshots themselves, because the greedy algorithm favors the initial vectors in the POD bases and these are the most 'energetic' ones.

The proposed sample index selection procedure is summarized in Algorithm 5. In Step 4, the sample index is chosen to minimize the mean squared gappy reconstruction error for the residual and Jacobian POD bases. In Step 6, the number of state vector entries n_{j} is minimized by selecting the indices that 'see' the same state vector entries. Therefore, adding these indices to the index set incurs no additional cost in terms of computing a state. Finally, it is noted that p_R and p_J are chosen so that the reconstruction in Steps 11–12 is executed only with the basis vectors that will actually be used online.

Remark

It should be noted that for the sake of computational efficiency, the index selection algorithm proposed above does not take into account the outputs of interest. Indeed, accounting for such

DOI: 10.1002/nme

outputs while choosing the indices of the residuals would require the solution of a computationally intensive optimization problem as the relationship between the outputs and the indices is not in general a simple one.

6. PERFORMANCE ASSESSMENT

Here, the performance of the non-linear model reduction method proposed in this paper is assessed with the geometrically non-linear finite element transient dynamic analysis of the clamped-free N_b-bay truss structure shown in Figure 4. Each bay contains 16 structural members and each member is represented by a three-dimensional non-linear bar element. The semi-discrete equations of dynamic equilibrium associated with this finite element representation of the global structure can be written as

$$M\ddot{u}(t)+f^{\text{int}}(u(t))=f^{\text{ext}}(t), \quad t\in[0,t_f], \tag{57}$$

where a dot denotes a time-derivative, $u(t)\in\mathbb{R}^N$ is the displacement vector, $M\in\mathbb{R}^{N\times N}$ is the mass matrix, $f^{\text{int}}\in\mathbb{R}^N$ is the vector of non-linear internal forces, and $f^{\text{ext}}\in\mathbb{R}^N$ is the vector of prescribed external forces and is set here to zero. The structure is excited by an initial vertical displacement represented by the arrow shown in Figure 4.

Algorithm 5 Greedy algorithm for computing sample indices.

Input: $\Phi_R(n_{x_R}, X_R)$, $\Phi_J(n_{x_J}, X_J)$, n_R, n_J, n_{i}
Output: $\mathscr{I}$

1: $\mathscr{I}=\emptyset$, $\bar{n}_{\text{i}}=0$, $m=1$
2: $\mathsf{R}\leftarrow\phi_R^1$, $\mathsf{J}\leftarrow\phi_J^1$
3: **while** $\bar{n}_{\text{i}}<n_{\text{i}}$ **do**
4: $\quad i\leftarrow\arg\max_{l\in\{1,\ldots,N\}/\mathscr{I}}((\mathsf{R}_l)^2+(\mathsf{J}_l)^2)$
5: $\quad\mathscr{I}\leftarrow\mathscr{I}\cup i$
6: $\quad\mathscr{K}\leftarrow\{k\in\{1,\ldots,N\}/\mathscr{I}\mid\mathscr{J}(k)\subset\mathscr{J}(\mathscr{I})\}$
7: $\quad\mathscr{I}\leftarrow\mathscr{I}\cup\mathscr{K}$
8: $\quad\bar{n}_{\text{i}}\leftarrow\bar{n}_{\text{i}}+1+|\mathscr{K}|$
9: $\quad m\leftarrow m+1$
10: $\quad p_R=\min(m-1,n_R)$, $p_J=\min(m-1,n_J)$
11: $\quad\mathsf{R}\leftarrow\phi_R^m-[\phi_R^1\ \ldots\ \phi_R^{p_R}]\phi_{Rr}^m$ where $\phi_{Rr}^m=\arg\min_{a\in\mathbb{R}^{\bar{n}_{\text{i}}}}\|[\hat{\phi}_R^1\ \ldots\ \hat{\phi}_R^{p_R}]a-\hat{\phi}_R^m\|_2$
12: $\quad\mathsf{J}\leftarrow\phi_J^m-[\phi_J^1\ \ldots\ \phi_J^{p_J}]\phi_{Jr}^m$, where $\phi_{Jr}^m=\arg\min_{a\in\mathbb{R}^{\bar{n}_{\text{i}}}}\|[\hat{\phi}_J^1\ \ldots\ \hat{\phi}_J^{p_J}]a-\hat{\phi}_J^m\|_2$
13: **end while**

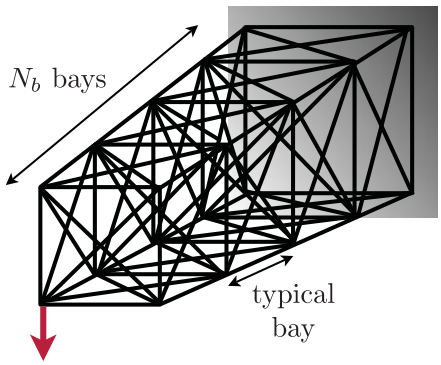

Figure 4. Clamped-free N_b-bay truss structure.

DOI: 10.1002/nme

Equation (57) above is discretized by the second-order time-accurate implicit midpoint rule that leads at each time-step to the residual

$$R(u^{n+1}) = M\left(\frac{u^{n+1}-u^n}{2} - \frac{\Delta t^n}{2}\dot{u}^n\right) + \left(\frac{\Delta t^n}{2}\right)^2 f^{\text{int}}\left(\frac{u^{n+1}+u^n}{2}\right)$$

$$-\left(\frac{\Delta t^n}{2}\right)^2 f^{\text{ext}}\left(t^n + \frac{\Delta t^n}{2}\right) \tag{58}$$

and associated Jacobian

$$\frac{\partial R}{\partial u}(u^{n+1}) = \frac{1}{2}\left(M + \left(\frac{\Delta t^n}{2}\right)^2 K_t\left(\frac{u^n+u^{n+1}}{2}\right)\right). \tag{59}$$

In Equations (58) and (59) above, the superscript n designates the time-instance t^n, $\Delta t^n = t^{n+1} - t^n$ denotes the computational time-step at time-instance t^n, and $K_t(u^n) \equiv \partial f^{\text{int}}/\partial u|_{u^n}$ denotes the tangent stiffness matrix evaluated at the state u^n.

In all numerical computations, the Model I and Model II Newton equations are solved directly using the Cholesky factorization of the Jacobian. The Model III reduced Newton equations are solved using the procedure described in Algorithm 3.

The output of interest z is chosen to be the downward velocity v at the tip of the structure for all time-steps—that is, $z = [v^0\ v^1\ \cdots\ v^{n_t}]$. To quantify the error in a prediction generated by Model II for this problem, the following time-averaged metric is used

$$e_{\text{I}} = \frac{\frac{1}{n_t}\sum_{n=0}^{n_t} |v^n - v_{\text{I}}^n|}{\max_{i,j\in\{0,\ldots,n_t\}} |v_{\text{I}}^i - v_{\text{I}}^j|} \times 100\%, \tag{60}$$

where the subscript I indicates that the error is measured with respect to the counterpart solution obtained using Model I. On the other hand, to quantify the error in a solution generated by Model III of this problem, two different error metrics are used: e_{I} above and e_{II} defined as

$$e_{\text{II}} = \frac{\frac{1}{n_t}\sum_{n=0}^{n_t} |v^n - v_{\text{II}}^n|}{\max_{i,j\in\{0,\ldots,n_t\}} |v_{\text{II}}^i - v_{\text{II}}^j|} \times 100\%. \tag{61}$$

The e_{II} metric is suitable for verifying the consistency conditions introduced in Section 4.3. In the following, Model III.i, $i = 0, 1, 2$ refers to Model III equipped with the ith snapshot collection procedure specified in Table I.

6.1. Verification of the consistency conditions

First, the number of bays of the structure is set to $N_b = 150$—in which case the total number of dofs of the finite element model becomes $N = 1800$. The time-interval of interest is sampled into $n_t = 150$ time-instances.

Three separate finite element analyses of the truss structure are performed: one associated with Model I; one with the reduced-order Model II without truncation of the POD basis; and one with the reduced-order Model III equipped with the consistent snapshot collection procedure 3 specified in Table I, also without truncation of the POD bases. Model III.3 is implemented with the sample index factor, defined as

$$\eta \equiv \frac{n_i}{n_R}, \tag{62}$$

set to $\eta = 1.0$. Note that for $\eta = 1.0$, the gappy reconstruction interpolates the residual and column-reduced Jacobian at each Newton iteration, while these reconstructions are done in the least-squares sense for $\eta > 1.0$.

DOI: 10.1002/nme

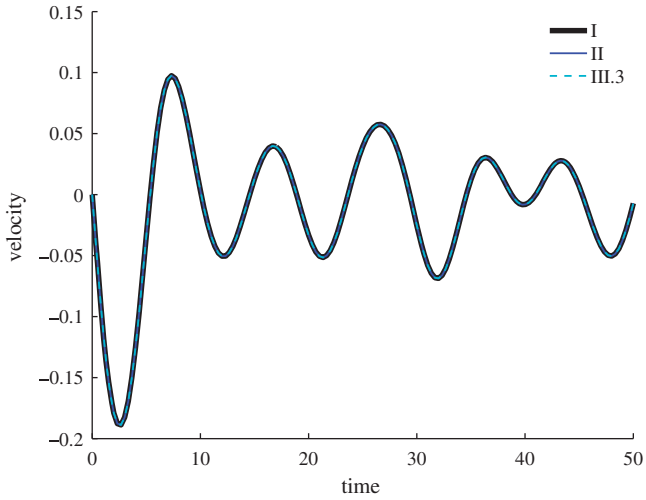

Figure 5. Time-history of the tip velocity of the structure for Models I, II, and III.3. Here, $N_b = 150$, $n_t = 150$, $\eta = 1.0$, and Models II and III.3 are implemented with no truncation of the POD bases.

In all cases, the iterative Newton process is declared to have converged when the two-norm of the absolute non-linear residual reaches the value $\varepsilon = 3\varepsilon_0$, where $\varepsilon_0 = 2.22 \times 10^{-16}$ is machine precision. This stringent convergence criterion is chosen because the objective of this numerical experiment is to demonstrate consistency—that is, to show numerically that the approximations do not introduce any error when the POD bases are not truncated. Therefore, controlling tightly the convergence of the Newton solver eliminates the possibility of attributing consistency errors to solver performance.

The predicted output of interest z is graphically reported in Figure 5 in the form of a time-history. For Model II, it is found that the output error measure is $e_{\mathrm{I}} = 1.83 \times 10^{-15}$—that is, on the order of machine precision. This is consistent with the theoretical result of Proposition 1. For Model III.3, it is found that $e_{\mathrm{II}} = 1.16 \times 10^{-15}$, which is consistent with the result of Proposition 2, and $e_{\mathrm{I}} = 1.14 \times 10^{-15}$.

6.2. *Effect on performance of satisfying the consistency conditions in the presence of truncation*

Next, the number of structural bays is increased to $N_b = 1000$, which generates $N = 1.2 \times 10^4$ dofs. The time-interval of interest is discretized into $n_t = 300$ time-instances.

To determine the dimension n_y of the POD basis for the state, an energy truncation criterion is employed with $n_y = n_{\mathscr{E}}(\Sigma_u, 0.99)$ (see Equation (19)). This gives $n_y = 26$, resulting in a ROM of dimension equal to 0.22% of that of Model I. Similarly, the number of vectors used for reconstructing the residual, n_R, and that used for reconstructing the column-reduced Jacobian, n_J, are fixed at 28. This corresponds to $n_{\mathscr{E}}(\Sigma_R, 0.99)$, where Σ_R is the matrix of singular values arising from the SVD of the residuals computed during the transient dynamic analysis of Model I.

First, the number of sample indices is set to the number of POD basis vectors for the residual and Jacobian. In this case, $n_i = n_R = n_J$ and the sample index factor is $\eta = 1.0$. One analysis is performed using Model II, and three other analyses are performed using Model III: one for each of the first three snapshot collection procedures specified in Table I. In all cases, the iterative Newton-type process is declared to have converged when the two-norm of the absolute non-linear residual reaches the value $\varepsilon = 10^{-5}$. An analysis of Model III using the snapshot collection procedure 3 specified in Table I is not performed here because of its excessive storage requirements.

The obtained output results are graphically depicted in Figure 6, and the performance results are reported in Table III. Model II is found to reproduce the output of Model I with the small relative error of 3.44%. However, it delivers almost no speed-up over the high-dimensional Model I, illustrating the need for a system approximation. Method III.0 is found to be unstable for this

DOI: 10.1002/nme

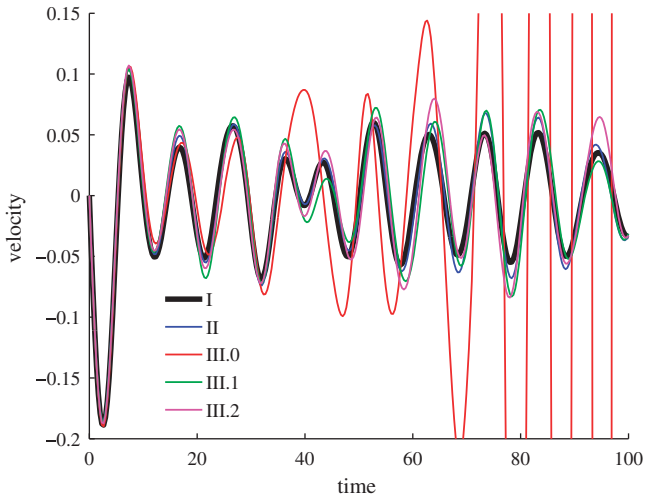

Figure 6. Time-history of the tip velocity of the structure for Models I, II, and III.0–2. Here, $N_b = 1000$, $n_t = 300$, $n_y = 26$, $\eta = 1.0$, and $n_R = n_J = 28$.

Table III. Performance results of Model II and Models III.0–2. Here, $N_b = 1000$, $n_t = 300$, $n_y = 26$, $\eta = 1.0$, and $n_R = n_J = 28$.

Model	Relative error e_I (%) for output v	Total number of Newton iterations	Speed-up over Model I
II	3.44	601	1.03
III.0	361	1076	120
III.1	6.31	603	158
III.2	6.01	891	122

Table IV. Effect of increasing the sample index factor while maintaining $n_R = n_J = 28$ constant. Here, $N_b = 1000$, $n_t = 300$, $n_y = 26$, and $\eta = 1.0$.

Sample index factor η	Relative error e_I (%) for output v			Speed-up over model I		
	Model III.0	III.1	III.2	Model III.0	III.1	III.2
1.0	361	6.31	6.01	120	158	122
1.5	4.95	3.77	3.63	137	152	119
2.0	4.03	3.50	3.44	129	136	116

problem, perhaps because it does not satisfy any of the consistency conditions of Proposition 2. On the other hand, both Methods III.1 and III.2 are found to be sufficiently accurate—they deliver solutions with relative errors of roughly 6%—and fast—they generate speed-ups greater than 100 over the high-dimensional Model I—thereby underscoring the importance of satisfying at least some of the consistency conditions developed in Section 4.3.

Next, similar analyses are performed using an increasing value of n_i while maintaining $n_R = n_J = 28$. The corresponding performance results are reported in Table IV. They reveal that in all cases, increasing the number of sample indices decreases the error and even stabilizes Model III.0. However, increasing n_i decreases the speed-up as expected (see Table II). The performance results reported in Table IV also reveal that the considered non-linear ROMs are more accurate when they satisfy more consistency conditions.

DOI: 10.1002/nme

Table V. Effect of increasing the sample index factor η for a fixed number of indices n_{i}. Here, $N_b = 1000$, $n_{\mathrm{t}} = 300$, $n_y = 26$, and $n_J = n_R$.

			Relative error e_{I} (%) for output v			Speed-up over Model I		
n_{i}	n_R	η	Model III.0	III.1	III.2	Model III.0	III.1	III.2
42	42	1.00	5.21	4.37	4.59	126	150	65
	37	1.14	4.16	4.13	3.52	125	138	138
	28	1.50	4.95	3.77	3.63	137	152	119
56	56	1.00	5.32	3.56	3.57	112	125	126
	37	1.50	4.32	3.83	3.51	118	134	132
	28	2.00	4.03	3.50	3.44	129	136	116
84	84	1.00	3.07	3.44	3.44	97.1	103	110
	56	1.50	3.72	3.46	3.47	94.9	102	110
	42	2.00	3.65	3.43	3.45	97.5	102	105

Finally, another suite of analyses is performed to examine the effect of varying η while maintaining the number of sample indices n_{i} constant. The obtained performance results, which are reported in Table V, indicate that increasing η (i.e. decreasing n_R and n_J) while maintaining n_{i} constant generally decreases the error without significantly affecting the speed-up. This suggests that performing least-squares gappy reconstruction ($\eta > 1.0$) as opposed to interpolation ($\eta = 1.0$) often improves accuracy without significantly affecting the CPU performance.

7. SUMMARY AND CONCLUSIONS

A Petrov–Galerkin projection method is proposed in this paper for non-linear model reduction to enable real-time prediction. The projections are designed to operate at the *discrete* level, for a static or dynamic computational model. The right reduced-order basis is chosen to be invariant and is constructed using the POD method. The left reduced-order basis is selected to minimize in a least-squares sense the residual of the linearized system of equations arising at each Newton iteration. This choice of left reduced-order basis, together with the decision to perform the non-linear model reduction at the discrete rather than the semi-discrete or continuous level, leads to the Gauss–Newton solution method, which is globally convergent. Numerical results obtained for difficult problems such as the prediction of turbulent flows reveal that the Petrov–Galerkin projection method generates non-linear ROMs that are robust and accurate. However, it does not deliver in general the speed-up expected from model reduction, because the computational cost associated with the projection at each Gauss–Newton iteration of the updated Jacobian and non-linear residual scales with the size of the higher dimensional model.

To avoid this computational cost, a system approximation is incorporated in the proposed non-linear model reduction method. At each linearized iteration, the non-linear residual and the action of the Jacobian on the right reduced-order basis are each approximated using the Gappy POD method and a unique POD basis. Therefore, the proposed non-linear model reduction method relies on three different POD bases: one for constructing the right reduced-order basis and two for approximating at each Gauss–Newton iteration the projected linearized system.

To design the snapshot collection procedures needed for constructing these three POD bases, the concept of consistent approximation is first extended to the context of model reduction. In this context, a snapshot-based approximation is defined to be consistent if, when implemented without data compression, it introduces no additional errors in the solution of the same problem for which snapshots were collected. Consistency requirements are formulated for the projection and system approximations, and sufficient conditions on the snapshots to be used for constructing the three POD bases are rigorously established for satisfying these consistency conditions.

DOI: 10.1002/nme

Finally, the Gappy POD method is equipped with a greedy algorithm for selecting sample indices that minimize the error in the reconstructed snapshots while reducing the computational complexity of the online computations.

The resulting non-linear model reduction method is illustrated with the solution of a set of geometrically non-linear structural dynamics problems. When equipped with a snapshot collection procedure that does not satisfy any of the established consistency conditions, the non-linear model reduction method is found to be unstable in most cases. On the other hand, when equipped with data acquisition procedures that meet the established consistency requirements, the non-linear model reduction method is found to accelerate the predictions based on the higher dimensional computational model by 2 orders of magnitude while delivering good accuracy.

APPENDIX A

Proof of Lemma 1

The least-squares problems (25) and (43) can be equivalently expressed as

$$\mathsf{p}^{\mathsf{k}} = \arg \min_{\mathsf{a} \in \mathrm{range}(J^{(k)}\Phi_y)} \|\mathsf{a} - R^{(k)}\|_2, \tag{A1}$$

$$\bar{\mathsf{p}}^{(\mathsf{k})} = \arg \min_{\mathsf{a} \in \mathrm{range}(\overline{J^{(k)}\Phi_y})} \|\mathsf{a} - \bar{R}^{(k)}\|_2, \tag{A2}$$

respectively, where $\mathsf{p}^{(\mathsf{k})} = J^{(k)}\Phi_y p^{(k)}$ and $\bar{\mathsf{p}}^{(\mathsf{k})} = \overline{J^{(k)}\Phi_y}\,\bar{p}^{(k)}$. From conditions 1 and 3 of Lemma 1, it follows that the above two problems are equivalent and therefore $\bar{\mathsf{p}}^{(\mathsf{k})} = \mathsf{p}^{(\mathsf{k})}$—that is,

$$J^{(k)}\Phi_y p^{(k)} = \overline{J^{(k)}\Phi_y}\,\bar{p}^{(k)}. \tag{A3}$$

Furthermore, from condition 2 of Lemma 1 and the above result, it follows that

$$\overline{J^{(k)}\Phi_y}\,p^{(k)} = \overline{J^{(k)}\Phi_y}\,\bar{p}^{(k)}. \tag{A4}$$

Finally, if $\overline{J^{(k)}\Phi_y}$ has full column rank, it follows that $p^{(k)} = \bar{p}^{(k)}$.

APPENDIX B

Proof of Lemma 2

I. If (44) holds, the non-linear residual can be expressed as $R^{(k)} = \Phi_R R_r^{(k)}$. This implies that $\hat{R}^{(k)} = \hat{\Phi}_R R_r^{(k)}$ and therefore $\|\hat{\Phi}_R R_r^{(k)} - \hat{R}^{(k)}\|_2 = 0$. If furthermore $\hat{\Phi}_R$ has full column rank, then $R_r^{(k)} = \arg\min_{a \in \mathbb{R}^{n_R}} \|\hat{\Phi}_R a - \hat{R}^{(k)}\|$ and therefore the gappy data reconstruction procedure $\tilde{R}^{(k)} = \Phi_R R_r^{(k)}$ reconstructs exactly the non-linear residual $R^{(k)}$.

II. The gappy approximation (33) where $J_r^{(k)}$ is computed as explained in (37) can be written as

$$\widetilde{J^{(k)}\Phi_y} = \Phi_J (\hat{\Phi}_J^{\mathrm{T}} \hat{\Phi}_J)^{-1} \hat{\Phi}_J^{\mathrm{T}} \widehat{J^{(k)}\Phi_y}. \tag{B1}$$

Then if (45) holds, the action of the Jacobian on the solution can be expanded as $J^{(k)}\Phi_y p^{(k)} = \Phi_J \mathsf{p}_r^{(k)}$. As this expression holds for all entries of $J^{(k)}\Phi_y p^{(k)}$, it also holds for the restricted entries

$$\widehat{J^{(k)}\Phi_y} p^{(k)} = \hat{\Phi}_J \mathsf{p}_r^{(k)}. \tag{B2}$$

DOI: 10.1002/nme

Post-multiplying Equation (B1) by $p^{(k)}$ and substituting Equation (B2) into the result gives

$$\widetilde{J^{(k)}\Phi_y}\,p^{(k)} = \Phi_J \mathsf{p}_r^{(k)}, \tag{B3}$$

and therefore

$$\widetilde{J^{(k)}\Phi_y}\,p^{(k)} = J^{(k)}\Phi_y p^{(k)}. \tag{B4}$$

III. Following the proof for I above, if (46) holds, then the gappy reconstruction of the column-reduced Jacobian is exact—that is, $\widetilde{J^{(k)}\phi_y^j} = J^{(k)}\phi_y^j$, $1 \leqslant j \leqslant n_y$—and therefore the weaker condition (42) is satisfied.

APPENDIX C

Proof of Proposition 2

From (47), (48), and (49), it follows that $R^{(k)} \in \text{span}\{x_R^i\}_{i=1}^{n_{x_R}}$, $J^{(k)}\Phi_y p^{(k)} \in \text{span}\{x_J^i\}_{i=1}^{n_{x_J}}$, and $J^{(k)}\phi_y^j \in \text{span}\{x_J^i\}_{i=1}^{n_{x_J}}$, $1 \leqslant j \leqslant n_y$. When the POD bases are not truncated—that is, when $\dim \Phi_J = n_{x_J}$ and $\dim \Phi_R = n_{x_R}$—$\text{span}\{x_R^i\}_{i=1}^{n_{x_R}} = \text{range}(\Phi_R)$ and $\text{span}\{x_J^i\}_{i=1}^{n_{x_J}} = \text{range}(\Phi_J)$. In this case, it follows from Lemma 2 that Lemma 1 holds, and therefore the system approximation is consistent.

ACKNOWLEDGEMENTS

The first author acknowledges the partial support by a National Science Foundation Graduate Fellowship and the partial support by a National Defense Science and Engineering Graduate Fellowship. The second and third authors acknowledge the partial support by the Motor Sports Division of the Toyota Motor Corporation under Agreement Number 48737, and the partial support by a research grant from the Academic Excellence Alliance program between King Abdullah University of Science and Technology (KAUST) and Stanford University. All authors also acknowledge the constructive comments received during the review process.

REFERENCES

1. Sirovich L. Turbulence and the dynamics of coherent structures. I—coherent structures. *Quarterly of Applied Mathematics* 1987; **45**:561–571.
2. Holmes P, Lumley J, Berkooz G. *Turbulence*, *Coherent Structures*, *Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, 1996.
3. Antoulas AC. *Approximation of Large-scale Dynamical Systems*. Society for Industrial and Applied Mathematics: Philadelphia, PA, 2005.
4. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering* 2007; **15**(3):1–47.
5. Veroy K, Prud'homme C, Rovas DV, Patera AT. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. *AIAA Paper 2003-3847*, *16th AIAA Computational Fluid Dynamics Conference*, Orlando, FL, 23–26 June 2003.
6. Nguyen NC, Veroy K, Patera AT. Certified real-time solution of parametrized partial differential equations. *Handbook of Materials Modeling*, Yip S (ed.). Springer: New York, 2005; 1529–1564.
7. Veroy K, Patera AT. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids* 2005; **47**(8):773–788.
8. Hall KC, Thomas JP, Dowell EH. Reduced-order modelling of unsteady small-disturbance flows using a frequency domain proper orthogonal decomposition technique. *AIAA Paper 99-0655*, *Proceedings of the 37th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1999.
9. LeGresley PA, Alonso JJ. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. *AIAA Paper 2000-25450*, *Fluids 2000 Conference and Exhibit*, Denver, CO, 19–22 June 2000.
10. Hall KC, Thomas JP, Dowell EH. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA Journal* 2000; **38**(2):1853–1862.

DOI: 10.1002/nme

11. Willcox K, Peraire J. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal* 2002; **40**(11):2323–2330.
12. Epureanu BI. A parametric analysis of reduced order models of viscous flows in turbomachinery. *Journal of Fluids and Structures* 2003; **17**:971–982.
13. Thomas JP, Dowell EH, Hall KC. Three-dimensional transonic aeroelasticity using proper orthogonal decomposition-based reduced order models. *Journal of Aircraft* 2003; **40**(3):544–551.
14. Kim T, Hong M, Bhatia KB, SenGupta G. Aeroelastic model reduction for affordable computational fluid dynamics-based flutter analysis. *AIAA Journal* 2005; **43**(12):2487–2495.
15. Lieu T, Farhat C, Lesoinne M. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5730–5742.
16. Lieu T, Farhat C. Adaptation of aeroelastic reduced-order models and application to an F-16 configuration. *AIAA Journal* 2007; **45**:1244–1269.
17. Amsallem D, Farhat C. An interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal* 2008; **46**:1803–1813.
18. Bergmann M, Cordier L, Brancher J-P. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids* 2005; **17**(9):80–101.
19. Han S, Feeny BF. Enhanced proper orthogonal decomposition for the modal analysis of homogeneous structures. *Journal of Vibration and Control* 2002; **8**(1):19–40.
20. Amabili M, Sarkar A, Paidoussis MP. Reduced-order models for nonlinear vibrations of cylindrical shells via the proper orthogonal decomposition method. *Journal of Fluids and Structures* 2003; **18**(2):227–250.
21. Kerschen G, Golinval J-C, Vakakis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dynamics* 2005; **41**:147–169.
22. Amsallem D, Carlberg K, Cortial J, Farhat C. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering* 2009; **78**:275–295.
23. Grepl MA, Nguyen NC, Veroy K, Patera AT, Liu GR. Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In *Proceedings of the Second Sandia Workshop of PDE-Constrained Optimization*, Biegler L, Ghattas O, Heinkenschloss M, Keyes D, van Bloemen Waanders B (eds). SIAM Computational Science and Engineering Book Series: Philadelphia, PA, 2007; 197–216.
24. Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis* 2007; **41**(3):575–605.
25. Nguyen NC, Peraire J. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *International Journal for Numerical Methods in Engineering* 2008; **76**:27–55.
26. Astrid P, Weiland S, Willcox K, Backx T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control* 2008; **53**(10):2237–2251.
27. Chaturantabut S, Sorensen DC. Discrete empirical interpolation for nonlinear model reduction. *Technical Report TR09-05*, Department of Computational and Applied Mathematics, Rice University, March 2009.
28. Galbally D, Fidkowski K, Willcox K, Ghattas O. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering* 2010; **81**(12): 1581–1608.
29. Everson R, Sirovich L. Karhunen–Loeve procedure for gappy data. *Journal of the Optical Society of America A* 1995; **12**(8):1657–1664.
30. Barrault M, Maday Y, Nguyen N, Patera A. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique Académie des Sciences* 2004; **339**(9):667–672.
31. Queipo N, Haftka R, Shyy W, Goel T, Vaidyanathan R, Tucker P. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences* 2005; **41**(1):1–28.
32. Rowley C, Colonius T, Murray R. Model reduction for compressible flows using POD and Galerkin projection. *Physica D*: *Nonlinear Phenomena* 2004; **189**(1–2):115–129.
33. Rathinam M, Petzold LR. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis* 2003; **41**(5):1893–1925.
34. Saad Y. *Iterative Methods for Sparse Linear Systems* (2nd edn). Society for Industrial and Applied Mathematics: Philadelphia, PA, 2003.
35. Golub GH, Loan CFV. *Matrix Computations* (3rd edn). Johns Hopkins University Press: Baltimore, MD, 1996.
36. Nocedal J, Wright SJ. *Numerical Optimization* (2nd edn). Springer: Berlin, 2006.
37. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics: Philadelphia, PA, 1995.
38. Bui-Thanh T, Willcox K, Ghattas O. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing* 2008; **30**(6):3270–3288.
39. Ahmed SR, Ramm G, Faitin G. Some salient features of the time-averaged ground vehicle wake. *SAE Paper 840300*. Society of Automotive Engineers Inc.: Warrendale, PA, 1984.
40. Hinterberger C, Garca-Villalba M, Rodi W. Large eddy simulation of flow around the Ahmed body. In *The Aerodynamics of Heavy Vehicles*: *Trucks*, *Buses*, *and Trains*, Lecture Notes in Applied and Computational Mechanics, McCallen JRR, Browand F (eds), vol. 19. Springer: Berlin, 2004.

DOI: 10.1002/nme

41. Bui-Thanh T, Murali D, Willcox K. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. *AIAA Paper 2003-4213*, *21st Applied Aerodynamics Conference*, Orlando, FL, 23–26 June 2003.
42. Bui-Thanh T, Damodaran M, Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal* 2004; **42**(8):1505–1516.
43. Venturi D, Karniadakis G. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics* 2004; **519**:315–336.
44. Willcox K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers and Fluids* 2006; **35**(2):208–226.
45. Robinson T, Eldred M, Willcox K, Haimes R. Strategies for multifidelity optimization with variable dimensional hierarchical models. *AIAA Paper 2006-1819*, *47th AIAA/ASME/ASCE/AHS/ASC Structures*, *Structural Dynamics*, *and Materials Conference*, Newport, RI, 1–4 May 2006.
46. Bos R, Bombois X, Van den Hof P. Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations. *Proceedings of the American Control Conference*, vol. 4, 2004; 3705–3710.

DOI: 10.1002/nme